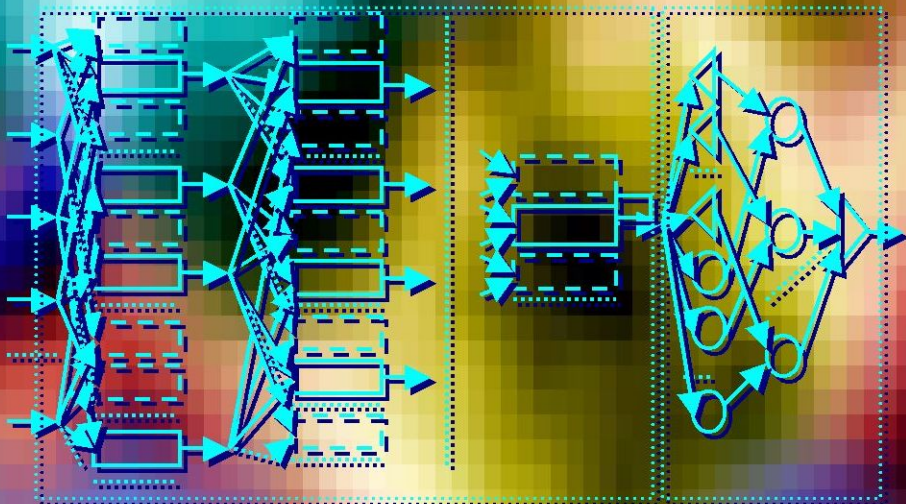


СУББОТІН С.О.
 ОЛІЙНИК А.О.
 ОЛІЙНИК О.О.

НЕІТЕРАТИВНІ, ЕВОЛЮЦІЙНІ ТА МУЛЬТИАГЕНТНІ МЕТОДИ СИНТЕЗУ НЕЧІТКОЛОГІЧНИХ І НЕЙРОМЕРЕЖНИХ МОДЕЛЕЙ

МОНОГРАФІЯ



Запорізький національний
 технічний університет

2009

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет

Субботін С. О.,
Олійник А. О.,
Олійник О. О.

**НЕІТЕРАТИВНІ, ЕВОЛЮЦІЙНІ ТА
МУЛЬТИАГЕНТНІ МЕТОДИ
СИНТЕЗУ НЕЧІТКОЛОГІЧНИХ
І НЕЙРОМЕРЕЖНИХ МОДЕЛЕЙ**

Монографія

*Під загальною редакцією
лауреата премії Президента України,
к. т. н., доцента С. О. Субботіна*

*Видання здійснено за підтримки Міжнародного проекту
«Європейсько-український ступінь магістра
з програмної інженерії» (JEP 26182–2005)
за програмою Tetris Tacis Європейської Комісії*

Запоріжжя
2009

ББК 32.973
С89
УДК 004.93:681.32

*Рекомендовано до друку Вченою радою
Запорізького національного технічного університету
(Протокол № 4 від 01.12.2008 р.)*

Рецензенти: доктор технічних наук, професор кафедри штучного інтелекту Харківського національного університету радіоелектроніки *Бодяньський Є. В.*;
доктор технічних наук, професор, декан математичного факультету Запорізького національного університету *Гоменюк С. І.*;
доктор технічних наук, професор кафедри обчислювальної техніки та програмування Національного технічного університету «Харківський політехнічний інститут» *Дмитрієнко В. Д.*

Субботін С. О., Олійник А. О., Олійник О. О.

С89 Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: Монографія / Під заг. ред. С. О. Субботіна. – Запоріжжя: ЗНТУ, 2009. – 375 с.

ISBN 978–966–7809–96–6

Книга містить систематизований виклад результатів дослідження неітеративних, еволюційних та мультиагентних методів синтезу моделей залежностей на основі штучних нейронних та нейро-нечітких мереж. Значну увагу приділено аналізу та класифікації еволюційних методів, а також мультиагентних методів Swarm Intelligence (колективного інтелекту). Запропоновано широкий арсенал нових методів для побудови моделей в задачах діагностики, оцінювання, прогнозування. Наведено приклади вирішення практичних завдань на основі досліджених методів.

Видання призначене для наукових працівників, аспірантів, практичних фахівців у галузі комп'ютерних наук, а також студентів комп'ютерних спеціальностей вищих навчальних закладів.

ББК 32.973

ISBN 978–966–7809–96–6

© ЗНТУ, 2009
© Субботін С. О.,
Олійник А. О.,
Олійник О. О., 2009

ЗМІСТ

ВСТУП	11
ЧАСТИНА I НЕІТЕРАТИВНІ МЕТОДИ СИНТЕЗУ НЕЙРО-НЕЧІТКИХ ТА НЕЙРОМЕРЕЖНИХ МОДЕЛЕЙ.....	15
Розділ 1 Синтез розпізнавальних моделей	15
1.1 Априорна інформація про навчальну вибірку.....	15
1.2 Редукція кількості нечітких термів	17
1.3 Об'єднання суміжних термів по ознаках.....	19
1.4 Метод донавчання нейро-нечітких мереж.....	21
1.5 Виділення нечітких термів на основі інтервалів значень ознак, що перетинаються для різних класів	23
1.6 Синтез тришарових розпізнавальних нейро-нечітких моделей.....	27
1.7 Метод синтезу класифікувальних нейро-нечітких мереж з урахуванням значимості термів ознак.....	29
1.8 Синтез чотиришарових розпізнавальних нейро-нечітких моделей з урахуванням інформативності ознак.....	33
1.9 Ідентифікація нечітких, нейро-нечітких і нейромережних моделей за прецедентами для вирішення задач розпізнавання образів.....	36
1.9.1 Метод ідентифікації параметрів нечітких моделей	36
1.9.2 Синтез структури й ідентифікація параметрів нейро-нечітких моделей.....	38
1.9.3 Синтез структури й ідентифікація параметрів нейромережних моделей.....	40
1.10 Синтез нейро-нечітких мереж з групуванням ознак.....	44
1.11 Методи синтезу нейромережних і нейро-нечітких розпізнавальних моделей з лінеаризацією, факторним групуванням та згорткою ознак.....	51
1.11.1 Постановка задачі синтезу нейронних та нейро-нечітких моделей.....	52
1.11.2 Аналіз інформації про ознаки і топологію навчальної вибірки.....	52
1.11.3 Синтез нейронної мережі	55
1.11.4 Синтез нейро-нечітких мереж.....	57
Розділ 2 Синтез ієрархічних логічно прозорих нейро-нечітких мереж	59
2.1 Метод синтезу ієрархічних логічно прозорих нейро-нечітких мереж	59
2.2 Модель ієрархічної логічно прозорої нейро-нечіткої мережі.....	62
2.3 Модифікації ієрархічної логічно прозорої нейро-нечіткої мережі	63

Розділ 3 Синтез нейро-нечіких апроксиматорів	69
3.1 Об'єднання нечіких термів у кластери	69
3.2 Нейро-нечітка кластер-регресійна апроксимація	71
3.3 Метод нейро-нечіткої класифікації за узагальненою віссю	74
3.3.1 Постановка задачі	75
3.3.2 Формування узагальненої вісі	76
3.3.3 Визначення параметрів розбиття за узагальненою ознакою	76
3.3.4 Формування ієрархії узагальнення ознак	77
3.3.5 Структурно-параметричний синтез нейро-нечітких моделей	79
3.4 Нейро-нечітка кластер-регресійна апроксимація за узагальненою віссю.....	84
Література до частини I	94
ЧАСТИНА II ЕВОЛЮЦІЙНІ МЕТОДИ СИНТЕЗУ НЕЙРОМЕРЕЖНИХ МОДЕЛЕЙ	98
Розділ 4 Генетичні методи.....	98
4.1 Генетичний пошук як метод оптимізації.....	98
4.2 Аналогія генетичних методів з поняттями генетики	102
4.3 Узагальнена схема роботи генетичних методів.....	102
4.4 Класифікація методів еволюційного пошуку	105
4.5 Моделі генетичного пошуку.....	108
4.5.1 Канонічні моделі.....	108
4.5.2 Genitor.....	110
4.5.3 Гібридний генетичний метод	110
4.5.4 Генетичний метод зі змінним часом життя особин.....	111
4.5.5 Мобільний генетичний метод.....	113
4.5.6 Паралельні та багаторівневі генетичні методи	115
4.5.6.1 Однопопуляційні генетичні методи	116
4.5.6.2 Острівна модель.....	116
4.5.6.3 Дрібноструктурні генетичні методи	118
4.5.6.4 Ієрархічні гібриди.....	118
4.5.6.5 Дворівневий генетичний метод.....	118
4.5.7 Генетичний пошук зі зменшенням розміру популяції	119
4.6 Ініціалізація та запуск генетичного пошуку	120
4.6.1 Кодування параметрів, що оптимізуються	120
4.6.2 Визначення цільової функції.....	123
4.6.3 Ініціалізація	125
4.7 Генетичні оператори	127
4.7.1 Відбір.....	127
4.7.1.1 Пропорційний відбір	127
4.7.1.2 Відбір ранжируванням	130
4.7.1.3 Турнірний відбір.....	130
4.7.1.4 Відбір з використанням порогу	130

4.7.2 Схрещування	131
4.7.2.1 Вибір батьківської пари.....	131
4.7.2.2 Оператори схрещування для бінарних хромосом	134
4.7.2.2.1 Точкове схрещування	134
4.7.2.2.2 Однорідне схрещування	134
4.7.2.2.3 Рівномірне схрещування	135
4.7.2.2.4 Порівняльне схрещування.....	135
4.7.2.2.5 Діагональне схрещування	135
4.7.2.3 Оператори схрещування для гомологічних числових хромосом.....	136
4.7.2.3.1 Плоске схрещування.....	137
4.7.2.3.2 Арифметичне схрещування.....	137
4.7.2.3.3 Геометричне схрещування	139
4.7.2.3.4 Змішане схрещування.....	139
4.7.2.3.5 Евристичне схрещування	139
4.7.2.3.6 Нечітке схрещування.....	140
4.7.2.3.7 SBX-схрещування	141
4.7.2.4 Оператори схрещування для негомологічних числових хромосом.....	141
4.7.2.4.1 Впорядковуючий оператор схрещування	142
4.7.2.4.2 Схрещування з частковим відображенням	143
4.7.2.4.3 Циклічне схрещування	144
4.7.2.4.4 Жадібний оператор схрещування	144
4.7.2.4.5 Схрещування методом дихотомії	145
4.7.2.4.6 Оператор сегрегації.....	146
4.7.2.4.7 Інші оператори схрещування	146
4.7.3 Мутація	147
4.7.3.1 Проста мутація	147
4.7.3.2 Мутація гомологічних числових хромосом.....	149
4.7.3.3 Мутація обміну.....	151
4.7.3.4 Інвертування.....	153
4.7.3.5 Транслокація, вставка і делеція	154
4.7.4 Формування нового покоління	155
4.7.5 Критерії зупинення	156
4.8 Теоретичний аналіз генетичного пошуку	157
4.8.1 Основні поняття	157
4.8.2 Аналіз впливу генетичних операторів	158
4.8.2.1 Вплив оператора відбору.....	158
4.8.2.2 Вплив оператора схрещування	159
4.8.2.3 Вплив оператора мутації	160

4.8.3 Теорема схем.....	160
4.8.4 Будівельні блоки.....	161
4.9 Програмні засоби генетичної оптимізації.....	162
Розділ 5 Еволюційні стратегії, генетичне та еволюційне програмування.....	165
5.1 Еволюційні стратегії.....	165
5.1.1 ($\mu + \lambda$) та (μ, λ) еволюційні стратегії.....	165
5.1.2 Оператор мутації.....	166
5.1.2.1 Правило «20% успіху».....	166
5.1.2.2 Мутація в багатоеlementних еволюційних стратегіях.....	167
5.1.3 Метод імітації відпалу.....	168
5.2 Генетичне програмування.....	170
5.2.1 Основні поняття.....	171
5.2.2 Етапи генетичного програмування.....	171
5.2.3 Оператори генетичного програмування.....	171
5.2.3.1 Схрещування.....	172
5.2.3.2 Мутація.....	172
5.3 Еволюційне програмування.....	173
5.3.1 Основні поняття.....	173
5.3.2 Етапи еволюційного програмування.....	174
5.3.3 Оператор мутації.....	174
5.3.4 Метод групового врахування аргументів.....	175
Розділ 6 Керування параметрами еволюційного пошуку.....	178
6.1 Неадаптивні методи керування параметрами.....	178
6.1.1 Використання параметрів-констант.....	179
6.1.2 Детерміноване керування параметрами.....	179
6.2 Адаптивні методи.....	180
6.2.1 Оцінювальні методи.....	180
6.2.2 Пристосовувальні методи.....	180
6.2.3 Структурно-орієнтовані методи.....	181
6.2.3.1 Розбиття популяції.....	181
6.2.3.2 Метод просторової популяційної структури.....	182
Розділ 7 Полімодальна еволюційна оптимізація.....	184
7.1 Методи ухилення від передчасної збіжності.....	184
7.1.1 Уповільнення генетичної збіжності.....	184
7.1.1.1 Методи заміщення.....	184
7.1.1.2 Просторові методи.....	186
7.1.1.2.1 Клітинні еволюційні методи.....	186
7.1.1.2.2 Мозаїчна модель.....	187
7.1.1.2.3 Релігійні еволюційні методи.....	188
7.1.2 Запобігання появі співпадаючих рішень.....	188
7.1.2.1 Методи відбору.....	188
7.1.2.1.1 Методи розділення.....	188

7.1.2.1.2	Методи керування різноманітністю в популяції.....	189
7.1.2.2	Методи розбиття простору пошуку.....	190
7.1.2.2.1	Еволюційні методи, що розгалужуються.....	190
7.1.2.2.2	Методи рівноваги, що зміщується.....	190
7.1.2.2.3	Багатонаціональні еволюційні методи.....	191
7.2	Методи відновлення.....	192
7.2.1	Методи множинного заміщення.....	192
7.2.1.1	Методи випадкової міграції.....	192
7.2.1.2	Еволюційне програмування із заміщенням особин.....	193
7.2.1.3	Критичні еволюційні методи, що пристосовуються.....	193
7.2.2	Методи перезапуску і фазові методи.....	193
7.2.2.1	СНС-метод.....	193
7.2.2.2	Еволюційний метод з керуванням різноманітністю в популяції.....	194
Розділ 8 Багатокритеріальний еволюційний пошук		
з обмеженнями.....		196
8.1	Постановка задачі та класифікація методів багатокритеріальної оптимізації.....	196
8.2	Априорні методи багатокритеріальної оптимізації.....	197
8.2.1	Метод агрегувальних функцій.....	197
8.2.2	Лексикографічне впорядкування.....	198
8.2.3	Мінімаксний метод.....	198
8.2.4	Метод досягнення заданого значення.....	198
8.3	Популяційний підхід.....	199
8.4	Багатокритеріальний еволюційний пошук, заснований на підході Парето.....	199
8.4.1	Основні поняття.....	200
8.4.2	Оцінювання хромосом.....	202
8.4.3	Методи ущільнення множини Парето.....	203
8.4.4	Методи багатокритеріального еволюційного пошуку.....	205
8.4.4.1	Генетичний метод з сортуванням за принципом недомінантності.....	205
8.4.4.2	Генетичний метод з перевіркою на недомінантність в групі хромосом.....	206
8.4.4.3	Багатокритеріальний еволюційний метод.....	207
8.4.4.4	Еволюційний метод з архівуванням недомінантних хромосом.....	207
8.4.4.5	Еволюційна стратегія з архівуванням недомінантних хромосом.....	208
8.4.4.6	Генетичний метод з сортуванням за принципом недомінантності 2.....	209
8.4.4.7	Генетичний метод з перевіркою на недомінантність в групі хромосом 2.....	209

8.4.4.8 Еволюційний метод з пошуком огинаючої в просторі цільових функцій	210
8.4.4.9 Мікрогенетичний метод	210
8.4.4.10 Порівняння методів багатокритеріального еволюційного пошуку, заснованого на підході Парето	211
8.5 Критерії оптимальності	211
8.6 Підтримка обмежень	214
8.6.1 Використання штрафних функцій	214
8.6.2 Перетворення простору пошуку	215
8.6.3 Використання спеціальних еволюційних операторів	216
8.6.4 Відновлення неприпустимих рішень	216
8.6.5 Застосування багатокритеріального підходу	216
8.6.5.1 Метод багатокритеріального генетичного пошуку з підтримкою обмежень	216
8.6.5.2 Векторний генетичний метод	217
8.6.5.3 Лінійний пошук і домінування Парето	217
8.6.5.4 Мінімаксна стратегія	218
Розділ 9 Еволюційний синтез нейромережних моделей	219
9.1 Загальна постановка задачі синтезу нейромережних моделей	219
9.1.1 Вибір системи інформативних ознак	222
9.1.2 Структурний синтез	224
9.1.3 Параметричний синтез	225
9.1.4 Оптимізація побудованої нейромоделі	227
9.2 Еволюційний відбір інформативних ознак	228
9.2.1 Кодування хромосом для виділення найбільш значимого набору ознак	229
9.2.2 Фітнес-функція	230
9.2.3 Послідовність еволюційного пошуку інформативного набору ознак	232
9.2.4 Керування параметрами еволюційного пошуку	232
9.2.5 Еволюційний метод з фіксацією частини простору пошуку	235
9.2.6 Методи еволюційного пошуку з використанням апріорної інформації про значущість ознак	239
9.2.6.1 Еволюційний пошук з групуванням ознак на основі індивідуальних оцінок значущості ознак	239
9.2.6.2 Еволюційний метод з кластеризацією ознак	242
9.2.6.3 Острівна модель еволюційного пошуку з урахуванням апріорної інформації про значущість ознак	244
9.3 Параметричний синтез нейромереж	245
9.4 Структурний синтез нейронних мереж	251
9.4.1 Подання інформації про структуру нейромережі в хромосомі	251
9.4.1.1 Пряме кодування	252

9.4.1.2	Параметричне подання	253
9.4.1.3	Комбіноване подання	253
9.4.1.4	Подання у вигляді породжуючих правил	254
9.4.1.5	Фрактальне подання	254
9.4.1.6	Популяційне подання	254
9.4.1.7	Дерево подання	255
9.4.2	Визначення фітнес-функції	255
9.4.3	Послідовність виконання структурного синтезу на основі методів еволюційної оптимізації	255
9.4.4	Еволюційні оператори для структурного синтезу нейромоделей	256
9.4.5	Метод полімодального еволюційного пошуку з кластеризацією хромосом	257
9.5	Структурно-параметричний синтез	259
9.6	Адаптація навчальних правил	263
9.7	Оптимізація структури нейромоделей	264
9.8	Критерії й методика порівняння еволюційних методів синтезу нейронних мереж	267
9.9	Автоматизована система еволюційного синтезу й оптимізації нейромережних моделей	271
	Література до частини II	277
	ЧАСТИНА III ІНТЕЛЕКТУАЛЬНІ МЕТОДИ МУЛЬТИАГЕНТНОЇ ОПТИМІЗАЦІЇ (SWARM INTELLIGENCE)	286
	Розділ 10 Основи swarm intelligence	286
10.1	Мультиагентні системи	287
10.2	Основні принципи колективного інтелекту	288
	Розділ 11 Метод мурашиних колоній	292
11.1	Біологічні основи	292
11.2	Метод мурашиних колоній	295
11.3	Різновиди методу мурашиних колоній	298
11.4	Використання методу мурашиних колоній до вирішення задачі відбору інформативних ознак	300
11.4.1	Модифікація на основі подання пунктів призначення ознаками	302
11.4.2	Модифікація на основі подання пунктів призначення у вигляді інформативності ознак	305
11.4.3	Модифікація з використанням операцій над чіткими множинами	307
11.4.4	Модифікація з використанням операцій над нечіткими множинами	309
11.4.5	Порівняння модифікацій методу мурашиних колоній, що використовуються до вирішення задачі відбору інформативних ознак	309

11.5 Кластер-аналіз на основі методу мурашиних колоній	310
11.6 Переваги й недоліки методу мурашиних колоній	312
Розділ 12 Метод бджолоїної колонії	314
12.1 Біологічні основи методу бджолоїної колонії	314
12.2 Формалізація поведінки бджіл у процесі фуражування.....	316
12.3 Методи на основі моделювання бджолоїної колонії для вирішення задач дискретної оптимізації	319
12.4 Багатовимірна оптимізація на основі методу бджолоїної колонії.....	323
12.5 Відбір інформативних ознак на основі методу бджолоїної колонії.....	327
12.6 Особливості методу бджолоїної колонії	330
Розділ 13 PSO–метод	332
13.1 Фундаментальні принципи PSO-методу	332
13.2 Підхід gbest PSO	333
13.3 Підхід lbest PSO	335
13.4 Компоненти швидкості частки в PSO-методі	336
13.5 Ініціалізація та закінчення PSO-методу	338
13.6 Вибір параметрів PSO-методу.....	339
13.7 Багатокритеріальний PSO-метод.....	340
13.8 PSO-метод з керуванням зміною швидкості часток	341
13.9 Порівняння різновидів PSO-методів.....	343
13.10 Напрями використання PSO-методу.....	344
Розділ 14 Метод оптимізації на основі моделювання переміщення бактерій.....	345
14.1 Біологічні основи.....	345
14.2 Метод оптимізації на основі моделювання переміщення бактерій (Bacterial Foraging Optimization, BFO).....	346
14.3 Метод оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок зв'язку між клітинами	350
14.4 Метод оптимізації на основі моделювання переміщення бактерій з групування за рахунок використання PSO-оператору	352
14.5 Області застосування й відповідні модифікації методу оптимізації на основі моделювання переміщення бактерій	355
14.6 Метод оптимізації на основі моделювання переміщення бактерій для відбору інформативних ознак	357
14.7 Переваги та недоліки BFO	359
Література до частини III	361
ВИСНОВКИ	371

ВСТУП

Побудова моделей кількісних та якісних залежностей є одним з головних засобів автоматизації керування складними технічними об'єктами і процесами, діагностики та прогнозування їхнього стану, оптимізації режимів їхнього функціонування. На практиці проблема побудови моделей, як правило, ускладнюється багатовимірністю, невизначеністю та нелінійністю модельованих об'єктів та їхніх характеристик, відсутністю (повною або частковою) експертного досвіду та аналітичного опису залежностей.

Тому актуальним є завдання побудови моделей залежностей за окремими спостереженнями модельованого об'єкта (процесу), поданими навчальною вибіркою, яка складається з екземплярів, що можуть уявлятися як точки у просторі ознак, що описують екземпляри.

На сьогодні ефективних засобів повного вирішення проблеми побудови моделей залежностей за точковими даними не існує. Відомо декілька основних підходів щодо вирішення цієї проблеми:

- поліноміальні регресійні моделі – використовуються для задач оцінювання, вимагають попереднього визначення структури залежності, великого обсягу експериментальних даних та ітеративного оптимізаційного корегування параметрів моделі;

- імовірнісні моделі – використовуються для задач класифікації та оцінювання, вимагають попереднього визначення умовних щільностей розподілу (на практиці це, як правило, зробити дуже складно) та великого обсягу даних;

- методи теорії розпізнавання образів – використовуються для задач класифікації, ґрунтуються на визначенні розташування класів у просторі ознак на основі кластер-аналізу або дискримінантних функцій, вимагають попереднього визначення виду функції відстані або дискримінантної функції, є ітеративними, проте є логічно прозорими;

- асоціативні правила – використовуються переважно для класифікації, можуть бути суперечливими, мають низький рівень узагальнення даних, процес їхнього виділення є дуже повільним;

- дерева рішень – високоієрархічні моделі для класифікації та оцінювання, є ефективними при невеликій кількості високоінформативних ознак, мають високий рівень логічної прозорості, проте будуються у ітеративному режимі;

- штучні нейронні мережі – використовуються для задач класифікації та оцінювання, мають високий рівень узагальнення даних та ієрархічну структуру, але характеризуються низьким рівнем логічної прозорості;

- нечіткологічні моделі – використовуються для задач класифікації та оцінювання, мають низький рівень узагальнення, проте характеризуються високою логічною прозорістю;

– нейро-нечіткі мережі – поєднують переваги нейронних мереж та нечіткої логіки, використовуються для задач класифікації та оцінювання, мають високий рівень узагальнення даних та ієрархічну структуру, а також характеризуються високим рівнем логічної прозорості.

Оскільки для нейронних мереж та нейро-нечітких мереж властиві такі риси, як здатність до навчання та самонавчання, адаптивність, здатність видобувати знання з даних, ізоморфізм топології навчальних даних, масовий паралелізм обчислень, стійкість до відмови окремих елементів, ієрархічність процесу обчислень, однорідність і легкість реалізації основних обчислювальних елементів, а також те, що вони є універсальними апроксиматорами (тобто принципово є спроможними апроксимувати будь-яку залежність, що може бути обчислена) питання побудови нейронних та нейро-нечітких моделей є вельми актуальними.

Традиційним і широко використовуваним підходом до синтезу нейро-моделей є вибір їхньої архітектури (топології та принципів передачі сигналів, типів нейроелементів) та визначення її параметрів (кількості шарів, кількостей нейронів в шарах, типів використовуваних вагових і активісційних функцій) людиною-користувачем із подальшим навчанням моделей (підбором значень їхніх вагових коефіцієнтів) на вибірках навчальних даних шляхом вирішення оптимізаційних задач мінімізації критерія помилки, для чого, як правило, для найбільш часто використовуваних на практиці нейромереж прямого поширення використовують градієнтні методи безумовної багатовимірної нелінійної оптимізації із використанням техніки зворотного поширення помилки для розрахунку часткових похідних цільової функції навчання. Такий підхід має кілька вад:

- залучення людини-користувача не забезпечує створення моделі в автоматичному режимі, а також робить процес її побудови та ефективність залежною від кваліфікації користувача;

- використання градієнтних методів обумовлює необхідність ітеративного підбору значень ваг, що є дуже повільним, вимагає обчислення похідних цільової функції навчання (які визначаються на основі повільних чисельних методів і містять похибки) та дуже залежать від початкової точки пошуку у просторі ваг (вимагають визначення початкової точки пошуку, шукають рішення тільки поблизу заданої початкової точки та можуть потрапляти через це до локальних мінімумів функції помилки).

Альтернативами до традиційного підходу синтезу нейромережних та нейро-нечітких моделей є використання неітеративних методів, які не вимагають підбору значень ваг та автоматично синтезують модель, а також ітеративних стохастичних методів пошуку, що здатні здійснювати формування структури мережі та підбір її параметрів в автоматичному режимі, але є методами глобального пошуку (мають здатність виходити

з локальних мінімумів та аналізують весь простір пошуку) і не вимагають обчислення градієнтів цільової функції навчання.

Саме тому предметом дослідження, описаного у цій книзі, автори обрали неітеративні та стохастичні (еволюційні та мультиагентні) методи синтезу нечіткологічних і нейромережних моделей.

Доцільність дослідження підтверджується Державною програмою розвитку промисловості України на 2003–2011 роки, схваленою Постановою Кабінету Міністрів України № 1174 від 28.07.2003 та Указом Президента України №102/2003 від 12.02.2003 «Про Концепцію державної промислової політики», що визначають як пріоритет інформатизації створення систем підтримки прийняття рішень та штучного інтелекту; Постановою Кабінету міністрів України №1896 від 10.12.2003, яка передбачає «... розроблення методології інтелектуального аналізу даних ... на основі застосування сучасних методів нечіткої логіки, штучного інтелекту та добування знань із баз даних»; Постановою Кабінету міністрів України №789 від 15 липня 1997 р. «Про першочергові заходи інформатизації», Законом України №75/98-ВР від 04.02.1998 «Про Концепцію Національної програми інформатизації», Законом України №76/98-ВР від 04.02.1998 «Про затвердження Завдань Національної програми інформатизації на 1998–2000 роки», Постановою Верховної Ради України № 914–XIV від 13.07.1999 «Про затвердження Завдань Національної програми інформатизації на 1999–2001 роки», які передбачають, зокрема, «створити діючі зразки та прототиби конкурентоспроможних засобів та систем: методичне та програмне забезпечення проектування і розроблення комп'ютеризованих систем для застосування в управлінні, програмно-технічні засоби підтримки експертного прийняття рішень, високопродуктивні оброблювачі інформації з нейромережною архітектурою, проблемно-орієнтовані робочі станції, нейрокомп'ютери та нейромережні технології, програмне забезпечення інформаційно-аналітичної обробки текстових, фактографічних та статистичних даних, конкурентоспроможні інформаційні технології формування аналітичних електронних оглядів і довідок, засоби інтелектуалізації широкого застосування».

Матеріал книги підготовлено в межах держбюджетної науково-дослідної роботи Запорізького національного технічного університету (ЗНТУ) «Науково-методичні основи та математичне забезпечення для автоматизації й моделювання процесів керування та підтримки прийняття рішень на основі процедур розпізнавання й еволюційної оптимізації в нейромережевому та нечіткологічному базисах» (№ держ. реєстрації 0106U008621) та завдяки частковій підтримці міжнародного проекту «Європейсько-український ступінь магістра з програмної інженерії» (JEP 26182-2005) програми «Темпус» Європейської Комісії, а також за договором творчої співдружності між

ЗНТУ та ТОВ «МПА Групп» в межах госпдоговірних науково-дослідних робіт ТОВ «МПА Групп» та італійської компанії Multinetworks s.r.l. «Розробка математичного та інформаційного забезпечення інтелектуальної системи візуального контролю транспортних засобів» (№ держ. реєстрації 0106U012013) та «Розроблення методів і програмних засобів на підставі навчання, розпізнавання, оптимізації та адаптації для прийняття рішень в автоматизованих системах управління транспортними засобами» (№ держ. реєстрації 0107U0006781).

Автори висловлюють щиру подяку своїм колегам, які допомогли їм отримати доступ до використаних при підготовці книги літературних джерел (заступниці директора Інституту прикладної інформатики Вроцлавської політехніки, доктору технічних наук, професору Г. Квасницькій, заступнику директора Інституту прикладної інформатики Вроцлавської політехніки, доктору філософії, професору П. Каженку, професору Інституту прикладної інформатики Вроцлавської політехніки, доктору філософії У. Марковській-Качмар, директору Інституту мехатроніки та технічної інформатики Технічного університету в Ліберці, доктору філософії, професору П. Тумі, професору Інституту прикладної інформатики Вроцлавської політехніки, доктору філософії Я. Квятковському, професору Харківського національного університету радіоелектроніки, д. т. н. Є. В. Бодянському, проректору ЗНТУ, доценту, к. т. н. О. М. Андрієнку, завідувачу залом електронної інформації наукової бібліотеки ЗНТУ Г. Г. Киричек), проректору ЗНТУ, директору Інституту інформатики та радіоелектроніки, професору, д. т. н. Д. М. Пізі за сприяння у виданні книги, а також рецензентам за зауваження й поради (професору Харківського національного університету радіоелектроніки, д. т. н. Є. В. Бодянському, декану математичного факультету Запорізького національного університету, професору, д. т. н. С. І. Гоменюку, професору Національного технічного університету «Харківський політехнічний інститут», д. т. н. В. Д. Дмитрієнку).

ЧАСТИНА I

НЕІТЕРАТИВНІ МЕТОДИ СИНТЕЗУ НЕЙРО-НЕЧІТКИХ ТА НЕЙРОМЕРЕЖНИХ МОДЕЛЕЙ

РОЗДІЛ 1

СИНТЕЗ РОЗПІЗНАВАЛЬНИХ МОДЕЛЕЙ

Нейро-нечіткі мережі [1, 8, 10, 13, 23] поєднують у собі концепції нечітких систем [9, 12, 16, 22, 24, 26, 29–31, 33, 46] та нейромереж [2, 14, 15, 17, 19, 20, 23, 25, 27–29, 32, 33, 45, 46, 51]: вони дозволяють одержувати моделі, у яких виведення здійснюється на основі апарата нечіткої логіки [13, 51], але відповідні функції належності налагоджуються з використанням методів навчання нейромереж (додатково зазначимо, що близькими до нейро-нечітких мереж є технології видобування асоціативних правил [7, 11, 15, 21, 47] та дерев рішень [4, 48]). Такі моделі є логічно прозорими, але відомі методи їхнього синтезу є дуже повільними і високо ітеративними. Крім того, точність і узагальнюючі властивості нейро-нечітких моделей, одержуваних на основі традиційних методів, є недостатніми і залежать як від тривалості навчання, так і від кількості термів для кожної з вхідних змінних, що задається користувачем.

Тому актуальною є розробка методів, що дозволяють у неітеративному режимі (без тривалої оптимізаційної підгонки параметрів) синтезувати ефективні нейро-нечіткі моделі з високими узагальнюючими властивостями.

Для синтезу нейро-нечітких моделей у неітеративному режимі пропонується враховувати апріорну інформацію про екземпляри навчальної вибірки, а скорочення надмірності та підвищення якості апроксимації і рівня узагальнення моделей пропонується досягати шляхом редукції блоків мережі, що дублюються.

1.1 Апріорна інформація про навчальну вибірку

Нехай ми маємо навчальну вибірку x , що складається з S екземплярів x^s , де s – номер екземпляра, $s = 1, 2, \dots, S$. Кожен s -ий екземпляр будемо характеризувати набором значень N ознак x^s_j , де j – номер ознаки s -го екземпляра, $j = 1, 2, \dots, N$. Крім того, кожному екземпляру x^s зіставимо цільову ознаку y^{s*} – номер класу s -го екземпляра.

До апріорної інформації про екземпляри навчальної вибірки можна віднести показники індивідуальної інформативності ознак, а також параметри, що характеризують межі областей групування екземплярів.

Розіб'ємо інтервал значень кожної ознаки екземплярів навчальної вибірки на інтервали, у яких номер класу залишається незмінним. Це дасть нам можливість визначити, з одного боку, скільки буде потрібно поділяючих площин, перпендикулярних вісі кожної ознаки, а, з іншого боку, дозволить визначити ліву і праву межі інтервалів для класів за віссю кожної ознаки. Кількість інтервалів, а також значення меж і номера класів інтервалів для кожної ознаки можна знайти, виконавши кроки 1–12 [17].

Крок 1. Ініціалізація. Задати навчальну вибірку екземплярів, подану у вигляді масиву даних x , у якому ознаки лінеаризовано за рядками, а екземпляри – за стовпцями, а також відповідний масив $y^* = \{y^*_i\}$, що містить номери класів, зіставлені екземплярам навчальної вибірки. Створити масив $\{D_j\}$, рівний за розміром кількості ознак N , елементи якого будуть містити кількість інтервалів для кожної ознаки. Установити: $D_j = 0$, $j = 1, 2, \dots, N$, де j – номер поточної ознаки. Занести кількість екземплярів навчальної вибірки до змінної S . Установити номер поточної ознаки: $i = 1$.

Крок 2. Якщо $i \leq N$, тоді перейти до кроку 3, інакше – перейти до кроку 12.

Крок 3. Занести до буфера ознаки x вектор значень i -ої ознаки з навчальної вибірки: $x(j) = x^*_j$; занести до буфера класу у копію масиву y^* : $y(s) = y^*_s$, $s = 1, 2, \dots, S$.

Крок 4. Відсортувати масиви x та y в порядку зростання масиву x (кроки 4.1–4.7 реалізують найпростіший метод бульбашкового сортування, який можна замінити на практиці більш швидким методом).

Крок 4.1 Установити номер поточного екземпляра: $s = 1$.

Крок 4.2 Якщо $s \leq S$, тоді перейти до кроку 4.3, інакше – до кроку 5.

Крок 4.3 Установити номер поточного екземпляра: $k = s+1$.

Крок 4.4 Якщо $k \leq S$, тоді перейти до кроку 4.5, у протилежному випадку – перейти до кроку 4.7.

Крок 4.5 Якщо $x(s) > x(k)$, тоді установити: $z = x(s)$, $x(s) = x(k)$, $x(k) = z$, $z = y(s)$, $y(s) = y(k)$, $y(k) = z$, де z – буферна змінна.

Крок 4.6 Установити: $k = k+1$. Перейти до кроку 4.4.

Крок 4.7 Установити: $s = s+1$. Перейти до кроку 4.2.

Крок 5. Установити: $s = 1$, $k = 1$.

Крок 6. Якщо $s \leq S$, тоді установити: $a^i = x(s)$, де a^i – буфер для збереження лівої межі k -го інтервалу i -ої ознаки, та перейти до кроку 7, у протилежному випадку – перейти до кроку 11.

Крок 7. Поки $s < S$ та $y(s) = y(s+1)$ виконувати: $s = s+1$.

Крок 8. Якщо $s = S$ та $y(s) = y(s-1)$, тоді установити: $K(i, k) = y(s)$, $A(i, k) = a^i$, $B(i, k) = x(s)$, $k = k+1$, $s = s+1$, перейти до кроку 10. Тут $K(i, k)$ – номер класу, зіставлений екземплярам навчальної вибірки, значення i -ої ознаки яких попадає усередину k -го інтервалу; $A(i, k)$ і $B(i, k)$ – ліва і права межі k -го інтервалу i -ої ознаки, відповідно.

Крок 9. Якщо $s < S$ та $y(s) \neq y(s+1)$, тоді установити: $K(i,k) = y(s)$, $A(i,k) = a^t$, $B(i,k) = x(s)$, $k = k + 1$, $s = s + 1$, $D_i = D_i + 1$, у протилежному випадку – установити: $K(i,k) = y(s)$, $A(i,k) = x(s)$, $B(i,k) = x(s)$, $k = k + 1$, $s = s + 1$.

Крок 10. Перейти до кроку 6.

Крок 11. Установити: $i = i + 1$, перейти до кроку 2.

Крок 12. Зупинення.

У результаті виконання кроків 112 для навчальної пари $\{x, y\}$ ми одержимо масив $\{D_j\}$, що містить для кожної ознаки кількість інтервалів, на які вона розбивається, а також масиви $\{A(i, k)\}$, $\{B(i, k)\}$ та $\{K(i, k)\}$, що містять інформацію про межі інтервалів і номери класів, зіставлених їм для всіх ознак.

На основі цих масивів сформуємо масив $\{K(q)\}$, що містить номери класів для інтервалів ознак, упорядкованих у порядку зростання номерів ознак і номерів інтервалів значень ознаки.

Даний метод поряд із визначенням меж інтервалів значень ознак дозволяє визначити й оцінки інформативності ознак. Як міру інформативності ознаки стосовно вихідного параметра (міру впливу ознаки на вихідний параметр) будемо використовувати кількість інтервалів, на які розбивається діапазон значень ознаки, таких, що екземпляри, зі значенням ознаки, які попали в один інтервал, відносяться до одного класу, а екземпляри суміжних інтервалів відносяться до різних класів (чим менше кількість інтервалів, тим більше інформативність ознаки і навпаки).

Показники інформативності ознак I_j , таким чином, будемо визначати за формулою:

$$I_j = \left(\min_{i=1,2,\dots,N} D_i \right) / D_j, \\ j = 1, 2, \dots, N.$$

1.2 Редукція кількості нечітких термів

Перед синтезом нейро-нечіткої моделі для забезпечення її простоти і високих узагальнюючих й апроксимаційних властивостей зробимо видалення надлишкових блоків визначення належності до інтервалів значень ознак (нечітких термів) [37].

Приймемо таке допущення. Інтервали значень ознак x_k та x_q $[A(i, k); B(i, k)]$ та $[A(j, q); B(j, q)]$ є еквівалентними тим сильніше, чим більше екземплярів, що потрапили в k -ий інтервал значень i -ої ознаки, потраплять у q -ий інтервал значень j -ої ознаки та будуть мати при цьому однакові номери класів.

Коефіцієнт еквівалентності між k -им інтервалом значень i -ої ознаки для s -го екземпляра та q -им інтервалом значень j -ої ознаки для g -го екземпляра визначимо за формулою:

$$n(x_i^s, x_j^g, k, q) = \begin{cases} 0, & K(i, k) \neq K(j, q), \\ 0, & B(i, k) > x_i^s \text{ або } x_i^s < A(i, k), \\ 0, & B(j, q) > x_j^g \text{ або } x_j^g < A(j, q), \\ 1, & K(i, k) = K(j, q), A(i, k) \leq x_i^s \leq B(i, k), A(j, q) \leq x_j^g \leq B(j, q), \end{cases}$$

$s = 1, 2, \dots, S; g = 1, 2, \dots, S; i = 1, 2, \dots, N; j = 1, 2, \dots, N; k = 1, 2, \dots, k_i; q = 1, 2, \dots, k_j.$

Кількість екземплярів з однаковими номерами класів, що потрапили одночасно до k -го інтервалу значень i -ої ознаки та до q -го інтервалу значень j -ої ознаки визначимо за формулою:

$$N(i, k, j, q) = \sum_{s=1}^S \sum_{\substack{g=1 \\ g \neq s}}^S n(x_i^s, x_j^g, k, q),$$

$i = 1, 2, \dots, N; j = 1, 2, \dots, N; k = 1, 2, \dots, k_i; q = 1, 2, \dots, k_j.$

Нехай $N_{i,k}$ – кількість екземплярів навчальної вибірки, що потрапили до k -го інтервалу значень i -ої ознаки.

Коефіцієнт взаємної еквівалентності між k -им інтервалом значень i -ої ознаки та q -им інтервалом значень j -ої ознаки $e_{i,k,j,q}$ визначимо за формулою:

$$e_{i,k,j,q} = \min \left\{ \frac{N(i, k, j, q)}{N_{i,k}}, \frac{N(i, k, j, q)}{N_{j,q}} \right\} = \frac{N(i, k, j, q)}{\min \{N_{i,k}, N_{j,q}\}},$$

$i = 1, 2, \dots, N; j = 1, 2, \dots, N; k = 1, 2, \dots, k_i; q = 1, 2, \dots, k_j.$

Коефіцієнт взаємної еквівалентності між i -ою та j -ою ознаками для всіх екземплярів вибірки визначимо за формулою:

$$e_{i,j} = \frac{\sum_{k=1}^{k_i} \sum_{q=1}^{k_j} e_{i,k,j,q}}{\max \{k_i, k_j\}},$$

$i = 1, 2, \dots, N; j = 1, 2, \dots, N.$

Редукцію кількості ознак і блоків визначення належності значень ознак до інтервалів для нейро-нечіткої моделі будемо здійснювати шляхом виконання послідовності кроків 1–7.

Крок 1. Ініціалізація. Задати $x = \{x_i^s\}$ та $y = \{y^s\}$, $s = 1, 2, \dots, S; i = 1, 2, \dots, N.$

Крок 2. Обчислити характеристики навчальної вибірки.

Крок 2.1 Знайти: $A(i, k), B(i, k), K(i, k), I_i, N_{i,k}, k_i.$

Крок 2.2 Визначити: $N(i, k, j, q), e_{i,k,j,q}, e_{i,j}.$

Крок 3. Ранжирувати ознаки в порядку убавання I_i . Установити: $i = N.$

Крок 4. Якщо $i > 1$, тоді виконувати кроки 4.1 та 4.2.

Крок 4.1 Для $\forall j, j \neq i, j = 1, 2, \dots, (i-1)$: якщо $e_{i,j} = 1$, тоді: видалити x_i , уста-

новити $N = N - 1.$

Крок 4.2 Установити: $i = i + 1.$ Перейти до кроку 4.

Крок 5. Установити: $i = N.$

Крок 6. Якщо $i \geq 1$, тоді виконати кроки 6.1 та 6.2.

Крок 6.1 Установити: $k = k_i$.

Крок 6.2 Якщо $k \geq 1$, тоді виконати кроки 6.2.1–6.2.3.

Крок 6.2.1 Розрахувати: $c = \sum_{j=1}^{N-1} \sum_{q=1}^{k_j} e_{i,k,j,q} \cdot e_{i,k,j,q} = 1$.

Крок 6.2.2 Якщо $c \geq 1$, тоді видалити k -ий інтервал i -ої ознаки та встановити: $k_i = k_i - 1$.

Крок 6.2.3 Установити: $k = k - 1$. Перейти до кроку 6.2.

Крок 7. Зупинення.

Після скорочення кількості ознак і блоків визначення належності значень ознак до інтервалів сформуємо набір правил виду:

Якщо $A(i, k) \leq x_i^s \leq B(i, k)$, то $y_i^s = K(i, k)$,

де $s = 1, 2, \dots, S$; $i = 1, 2, \dots, N$; $k = 1, 2, \dots, k_i$, y_i^s – номер класу, до якого належить s -ий екземпляр за i -ою ознакою.

1.3 Об'єднання суміжних термів по ознаках

При побудові нейро-нечітких моделей на основі термів, сформованих за допомогою методу 1.1, можлива ситуація, коли деякі ознаки будуть містити інтервали значень, до яких потрапляє відносно невелика кількість екземплярів одного класу, а в суміжні до них інтервали буде потрапляти істотно більша кількість екземплярів іншого класу. Ця ситуація найчастіше може пояснюватися тим, що класи містять взаємопроникнення або в даних присутні викиди (нетипові чи помилкові значення).

Для підвищення узагальнюючих й апроксимаційних властивостей, а також спрощення та збільшення швидкості роботи нейро-нечітких моделей є доцільним здійснити виключення інтервалів з невеликим числом екземплярів і виконати об'єднання суміжних з ними інтервалів шляхом виконання таких кроків.

Крок 1. Ініціалізація. Задати S – кількість екземплярів навчальної вибірки, N – кількість ознак, що характеризують екземпляри навчальної вибірки, масиви для меж і номерів класів інтервалів значень ознак $\{A(j, k)\}$, $\{B(j, k)\}$, $\{K(j, k)\}$, де j – номер ознаки, $j = 1, 2, \dots, N$, k – номер інтервалу значень j -ої ознаки, $k = 1, 2, \dots, k_j$; k_j – кількість інтервалів, на які розбивається діапазон значень j -ої ознаки. Задати граничне значення α .

Крок 2. Для всіх інтервалів всіх ознак знайти $S_{j,k}$ – кількість екземплярів у k -му інтервалі значень j -ої ознаки, $k = 1, 2, \dots, k_j$, $j = 1, 2, \dots, N$. Для кожної ознаки визначити середню кількість екземплярів, що потрапляють в інтервал її значень:

$$\bar{S}_j = \frac{1}{k_j} \sum_{k=1}^{k_j} S_{j,k},$$

$$j = 1, 2, \dots, N.$$

Крок 3. Для кожного інтервалу значень кожної ознаки визначити коефіцієнти достовірності номера класу:

$$\mu_{j,k}^K = \min \left\{ 1; \frac{S_{j,k}}{\bar{S}_j} \right\}, \quad k = 1, 2, \dots, k_j, j = 1, 2, \dots, N.$$

Крок 4. Виконати видалення суміжних інтервалів.

Крок 4.1 Прийняти: $j = 1$.

Крок 4.2 Якщо $j > N$, тоді перейти до кроку 5, у протилежному випадку – перейти до кроку 4.3.

Крок 4.3 Прийняти: $k = k_j - 1$.

Крок 4.4 Якщо $k > 1$, тоді перейти до кроку 4.8, у протилежному випадку – перейти до кроку 4.5.

Крок 4.5 Якщо $K(j, k-1) = K(j, k+1) \neq K(j, k)$ та $\alpha < S_{j,k-1} + S_{j,k+1} - S_{j,k}$, тоді перейти до кроку 4.6, у протилежному випадку – перейти до кроку 4.7.

Крок 4.6. Прийняти: $B(j, k-1) = B(j, k+1)$, $\mu_{j,k}^K = \mu_{j,k-1}^K + \mu_{j,k+1}^K - \mu_{j,k}^K$. Видалити інтервал $k+1$, видалити інтервал k . Прийняти: $k_j = k_j - 2$, $k = k - 1$.

Крок 4.7. Прийняти: $k = k - 1$. Перейти до кроку 4.4.

Крок 4.8. Прийняти: $j = j + 1$. Перейти до кроку 4.2.

Крок 5. Зупинення.

Умову видалення інтервалу на кроці 4.5 можна замінити на більш гнучку: якщо $K(j, k-1) = K(j, k+1) \neq K(j, k)$ та $S_{j,k-1} - S_{j,k} > \alpha$ та $S_{j,k+1} - S_{j,k} > \alpha$, тоді ...

Запропоновані раніше умови видалення інтервалу враховували тільки кількість екземплярів, що потрапили в суміжні інтервали, але ігнорували розмір інтервалів. Для підвищення точності при видаленні й об'єднанні інтервалів доцільно виходити з певної величини, що характеризує щільність інтервалів, тобто відношення кількості екземплярів, що потрапили в інтервал, до його довжини.

Умову видалення інтервалу з грубим обліком щільності інтервалів $\rho_{j,k}$ можна записати у такий спосіб: якщо $K(j, k-1) = K(j, k+1) \neq K(j, k)$ та $\rho_{j,k-1} + \rho_{j,k+1} - \rho_{j,k} > \alpha$, тоді ..., де $\rho_{j,k} = S_{j,k} / (B(j,k) - A(j,k))$.

Визначимо функцію, що враховує щільність екземплярів для k -го інтервалу j -ої ознаки:

$$\rho_{j,k}(x_j) = S_{j,k} e^{-\frac{(x_j - \bar{x}_{j,k})^2}{2\sigma_{j,k}^2}},$$

$$\bar{x}_{j,k} = \frac{1}{S_{j,k}} \sum_{s=1}^S x_j^s, A(j,k) \leq x_j^s \leq B(j,k),$$

$$\sigma_{j,k}^2 = \frac{1}{S_{j,k} - 1} \sum_{s=1}^S (x_j^s - \bar{x}_{j,k})^2, A(j,k) \leq x_j^s \leq B(j,k).$$

Тоді умову видалення інтервалу з урахуванням щільності інтервалів можна представити в такий спосіб: якщо $K(j, k-1) = K(j, k+1) \neq K(j, k)$ та

$$\int_{A(j,k-1)}^{B(j,k+1)} \rho_{j,k-1}(x_j) dx_j + \int_{A(j,k-1)}^{B(j,k+1)} \rho_{j,k+1}(x_j) dx_j - \int_{A(j,k-1)}^{B(j,k+1)} \rho_{j,k}(x_j) dx_j > \alpha, \text{ тоді } \dots$$

Оскільки
$$\int_a^b \rho_{j,k}(x_j) dx_j = S_{j,k} \left(e^{-\frac{(b-\bar{x}_{j,k})^2}{2\sigma_{j,k}^2}} - e^{-\frac{(a-\bar{x}_{j,k})^2}{2\sigma_{j,k}^2}} \right),$$

дану умову можна подати в такий спосіб: якщо $K(j, k-1) = K(j, k+1) \neq K(j, k)$ та

$$S_{j,k-1} \left(e^{-\frac{(B(j,k+1)-\bar{x}_{j,k+1})^2}{2\sigma_{j,k-1}^2}} - e^{-\frac{(A(j,k-1)-\bar{x}_{j,k-1})^2}{2\sigma_{j,k-1}^2}} \right) + S_{j,k+1} \left(e^{-\frac{(B(j,k+1)-\bar{x}_{j,k+1})^2}{2\sigma_{j,k+1}^2}} - e^{-\frac{(A(j,k-1)-\bar{x}_{j,k+1})^2}{2\sigma_{j,k+1}^2}} \right) - S_{j,k} \left(e^{-\frac{(B(j,k+1)-\bar{x}_{j,k})^2}{2\sigma_{j,k}^2}} - e^{-\frac{(A(j,k-1)-\bar{x}_{j,k})^2}{2\sigma_{j,k}^2}} \right) > \alpha, \text{ тоді } \dots$$

1.4 Метод донавчання нейро-нечітких мереж

Нехай ми маємо нейро-нечітку мережу, параметри функцій належностей нечітких термів якої A, B, K, D , а також правила сформовані за вибіркою $\langle x, y \rangle$, $x = \{x^s\}$, $x^s = \{x_j^s\}$, $y = \{y^s\}$, $s = 1, 2, \dots, S$; $j = 1, 2, \dots, N$. Нехай також є вибірка $\langle \hat{x}, \hat{y} \rangle$, $\hat{x} = \{\hat{x}^s\}$, $\hat{x}^s = \{\hat{x}_j^s\}$, $\hat{y} = \{\hat{y}^s\}$, $s = 1, 2, \dots, \hat{S}$; $j = 1, 2, \dots, N$. Необхідно донавчити нейро-нечітку мережу на основі вибірки $\langle \hat{x}, \hat{y} \rangle$, за можливістю зберігаючи при цьому знання, отримані за вибіркою $\langle x, y \rangle$.

Для донавчання пропонується використовувати такий метод.

Етап 1. Донавчання нечітких термів.

Крок 1. Ініціалізація. Задати: параметри A, B, K, D , вибірку $\langle \hat{x}, \hat{y} \rangle$.

Прийняти: $\hat{A} = A, \hat{B} = B, \hat{K} = K, \hat{D} = D$. Установити номер поточної ознаки: $j = 1$.

Крок 2. Якщо $j > N$, тоді перейти до кроку 11.

Крок 3. Установити: $k = 1$.

Крок 4. Якщо $k > \hat{D}_j$, тоді перейти до кроку 10.

Крок 5. Установити: $s = 1$.

Крок 6. Якщо $s > \hat{S}$, тоді перейти до кроку 10.

Крок 7. Проаналізувати положення поточного екземпляра відносно поточного інтервалу поточної ознаки.

Крок 7.1 Якщо $\hat{A}(j,k) \leq \hat{x}_j^s \leq \hat{B}(j,k)$ та $\hat{y}^s = \hat{K}(j,k)$, тоді перейти до кроку 8.

Крок 7.2 Якщо $\hat{A}(j,k) \leq \hat{x}_j^s \leq \hat{B}(j,k)$ та $\hat{y}^s \neq \hat{K}(j,k)$, тоді додати два нових інтервали: $\hat{D}_j = \hat{D}_j + 2$; $\hat{A}(j,i) = \hat{A}(j,i-2)$, $\hat{B}(j,i) = \hat{B}(j,i-2)$, $\hat{K}(j,i) = \hat{K}(j,i-2)$, $i = \hat{D}_j, \hat{D}_j - 1, \dots, k+4, k+3$; $\hat{A}(j,k+2) = \hat{x}_j^s + \delta$, $\hat{B}(j,k+2) = \hat{B}(j,k)$, $\hat{K}(j,k+2) = \hat{K}(j,k)$, $\hat{A}(j,k+1) = \hat{x}_j^s$, $\hat{B}(j,k+1) = \hat{x}_j^s$, $\hat{K}(j,k+1) = \hat{y}^s$, $\hat{B}(j,k) = \hat{x}_j^s - \delta$, де δ – деяке невелике значення: $\delta \geq 0$, $\delta \ll |\hat{B}(j,k) - \hat{A}(j,k)|$; перейти до кроку 8.

Крок 7.3 Якщо $\hat{B}(j,k-1) < \hat{x}_j^s < \hat{A}(j,k)$ та $\hat{y}^s = \hat{K}(j,k)$, тоді прийняти $\hat{A}(j,k) = \hat{x}_j^s$, перейти до кроку 8.

Крок 7.4 Якщо $\hat{B}(j,k-1) < \hat{x}_j^s < \hat{A}(j,k)$ та $\hat{y}^s = \hat{K}(j,k-1)$, тоді прийняти: $\hat{B}(j,k-1) = \hat{x}_j^s$, перейти до кроку 8.

Крок 7.5 Якщо $\hat{B}(j,k) < \hat{x}_j^s < \hat{A}(j,k+1)$ та $\hat{K}(j,k) \neq \hat{y}^s \neq \hat{K}(j,k+1)$, тоді додати новий інтервал: $\hat{D}_j = \hat{D}_j + 1$; $\hat{A}(j,i) = \hat{A}(j,i-1)$, $\hat{B}(j,i) = \hat{B}(j,i-1)$, $\hat{K}(j,i) = \hat{K}(j,i-1)$, $i = \hat{D}_j, \hat{D}_j - 1, \dots, k+4, k+2$; $\hat{A}(j,k+1) = \hat{x}_j^s$, $\hat{B}(j,k+1) = \hat{x}_j^s$, перейти до кроку 8.

Крок 8. Установити: $s = s + 1$. Перейти до кроку 6.

Крок 9. Установити: $k = k + 1$. Перейти до кроку 4.

Крок 10. Установити: $j = j + 1$. Перейти до кроку 2.

Крок 11. Зупинення.

Етап 2. Довизначення правил. Відповідно до принципів побудови архітектури нейро-нечіткої мережі внести до неї довизначені терми, зберегти попередні правила, сформувати і додати правила для нових екземплярів.

1.5 Виділення нечітких термів на основі інтервалів значень ознак, що перетинаються для різних класів

Метод 1.1 припускає, що інтервали різних класів за ознаками не перетинаються, однак для задач із взаємним проникненням і складним поділом класів це призводить до виділення великої кількості інтервалів в одномірних проекціях на осях ознак і, отже, зниженню узагальнюючої здатності методу. Використання методу редукції нечітких термів (скорочення інтервалів) 1.2 дозволяє усунути ці недоліки, але лише частково.

Для забезпечення можливості вирішення задач зі складним поділом класів і побудови моделей з високою узагальнюючою здатністю пропонується за ознаками для кожного класу окремо виділяти компактні групи екземплярів, на основі значень відповідних координат яких визначати параметри інтервалів класів. Цей метод буде дозволяти за тією ж самою ознакою для різних класів виділяти цілком або частково пересічні інтервали, на основі яких можна буде більш точно визначити процедури розпізнавання.

Для знаходження параметрів інтервалів на основі запропонованого методу необхідно виконати кроки 1–21.

Крок 1. Ініціалізація. Задати навчальну вибірку $\langle x, y \rangle$. Установити номер поточної ознаки: $j=1$.

Крок 2. Якщо $j > N$, тоді перейти до кроку 19.

Крок 3. Сформувати буферні масиви x_t та y_t : $x_t = x_j, y_t = y, \forall s$.

Крок 4. Упорядкувати пари елементів масивів x_t та y_t у порядку неспадання значень x_t . Кроки 4.1–4.7 реалізують найпростіший метод бульбашкового сортування, що на практиці може бути замінений більш швидкодійним методом.

Крок 4.1 Установити номер поточного екземпляра: $s = 1$.

Крок 4.2 Якщо $s \leq S$, тоді перейти до кроку 4.3, у протилежному випадку – перейти до кроку 5.

Крок 4.3 Установити номер поточного екземпляра: $k = s+1$.

Крок 4.4 Якщо $k \leq S$, тоді перейти до кроку 4.5, у протилежному випадку – перейти до кроку 4.7.

Крок 4.5 Якщо $x_t^s > x_t^k$, тоді установити: $z = x_t^s, x_t^s = x_t^k, x_t^k = z, z = y_t^s, y_t^s = y_t^k, y_t^k = z$, де z – буферна змінна.

Крок 4.6 Установити: $k = k+1$. Перейти до кроку 4.4.

Крок 4.7 Установити: $s = s+1$. Перейти до кроку 4.2.

Крок 5. Установити номер поточного класу $q=1$.

Крок 6. Якщо $q > K$, тоді перейти до кроку 18.

Крок 7. Знайти відстані за віссю j -ої ознаки між екземплярами вибірки, що належать до q -го класу, $\forall s, p = 1, 2, \dots, S, s \neq p$:

$$r_j^q(s, p) = \begin{cases} |x_i^s - x_i^p|, & y_i^s = y_i^p = q, \\ -1, & \neg(y_i^s = y_i^p = q). \end{cases}$$

Крок 8. Знайти середню відстань за віссю j -ої ознаки між екземплярами вибірки, що належать до q -го класу:

$$\bar{r}_j^q = \frac{2}{S_q^2 - S_q} \sum_{s=1}^S \sum_{p=s+1}^S r_j^q(s, p), \quad r_j^q(s, p) \geq 0,$$

де S_q – кількість екземплярів, що належать до q -го класу.

Крок 9. Установити номер поточного інтервалу значень j -ої ознаки екземплярів, що належать до q -го класу: $k=0$. Установити номер поточно-го екземпляра $s=1$. Установити кількість інтервалів значень j -ої ознаки для екземплярів q -го класу: $D(j, q) = 0$.

Крок 10. Якщо $s > S$, тоді перейти до кроку 17.

Крок 11. Якщо $y_i^s \neq q$, тоді прийняти: $s = s+1$, перейти до кроку 10.

Крок 12. Установити: $k + 1$, $p = s + 1$, $A(j, q, k) = x_i^s$, $B(j, q, k) = x_i^s$, $n(j, q, k) = 1$, $D(j, q) = D(j, q) + 1$. Тут $A(j, q, k)$ та $B(j, q, k)$ – відповідно, ліве і праве граничні значення k -го інтервалу j -ої ознаки для екземплярів q -го класу, $n(j, q, k)$ – кількість екземплярів q -го класу, що потрапили в k -ий інтервал j -ої ознаки для екземплярів q -го класу.

Крок 13. Якщо $p > S$, тоді перейти до кроку 17.

Крок 14. Якщо $y_i^p \neq q$, тоді прийняти: $p = p+1$, перейти до кроку 13.

Крок 15. Якщо $r_j^q(s, p) \leq \alpha_q \bar{r}_j^q$, де α_q – коефіцієнт, що залежить від кількості екземплярів q -го класу, тоді прийняти: $B(j, q, k) = x_i^p$, $n(j, q, k) = n(j, q, k) + 1$, $p = p+1$, перейти до кроку 13.

Для розрахунку α_q пропонується використовувати одну з формул:

$$\alpha_q = \frac{1}{1 + \log_2 S_q} \quad \text{або} \quad \alpha_q = \frac{\alpha}{1 + S_q}, \quad 1 \leq \alpha \leq S.$$

Крок 16. Установити: $s=p$. Перейти до кроку 10.

Крок 17. Установити: $q=q+1$. Перейти до кроку 6.

Крок 18. Установити: $j=j+1$. Перейти до кроку 2.

Крок 19. Для усіх виділених інтервалів значень ознак визначити $n(j, q, k)$ – кількість влучень екземплярів, що не належать до q -го класу, у k -ий інтервал значень j -ої ознаки для екземплярів q -го класу.

Крок 19.1 Установити: $j = 1$.

Крок 19.2 Якщо $j > N$, тоді перейти до кроку 20.

Крок 19.3 Установити: $q = 1$.

Крок 19.4 Якщо $q > K$, тоді перейти до кроку 19.13.

Крок 19.5 Установити: $k = 1$, $n(j, q, k) = 0$.

Крок 19.6 Якщо $k > D(j, q)$, тоді перейти до кроку 19.12.

Крок 19.7 Установити: $s=1$.

Крок 19.8 Якщо $s > S$, тоді перейти до кроку 19.11.

Крок 19.9 Якщо $A(j, q, k) \leq x_j^s \leq B(j, q, k)$ та $y^s \neq q$, тоді прийняти:
 $n'(j, q, k) = n'(j, q, k) + 1$.

Крок 19.10 Установити: $s = s + 1$. Перейти до кроку 19.8.

Крок 19.11 Установити: $k = k + 1$. Перейти до кроку 19.6.

Крок 19.12 Установити: $q = q + 1$. Перейти до кроку 19.4.

Крок 19.13 Установити: $j = j + 1$. Перейти до кроку 19.2.

Крок 20. Визначити показники значимості інтервалів та ознак.

Крок 20.1 Діагностичну цінність k -го інтервалу j -ої ознаки для екземплярів q -го класу пропонується визначити за формулою:

$$I(j, q, k) = \begin{cases} S_q^{-1}(n(j, q, k) - n'(j, q, k)), n(j, q, k) > n'(j, q, k); \\ 0, n(j, q, k) = n'(j, q, k); \\ -(S - S_q)^{-1}(n(j, q, k) - n'(j, q, k)), n(j, q, k) < n'(j, q, k), \end{cases}$$

$$j = 1, 2, \dots, N; q = 1, 2, \dots, K; k = 1, 2, \dots, D(j, q).$$

Крок 20.2 Діагностичну цінність j -ої ознаки для екземплярів q -го класу пропонується визначити за формулою:

$$I(j, q) = \frac{1}{D(j, q)} \sum_{k=1}^{D(j, q)} \frac{1}{1 + e^{-I(j, q, k)}}, j = 1, 2, \dots, N; q = 1, 2, \dots, K.$$

Крок 20.3 Діагностичну цінність j -ої ознаки пропонується визначити за формулою:

$$I_j = \frac{1}{K} \sum_{q=1}^K I(j, q), j = 1, 2, \dots, N.$$

Крок 21. Зупинення.

Після знаходження інтервалів й їхніх параметрів можна визначити нечіткі терми.

Як функції належності до нечітких термів можна використовувати П-образні функції:

$$\mu_{j, q, k} (x_j^s) = \mu_{j, q, k_S} (x_j^s) \mu_{j, q, k_Z} (x_j^s),$$

де $\mu_{j, q, k_S} (x_j^s)$ – S-образна функція, а $\mu_{j, q, k_Z} (x_j^s)$ – Z-образна функція:

$$\mu_{j, q, k_S} (x_j^s) = \begin{cases} 0, x_j^s < 0,5(A(j, q, k) + B(j, q, k - 1)), \\ \frac{1}{2} + \frac{1}{2} \cos \left(\frac{x_j^s - A(j, q, k)}{0,5(A(j, q, k) - B(j, q, k - 1))} \pi \right), 0,5(A(j, q, k) + B(j, q, k - 1)) \leq x_j^s \leq A(j, q, k), \\ 1, x_j^s > A(j, q, k); \end{cases}$$

$$\mu_{j,q,k_Z}(x_j^s) = \begin{cases} 1, x_j^s < B(j,q,k), \\ \frac{1}{2} + \frac{1}{2} \cos \left(\frac{x_j^s - B(j,q,k)}{0,5(A(j,q,k+1) - B(j,q,k))} \pi \right), B(j,q,k) \leq x_j^s \leq 0,5(B(j,q,k) + A(j,q,k+1)), \\ 0, x_j^s > 0,5(B(j,q,k) + A(j,q,k+1)); \end{cases}$$

або трапецієподібні функції:

$$\mu_{j,q,k}(x_j^s) = \begin{cases} 0, x_j^s \leq 0,5(A(j,q,k) + B(j,q,k-1)), \\ \frac{x_j^s - 0,5(A(j,q,k) + B(j,q,k-1))}{0,5(A(j,q,k) - B(j,q,k-1))}, 0,5(A(j,q,k) + B(j,q,k-1)) \leq x_j^s < A(j,q,k), \\ 1, A(j,q,k) \leq x_j^s \leq B(j,q,k), \\ \frac{0,5(A(j,q,k+1) + B(j,q,k)) - x_j^s}{0,5(A(j,q,k+1) - B(j,q,k))}, B(j,q,k) \leq x_j^s < 0,5(B(j,q,k) + A(j,q,k+1)), \\ 0, 0,5(B(j,q,k) + A(j,q,k+1)) \leq x_j^s, \end{cases}$$

Також можливо використовувати гаусіан:

$$\mu_{j,q,k}(x_j^s) = \exp \left(- \frac{(x_j^s - 0,5(B(j,q,k) - A(j,q,k)))^2}{2(\alpha_q)^2} \right).$$

Приналежність s -го екземпляра x^s до q -го класу за j -ою ознакою пропонується оцінювати за формулою:

$$\mu_{q,j}(x^s) = \max_{k=1,2,\dots,D(j,q)} (I(j,q,k) \mu_{q,j,k}(x^s)), q = 1, 2, \dots, K; j=1,2,\dots,N.$$

Приналежність s -го екземпляра x^s до q -го класу пропонується оцінювати за формулою:

$$\mu_q(x^s) = \max_{j=1,2,\dots,N} (I(j,q) \mu_{q,j}(x^s))$$

$$\text{або } \mu_q(x^s) = \frac{1}{N} \sum_{j=1}^N I(j,q) \mu_{q,j}(x^s)$$

$$\text{або } \mu_q(x^s) = f \left(\sum_{j=1}^N I(j,q) \mu_{q,j}(x^s) \right),$$

де $f(x) = 1/(1+e^{-x})$, $q = 1, 2, \dots, K$.

Використовуючи запропонований метод і формули для визначення належностей розпізнаваного екземпляра, можна модифікувати відповідним чином наведені далі методи синтезу нейро-нечітких мереж.

1.6 Синтез тришарових розпізнавальних нейро-нечітких моделей

Визначивши характеристики екземплярів навчальної вибірки і виконавши редукцію нечітких термів, задамо функції належності для інтервалів значень ознак $\mu_{i,k}(x_i)$, де i – номер ознаки, k – номер інтервалу значень i -ої ознаки.

Як функції належності можна використовувати трапецієподібні функції:

$$\mu_{i,k}(x_i) = \begin{cases} 0, & x_i \leq 0,5(A(i,k) + B(i,k-1)), \\ \frac{x_i - 0,5(A(i,k) + B(i,k-1))}{0,5(A(i,k) - B(i,k-1))}, & 0,5(A(i,k) + B(i,k-1)) \leq x_i < A(i,k), \\ 1, & A(i,k) \leq x_i \leq B(i,k), \\ \frac{0,5(A(i,k+1) + B(i,k)) - x_i}{0,5(A(i,k+1) - B(i,k))}, & B(i,k) \leq x_i < 0,5(B(i,k) + A(i,k+1)), \\ 0, & 0,5(B(i,k) + A(i,k+1)) \leq x_i, \end{cases}$$

або П-образні функції: $\mu_{i,k}(x_i) = \mu_{i,kS}(x_i)\mu_{i,kZ}(x_i)$, де $\mu_{i,kS}(x_i)$ – S-образна функція, а $\mu_{i,kZ}(x_i)$ – Z-образна функція:

$$\mu_{i,kS}(x_i) = \begin{cases} 0, & x_i < 0,5(A(i,k) + B(i,k-1)), \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x_i - A(i,k)}{0,5(A(i,k) - B(i,k-1))} \pi\right), & 0,5(A(i,k) + B(i,k-1)) \leq x_i \leq A(i,k), \\ 1, & x_i > A(i,k); \end{cases}$$

$$\mu_{i,kZ}(x_i) = \begin{cases} 1, & x_i < B(i,k), \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x_i - B(i,k)}{0,5(A(i,k+1) - B(i,k))} \pi\right), & B(i,k) \leq x_i \leq 0,5(B(i,k) + A(i,k+1)), \\ 0, & x_i > 0,5(B(i,k) + A(i,k+1)). \end{cases}$$

Далі задамо спосіб знаходження належностей $\mu^0(x^s)$ та $\mu^1(x^s)$ розпізнаваного екземпляра x^s до класів 0 та 1, відповідно:

$$\mu^0(x^s) = \max \mu_{i,k}(x_i), K(i,k) = 0; \quad \mu^1(x^s) = \max \mu_{i,k}(x_i), K(i,k) = 1,$$

$$i = 1, 2, \dots, N; k = 1, 2, \dots, k_i.$$

Після чого визначимо спосіб дефаззифікації:

$$y^s = \begin{cases} 1, & \mu^1(x^s) > \mu^0(x^s), \\ 0, & \mu^1(x^s) \leq \mu^0(x^s). \end{cases}$$

Запропонований метод дозволить синтезувати розпізнавальні моделі на основі тришарової нейро-нечіткої мережі [39], схему якої зображено на рис. 1.1.

На входи мережі надходять значення ознак розпізнаваного екземпляра. Вузли першого шару мережі визначають належності розпізнаваного екземпляра до інтервалів значень ознак. Вузли другого шару мережі визначають належності розпізнаваного екземпляра до класів. Єдиний вузол третього шару здійснює дефаззифікацію.

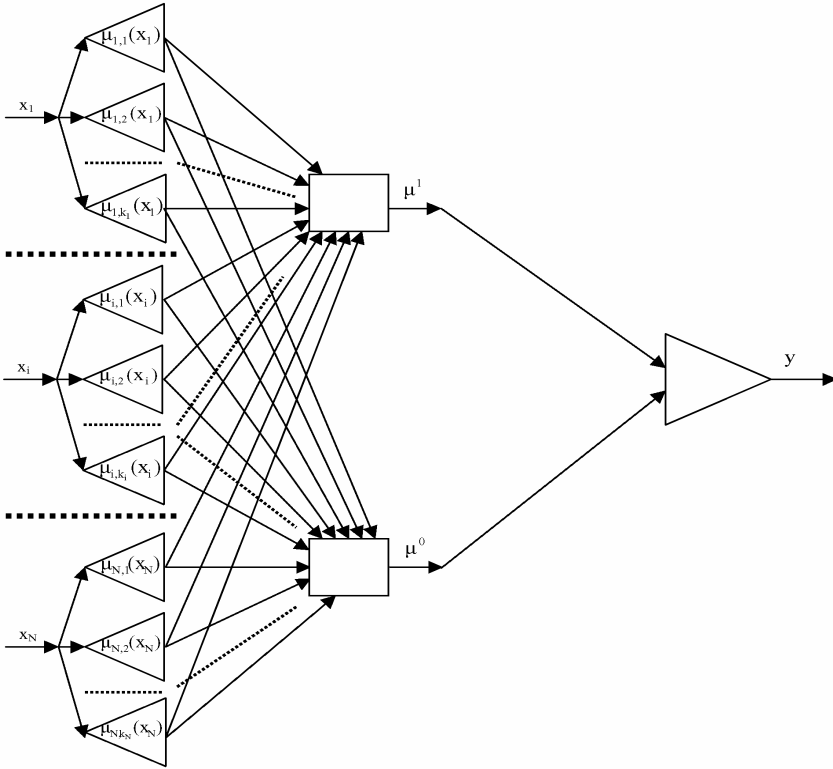


Рисунок 1.1 – Схема нейро-нечіткої мережі

Нейрони нейро-нечіткої мережі, синтезованої на основі запропонованого методу, будуть мати функції постсинаптичного потенціалу і функції активації, що задаються формулами:

$$\varphi^{(3,1)}(w^{(3,1)}, x^{(3,1)}) = \sum_{j=1}^2 w_j^{(3,1)} x_j^{(3,1)} + w_0^{(3,1)},$$

$$\psi^{(3,1)}(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0, \end{cases}$$

$$\varphi_j^{(2,i)}(w^{(2,i)}, x^{(2,i)}) = \min(w_j^{(2,i)}, x_j^{(2,i)}), \quad \psi^{(2,i)}(x) = \max \varphi_j^{(2,i)}(w_j^{(2,i)}, x_j^{(2,i)}), \quad i=1, 2,$$

де $\psi^{(\eta,i)}(x)$ – функція активації i -го нейрона η -го шару мережі, $\varphi_j^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)})$ – функція постсинаптичного потенціалу j -го входу i -го ней-

рона η -го шару мережі, $w^{(\eta,i)}, x^{(\eta,i)}$ – набори вагових коефіцієнтів та вхідних значень i -го нейрона η -го шару мережі, відповідно.

Вагові коефіцієнти нейронів $w_j^{(\eta,i)}$, де j – номер входу, i – номер нейрона, η – номер шару, будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} 0, \eta = 2, i = 1, K(p, q) = 0, j = z(p, q), p = 1, 2, \dots, N, q = 1, 2, \dots, k_p, \\ 0, \eta = 2, i = 2, K(p, q) = 1, j = z(p, q), p = 1, 2, \dots, N, q = 1, 2, \dots, k_p, \\ 1, \eta = 2, i = 1, K(p, q) = 1, j = z(p, q), p = 1, 2, \dots, N, q = 1, 2, \dots, k_p, \\ 1, \eta = 2, i = 2, K(p, q) = 0, j = z(p, q), p = 1, 2, \dots, N, q = 1, 2, \dots, k_p, \\ 0, \eta = 2, i = 1, 2, j = 0, \\ 0, \eta = 3, i = 1, j = 0, \\ 1, \eta = 3, i = 1, j = 1, \\ -1, \eta = 3, i = 1, j = 2, \end{cases}$$

$$\text{де } z(p, q) = q + \sum_{v=1}^{p-1} k_v.$$

Запропонований метод синтезу нейро-нечітких моделей налагоджує параметри функцій належності ітеративно в процесі синтезу нейро-нечіткої моделі на основі попередньо визначених параметрів інтервалів значень ознак, на відміну від традиційного підходу, коли параметри функцій належності налагоджуються шляхом ітеративної оптимізації.

1.7 Метод синтезу класифікувальних нейро-нечітких мереж з урахуванням значимості термів ознак

Класичні методи синтезу нейро-нечітких моделей [1, 50], як правило, припускають виконання настроювання їхніх вагових коефіцієнтів і параметрів функцій належності на основі ітеративної оптимізації, реалізованої на основі модифікацій методу зворотного поширення помилки.

Істотними недоліками класичних методів синтезу нейро-нечітких моделей є низька швидкість навчання, обумовлена ітеративною корекцією параметрів, надзвичайно висока залежність якості та часу навчання від вибору початкової точки пошуку в просторі керованих змінних, можливість потрапляння до локальних мінімумів цільової функції навчання.

Як альтернатива до класичних методів ітеративного настроювання нейро-нечітких моделей раніше автором були запропоновані неітеративні методи [35, 38], засновані на виділенні для кожної ознаки інтервалів значень, у яких номер класу залишається незмінним. При цьому кількість виділюваних інтервалів для ознак виявляється різною, а самі інтервали мають різну ширину.

Перевагами даного підходу є високий рівень ізоморфності нейро-нечіткої мережі топології навчальної множини і легкість подальшого аналізу людиною. Однак апаратна реалізація нейро-нечітких мереж, що налагоджуються за допомогою даних методів, є трохи ускладненою, оскільки кількість інтервалів для ознак є апіорно невідомою.

Тому є доцільним створення методу, що дозволить в неітеративному режимі синтезувати нейро-нечіткі моделі, що легко реалізуються апаратно і програмно й у той же час є зручними для сприйняття й аналізу людиною.

Нехай ми маємо навчальну вибірку $\langle x, y \rangle$, що складається з множини екземплярів $\{x^s\} = \{x_j^s\}$, $s = 1, 2, \dots, S$, $j = 1, 2, \dots, N$, де S – кількість екземплярів у вибірці, N – кількість ознак, що характеризують екземпляри, x_j^s – значення j -ої ознаки, що характеризує s -ий екземпляр вибірки x^s , і набору значень цільової (вихідної) ознаки $y = \{y^s\}$, зіставлених кожному екземпляру вибірки, де y^s – значення цільової ознаки для екземпляра x^s , $y^s \in \{1, 2, \dots, K\}$, K – кількість класів.

Тоді задача синтезу моделі залежності $y(x)$ за вибіркою $\langle x, y \rangle$ на основі нейро-нечіткої мережі полягає у визначенні кількості нейроелементів у мережі, задаванні їхніх дискримінантних (постсинаптичних, вагових) і активаційних функцій, визначенні топології мережі і розрахунку вагових коефіцієнтів міжнейронних зв'язків.

Нечіткі терми будемо формувати для рівних за шириною інтервалів значень кожної з ознак. При цьому функції належності будемо задавати таким чином, щоб з однієї сторони враховувати положення розпізнаваного екземпляра щодо центра інтервалу терма, а з іншої сторони забезпечити часткове перекриття термів для урахування нелінійностей меж класів.

Після формування нечітких термів є доцільним визначення оцінки значимості (важливості, інформативності) відповідних термів для визначення належності розпізнаваного екземпляра до кожного класу.

Для формування нечітких термів і оцінювання їхньої значимості пропонується виконувати послідовність кроків 1–6.

Крок 1. Ініціалізація. Задати навчальну вибірку $\langle x, y \rangle$. Задати кількість рівнів квантування значень ознак q і масштабувальний коефіцієнт ширини вікна активації нечітких термів δ , $0 \leq \delta < 0,5$.

Крок 2. Розбити діапазон значень кожної ознаки на q рівних інтервалів, шириною $q_i = (\max(x_i) - \min(x_i))/q$.

Крок 3. Сформувати нечіткі терми для кожної ознаки. Для чого задати $\mu_{i,j}(x_i^s)$ – функцію належності розпізнаваного s -го екземпляра вибірки x^s за i -ою ознакою до j -го терму i -ої ознаки відповідно до формули:

$$\mu_{i,j}(x_i^s) = \begin{cases} 0, x_i^s \leq q_i(j-\delta-1); \\ 0, x_i^s \geq q_i(j+\delta); \\ \frac{x_i^s - q_i(j-\delta-1)}{q_i(0,5-j+\delta)}, q_i(j-\delta-1) < x_i^s \leq q_i(2j-1,5); \\ \frac{x_i^s - q_i(j+\delta)}{q_i(j-1,5-\delta)}, q_i(2j-1,5) < x_i^s \leq q_i(j+\delta), \end{cases}$$

$$i = 1, 2, \dots, N; j = 1, 2, \dots, q.$$

Крок 4. Знайти для кожної ознаки кожного екземпляра навчальної вибірки значення функцій належності екземпляра до нечітких термів $\mu_{i,j}(x_i^s)$, $s = 1, 2, \dots, S$; $i = 1, 2, \dots, N$; $j = 1, 2, \dots, q$.

Крок 5. Визначити значення коефіцієнтів, що характеризують апріорну приналежність j -го терма i -ої ознаки до k -го класу:

$$\alpha_{i,j,k} = \frac{\sum_{s=1}^S \mu_{i,j}(x_i^s), y^s = k}{\sum_{s=1}^S \mu_{i,j}(x_i^s)}.$$

Крок 6. Зупинення.

На основі інформації про виділені нечіткі терми можливо синтезувати тришарову нейро-нечітку мережу [38], схему якої зображено на рис. 1.2.

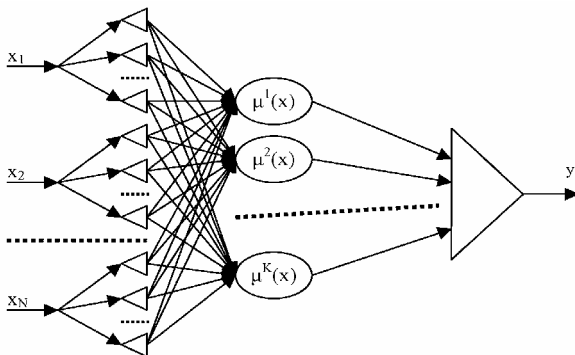


Рисунок 1.2 – Схема тришарової нейро-нечіткої мережі

На входи нейро-нечіткої мережі надходять значення відповідних ознак розпізнаваного екземпляра.

Нейрони першого шару мережі здійснюють обчислення значень функцій належності розпізнаваного екземпляра до нечітких термів ознак.

Нейрони другого шару мережі поєднують належності розпізнаваного екземпляра до термів ознак у належності до термів класів з урахуванням

значимості термів ознак для визначення відповідного класу (на рисунку позначені як $\mu^k(x)$).

Єдиний нейрон останнього шару мережі здійснює об'єднання належностей до класів і дефаззифікацію результату.

Дискримінантні функції нейронів нейро-нечіткої мережі будуть задаватися формулами:

$$\varphi^{(2,i)}(w_j^{(2,i)}, x_j^{(2,i)}) = w_j^{(2,i)} x_j^{(2,i)}, i = 1, 2, \dots, K, j = 1, 2, \dots, Nq,$$

$$\varphi^{(3,1)}(w_j^{(3,1)}, x_j^{(3,1)}) = \frac{\sum_j w_j^{(2,i)} x_j^{(2,i)}}{\sum_j x_j^{(2,i)}}, j = 1, 2, \dots, K,$$

де $\varphi^{(n,i)}(w^{(n,i)}, x^{(n,i)})$ – дискримінантна функція i -го нейрона η -го шару, $w^{(n,i)}$ – набір вагових коефіцієнтів i -го нейрона η -го шару, $w_j^{(n,i)} = \{w_j^{(n,i)}\}$, $w_j^{(n,i)}$ – ваговий коефіцієнт j -го входу i -го нейрона η -го шару, $x^{(n,i)}$ – набір входів i -го нейрона η -го шару, $x^{(n,i)} = \{x_j^{(n,i)}\}$, $x_j^{(n,i)}$ – значення на j -му вході i -го нейрона η -го шару мережі.

Функції активації нейронів нейро-нечіткої мережі будуть задаватися формулами:

$$\psi^{(2,i)}(\varphi^{(2,i)}) = \max_j \{\varphi_j^{(2,i)}\} \text{ або } \psi^{(2,i)}(\varphi^{(2,i)}) = \frac{1}{Nq} \sum_j \varphi_j^{(2,i)},$$

$$i = 1, 2, \dots, K, j = 1, 2, \dots, Nq,$$

$$\psi^{(3,1)}(\varphi^{(3,1)}) = \text{round}(\varphi^{(3,1)}),$$

де $\psi^{(n,i)}(x)$ – функція активації i -го нейрона η -го шару мережі.

Вагові коефіцієнти нейронів нейро-нечіткої мережі будуть визначатися за формулою:

$$w_j^{(n,i)} = \begin{cases} \alpha_{i,j,k}, \eta = 2, i = 1, 2, \dots, K, j = (p-1)N + r, p = 1, 2, \dots, N, r = 1, 2, \dots, q; \\ j, \eta = 3, i = 1, j = 1, 2, \dots, K. \end{cases}$$

Запропонований метод синтезу класифікувальних нейро-нечетких моделей дозволяє в неітеративному режимі налагоджувати параметри нейро-нечіткої моделі. Нейро-нечіткі мережі, синтезовані на основі запропонованого методу є логічно прозорими (зручні для подальшого аналізу і сприйняття людиною), а також легко реалізуються апаратно і програмно, що дозволяє рекомендувати запропонований метод для широкого застосування на практиці.

1.8 Синтез чотиришарових розпізнавальних нейро-нечітких моделей з урахуванням інформативності ознак

Метод синтезу чотиришарових нейро-нечітких моделей з використанням характеристик навчальної вибірки й оцінок інформативності ознак подамо як неітеративну послідовність кроків 1–7 [35, 38].

Крок 1. Ініціалізація. Задати навчальну вибірку $x = \{x_j^s\}$, $j=1, 2, \dots, N$, $s=1, 2, \dots, S$, та значення цільової ознаки $y = \{y^{s*}\}$, зіставлені екземплярам навчальної вибірки.

Крок 2. Знайти $A(i,k)$, $B(i,k)$, $K(i,k)$, D_j , $K(j)$, використовуючи метод 1.1. Для кожної q -ої ознаки визначити k_q – кількість інтервалів, на які розбивається її діапазон значень.

Крок 3. Використовуючи метод 1.1 визначити показники інформативності ознак I_j , $j = 1, 2, \dots, N$.

Крок 4. Задати функції належності для інтервалів значень ознак $\mu_{i,k}(x_i)$, де i – номер ознаки, k – номер інтервалу значень i -ої ознаки. Як функції належності пропонується використовувати трапецієподібні функції або П-образні функції.

Крок 5. Сформувати правила, що поєднують належності до інтервалів значень ознак у належності до класів одномірних нечітких класифікацій за відповідними ознаками:

$$\mu_i^0 = \min \mu_{i,k}(x_i), K(i,k) = 0, k = 1, 2, \dots, k_i, i = 1, 2, \dots, N;$$

$$\mu_i^1 = \min \mu_{i,k}(x_i), K(i,k) = 1, k = 1, 2, \dots, k_i, i = 1, 2, \dots, N.$$

Крок 6. Сформувати правила, що поєднують належності до класів одномірних класифікацій за ознаками:

$$\mu^0(y) = \max \mu_i^0, \mu^1(y) = \max \mu_i^1,$$

$$i = 1, 2, \dots, N.$$

Крок 7. Задати правило для визначення чіткого результату класифікації:

$$y = \begin{cases} 0, & \mu^0(y) \geq \mu^1(y), \\ 1, & \mu^0(y) < \mu^1(y). \end{cases}$$

Запропонований метод дозволить синтезувати нейро-нечітку мережу, вагові коефіцієнти нейронів якої $w_j^{(\eta,i)}$, де j – номер входу, i – номер нейрона, η – номер шару, будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} 0, \eta = 2, i = 1, 3, \dots, (2N-1), z(j-1) < j < (1+z(j)), K(j) = 0, j = 1, 2, \dots, z(N), \\ 0, \eta = 2, i = 2, 4, \dots, 2N, z(j-1) < j < (1+z(j)), K(j) = 1, j = 1, 2, \dots, z(N), \\ 1, \eta = 2, i = 1, 3, \dots, (2N-1), j \leq z(j-1) \text{ або } j \geq (1+z(j)), j = 1, 2, \dots, z(N), \\ 1, \eta = 2, i = 2, 4, \dots, 2N, j \leq z(j-1) \text{ або } j \geq (1+z(j)), j = 1, 2, \dots, z(N), \\ 1, \eta = 2, i = 1, 3, \dots, (2N-1), z(j-1) < j < (1+z(j)), K(j) \neq 0, j = 1, 2, \dots, z(N), \\ 1, \eta = 2, i = 2, 4, \dots, 2N, z(j-1) < j < (1+z(j)), K(j) \neq 1, j = 1, 2, \dots, z(N), \\ 0, \eta = 3, i = 1, j = 2, 4, \dots, 2N, \\ 0, \eta = 3, i = 2, j = 1, 3, \dots, (2N-1), \\ 1, \eta = 3, i = 2, j = 2, 4, \dots, 2N, \\ 1, \eta = 3, i = 1, j = 1, 3, \dots, (2N-1), \\ 0, \eta = 4, i = 1, j = 0, \\ 1, \eta = 4, i = 1, j = 1, \\ -1, \eta = 4, i = 1, j = 2, \end{cases}$$

$$\text{де } z(p) = \sum_{q=1}^p k_q.$$

Функції постсинаптичного потенціалу, а також функції активації нейронів мережі будуть задаватися формулами:

$$\varphi^{(4,1)}(w^{(4,1)}, x^{(4,1)}) = \sum_{j=1}^2 w_j^{(4,1)} x_j^{(4,1)} + w_0^{(4,1)}, \quad \psi^{(4,1)}(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0, \end{cases}$$

$$\varphi^{(3,i)}(w^{(3,i)}, x^{(3,i)}) = \min_{j=1,2,\dots,2N} (w_j^{(3,i)} x_j^{(3,i)}), \quad \psi^{(3,i)}(x) = \max_{j=1,2,\dots,2N} \varphi^{(3,i)}(w_j^{(3,i)}, x_j^{(3,i)}), \quad i=1, 2,$$

$$\varphi^{(2,i)}(w^{(2,i)}, x^{(2,i)}) = \max_j (w_j^{(2,i)} x_j^{(2,i)}), \quad \psi^{(2,i)}(x) = \min_j \varphi^{(2,i)}(w_j^{(2,i)}, x_j^{(2,i)}), \quad i=1, 2, \dots, 2N,$$

де $\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)})$ – функція постсинаптичного потенціалу i -го нейрона η -го шару мережі, $\psi^{(\eta,i)}(x)$ – функція активації i -го нейрона η -го шару мережі, $w^{(\eta,i)}, x^{(\eta,i)}$ – набори вагових коефіцієнтів і вхідних значень i -го нейрона η -го шару мережі, відповідно.

Схему нейро-нечіткої мережі, синтезованої на основі запропоновано-го методу, зображено на рис. 1.3.

Як входи мережі використовуються ознаки розпізнаваного екземпляра x_j . Перший шар мережі складають функції належності розпізнаваного екземпляра до інтервалів значень ознак. Вузли другого шару поєднують належності до інтервалів значень ознак у належності до класів одномірних нечітких класифікацій за відповідними ознаками. Вузли третього шару мережі поєднують належності до класів одномірних класифікацій за ознаками. Єдиний вузол четвертого шару здійснює дефаззифікацію.

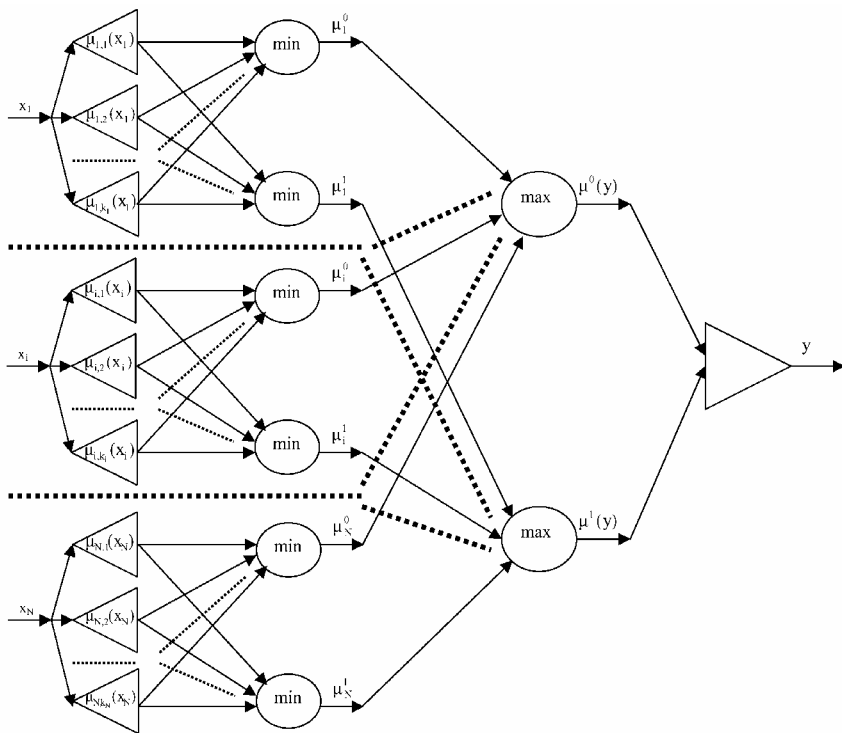


Рисунок 1.3 – Схема нейро-нечіткої мережі

Зазначимо, що в запропонованих методі синтезу і моделі нейро-нечіткої мережі параметри функцій належності налагоджуються неітеративно в процесі синтезу нейро-нечіткої моделі на основі попередньо визначених параметрів інтервалів значень ознак, на відміну від традиційного підходу, коли параметри функцій належності налагоджуються шляхом ітеративної оптимізації.

При визначенні вагових коефіцієнтів нейро-нечіткої мережі, наведеної вище, не використовувалася інформація про інформативність ознак. Однак її можна ввести в мережу за допомогою перевизначення значень ваг третього шару:

$$w_j^{(3,i)} = \begin{cases} I_j, & i = 2, j = 2, 4, \dots, 2N; \\ I_j, & i = 1, j = 1, 3, \dots, (2N - 1). \end{cases}$$

Оцінки інформативності ознак I_j можна використовувати як в абсолютному виді, так і в пронормованому: $I_{j \text{ норм.}} = \frac{I_j - \min_j(I_j)}{\max_j(I_j) - \min_j(I_j)}$.

1.9 Ідентифікація нечітких, нейро-нечітких і нейромережних моделей за прецедентами для вирішення задач розпізнавання образів

Основною проблемою при побудові нейронних і нейро-нечітких моделей є ідентифікація їхньої структури і значень параметрів, що налагоджуються.

Традиційно ця проблема вирішується шляхом задавання користувачем типу структури, кількості шарів і нейронів у шарах мереж, а також наступного ітеративного навчання моделей за прецедентами, поданими навчальною вибіркою, виконуваного, як правило, на основі методу зворотного поширення помилки [1, 10]. Недоліками такого підходу є висока залежність від користувача (недостатньо високий рівень автоматизації процесу синтезу моделей), а також ітеративність (і, як наслідок, низька швидкість) налаштування значень керованих параметрів моделей, залежність від вибору початкової крапки і локальний характер пошуку традиційних методів навчання.

Для усунення цих недоліків у [35, 38] запропоновано методи синтезу нейро-нечітких моделей у неітеративному режимі, а в [17] – неітеративні методи синтезу нейромережних моделей. Ці методи дозволяють враховувати інформацію про індивідуальну інформативність ознак. Однак, ці нейромережні та нейро-нечіткі моделі, як правило, складно претворюються з чіткого в нечітке і з нечіткого в чітке нейронні подання. Це може ускладнювати їхнє використання на практиці, оскільки апаратна реалізація чітких нейромоделей є простішою, а логічна прозорість нейро-нечітких моделей є більшою.

Тому виявляється доцільним створення методу, який дозволить ідентифікувати структуру і параметри нечітких, нейро-нечітких і нейромережних моделей у неітеративному режимі, що дозволить враховувати інформацію про інформативність ознак, а також забезпечить можливість взаємного перетворення синтезованих нечітких, нейро-нечітких і нейромережних моделей.

1.9.1. Метод ідентифікації параметрів нечітких моделей

Крок 1. Задати навчальну вибірку $\langle x, y \rangle$, де $x = \{x^s\}$, x^s – s -ий екземпляр вибірки, s – номер екземпляра вибірки, $s = 1, 2, \dots, S$, $x^s = \{x_j^s\}$, x_j^s – значення j -ої ознаки для екземпляра x^s , $j = 1, 2, \dots, N$, $y = \{y^s\}$, y^s – значення вихідної ознаки (номер класу), зіставлене екземпляру x^s , $y^s \in \{q\}$, $q = 1, 2, \dots, K$, q – номер класу, K – кількість класів, $K > 1$.

Крок 2. Сформувані нечіткі терми, для чого виконати кроки 2.1–2.3.

Крок 2.1 Розбити діапазони значень кожної ознаки на інтервали.

Для цього діапазон значень кожної змінної можна розбити на рівні за довжиною інтервали, кількість яких заздалегідь є заданою користувачем [22]. Однак такий підхід може привести до формування інтервалів, у які потраплять екземпляри різних класів, що знизить точність моделі. Рішен-

ням даної проблеми може бути використання експертної інформації про те, які інтервали значень ознак відповідають яким класам. Однак такий підхід буде сильно залежати від людини і буде непридатним при вирішенні задач, де відсутній експертний досвід. Як альтернативу двом розглянутим підходам пропонується використовувати метод [17], що не потребує експертної інформації, який оперує тільки набором прецедентів (навчальною вибіркою) і формує розбиття ознак на інтервали різної довжини, забезпечуючи високу точність нечіткої моделі.

Крок 2.2. Для кожного j -го інтервалу i -ої ознаки ($i = 1, 2, \dots, N; j = 1, 2, \dots, J_i$) визначити: значення лівої межі інтервалу A_{ij} ; значення правої межі інтервалу B_{ij} ; N_{ij} – кількість екземплярів навчальної вибірки, що потрапили в інтервал; N_{ijq} – кількість екземплярів навчальної вибірки, що потрапили в інтервал і належать до класу q ; номер домінуючого класу в інтервалі K_{ij} :

$$K_{ij} = \arg \max_{q=1,2,\dots,K} N_{ijq}$$

Крок 2.3 Для кожного інтервалу значень кожної ознаки задати функції належності до нечітких термів. Для цього пропонується використовувати функцію різниці між сигмоїдними функціями:

$$\mu_{ij}(x) = \frac{1}{1 + e^{-\gamma(x_i - A_{ij})}} - \frac{1}{1 + e^{-\gamma(x_i - B_{ij})}},$$

де γ – деяка константа, що задає крутість сигмоїдної функції, $\gamma > 1$.

Крок 3. Визначити інформативність j -го інтервалу i -ої ознаки для q -го класу як частку екземплярів, що належать до q -го класу, серед екземплярів, що потрапили в j -ий інтервал i -ої ознаки:

$$I_{ijq} = \frac{N_{ijq}}{\sum_{p=1}^K N_{ijp}},$$

$$i = 1, 2, \dots, N; j = 1, 2, \dots, J_i; q = 1, 2, \dots, K.$$

Крок 4. Визначити інформативність i -ої ознаки для q -го класу по всіх інтервалах за однією з наступних формул:

$$I_{iq} = \frac{1}{J_i} \sum_{j=1}^{J_i} I_{ijq} \quad \text{або} \quad I_{iq} = \max_{j=1,2,\dots,J_i} I_{ijq} \quad \text{або} \quad I_{iq} = \frac{\sum_{j=1}^{J_i} (I_{ijq} \tau(K_{ij}, q))}{\sum_{j=1}^{J_i} \tau(K_{ij}, q)},$$

де $i = 1, 2, \dots, N; q = 1, 2, \dots, K$, $\tau(a, b) = \begin{cases} 0, a \neq b; \\ 1, a = b. \end{cases}$

Крок 5. Визначити правила для оцінювання належності розпізнаваного екземпляра x^* до для кожного q -го класу з урахуванням інформативностей ознак ($q = 1, 2, \dots, K$):

$$\mu^q(x^*) = \frac{1}{\sum_{i=1}^N I_{iq}} \sum_{i=1}^N \left(I_{iq} \frac{\sum_{j=1}^{J_i} (\mu_{ij}(x^*) I_{ijq})}{\sum_{j=1}^{J_i} (I_{ijq})} \right) \quad \text{або} \quad \mu^q(x^*) = \frac{1}{\sum_{i=1}^N I_{iq}} \sum_{i=1}^N \left(\frac{\sum_{j=1}^{J_i} (\mu_{ij}(x^*) I_{ijq})}{\sum_{j=1}^{J_i} (I_{ijq})} \right) \quad \text{або}$$

$$\mu^q(x^*) = \max_{i=1,2,\dots,N} \left(\frac{\sum_{j=1}^{J_i} (\mu_{ij}(x^*) I_{ijq})}{\sum_{j=1}^{J_i} (I_{ijq})} \right) \quad \text{або} \quad \mu^q(x^*) = \min_{i=1,2,\dots,N} \left(\frac{\sum_{j=1}^{J_i} (\mu_{ij}(x^*) I_{ijq})}{\sum_{j=1}^{J_i} (I_{ijq})} \right).$$

Крок 6. Задати правило дефаззифікації для визначення чіткого номера класу розпізнаваного екземпляра x^* :

$$q = \arg \max_{i=1,2,\dots,N} \mu^q(x^*).$$

1.9.2. Синтез структури й ідентифікація параметрів нейро-нечітких моделей

Запропонований метод можна використовувати для синтезу структури й ідентифікації параметрів нейро-нечітких моделей. Система нечіткого виведення, синтезована на основі розробленого методу, може бути реалізована у вигляді чотиришарової нейро-нечіткої мережі, схему якої зображено на рис. 1.4.

Як нейрони першого шару нейро-нечіткої мережі виступають блоки визначення значень функцій належності розпізнаваного екземпляра до нечітких термів ознак, сформованих на кроці 2 запропонованого методу.

Нейрони другого шару здійснюють обчислення належностей розпізнаваного екземпляра до q -го класу за j -им термом i -ої ознаки.

Нейрони третього шару визначають приналежність розпізнаваного екземпляра до кожного з класів – термів вихідної ознаки.

Єдиний нейрон останнього вихідного шару мережі здійснює дефаззифікацію значення вихідної ознаки відповідно до формули кроку 6 розробленого методу.

Дискримінантні функції нейронів мережі будуть задаватися формулами:

$$\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)}) = \sum_{j=1}^{J_\Sigma} w_j^{(\eta,i)} x_j^{(\eta,i)} + w_0^{(\eta,i)},$$

$$\eta = 2: i = 1, 2, \dots, J_\Sigma K; \quad \eta = 3: i = 1, 2, \dots, K,$$

де η – номер шару, i – номер нейрона, j – номер входу нейрона, $x^{(\eta,i)}$ – значення сигналу на j -му вході i -го нейрона η -го шару, $w^{(\eta,i)}_j$ – вага j -го входу i -го нейрона η -го шару, $J_\Sigma = \sum_{a=1}^N J_a$.

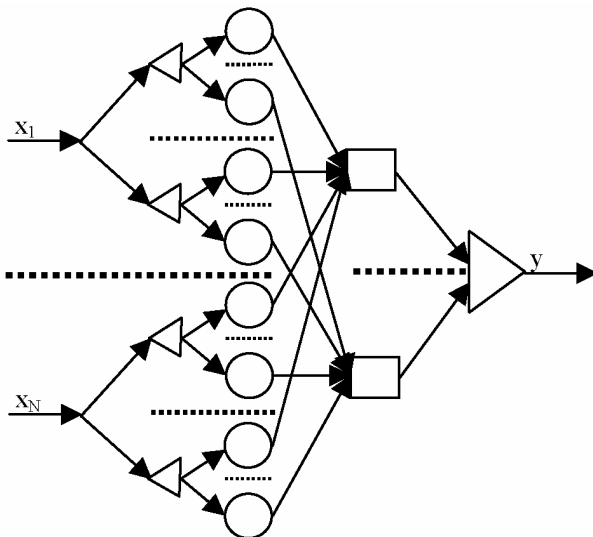


Рисунок 1.4 – Схема нейро-нечіткої мережі

Функції активації нейронів мережі будуть задаватися формулами:

$$\psi^{(\eta,i)}(x) = x, \quad \eta = 2: i = 1, 2, \dots, J_\Sigma K; \quad \eta = 3: i = 1, 2, \dots, K.$$

Вагові коефіцієнти нейронів другого і третього шарів мережі будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} 0, & \eta = 2, i = 1, 2, \dots, J_\Sigma K, j = 0; \\ I_{abq}, & \eta = 2, i = \beta_{abq}, j = (\beta_{abq} - q)/K, a = 1, 2, \dots, N, b = 1, 2, \dots, J_a, q = 1, 2, \dots, K; \\ 0, & \eta = 2, i = \beta_{abq}, j \neq (\beta_{abq} - q)/K, a = 1, 2, \dots, N, b = 1, 2, \dots, J_a, q = 1, 2, \dots, K; \\ 0, & \eta = 3, i = 1, 2, \dots, K, j = 0; \\ 0, & \eta = 3, i = 1, 2, \dots, K, j \neq \beta_{abi}, a = 1, 2, \dots, N, b = 1, 2, \dots, J_a, j = 1, 2, \dots, J_\Sigma K; \\ \alpha_{ai}, & \eta = 3, i = 1, 2, \dots, K, j = \beta_{abi}, a = 1, 2, \dots, N, b = 1, 2, \dots, J_a, j = 1, 2, \dots, J_\Sigma K, \end{cases}$$

$$\text{де } \alpha_{iq} = \frac{I_{iq}}{\sum_{j=1}^{J_i} (I_{ijq})}, \quad \beta_{abq} = K \left(b - 1 + \sum_{j=1}^{a-1} J_j \right) + q.$$

1.9.3. Синтез структури й ідентифікація параметрів нейромережних моделей

Поряд із синтезом нейро-нечітких моделей запропонований метод можна використовувати для синтезу структури і настроювання параметрів чітких нейронних мереж.

На рис. 1.5 зображено схему чотиришарової нейронної мережі прямого поширення, синтезованої на основі запропонованого методу.

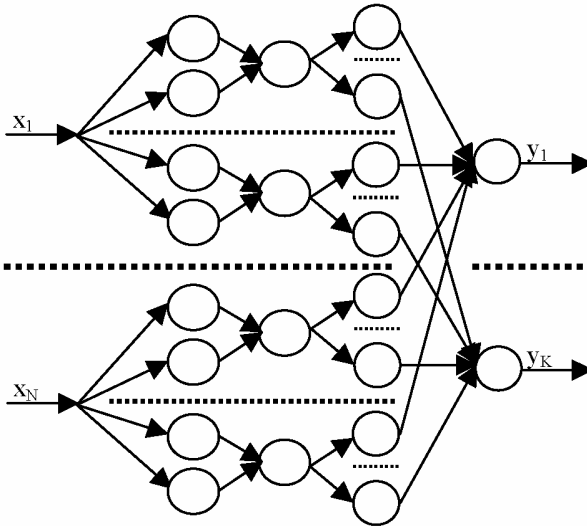


Рисунок 1.5 – Схема чотиришарової нейронної мережі

Нейрони першого і другого шарів поєднуються в групи, що працюють аналогічно блокам визначення належностей розпізнаваного екземпляра до нечітких термів. Нейрони третього шару здійснюють обчислення належностей розпізнаваного екземпляра до q -го класу за j -им термом i -ої ознаки, аналогічно роботі нейронів другого шару нейро-нечіткої мережі, розглянутої вище. Нейрони четвертого шару визначають приналежність розпізнаваного екземпляра до кожного з класів – термів вихідної ознаки. Розпізнаваний екземпляр відносять до того класу, рівень вихідного сигналу в нейрона якого на останньому шарі мережі виявиться максимальним.

Дискримінантні функції нейронів мережі будуть задаватися формулами:

$$\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)}) = \sum_{j=1}^{J_{\Sigma}} w_j^{(\eta,i)} x_j^{(\eta,i)} + w_0^{(\eta,i)},$$

$$\eta = 1: i = 1, 2, \dots, 2J_{\Sigma}; \eta = 2: i = 1, 2, \dots, J_{\Sigma};$$

$$\eta = 3: i = 1, 2, \dots, J_{\Sigma}K; \eta = 4: i = 1, 2, \dots, K.$$

Функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \frac{1}{1 + e^{-x}}, \eta = 1: i = 1, 2, \dots, 2J_{\Sigma};$$

$$\psi^{(\eta,i)}(x) = x, \eta = 2: i = 1, 2, \dots, J_{\Sigma};$$

$$\eta = 3: i = 1, 2, \dots, J_{\Sigma}K; \eta = 4: i = 1, 2, \dots, K.$$

Вагові коефіцієнти нейронів мережі будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} \gamma A_{ab}, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots, 2J_{\Sigma}-1, a=1, 2, \dots, N, j=0; \\ -\gamma, \eta=1, j=a, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots, 2J_{\Sigma}-1, a=1, 2, \dots, N; \\ 0, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots, 2J_{\Sigma}-1, j \neq a, a=1, 2, \dots, N, j=1,2,\dots,N; \\ \gamma B_{ab}, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5i, i=2,4,\dots, 2J_{\Sigma}, a=1, 2, \dots, N, j=0; \\ -\gamma, \eta=1, j=a, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots, 2J_{\Sigma}-1, a=1, 2, \dots, N; \\ 0, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots, 2J_{\Sigma}-1, j \neq a, a=1, 2, \dots, N, j=1,2,\dots,N; \\ 1, \eta=2, i=1,2,\dots, J_{\Sigma}, j=2i-1; \\ -1, \eta=2, i=1,2,\dots, J_{\Sigma}, j=2i; \\ 0, \eta=2, i=1,2,\dots, J_{\Sigma}, j \neq 2i-1, j \neq 2i, j=1,2,\dots, 2J_{\Sigma}; \\ 0, \eta=2, i=1,2,\dots, J_{\Sigma}, j=0; \\ 0, \eta=3, i=1,2,\dots, J_{\Sigma}K, j=0; \\ I_{abq}, \eta=3, i=\beta_{abq}j = (\beta_{abq} - q)/K, a=1,2,\dots,N, b=1,2,\dots,J_a, q=1,2,\dots,K; \\ 0, \eta=3, i=\beta_{abq}, j \neq (\beta_{abq} - q)/K, a=1,2,\dots,N, b=1,2,\dots,J_a, q=1,2,\dots,K; \\ 0, \eta=4, i=1,2,\dots,K, j=0; \\ 0, \eta=4, i=1,2,\dots,K, j \neq \beta_{abi}, a=1,2,\dots,N, b=1,2,\dots,J_a, j=1,2,\dots,J_{\Sigma}K; \\ \alpha_{ai}, \eta=4, i=1,2,\dots,K, j=\beta_{abi}, a=1,2,\dots,N, b=1,2,\dots,J_a, j=1,2,\dots, J_{\Sigma}K, \end{cases}$$

$$\text{де } v_j = \sum_{k=1}^j J_k.$$

Розглянута мережа може бути спрощена шляхом об'єднання другого і третього шарів, здійснюваного за допомогою переносу операцій, виконуваних нейронами другого шару, у нейрони третього шару. Таким чином, чотиришарову мережу (рис. 1.5) можна перетворити у тришарову мережу, схему якої зображено на рис. 1.6.

Нейрони першого шару поєднуються в групи, що відповідають блокам визначення належностей розпізнаваного екземпляра до нечітких термів. Нейрони другого шару здійснюють обчислення належностей розпізнаваного екземпляра до q -го класу за j -им термом i -ої ознаки, аналогічно роботі

нейронів другого і третього шарів чотиришарової нейромережі, розглянутої вище. Нейрони третього шару визначають приналежність розпізнаваного екземпляра до кожного з класів – термів вихідної ознаки. Розпізнаваний екземпляр відносять до того класу, рівень вихідного сигналу в нейрона якого на останньому шарі мережі виявиться максимальним.

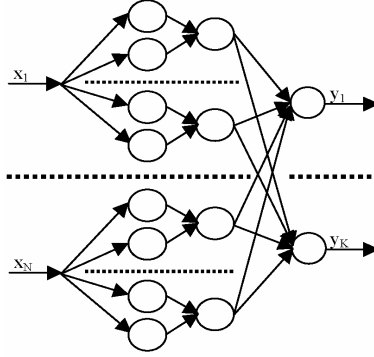


Рисунок 1.6 – Схема тришарової нейронної мережі

Дискримінантні функції нейронів мережі будуть задаватися формулами:

$$\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)}) = \sum_{j=1}^{J_{\Sigma}} w_j^{(\eta,i)} x_j^{(\eta,i)} + w_0^{(\eta,i)},$$

$\eta = 1: i = 1, 2, \dots, 2J_{\Sigma}; \eta = 2: i = 1, 2, \dots, J_{\Sigma}K; \eta = 3: i = 1, 2, \dots, K.$

Функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \frac{1}{1 + e^{-x}}, \quad \eta = 1: i = 1, 2, \dots, 2J_{\Sigma};$$

$$\psi^{(\eta,i)}(x) = x, \quad \eta = 2: i = 1, 2, \dots, J_{\Sigma}K; \eta = 3: i = 1, 2, \dots, K.$$

Вагові коефіцієнти нейронів мережі будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} \gamma A_{ab}, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots,2J_{\Sigma}-1, a=1,2,\dots,N, j=0; \\ -\gamma, \eta=1, j=a, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots,2J_{\Sigma}-1, a=1,2,\dots,N; \\ 0, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots,2J_{\Sigma}-1, j \neq a, a=1,2,\dots,N, j=1,2,\dots,N; \\ \gamma B_{ab}, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5i, i=2,4,\dots,2J_{\Sigma}, a=1,2,\dots,N, j=0; \\ -\gamma, \eta=1, j=a, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots,2J_{\Sigma}-1, a=1,2,\dots,N; \\ 0, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots,2J_{\Sigma}-1, j \neq a, a=1,2,\dots,N, j=1,2,\dots,N; \\ 0, \eta=2, i=1,2,\dots,J_{\Sigma}K, j=0; \\ I_{abq}, \eta=2, i=\beta_{abq}, (j+1)=2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots,J_a, j=1,3,\dots,2J_{\Sigma}-1, q=1,2,\dots,K; \\ -I_{abq}, \eta=2, i=\beta_{abq}, j=2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots,J_a, j=2,4,\dots,2J_{\Sigma}, q=1,2,\dots,K; \\ 0, \eta=2, i=\beta_{abq}, (j+1) \neq 2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots,J_a, j=1,3,\dots,2J_{\Sigma}-1, q=1,2,\dots,K; \\ 0, \eta=2, i=\beta_{abq}, j \neq 2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots,J_a, j=2,4,\dots,2J_{\Sigma}, q=1,2,\dots,K; \\ 0, \eta=3, i=1,2,\dots,K, j=0; \\ 0, \eta=3, i=1,2,\dots,K, j \neq \beta_{abi}, a=1,2,\dots,N, b=1,2,\dots,J_a, j=1,2,\dots,J_{\Sigma}K; \\ \alpha_{ai}, \eta=3, i=1,2,\dots,K, j=\beta_{abi}, a=1,2,\dots,N, b=1,2,\dots,J_a, j=1,2,\dots,J_{\Sigma}K. \end{cases}$$

Розглянута мережа може бути спрощена шляхом об'єднання другого і третього шарів, здійснюваного за допомогою переносу операцій, виконуваних нейронами другого шару, у нейрони третього шару. Таким чином, тришарову мережу (рис. 1.6) можна перетворити в двошарову мережу, схему якої зображено на рис. 1.7.

Нейрони першого шару поєднуються в групи, що відповідають блокам визначення належностей розпізнаваного екземпляра до нечітких термів. Нейрони другого шару визначають приналежність розпізнаваного екземпляра до кожного з класів – термів вихідної ознаки. При цьому вони включають операції, виконувані нейронами другого, третього і четвертого шарів чотиришарової мережі, розглянутої вище. Розпізнаваний екземпляр відносять до того класу, рівень вихідного сигналу в нейрона якого на останньому шарі мережі виявиться максимальним.

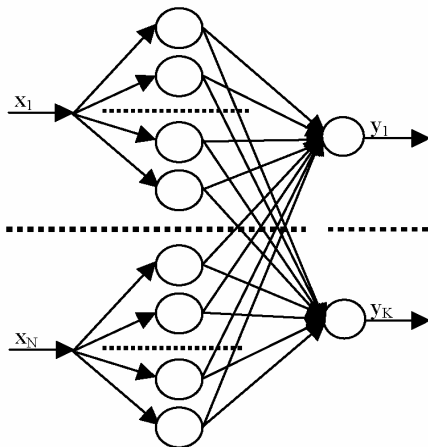


Рисунок 1.7 – Схема двошарової нейронної мережі

Дискримінантні функції нейронів мережі будуть задаватися формулами:

$$\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)}) = \sum_{j=1}^{J_{\Sigma}} w_j^{(\eta,i)} x_j^{(\eta,i)} + w_0^{(\eta,i)},$$

$$\eta = 1: i = 1, 2, \dots, 2J_{\Sigma}; \quad \eta = 2: i = 1, 2, \dots, K.$$

Функції активації будуть визначатися формулами:

$$\psi^{(\eta,i)}(x) = \frac{1}{1 + e^{-x}}, \quad \eta = 1: i = 1, 2, \dots, 2J_{\Sigma};$$

$$\psi^{(\eta,i)}(x) = x, \quad \eta = 2: i = 1, 2, \dots, K.$$

Вагові коефіцієнти нейронів мережі будуть визначатися за формулою:

$$w_j^{(q,i)} = \begin{cases} \gamma A_{ab}, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(\vartheta+1), i=1,3,\dots,2J_\Sigma-1, a=1,2,\dots,N, j=0; \\ -\gamma, \eta=1, j=a, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots,2J_\Sigma-1, a=1,2,\dots,N; \\ 0, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(i+1), i=1,3,\dots,2J_\Sigma-1, j \neq a, a=1,2,\dots,N, j=1,2,\dots,N; \\ \gamma B_{ab}, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5i, i=2,4,\dots,2J_\Sigma, a=1,2,\dots,N, j=0; \\ -\gamma, \eta=1, j=a, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(\vartheta+1), i=1,3,\dots,2J_\Sigma-1, a=1,2,\dots,N; \\ 0, \eta=1, b=p-v_{j-1}, v_{j-1} < p \leq v_j, p=0,5(\vartheta+1), i=1,3,\dots,2J_\Sigma-1, j \neq a, a=1,2,\dots,N, j=1,2,\dots,N; \\ \alpha_{ai} I_{abir}, \eta=2, (j+1)=2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots, J_a, j=1,3,\dots, 2J_\Sigma-1, i=1,2,\dots,K; \\ -\alpha_{ai} I_{abir}, \eta=2, j=2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots, J_a, j=2,4,\dots, 2J_\Sigma, i=1,2,\dots,K; \\ 0, \eta=2, (j+1) \neq 2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots, J_a, j=1,3,\dots, 2J_\Sigma-1, i=1,2,\dots,K; \\ 0, \eta=2, j \neq 2(\beta_{abq}-q)/K, a=1,2,\dots,N, b=1,2,\dots, J_a, j=2,4,\dots, 2J_\Sigma, i=1,2,\dots,K; \\ 0, \eta=2, i=1,2,\dots,K, j=0. \end{cases}$$

Запропонований метод ідентифікації параметрів нечітких моделей дозволяє на основі попередньо заданих термів ознак здійснити неітеративний синтез нечітких, нейро-нечітких і нейромережних моделей якісних залежностей. Метод враховує інформацію про інформативність ознак, а також забезпечує можливість взаємного перетворення синтезованих нечітких, нейро-нечітких та нейромережних моделей за рахунок використання однакової інформації при синтезі їхньої структури і настроюванні значень ваг.

Подання моделей, ідентифікованих на основі розробленого методу, у вигляді нечітких систем або нейро-нечітких мереж доцільно застосовувати, коли при вирішенні задачі важливо одержати можливість подальшого аналізу моделі і процесу прийняття рішень, оскільки нечітке подання має більшу логічну прозорість у порівнянні з нейромережними моделями.

Нейромережне подання моделей на основі запропонованого методу, доцільно застосовувати, коли швидкодія є важливішою за можливість аналізу моделі людиною. При цьому найбільш високий рівень логічної прозорості буде забезпечувати чотиришарова мережа (за рахунок більшої ієрархічності), а найбільш високу швидкість – двошарова мережа.

Як при апаратній, так і при програмній реалізації, запропоновані нейро-нечіткі та нейромережні моделі можна істотно спростити, якщо вважати в мережах відсутніми зв'язки, ваги яких дорівнюють нулю.

1.10 Синтез нейро-нечітких мереж з групуванням ознак

При побудові розпізнавальних моделей, особливо в задачах класифікації зображень і діагностики, не рідко виникає ситуація, коли окремо деякі ознаки слабо впливають на вихідну ознаку, а спільно впливають на неї сильно. Це може пояснюватися тим, що окремі ознаки, що доступні нами для спостереження і вимірювання, є непрямими проявами більш важливих ознак, недоступних для спостереження і (або) вимірювання, названих факторами.

Спроба побудови моделі за непрямыми ознаками може бути успішною, але така модель найчастіше вкрай незручна для подальшого аналізу, оскільки адекватно не відображає реальні залежності між ознаками або виділяє фактори неявно.

Тому є доцільним перед побудовою моделі розбивати ознаки на групи у залежності від значення певної міри їхнього взаємного зв'язку, здійснювати узагальнення ознак кожної групи шляхом розрахунку значення деякої згортки і далі будувати модель залежності вихідної ознаки від значень згорток для кожної групи ознак. Це дозволить не тільки спростити подальший аналіз моделі, але й істотно скоротити розмірність навчальної вибірки, а, отже, спростити розпізнавальну модель.

Групування ознак пропонується здійснювати шляхом виконання кроків 1–10 [43].

Крок 1. Задати навчальну вибірку $\langle x, y \rangle$, $x = \{x^s\}$, $x^s = \{x_j^s\}$, $y = \{y^s\}$, де x^s – s -ий екземпляр вибірки, x_j^s – значення j -ої ознаки s -го екземпляра вибірки, y^s – значення цільової ознаки s -го екземпляра вибірки.

Крок 2. Визначити значення показника інформативності для кожної ознаки I_j , $j = 1, 2, \dots, N$. Для цього можна використовувати метод 1.1, модуль коефіцієнта парної кореляції, коефіцієнт кореляції Фехнера, а також коефіцієнт кореляції знаків.

Крок 3. Для $i, j = 1, 2, \dots, N$, $i \neq j$, знайти відстані між ознаками в просторі екземплярів навчальної вибірки $d(x_j, x_i)$:

$$d(x_j, x_i) = \sqrt{\sum_{s=1}^S (x_j^s - x_i^s)^2}.$$

Крок 4. За допомогою методу 1.2 знайти коефіцієнти еквівалентності ознак e_{ji} , $i, j = 1, 2, \dots, N$, $i \neq j$. Прийняти: $e_{ji} = e_{ji}^{-1}$.

Крок 5. Установити номер поточної групи: $g = 0$.

Крок 6. Якщо $\exists x_j: \forall I_j \neq -1$ та $\exists d(x_j, x_i) \neq -1$, $i, j = 1, 2, \dots, N$, $i \neq j$, тоді перейти до кроку 7, у протилежному випадку – перейти до кроку 10.

Крок 7. Знайти ознаку $x_i: I_i = \max I_j$, $j = 1, 2, \dots, N$.

Крок 8. Установити: $g = g + 1$. Додати нову групу ознак G^g . Включити в групу G^g ознаку x_i . Установити: $I_i = -1$.

Крок 9. Якщо $\exists x_j: \forall I_j \neq -1$, $\exists d(x_i, x_j) \neq -1$, $F(x_i, x_j) = 1$, $i, j = 1, 2, \dots, N$, $i \neq j$, де F – певний булевий оператор, що визначає приналежність ознак до однієї групи, тоді включити в групу G^g ознаку x_j , установити: $I_j = -1$, $d(x_j, x_k) = -1$, $d(x_k, x_j) = -1$, $k = 1, 2, \dots, N$, перейти до кроку 9, у протилежному випадку – установити: $I_i = -1$, $d(x_i, x_k) = -1$, $d(x_k, x_i) = -1$, $k = 1, 2, \dots, N$, перейти до кроку 6.

Крок 10. Зупинення.

Оператор F може бути визначений одним з таких способів:

$$\begin{aligned}
F(x_i, x_j) &= \begin{cases} 1, d(x_i, x_j) < \alpha \bar{d}, \\ 0, d(x_i, x_j) \geq \alpha \bar{d}; \end{cases} & F(x_i, x_j) &= \begin{cases} 1, e_{ij} < \beta \bar{e}, \\ 0, e_{ij} \geq \beta \bar{e}; \end{cases} \\
F(x_i, x_j) &= \begin{cases} 1, \hat{d}(x_i, x_j) < \bar{\gamma} \hat{d}, \\ 0, \hat{d}(x_i, x_j) \geq \bar{\gamma} \hat{d}; \end{cases} & F(x_i, x_j) &= \begin{cases} 1, d(x_i, x_j) < \alpha \bar{d}, e_{ij} < \beta \bar{e}, \\ 0, d(x_i, x_j) \geq \alpha \bar{d} \text{ або } e_{ij} \geq \beta \bar{e}; \end{cases} \\
F(x_i, x_j) &= \begin{cases} 1, d(x_i, x_j) < \alpha \bar{d}, \hat{d}(x_i, x_j) < \bar{\gamma} \hat{d}, \\ 0, d(x_i, x_j) \geq \alpha \bar{d} \text{ або } \hat{d}(x_i, x_j) \geq \bar{\gamma} \hat{d}; \end{cases} & F(x_i, x_j) &= \begin{cases} 1, e_{ij} < \beta \bar{e}, \hat{d}(x_i, x_j) < \bar{\gamma} \hat{d}, \\ 0, e_{ij} \geq \beta \bar{e} \text{ або } \hat{d}(x_i, x_j) \geq \bar{\gamma} \hat{d}; \end{cases} \\
F(x_i, x_j) &= \begin{cases} 1, d(x_i, x_j) < \alpha \bar{d}, e_{ij} < \beta \bar{e}, \hat{d}(x_i, x_j) < \bar{\gamma} \hat{d}, \\ 0, d(x_i, x_j) \geq \alpha \bar{d} \text{ або } e_{ij} \geq \beta \bar{e} \text{ або } \hat{d}(x_i, x_j) \geq \bar{\gamma} \hat{d}. \end{cases}
\end{aligned}$$

Параметри $\bar{d}, \hat{d}, \bar{\gamma}, \bar{e}$ можуть бути визначені за відповідними формулами:

$$\begin{aligned}
\bar{d} &= \frac{1}{0,5N^2 - N} \sum_{i=1}^N \sum_{j=i+1}^N d(x_i, x_j), & \bar{e} &= \frac{1}{0,5N^2 - N} \sum_{i=1}^N \sum_{j=i+1}^N e_{ij}, \\
\bar{\gamma} \hat{d} &= \frac{1}{0,5N^2 - N} \sum_{i=1}^N \sum_{j=i+1}^N \hat{d}(x_i, x_j),
\end{aligned}$$

$$\hat{d}(x_i, x_j) = \sqrt{((i-1) \bmod \text{width} - (j-1) \bmod \text{width})^2 + ((i-1) \operatorname{div} \text{width} - (j-1) \operatorname{div} \text{width})^2},$$

де width – ширина зображення, div – операція цілочислового ділення, \bmod – операція «залишок від цілочислового ділення».

Після групування ознак можна побудувати нейро-нечіткі моделі.

У випадку, коли метою аналізу є визначення семантики схованих ознак (факторів, що відповідають групам) доцільно синтезувати мережу, зображену на рис. 1.8. Запропонована мережа може мати альтернативне подання (рис. 1.9), яке доцільно використовувати, якщо метою аналізу є установлення впливу факторів на конкретний клас.

На входи запропонованих мереж будуть подаватися значення відповідних ознак розпізнаваного екземпляра. На першому шарі обох мереж розташовані блоки визначення належності розпізнаваного екземпляра до нечітких термів. Нечіткі терми можна сформуванати також, як і в методі 1.5. Другий шар містить нейрони, що визначають приналежність розпізнаваного екземпляра за групами ознак кожного класу. Кількість нейронів другого шару $N_2 = VQ$, де V – кількість груп ознак, Q – кількість класів. Третій шар мережі містить нейрони, що поєднують належності за різними групами ознак у належності до класів. Кількість нейронів третього шару $N_3 = Q$.

Єдиний нейрон четвертого шару мережі здійснює об'єднання належностей до класів і виконує дефазифікацію.

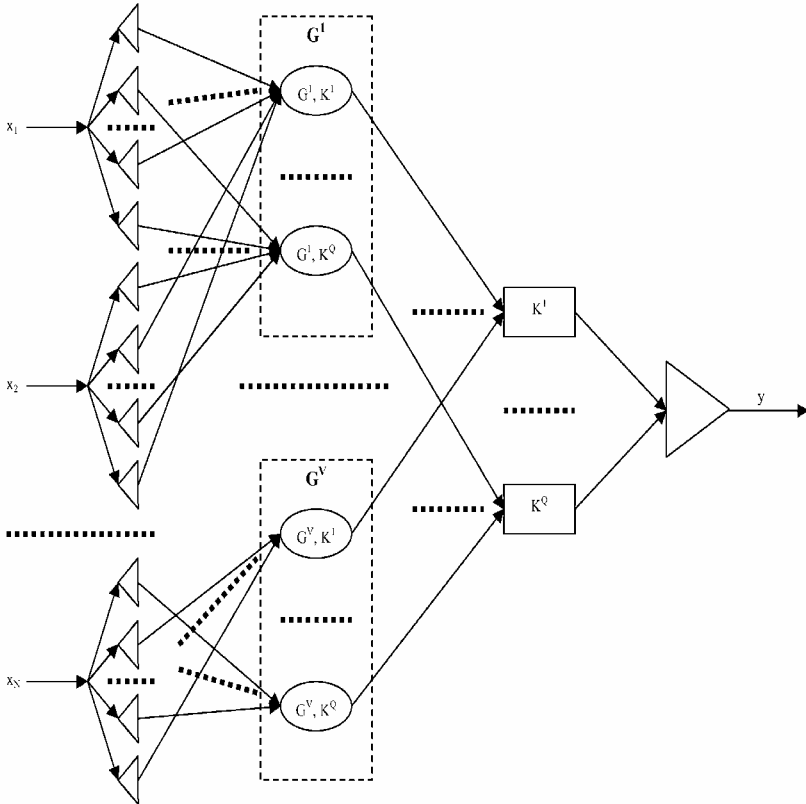


Рисунок 1.8 – Нейро-нечітка мережа з групуванням ознак

Вагові коефіцієнти нейронів $w_j^{(\eta i)}$, де j – номер входу, i – номер нейрона, η – номер шару, нейро-нечіткої мережі з групуванням ознак будуть визначатися за формулою:

$$w_j^{(\eta i)} = \begin{cases} j, \eta = 4, i = 1, j = 1, 2, \dots, K, \\ 0, \eta = 3, i = 1, 2, \dots, Q, v = 1, 2, \dots, V, j = (v-1)Q + i, \\ 1, \eta = 3, i = 1, 2, \dots, Q, v = 1, 2, \dots, V, j \neq (v-1)Q + i, \\ 1, \eta = 2, x_p \in G^v \text{ й } K(p, g) = q, j = z(p) + g, g = 1, 2, \dots, D_p, p = 1, 2, \dots, N, \\ \quad i = (v-1)Q + q, q = 1, 2, \dots, Q, v = 1, 2, \dots, V, \\ 0, \eta = 2, x_p \in G^v \text{ або } K(p, g) = q, j = z(p) + g, g = 1, 2, \dots, D_p, p = 1, 2, \dots, N, \\ \quad i = (v-1)Q + q, q = 1, 2, \dots, Q, v = 1, 2, \dots, V, \end{cases}$$

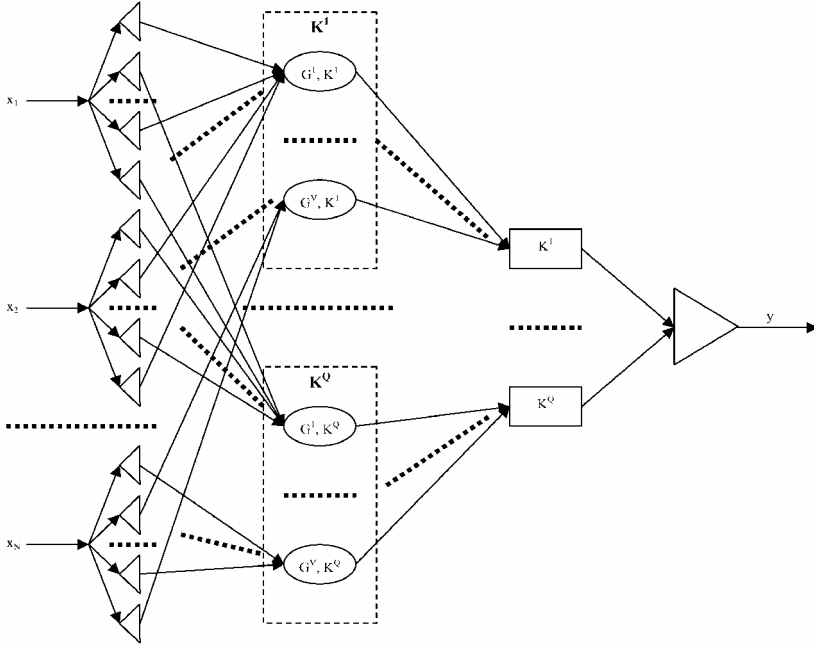


Рисунок 1.9 – Альтернативне подання нейро-нечіткої мережі з групуванням ознак

$$z(p) = \sum_{r=1}^{p-1} D_r, \quad z = \sum_{j=1}^N D_j.$$

Дискримінантні функції нейронів обох мереж будуть визначатися як:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min \{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, i = 1, 2, \dots, VQ;$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \max \{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=3, i = 1, 2, \dots, Q;$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \frac{\sum_{j=1}^K w_j^{(\eta,i)} x_j^{(\eta,i)}}{\sum_{j=1}^K x_j^{(\eta,i)}}, \quad \eta = 4, i = 1.$$

Функції активації нейронів обох мереж будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=2, i = 1, 2, \dots, VQ;$$

$$\psi^{(\eta,i)}(x) = \min_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=3, i = 1, 2, \dots, Q;$$

$$\psi^{(\eta,i)}(x) = \text{round}(x), \quad \eta=4, i = 1.$$

Для альтернативного подання мережі вагові коефіцієнти нейронів $w_j^{(\eta,i)}$ будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} j, \eta = 4, i = 1, j = 1, 2, \dots, K, \\ 0, \eta = 3, i = 1, 2, \dots, Q, v = 1, 2, \dots, V, j = (i-1)V + v, \\ 1, \eta = 3, i = 1, 2, \dots, Q, v = 1, 2, \dots, V, j \neq (i-1)V + v, \\ 1, \eta = 2, x_p \in G^v \text{ й } K(p, g) = q, j = z(p) + g, g = 1, 2, \dots, D_p, p = 1, 2, \dots, N, \\ \quad i = (q-1)V + v, v = 1, 2, \dots, V, q = 1, 2, \dots, Q, \\ 0, \eta = 2, x_p \in G^v \text{ або } K(p, g) = q, j = z(p) + g, g = 1, 2, \dots, D_p, p = 1, 2, \dots, N, \\ \quad i = (q-1)V + v, v = 1, 2, \dots, V, q = 1, 2, \dots, Q. \end{cases}$$

У випадку, якщо всі ознаки, віднесені до групи G^g , є однотипними (наприклад, є відліками сигналу чи рівнями інтенсивності точок зображення), їх можна замінити на узагальнену ознаку, що являє собою одну зі згорток:

$$x_i = \sum_{j=1}^N x_j, x_j \in G^g, \quad x_i = \frac{1}{N_G} \sum_{j=1}^N x_j, x_j \in G^g,$$

$$x_i = \min_{j=1,2,\dots,N} x_j, x_j \in G^g, \quad x_i = \max_{j=1,2,\dots,N} x_j, x_j \in G^g.$$

Поділ ознак на групи й об'єднання ознак груп за допомогою згортки дозволить синтезувати нейро-нечіткі мережі, зображені на рис. 1.10.

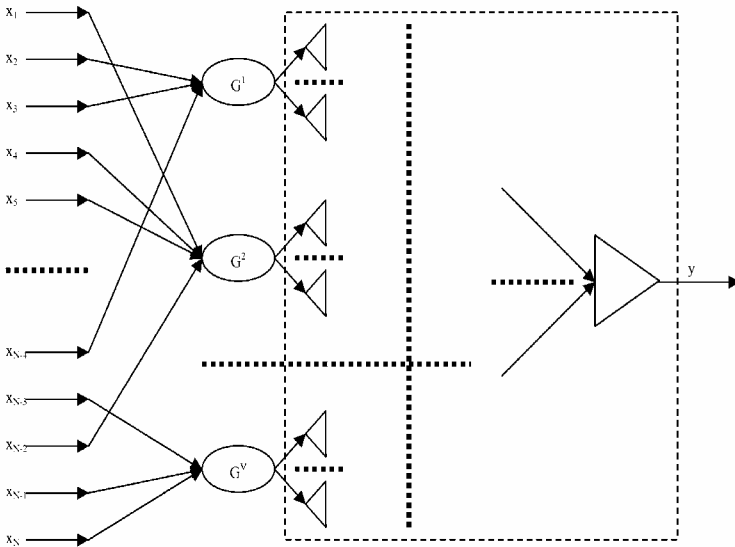


Рисунок 1.10 – Нейро-нечіткі мережа з групуванням та згорткою ознак

На входи нейро-нечіткої мережі будуть подаватися значення відповідних ознак розпізнаваного екземпляра.

На першому шарі мережі розташовані нейрони, що здійснюють об'єднання ознак у групи та розраховують значення згорток для кожної групи ознак. Кількість нейронів першого шару $N_1=V$.

Другий шар мережі містить блоки визначення належності розпізнаваного екземпляра до нечітких термів значень згорток. Нечіткі терми для інтервалів значень згорток можна сформулювати також, як і в методі 1.5, замінивши ознаки на згортки.

Третій і наступний шари мережі реалізують об'єднання належностей до нечітких термів у належності до класів. Ці шари можуть бути подані розпізнавальними нейро-нечіткими мережами різних типів.

Єдиний нейрон останнього шару мережі здійснює об'єднання належностей до класів і виконує дефазифікацію.

Як дискримінантну функцію для сум-згорток нейрони першого шару будуть використовувати:

$$\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)}) = w_0^{(\eta,i)} + \sum_{j=1}^{N_{\eta-1}} w_j^{(\eta,i)} x_j^{(\eta,i)}.$$

Як функцію активації для сум-згорток нейрони першого шару будуть використовувати: $\psi(x) = x$.

Для сум-згорток вагові коефіцієнти нейронів першого шару будуть задаватися формулою:

$$w_j^{(\eta,i)} = \begin{cases} 0, \eta = 1, i = 1, 2, \dots, V, j = 0, \\ \alpha, x_j \in G^i, \eta = 1, i = 1, 2, \dots, V, j = 1, 2, \dots, N, \\ 0, x_j \notin G^i, \eta = 1, i = 1, 2, \dots, V, j = 1, 2, \dots, N, \end{cases}$$

де $\alpha = 1$ – для суми значень ознак і $\alpha = N_G^{-1}$ – для середнього арифметичного значень ознак.

Як дискримінантну функцію нейрони першого шару для згортки \max будуть використовувати: $\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}$.

Як функцію активації для згортки \max нейрони першого шару будуть використовувати: $\psi^{(\eta,i)} = \max_i\{\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)})\}$.

При використанні згортки \max вагові коефіцієнти нейронів першого шару будуть задаватися формулою:

$$w_j^{(\eta,i)} = \begin{cases} 1, x_j \in G^i, \eta = 1, i = 1, 2, \dots, V, j = 1, 2, \dots, N, \\ 0, x_j \notin G^i, \eta = 1, i = 1, 2, \dots, V, j = 1, 2, \dots, N. \end{cases}$$

Як дискримінантну функцію нейрони першого шару для згортки \min будуть використовувати: $\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \max\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}$.

Як функцію активації для згортки \min нейрони першого шару будуть використовувати: $\psi^{(\eta,i)} = \min_i\{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}$.

При використанні згортки \min вагові коефіцієнти нейронів першого шару будуть задаватися формулою:

$$w_j^{(\eta,i)} = \begin{cases} 0, & x_j \in G^i, \eta=1, i=1, 2, \dots, V, j=1, 2, \dots, N, \\ 1, & x_j \notin G^i, \eta=1, i=1, 2, \dots, V, j=1, 2, \dots, N. \end{cases}$$

Параметри нейронів інших шарів мережі будуть визначатися правилами для відповідних типів нейро-нечітких мереж.

При вирішенні задач розпізнавання для випадку трьох і більше класів доцільною може виявитися побудова не єдиної моделі з виходом, що приймає три і більше значення, а синтез окремих моделей для визначення належностей до кожного класу. Це дозволить формувати набори груп ознак своїх для кожного класу.

Наприклад, у задачах розпізнавання зображень може виявитися, що одна і та ж ознака може відноситися до різних груп ознак для різних класів. Побудова окремих моделей для кожного класу дозволить зберегти внутрикласове семантичне навантаження ознак, у той час, як побудова єдиної моделі для всіх класів може привести до часткової втрати семантики ознак.

Для побудови окремих моделей для кожного класу необхідно у вихідній вибірці даних замінити значення цільової ознаки на «1», якщо екземпляр належить до класу, для визначення якого будується модель, і на «0» – у протилежному випадку. Після чого можна застосувати для побудови моделі по кожному класу той самий метод, що і для виходу, який приймає багато значень.

1.11 Методи синтезу нейромережних і нейро-нечітких розпізнавальних моделей з лінеаризацією, факторним групуванням та згорткою ознак

У багатьох практичних задачах екземпляри характеризуються наборами ознак, що містять взаємопов'язані ознаки. Виділення груп взаємопов'язаних ознак є вельми важливим завданням, оскільки дозволяє поліпшити розуміння предметної області, а також спростити модель, що синтезується, за рахунок об'єднання (згортки) взаємопов'язаних ознак. З іншого боку дуже актуальним завданням є лінеаризація взаємопов'язаних ознак, що дозволяє спростувати процес побудови моделі. Тому доцільним є створення методів, що дозволять синтезувати нейронні та нейро-нечіткі моделі в неітеративному режимі з лінеаризацією, факторним групуванням та згорткою ознак.

1.11.1 Постановка задачі синтезу нейронних та нейро-нечітких моделей

Нехай ми маємо навчальну вибірку $\langle x, y \rangle$, що складається з множини екземплярів $\{x^s\} = \{x_j^s\}$, $s = 1, 2, \dots, S, j = 1, 2, \dots, N$, де S – кількість екземплярів у вибірці, N – кількість ознак, що характеризують екземпляри, x_j^s – значення j -ої ознаки, що характеризує s -ий екземпляр вибірки x^s , і набору значень цільової (вихідної) ознаки $y = \{y^s\}$, зіставлених кожному екземпляру вибірки, де y^s – значення цільової ознаки для екземпляра x^s .

Тоді задача синтезу моделі залежності $y(x)$ за вибіркою $\langle x, y \rangle$ на основі нейронної (нейро-нечіткої) мережі полягає у визначенні кількості нейроелементів у мережі, задаванні їхніх дискримінантних й активаційних функцій, визначенні топології мережі і розрахунку вагових коефіцієнтів міжнейронних зв'язків.

1.11.2 Аналіз інформації про ознаки і топологію навчальної вибірки

Для синтезу логічно прозорих нейромоделей у неітеративному режимі доцільно проводити аналіз інформації про ознаки і топологію навчальної вибірки, для чого пропонується виконувати наступну послідовність кроків.

Крок 1. Задати навчальну вибірку $\langle x, y \rangle$. Задати межове значення r_m , $0 < r_m < 1$. Розрахувати логарифми значень ознак для всіх екземплярів і розширити набір ознак $x: x_{j+i}^s = \ln(x_j^s), j = 1, 2, \dots, N, i = 1, 2, \dots, N, s = 1, 2, \dots, S$.

Крок 2. Знайти мінімальні $\min(x_i)$ і максимальні $\max(x_i)$ значення ознак. Пронормувати значення ознак для $i = 1, 2, \dots, 2N, s = 1, 2, \dots, S$:

$$x_i^s = \frac{x_i^s - \min(x_i)}{\max(x_i) - \min(x_i)}.$$

Крок 3. Знайти коефіцієнти, що характеризують тісноту зв'язку ознак і їхніх логарифмів, $r_{ij}, i, j = 1, 2, \dots, 2N$. Наприклад, коефіцієнти парної кореляції:

$$r_{ij} = \frac{\sum_{s=1}^S (x_i^s - \bar{x}_i)(x_j^s - \bar{x}_j)}{\sqrt{\sum_{s=1}^S (x_i^s - \bar{x}_i)^2 \sum_{s=1}^S (x_j^s - \bar{x}_j)^2}}, \quad \bar{x}_i = \frac{1}{S} \sum_{s=1}^S x_i^s.$$

Крок 4. Згрупувати ознаки.

Крок 4.1 Установити кількість груп ознак: $G = 0$.

Крок 4.2 Знайти в матриці $r = \{r_{ij}\}$ елемент із максимальним за модулем значенням: $r_{\max} = \max(|r_{ij}|), i = 1, 2, \dots, 2N, j = i+1, \dots, 2N$. Визначити знак r_{ij} .

Крок 4.3 Якщо $r_{\max} \leq r_m$, тоді перейти до кроку 4.5.

Крок 4.4 Якщо $r_{\max} > r_m$, тоді виконати кроки 4.4.1–4.4.4.

Крок 4.4.1 Установити: $G = G + 1, h(G, i) = 1, h(G, j) = 1$.

Крок 4.4.2 Для всіх $k = 1, 2, \dots, 2N$, $k \neq i$, $k \neq i+N$, $k \neq i-N$, $k \neq j$, $k \neq j+N$, $k \neq j-N$: якщо $|r_{ik}| > r_m$ та $|r_{jk}| > r_m$, тоді установити: $h(G, k)=1$, $r_{kp} = 0$, $r_{pk} = 0$, $p = 1, 2, \dots, 2N$.

Крок 4.4.3 Установити: $r_{ip} = 0$, $r_{pi} = 0$, $r_{jp} = 0$, $r_{pj} = 0$, $p = 1, 2, \dots, 2N$.

Крок 4.4.4. Перейти до кроку 4.2.

Крок 4.5 Якщо $\exists p$, $r_{pj} \neq 0$, $r_{jp} \neq 0$, $p = 1, 2, \dots, 2N$, $j = 1, 2, \dots, N$, $j \neq p$, тоді прийняти: $G=G+1$, $h(G, j)=1$, $r_{jk} = 0$, $r_{kj} = 0$, $k = 1, 2, \dots, 2N$, перейти до кроку 4.5, у протилежному випадку – перейти до кроку 4.6.

Крок 4.6 Установити $h(g, i)=0$, для всіх $g = 1, 2, \dots, G$, $i = 1, 2, \dots, 2N$, для який значення $h(g, i)$ не було встановлено.

Крок 4.7 Для $\forall j, j = 1, 2, \dots, N$, установити:

$$g_i = \begin{cases} 0, i = j, \exists g : h(g, j) = 1, j \leq N; \\ 1, i = j - N, \exists g : h(g, j) = 1, j > N. \end{cases}$$

Крок 4.8 Для $j = N+1, \dots, 2N$, $g = 1, 2, \dots, G$, установити: $h(g, j-N)=1$, $h(g, j) = 0$.

Крок 5. Знайти коефіцієнти I_j , що характеризують тісноту зв'язку ознак і номера класу, такі, що $0 \leq I_j \leq 1$, і зі збільшенням значення I_j зростає значимість ознаки для визначення класу. Для цього можна використовувати методи, розглянуті в [5].

Крок 6. Виконати кластер-аналіз навчальної вибірки: розбити її на кластери C^q , $q = 1, 2, \dots, Q$, де Q – кількість кластерів, визначити координати центрів кластерів C^q , $j = 1, 2, \dots, N$, $q = 1, 2, \dots, Q$. Кластер-аналіз можна виконати, використовуючи методи [2, 12]. Визначити чітку приналежність кластерів до кожного з K класів.

Крок 7. Розбити вісь значень кожної ознаки на Q інтервалів (термів) і визначити їхні параметри для синтезу нейронних та нейро-нечітких мереж.

Крок 7.1 Установити номер поточної ознаки: $i = 1$.

Крок 7.2 Якщо $i > N$, тоді перейти до кроку 7.6, у протилежному випадку – перейти до кроку 7.3.

Крок 7.3 Установити номер поточного інтервалу значень поточної ознаки $q=1$.

Крок 7.4 Якщо $q > Q$, тоді перейти до кроку 7.5, у протилежному випадку – установити для поточного q :

$$v = \arg \min_{s=1,2,\dots,S} \{x_i^s | q^s = q\}, \quad u = \arg \max_{s=1,2,\dots,S} \{x_i^s | q^s = q\},$$

$$K(i, q) = y^v, \quad Q(i, q) = q^v, \quad a(i, q) = x_i^v, \quad b(i, q) = x_i^u, \quad c(i, q) = C_i^{q^v}.$$

Прийняти: $q = q + 1$. Перейти до кроку 7.4.

Крок 7.5 Прийняти: $i = i + 1$. Перейти на кроку 7.2.

Крок 7.6 На основі параметрів $K(i, q)$ і $Q(i, q)$ визначити $K(j)$ та $Q(j)$ – номери класу і кластера для j -го терму, $j = 1, 2, \dots, NQ$:

$$K(j) = K(1 + ((j-1) \operatorname{div} Q, ((j+1) \bmod Q) - 1)),$$

$$Q(j) = Q(1 + ((j-1) \operatorname{div} Q, ((j+1) \bmod Q) - 1)).$$

Крок 8. Задати функції належності розпізнаваного екземпляра до нечітких термів. Для цього пропонується використовувати трапецієподібні функції:

$$\mu_{i,q}(x_i) = \begin{cases} 0, & x_i \leq 0,5(a(i, q) + b(i, q - 1)), \\ \frac{x_i - 0,5(a(i, q) + b(i, q - 1))}{0,5(a(i, q) - b(i, q - 1))}, & 0,5(a(i, q) + b(i, q - 1)) \leq x_i < a(i, q), \\ 1, & a(i, q) \leq x_i \leq b(i, q), \\ \frac{0,5(a(i, q + 1) + b(i, q)) - x_i}{0,5(a(i, q + 1) - b(i, q))}, & b(i, q) \leq x_i < 0,5(b(i, q) + a(i, q + 1)), \\ 0, & 0,5(b(i, q) + a(i, q + 1)) \leq x_i, \end{cases}$$

або Π -образні функції: $\mu_{i,q}(x_i) = \mu_{i,q_S}(x_i)\mu_{i,q_Z}(x_i)$, де $\mu_{i,q}(x_i)$ – функція належності розпізнаваного екземпляра за ознакою x_i до q -го терму i -ої ознаки, $\mu_{i,q_S}(x_i)$ – S -образна функція, а $\mu_{i,q_Z}(x_i)$ – Z -образна функція:

$$\mu_{i,q_S}(x_i) = \begin{cases} 0, & x_i < 0,5(a(i, q) + b(i, q - 1)), \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x_i - a(i, q)}{0,5(a(i, q) - b(i, q - 1))} \pi\right), & 0,5(a(i, q) + b(i, q - 1)) \leq x_i \leq a(i, q), \\ 1, & x_i > a(i, q); \end{cases}$$

$$\mu_{i,q_Z}(x_i) = \begin{cases} 1, & x_i < b(i, q), \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x_i - b(i, q)}{0,5(a(i, q + 1) - b(i, q))} \pi\right), & b(i, q) \leq x_i \leq 0,5(b(i, q) + a(i, q + 1)), \\ 0, & x_i > 0,5(b(i, q) + a(i, q + 1)), \end{cases}$$

або трикутні функції:

$$\mu_{i,q}(x_i) = \begin{cases} 0, & x_i < a(i, q); \\ 0, & x_i > b(i, q); \\ \frac{x_i - a(i, q)}{c(i, q) - a(i, q)}, & a(i, q) \leq x_i \leq c(i, q); \\ \frac{x_i - b(i, q)}{c(i, q) - b(i, q)}, & c(i, q) < x_i \leq b(i, q), \end{cases}$$

або функції Гауса: $\mu_{i,q}(x_i) = \exp\left(-\frac{(x_i - c(i, q))^2}{\sigma^2}\right)$.

1.11.3 Синтез нейронної мережі

На основі результатів проведеного аналізу інформації про ознаки за навчальною вибіркою може бути синтезована нейронна мережа прямого поширення (рис. 1.11).

Нейрони першого шару мережі здійснюють нормування вхідних сигналів, відображаючи їх у діапазон $[0, 1]$, а також виконують, де це необхідно, логарифмування нормованого сигналу.

Нейрони другого шару групують перетворені нейронами першого шару сигнали і знаходять зважену суму сигналів групи з урахуванням оцінок їхнього індивідуального впливу на вихідну ознаку (номер класу).

Нейрони третього шару мережі визначають відстані від розпізнаваного екземпляра до центрів кластерів у просторі згрупованих сигналів і знаходять значення функції Гауса, аргументом якої є визначена відстань.

Нейрони четвертого шару відповідають класам і видають на виході значення «1», якщо розпізнаваний екземпляр відноситься до відповідного класу, і «0» – у протилежному випадку.

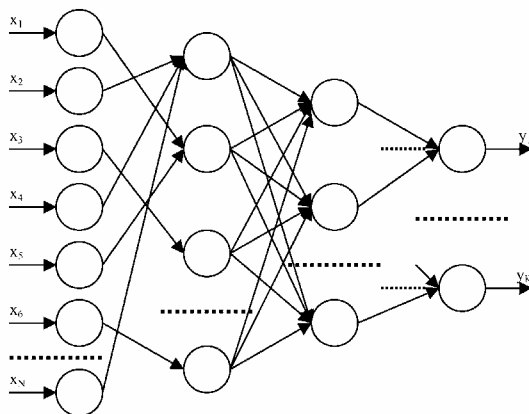


Рисунок 1.11 – Схема нейронної мережі

Дискримінантні функції нейронів мережі будуть задаватися формулами:

$$\varphi^{(1,i)}(w^{(1,i)}, x^{(1,i)}) = w_0^{(1,i)} + w_1^{(1,i)} x_1^{(1,i)}, \quad i = 1, 2, \dots, N;$$

$$\varphi^{(2,i)}(w^{(2,i)}, x^{(2,i)}) = w_0^{(1,i)} + \sum_{j=1}^N w_j^{(1,i)} x_j^{(1,i)}, \quad i = 1, 2, \dots, G;$$

$$\varphi^{(3,i)}(w^{(3,i)}, x^{(3,i)}) = \sum_{j=1}^G (w_j^{(3,i)} - x_j^{(3,i)})^2, \quad i = 1, 2, \dots, Q;$$

$$\varphi^{(4,i)}(w^{(4,i)}, x^{(4,i)}) = w_0^{(4,i)} + \sum_{j=1}^Q w_j^{(4,i)} x_j^{(4,i)}, \quad i = 1, 2, \dots, K,$$

де $\varphi^{(\eta,i)}(w^{(\eta,i)}, x^{(\eta,i)})$ – дискримінантна функція i -го нейрона η -го шару, $w^{(\eta,i)}$ – набір вагових коефіцієнтів i -го нейрона η -го шару, $w^{(\eta,i)} = \{w_j^{(\eta,i)}\}$, $w_j^{(\eta,i)}$ – ваговий коефіцієнт j -го входу i -го нейрона η -го шару, $x^{(\eta,i)}$ – набір входів i -го нейрона η -го шару, $x^{(\eta,i)} = \{x_j^{(\eta,i)}\}$, $x^{(\eta,i)}$ – значення на j -му вході i -го нейрона η -го шару нейромережі.

Функції активації нейронів будуть задаватися формулами:

$$\psi^{(1,i)}(x) = \begin{cases} x, & g_i = 0; \\ \ln(x), & g_i = 1, \end{cases} \quad i = 1, 2, \dots, N;$$

$$\psi^{(2,i)}(x) = x, \quad i = 1, 2, \dots, G;$$

$$\psi^{(3,i)}(x^2) = e^{-x^2}, \quad i = 1, 2, \dots, Q;$$

$$\psi^{(4,i)}(x) = \begin{cases} 0, & x \leq 0; \\ 1, & x > 0, \end{cases} \quad i = 1, 2, \dots, K,$$

де $\psi^{(\eta,i)}(x)$ – функція активації i -го нейрона η -го шару нейромережі.

Вагові коефіцієнти нейронів мережі пропонується розраховувати за формулою:

$$w_j^{(\eta,i)} = \begin{cases} 1 + \min(x_p)(\max(x_p) - \min(x_p))^{-1}, & p = i + N, g_i = 1, \eta = 1, i = 1, 2, \dots, N, j = 0; \\ \min(x_i)(\max(x_i) - \min(x_i))^{-1}, & g_i = 0, \eta = 1, i = 1, 2, \dots, N, j = 0; \\ (\max(x_p) - \min(x_p))^{-1}, & p = i + N, g_i = 1, \eta = 1, i = 1, 2, \dots, N, j = 1; \\ (\max(x_i) - \min(x_i))^{-1}, & g_i = 0, \eta = 1, i = 1, 2, \dots, N, j = 1; \\ 0, & \eta = 1, i = 1, 2, \dots, N, j = 0; \\ I_j h(i, j) z(i), & g_j = 0, \eta = 2, i = 1, 2, \dots, G, j = 1, 2, \dots, N; \\ (\ln 2)^{-1} I_j h(i, j) z(i), & g_j = 1, \eta = 2, i = 1, 2, \dots, G, j = 1, 2, \dots, N; \\ C_j^i, & \eta = 3, i = 1, 2, \dots, Q, j = 1, 2, \dots, G; \\ 0, & \eta = 4, i = 1, 2, \dots, K, j = 0; \\ 1, & C^j \in K_i, \eta = 4, i = 1, 2, \dots, K, j = 1, 2, \dots, Q; \\ -1, & C^j \notin K_i, \eta = 4, i = 1, 2, \dots, K, j = 1, 2, \dots, Q; \end{cases}$$

$$\text{де } z(i) = \left(\sum_{k=1}^N I_k h(i, k) \right)^{-1}.$$

1.11.4 Синтез нейро-нечітких мереж

Поряд із синтезом чітких нейромереж прямого поширення інформація, отримана в результаті аналізу інформації про ознаки, може бути використана для синтезу шестишарової (рис. 1.12) і п'ятишарової (рис. 1.13) нейро-нечітких мереж.

Перший і другий шари обох нейро-нечітких мереж будуть еквівалентні відповідним шарам вище описаної нейронної мережі.

Нейрони третього шару обох мереж здійснюють обчислення значень функцій належності розпізнаваного екземпляра до нечітких термів ознак.

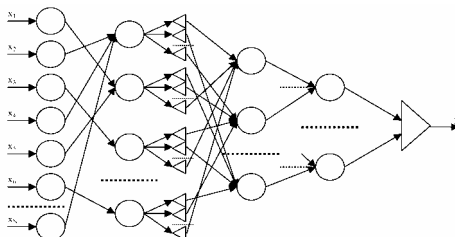


Рисунок 1.12 – Схема шестишарової нейро-нечіткої мережі

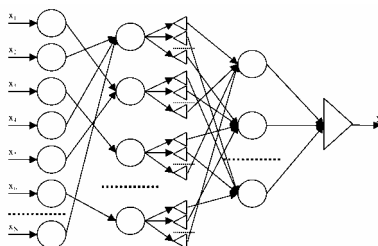


Рисунок 1.13 – Схема п'ятишарової нейро-нечіткої мережі

Нейрони четвертого шару шестишарової мережі здійснюють об'єднання нечітких термів у кластери, а нейрони п'ятого шару поєднують належності екземпляра до кластерів у належності до класів.

Нейрони четвертого шару п'ятишарової мережі здійснюють безпосереднє об'єднання нечітких термів у класи без кластерного групування.

Єдиний нейрон останнього шару обох мереж здійснює об'єднання належностей до класів і дефазифікацію результату.

Дискримінантні функції нейронів шестишарової нейро-нечіткої мережі будуть задаватися формулами:

$$\varphi_j^{(4,i)}(w_j^{(4,i)}, x_j^{(4,i)}) = \min\{w_j^{(4,i)}, x_j^{(4,i)}\} \quad i = 1, 2, \dots, Q, j = 1, 2, \dots, N_3;$$

$$\begin{aligned}\varphi_j^{(5,i)}(w_j^{(5,i)}, x_j^{(5,i)}) &= \min\{w_j^{(5,i)}, x_j^{(5,i)}\}, \quad i = 1, 2, \dots, K, j = 1, 2, \dots, Q; \\ \varphi_j^{(6,1)}(w_j^{(6,1)}, x_j^{(6,1)}) &= \min\{w_j^{(6,1)}, x_j^{(6,1)}\}, \quad j = 1, 2, \dots, K,\end{aligned}$$

де N_η – кількість нейронів у η -му шарі мережі.

Функції активації нейронів шестишарової нейро-нечіткої мережі будуть задаватися формулами:

$$\begin{aligned}\psi^{(4,i)}(\varphi_j^{(4,i)}) &= \max\{\varphi_j^{(4,i)}\}, \quad i = 1, 2, \dots, Q, j = 1, 2, \dots, N_3; \\ \psi^{(5,i)}(\varphi_j^{(5,i)}) &= \max\{\varphi_j^{(5,i)}\}, \quad i = 1, 2, \dots, K, j = 1, 2, \dots, Q; \\ \psi^{(6,1)}(\varphi_j^{(6,1)}) &= \arg \max\{\varphi_j^{(6,1)}\}, \quad j = 1, 2, \dots, K.\end{aligned}$$

Вагові коефіцієнти нейронів шестишарової нейро-нечіткої мережі будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} 1, Q(j) = i, \eta = 4, i = 1, 2, \dots, Q, j = 1, 2, \dots, N_3; \\ 0, Q(j) \neq i, \eta = 4, i = 1, 2, \dots, Q, j = 1, 2, \dots, N_3; \\ 1, C^j \in i, \eta = 5, i = 1, 2, \dots, K, j = 1, 2, \dots, Q; \\ 0, C^j \notin i, \eta = 5, i = 1, 2, \dots, K, j = 1, 2, \dots, Q; \\ 0, \eta = 6, i = 1, j = 1, 2, \dots, K.\end{cases}$$

Дискримінантні функції нейронів п'ятишарової нейро-нечіткої мережі будуть задаватися формулами:

$$\begin{aligned}\varphi_j^{(4,i)}(w_j^{(4,i)}, x_j^{(4,i)}) &= \min\{w_j^{(4,i)}, x_j^{(4,i)}\}, \quad i = 1, 2, \dots, K, j = 1, 2, \dots, N_3; \\ \varphi_j^{(5,1)}(w_j^{(5,1)}, x_j^{(5,1)}) &= \min\{w_j^{(5,1)}, x_j^{(5,1)}\}, \quad j = 1, 2, \dots, K,\end{aligned}$$

де N_η – кількість нейронів у η -му шарі мережі.

Функції активації нейронів шестишарової мережі будуть задаватися формулами:

$$\begin{aligned}\psi^{(4,i)}(\varphi_j^{(4,i)}) &= \max\{\varphi_j^{(4,i)}\}, \quad i = 1, 2, \dots, K, j = 1, 2, \dots, N_3; \\ \psi^{(5,1)}(\varphi_j^{(5,1)}) &= \arg \max\{\varphi_j^{(5,1)}\}, \quad j = 1, 2, \dots, K.\end{aligned}$$

Вагові коефіцієнти нейронів шестишарової нейро-нечіткої мережі будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} 1, K(j) = i, \eta = 4, i = 1, 2, \dots, K, j = 1, 2, \dots, N_3; \\ 0, K(j) \neq i, \eta = 4, i = 1, 2, \dots, K, j = 1, 2, \dots, N_3; \\ 0, \eta = 5, i = 1, j = 1, 2, \dots, K.\end{cases}$$

РОЗДІЛ 2

СИНТЕЗ ІЄРАРХІЧНИХ ЛОГІЧНО ПРОЗОРИХ НЕЙРО-НЕЧІТКИХ МЕРЕЖ

У задачах технічної та біомедичної діагностики, а також при побудові систем підтримки прийняття рішень в економіці й управлінні виникає необхідність розуміння того, яким чином приймається те чи інше рішення.

Класичні моделі нейро-нечітких мереж характеризуються високою складністю для аналізу і сприйняття людиною, оскільки нейроелементи схованих шарів цих мереж містять велику кількість входів, а кількість зв'язків між нейронами також є великою. Тому виникає необхідність у розробці методу, що дозволяє синтезувати нейро-нечіткі мережі, що мають властивість логічної прозорості, тобто є зручними для аналізу і сприйняття людиною.

Логічну прозорість нейро-нечітких мереж можливо забезпечити шляхом побудови ієрархії й узагальнення правил прийняття рішень.

2.1 Метод синтезу ієрархічних логічно прозорих нейро-нечітких мереж

Для синтезу ієрархічної нейро-нечіткої моделі на основі вихідної навчальної вибірки необхідно для кожного класу (або інтервалу значень цільової змінної – для задач з дійсним виходом) сформувати набір правил, до умов і висновків яких входять номери інтервалів значень ознак. Це дозволить сформувати нечіткі терми і базу знань для побудови нейро-нечіткої мережі.

Для отриманого набору правил кожного класу варто сформувати ієрархію прийняття рішення про віднесення розпізнаваного екземпляра до даного класу. Це пропонується здійснювати шляхом групування й об'єднання правил, що відрізняються тільки значенням терму, окремо для кожної ознаки в порядку збільшення інформативності ознак.

Метод синтезу ієрархічних логічно прозорих нейро-нечітких мереж, що реалізує дані розуміння, полягає у виконанні послідовності таких кроків [41, 42].

Крок 1. Ініціалізація. Задати навчальну вибірку $\langle x, y \rangle$, $x = \{x^s\}$, $x^s = \{x_j^s\}$, $y = \{y^s\}$, $s = 1, 2, \dots, S$, $j = 1, 2, \dots, N$, де N – кількість ознак, S – кількість екземплярів навчальної вибірки, x_j^s – значення j -ої ознаки s -го екземпляра, y^s – номер класу, зіставлений s -му екземпляру навчальної вибірки. Для задач з дійсним виходом номер класу визначається як номер інтервалу значень вихідної змінної, до якого відноситься s -ий екземпляр.

Крок 2. Визначити значення характеристик навчальної вибірки і сформувати структури даних.

Крок 2.1 На основі навчальної вибірки за допомогою методу 1.1 знайти $\{A(j,k)\}$, $\{B(j,k)\}$, $\{K(j,k)\}$, $\{I_j\}$. де k – номер інтервалу значень j -ої ознаки, $k = 1, 2, \dots, k_j$; k_j – кількість інтервалів, на які розбивається діапазон значень j -ої ознаки.

Крок 2.2 Визначити загальну кількість інтервалів для всіх ознак навчальної вибірки:

$$Z = \sum_{j=1}^N k_j.$$

Крок 2.3 Сформувати масив вузлів $\{u(p)\}$, $p = 1, 2, \dots, Z$, яким послідовно зіставити інтервали значень ознак у порядку збільшення номера ознаки і номера інтервалу значень ознаки. Додати вузол $u(Z+1)$ для об'єднання нечітких термів вихідної змінної. Установити загальну кількість вузлів: $z = Z+1$.

Крок 2.4 Сформувати матрицю зв'язків між вузлами $\{v(i,j)\}$, де $v(i,j)=0$, якщо зв'язок між i -им та j -им вузлами відсутній, $v(i,j) = 1$, якщо є зв'язок, спрямований від i -го вузла до j -го вузла. Установити: $v(i,j) = 0$, $i, j=1, 2, \dots, z$.

Крок 2.5 Сформувати масив типів вузлів $\{t(j)\}$, де $t(j) = 0$, якщо j -ий вузол має тип «ТА»; $t(j) = 1$, якщо j -ий вузол має тип «АБО»; $t(j) = -1$, якщо j -ий вузол є вхідним нечітким термом або вихідним вузлом мережі. Задати типи вузлів: $t(j)=-1, j=1, 2, \dots, z$.

Крок 2.6 Сформувати масив номерів рівнів ієрархії для вузлів $\{h(j)\}$, де $h(j)$ – номер рівня ієрархії для j -го вузла. Задати рівні ієрархії для вузлів: $h(j)=1, j=1, 2, \dots, z-1$; $h(z)=0$.

Крок 2.7 Сформувати нечіткі терми – задати функції належності для інтервалів значень ознак $\mu_{i,k}(x_i)$, де i – номер ознаки, k – номер інтервалу значень i -ої ознаки. Як функції належності пропонується використовувати трапецієподібні функції або П-образні функції.

Крок 3. Установити поточний номер класу $q = 1$.

Крок 4. Якщо $q > K$, де K – кількість класів, на які поділяються екземпляри навчальної вибірки, тоді перейти до кроку 16.

Крок 5. На основі навчальної вибірки і параметрів, визначених за допомогою методу 1.1, сформувати матрицю правил r . Установити номер поточного рівня ієрархії: $h = 2$.

Крок 5.1 Установити: $s = 1, s_t = 1$.

Крок 5.2 Якщо $s > S$, тоді перейти до кроку 6.

Крок 5.3 Якщо $y^s = q$, тоді для $\forall j=1, 2, \dots, N$, занести у комірку $r(s_t, j)$ номер інтервалу, у який потрапило значення j -ої ознаки s -го екземпляра вибірки, установити: $s_t = s_t + 1$. Номера інтервалів значень ознак запису-

ються в суцільній нумерації по всіх ознаках у порядку зростання номера ознаки і номера інтервалу значень усередині ознаки.

Крок 5.4 Установити: $s = s + 1$. Перейти до кроку 5.2

Крок 6. Для кожного стовпця матриці r знайти $n(j)$ – кількість інтервалів j -ої ознаки, що належать до q -го класу, $j = 1, 2, \dots, N$. Упорядкувати стовпці матриці r у порядку зростання $n(j)$.

Крок 7. Установити: $j = N$, $s_i = s_i - 1$.

Крок 8. Якщо $j < 2$, тоді перейти до кроку 12.

Крок 9. Для j -ої ознаки знайти в r та узагальнити правила з однаковими частинами лівіше j -го стовпця.

Крок 9.1 Установити: $s = 1$.

Крок 9.2 Якщо $s > s_i$, тоді перейти до кроку 10.

Крок 9.3 Для інтервалу $r(s, j)$ знайти всі інтервали тієї ж j -ої ознаки, що мають однакові з ним ліві частини правил – рядків матриці r . Занести номера вузлів для цих інтервалів у вектор $\kappa = \{\kappa_i\}$, де κ_i – i -ий елемент вектора κ .

Крок 9.4 Визначити ℓ – довжину вектора κ . Якщо $\ell > 0$, тоді: додати новий вузол $u(z+1)$; прийняти: $z = z + 1$, $h(z) = h + 1$; установити для вузла z тип «АБО»: $t(z) = 1$; додати зв'язок: $v(\kappa_i, z) = 1$; для $p = 1, 2, \dots, s_i$: якщо $r(p, j) \in \kappa$, тоді установити: $r(p, j) = z$.

Крок 9.5 Залишити в матриці r з кожної групи однакових правил тільки одне правило. Зкорегувати відповідним чином s_i .

Крок 9.6 Установити: $s = s + 1$. Перейти до кроку 9.2.

Крок 10. Для $s = 1, 2, \dots, s_i$: додати новий вузол $u(z+1)$; прийняти: $z = z + 1$, $h(z) = h + 2$; установити для вузла z тип «ТА»: $t(z) = 0$; додати зв'язок: $v(r(s, j), z) = 1$, $v(r(s, j-1), z) = 1$; установити: $r(s, j-1) = z$.

Крок 11. Установити: $h = h + 2$, $j = j - 1$. Перейти до кроку 8.

Крок 12. Залишити в матриці r з кожної групи однакових правил тільки одне правило. Зкорегувати відповідним чином s_i .

Крок 13. Якщо $s_i > 1$, тоді: додати новий вузол $u(z+1)$; прийняти: $z = z + 1$, $h(z) = h + 1$; установити для вузла z тип «АБО»: $t(z) = 0$; додати зв'язок: $v(s, z) = 1$; для $s = 1, 2, \dots, s_i$; установити: $r(1, 1) = z$; видалити з r усі правила, крім першого.

Крок 14. Установити зв'язок: $v(r(1, 1), Z+q) = 1$.

Крок 15. Установити: $q = q + 1$. Перейти до кроку 4.

Крок 16. Установити:

$$h(Z+1) = 1 + \max_{i=1,2,\dots,z} h(i).$$

Для кожного рівня ієрархії $\eta = 1, 2, \dots, h(Z+1)$, визначити N_η – кількість вузлів, що знаходяться на рівні ієрархії η .

Крок 17. Зупинення.

2.2 Модель ієрархічної логічно прозорої нейро-нечіткої мережі

Схему ієрархічної логічно прозорої нейро-нечіткої мережі, синтезованої на основі запропонованого методу, подано на рис. 2.1.

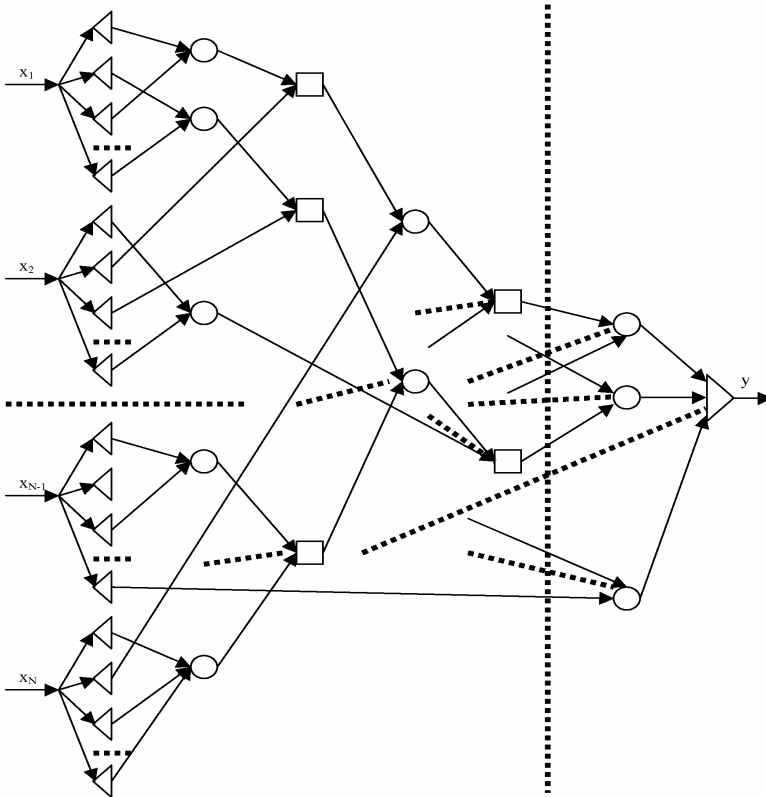


Рисунок 2.1 – Схема ієрархічної нейро-нечіткої мережі

На входи мережі подаються чіткі значення ознак розпізнаваного екземпляра. Перший шар мережі містить вузли, що визначають належності значень ознак до нечітких термів – інтервалів значень ознак. Сховані шари мережі реалізують ієрархічне виведення рішення про приналежність до нечітких термів вихідної змінної. Парні сховані шари (на рис. 2.1 позначені окружностями) містять нейрони типу «АБО», непарні сховані шари (на рис. 2.1 позначені квадратами) містять нейрони типу «ТА». Останній

шар мережі містить один нейрон, що виконує об'єднання нечітких значень належностей розпізнаваного екземпляра до класів і приведення результату до чіткого значення.

Дискримінантні функції нейронів будуть визначатися у такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, 4, \dots, h(Z+1)-1;$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \max\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=3, 5, \dots, h(Z+1)-2;$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \frac{\sum_{j=1}^K w_j^{(\eta,i)} x_j^{(\eta,i)}}{\sum_{j=1}^K x_j^{(\eta,i)}}, \quad \eta = h(Z+1), i = 1.$$

Функції активації нейронів мережі будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=2, 4, \dots, h(Z+1)-1;$$

$$\psi^{(\eta,i)}(x) = \min_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=3, 5, \dots, h(Z+1)-2;$$

для задач класифікації: $\psi^{(\eta,i)}(x) = \text{round}(x)$, $\eta = h(Z+1)$, $i = 1$;

для задач оцінювання: $\psi^{(\eta,i)}(x) = x$, $\eta = h(Z+1)$, $i = 1$.

Вагові коефіцієнти нейронів $w_j^{(\eta,i)}$, де j – номер входу, i – номер нейрона, η – номер шару, будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} j, \eta = h(Z+1), i = 1, j = 1, 2, \dots, K; \\ v(j, i), \eta = 2, 4, \dots, h(Z+1)-1, i = 1, 2, \dots, N_\eta, j = 1, 2, \dots, N_{\eta-1}; \\ 1 - v(j, i), \eta = 3, 5, \dots, h(Z+1)-2, i = 1, 2, \dots, N_\eta, j = 1, 2, \dots, N_{\eta-1}. \end{cases}$$

Мережа, синтезована на основі наведених вище формул, буде реалізувати виведення на основі ієрархічної max-min композиції.

2.3 Модифікації ієрархічної логічно прозорої нейро-нечіткої мережі

Дану мережу можна також модифікувати для реалізації виведення на основі ієрархічної max-max композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть, як і в попередньому випадку, містити нейрони типу «АБО», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити також нейрони типу «АБО».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, 3, \dots, h(Z+1)-1,$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=2, 3, \dots, h(Z+1)-1.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою: $w_j^{(\eta,i)} = v(j,i), \eta=2, 3, \dots, h(Z+1)-1, i=1, 2, \dots, N_\eta, j=1, 2, \dots, N_{\eta-1}$.

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної max-prod композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть, як і в попередньому випадку, містити нейрони типу «АБО», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «PROD».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, 4, \dots, h(Z+1)-1;$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \max\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=3, 5, \dots, h(Z+1)-2;$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=2, 4, \dots, h(Z+1)-1;$$

$$\psi^{(\eta,i)}(x) = \prod_j \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta=3, 5, \dots, h(Z+1)-2.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} v(j,i), \eta=2, 4, \dots, h(Z+1)-1, i=1, 2, \dots, N_\eta, j=1, 2, \dots, N_{\eta-1}, \\ 1-v(j,i), \eta=3, 5, \dots, h(Z+1)-2, i=1, 2, \dots, N_\eta, j=1, 2, \dots, N_{\eta-1}. \end{cases}$$

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної max-average композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть, як і в попередньому випадку, містити нейрони типу «АБО», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «AVERAGE».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, 3, \dots, h(Z+1)-1,$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=2, 4, \dots, h(Z+1)-1;$$

$$\psi^{(\eta,i)}(x) = \frac{1}{N_{\eta-1}} \sum_{j=1}^{N_{\eta-1}} \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta=3, 5, \dots, h(Z+1)-2.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою: $w_j^{(\eta,i)} = v(j,i), \eta = 2, 3, \dots, h(Z+1)-1, i = 1, 2, \dots, N_\eta, j = 1, 2, \dots, N_{\eta-1}$.

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної average-max композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть містити нейрони типу «AVERAGE», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «АБО».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min \{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta = 2, 3, \dots, h(Z+1)-1,$$

а функції активації будуть визначатися за формулами:

$$\Psi^{(\eta,i)}(x) = \frac{1}{N_{\eta-1}} \sum_{j=1}^{N_{\eta-1}} \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta = 2, 4, \dots, h(Z+1)-1;$$

$$\Psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta = 3, 5, \dots, h(Z+1)-2.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою: $w_j^{(\eta,i)} = v(j,i), \eta = 2, 3, \dots, h(Z+1)-1, i = 1, 2, \dots, N_\eta, j = 1, 2, \dots, N_{\eta-1}$.

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної average-min композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть містити нейрони типу «AVERAGE», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «ТА».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min \{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta = 2, 4, \dots, h(Z+1)-1,$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \max \{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta = 3, 5, \dots, h(Z+1)-2;$$

а функції активації будуть визначатися за формулами:

$$\Psi^{(\eta,i)}(x) = \frac{1}{N_{\eta-1}} \sum_{j=1}^{N_{\eta-1}} \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta = 2, 4, \dots, h(Z+1)-1;$$

$$\Psi^{(\eta,i)}(x) = \min_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta = 3, 5, \dots, h(Z+1)-2.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися як:

$$w_j^{(\eta,i)} = \begin{cases} v(j,i), \eta = 2, 4, \dots, h(Z+1)-1, i = 1, 2, \dots, N_\eta, j = 1, 2, \dots, N_{\eta-1}, \\ 1 - v(j,i), \eta = 3, 5, \dots, h(Z+1)-2, i = 1, 2, \dots, N_\eta, j = 1, 2, \dots, N_{\eta-1}. \end{cases}$$

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної average-prod композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть міс-

тити нейрони типу «AVERAGE», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «PROD».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\begin{aligned}\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) &= \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, 4, \dots, h(Z+1)-1, \\ \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) &= \max\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=3, 5, \dots, h(Z+1)-2;\end{aligned}$$

а функції активації будуть визначатися за формулами:

$$\begin{aligned}\psi^{(\eta,i)}(x) &= \frac{1}{N_{\eta-1}} \sum_{j=1}^{N_{\eta-1}} \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta=2, 4, \dots, h(Z+1)-1; \\ \psi^{(\eta,i)}(x) &= \prod_j \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta=3, 5, \dots, h(Z+1)-2.\end{aligned}$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} v(j, i), \eta=2, 4, \dots, h(Z+1)-1, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}, \\ 1-v(j, i), \eta=3, 5, \dots, h(Z+1)-2, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}. \end{cases}$$

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної average-average композиції. Парні (на рис. 2.1 позначені окружностями) і непарні (на рис. 2.1 позначені квадратами) сховані шари будуть містити нейрони типу «AVERAGE».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, 3, \dots, h(Z+1)-1,$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \frac{1}{N_{\eta-1}} \sum_{j=1}^{N_{\eta-1}} \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta=2, 3, \dots, h(Z+1)-1.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою: $w_j^{(\eta,i)} = v(j, i), \eta=2, 3, \dots, h(Z+1)-1, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}$.

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної sum-prod композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть містити нейрони типу «SUM», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «PROD».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(x) = \sum_{j=1}^{N_{\eta-1}} w_j^{(\eta,i)} x_j^{(\eta,i)}, \quad \eta=2, 4, \dots, h(Z+1)-1,$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \max \{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=3, 5, \dots, h(Z+1)-2;$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \frac{1}{1+e^{-x}}, \quad \eta=2, 4, \dots, h(Z+1)-1,$$

$$\psi^{(\eta,i)}(x) = \prod_j \varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}), \quad \eta=3, 5, \dots, h(Z+1)-2.$$

Ваги нейронів схованих шарів будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} v(j,i), \eta=2, 4, \dots, h(Z+1)-1, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}, \\ 1-v(j,i), \eta=3, 5, \dots, h(Z+1)-2, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}. \end{cases}$$

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної max-sum композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть містити нейрони типу «АБО», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «SUM».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=2, 4, \dots, h(Z+1)-1;$$

$$\varphi^{(\eta,i)}(x) = \sum_{j=1}^{N_{\eta-1}} w_j^{(\eta,i)} x_j^{(\eta,i)}, \quad \eta=3, 5, \dots, h(Z+1)-2;$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=2, 4, \dots, h(Z+1)-1,$$

$$\psi^{(\eta,i)}(x) = \frac{1}{1+e^{-x}}, \quad \eta=3, 5, \dots, h(Z+1)-2.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою: $w_j^{(\eta,i)} = v(j,i), \eta=2, 3, \dots, h(Z+1)-1, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}$.

За допомогою запропонованого методу також можна синтезувати мережу, що реалізує виведення на основі ієрархічної sum-max композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть містити нейрони типу «SUM», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «АБО».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(x) = \sum_{j=1}^{N_{\eta-1}} w_j^{(\eta,i)} x_j^{(\eta,i)}, \quad \eta=2, 4, \dots, h(Z+1)-1;$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \min\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=3, 5, \dots, h(Z+1)-2;$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \frac{1}{1+e^{-x}}, \quad \eta=2, 4, \dots, h(Z+1)-1,$$

$$\psi^{(\eta,i)}(x) = \max_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=3, 5, \dots, h(Z+1)-2.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою: $w_j^{(\eta,i)} = v(j,i), \eta=2, 3, \dots, h(Z+1)-1, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}$.

За допомогою запропонованого методу також можна синтезувати мережа, що реалізує виведення на основі ієрархічної sum-min композиції. Парні сховані шари (на рис. 2.1 позначені окружностями) будуть містити нейрони типу «SUM», а непарні сховані шари (на рис. 2.1 позначені квадратами) будуть містити нейрони типу «ГЛ».

Параметри нейронів першого й останнього шарів будуть визначатися також як і раніше. Для нейронів схованих шарів дискримінантні функції будуть визначатися в такий спосіб:

$$\varphi^{(\eta,i)}(x) = \sum_{j=1}^{N_{\eta-1}} w_j^{(\eta,i)} x_j^{(\eta,i)}, \quad \eta=2, 4, \dots, h(Z+1)-1,$$

$$\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)}) = \max\{w_j^{(\eta,i)}, x_j^{(\eta,i)}\}, \quad \eta=3, 5, \dots, h(Z+1)-2,$$

а функції активації будуть визначатися за формулами:

$$\psi^{(\eta,i)}(x) = \frac{1}{1+e^{-x}}, \quad \eta=2, 4, \dots, h(Z+1)-1,$$

$$\psi^{(\eta,i)}(x) = \min_j \{\varphi^{(\eta,i)}(w_j^{(\eta,i)}, x_j^{(\eta,i)})\}, \quad \eta=3, 5, \dots, h(Z+1)-2.$$

Вагові коефіцієнти нейронів схованих шарів будуть визначатися за формулою:

$$w_j^{(\eta,i)} = \begin{cases} v(j,i), \eta=2, 4, \dots, h(Z+1)-1, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}, \\ 1-v(j,i), \eta=3, 5, \dots, h(Z+1)-2, i=1, 2, \dots, N_{\eta}, j=1, 2, \dots, N_{\eta-1}. \end{cases}$$

Запропонований метод синтезу нейро-нечітких мереж узагальнює класичні методи нечіткого виведення і дозволяє будувати логічно прозорі нейро-нечіткі мережі, що є зручними не тільки для аналізу і сприйняття, але також мають добрі узагальнюючі властивості та легко реалізуються апаратно за рахунок спрощення структури обробляючих елементів і скорочення кількості зв'язків між нейроелементами.

РОЗДІЛ 3

СИНТЕЗ НЕЙРО-НЕЧІТКИХ АПРОКСИМАТОРІВ

Поряд із вирішенням завдань автоматичної класифікації за ознаками нейро-нечіткі мережі знаходять використання при вирішенні завдань апроксимації кількісних залежностей. Проте використання класичних методів синтезу нейро-нечітких апроксиматорів кількісних залежностей вимагає вирішення задачі багатовимірної нелінійної оптимізації функціоналу помилки, що на практиці досягається використанням градієнтних методів пошуку на основі техніки зворотного поширення помилки. Такі методи є високо ітеративними і дуже повільними, залежать від вибору початкової точки пошуку та не завжди забезпечують прийнятне вирішення задачі.

Тому є дуже актуальною розробка методів, що дозволять синтезувати нейро-нечіткі апроксиматори залежностей без використання ітеративного корегування ваг та обчислення похідних функції помилки.

3.1 Об'єднання нечітких термів у кластери

Нехай ми маємо вибірку даних $\langle x, y \rangle$. Кількість класів (можливих значень цільової змінної) позначимо як K . Кожен клас може містити один або декілька кластерів. Необхідно знайти таке розбиття навчальної вибірки на кластери, що буде у певному розумінні оптимальним [36].

Крок 1. Ініціалізація. Задати навчальну вибірку $\langle x, y \rangle$, $x = \{x^s\}$, $x_s = \{x^s_j\}$, $y = \{y^s\}$, де j – номер ознаки ($j = 1, 2, \dots, N$), N – кількість ознак, S – кількість екземплярів навчальної вибірки, x^s_j – значення j -ої ознаки для s -го екземпляра. Прийняти: $Q = S$, де Q – кількість кластерів.

Крок 2. За допомогою методу 1.1 знайти $A(j,k)$, $B(j,k)$, $K(j,k)$, I_j , де k – номер інтервалу значень j -ої ознаки, $k=1,2,\dots,k_j$; k_j – кількість інтервалів, на які розбивається діапазон значень j -ої ознаки. Визначити $n(j,k,p)$ – кількість екземплярів навчальної вибірки, що потрапили в k -ий інтервал значень j -ої ознаки та належать до p -го класу. Сформувати функції належності для нечітких термів (інтервалів значень ознак) $\mu_{j,k}$, де j – номер ознаки, k – номер інтервалу значень j -ої ознаки. Як функції належності пропонується використовувати трапецієподібні функції або П-образні функції.

Крок 3. Для вибірки x сформувати кластери $\{C^q\}$, $C^q_j = k$, де q – номер кластера, j – номер ознаки q -го кластера, k – номер інтервалу значень j -ої ознаки, у який потрапив екземпляр x^q . Кожному кластеру C^q зіставити номер класу y_{q^j} , до якого він належить.

Крок 4. Визначити відстані між кластерами:

$$R(M, L) = R(L, M) = \sum_{j=1}^N |C_j^M - C_j^L|, M = 1, 2, \dots, Q; L = 1, 2, \dots, Q.$$

Крок 5. Для кластерів кожного класу виконувати об'єднання поки не задовольняється критерій закінчення пошуку.

Крок 5.1 Установити: $q = 1$.

Крок 5.2 Якщо $q > K$, тоді перейти до кроку 6.

Крок 5.3 Установити: $M = 1$.

Крок 5.4 Якщо $M > Q$, тоді перейти до кроку 5.10.

Крок 5.5 Установити: $L = M + 1$.

Крок 5.6 Якщо $L > Q$, тоді перейти до кроку 5.9.

Крок 5.7 Якщо $y_M = y_L = q$ та $R(M, L) \leq R(\alpha, \beta)$ та $y_\alpha = y_\beta = q$, $\alpha = 1, \dots, Q$, $\beta = 1, \dots, Q$, $\alpha \neq \beta$; $\alpha \neq M$ при $\beta = L$; $\alpha \neq L$ при $\beta = M$; $\beta \neq M$ при $\alpha = L$; $\beta \neq L$ при $\alpha = M$, тоді виконати кроки 5.7.1-5.7.4; у протилежному випадку – перейти до кроку 5.8.

Крок 5.7.1 Об'єднати кластери C^M та C^L :

$$C_j^M = \begin{cases} C_j^M, C_j^M = C_j^L, \\ C_j^M, C_j^M \neq C_j^L, n(j, C_j^M, q) \geq n(j, C_j^L, q), \\ C_j^L, C_j^M \neq C_j^L, n(j, C_j^M, q) < n(j, C_j^L, q). \end{cases}$$

Прийняти для $v = L, \dots, Q-1$: $C^v = C^{v+1}$, $R(v, u) = R(v+1, u)$, $R(u, v) = R(u, v+1)$, $u = 1, 2, \dots, Q$. Установити: $Q = Q-1$. Для $L = 1, 2, \dots, Q$, знайти:

$$R(M, L) = R(L, M) = \sum_{j=1}^N |C_j^M - C_j^L|.$$

Крок 5.7.2 Для поточного розбиття на кластери знайти чіткий номер класу y^{s*} , $s = 1, 2, \dots, S$.

Крок 5.7.2.1 Знайти значення функцій належності розпізнаваного екземпляра x^s до нечітких термів (інтервалів значень ознак) $\mu_{j,k}(x^s)$.

Крок 5.7.2.2 Знайти значення функцій належності розпізнаваного екземпляра x^s до нечітких термів q -го кластера, p -го класу:

$$\mu_{j,k}^{p,q}(x^s) = \begin{cases} \mu_{j,k}(x^s), K(j,k) = y_q = p, \\ 0, \neg(K(j,k) = y_q = p); \end{cases}$$

$$p = 1, 2, \dots, K; q = 1, 2, \dots, Q; j = 1, 2, \dots, N; k = 1, 2, \dots, k_j.$$

Крок 5.7.2.3 Знайти значення функцій належності розпізнаваного екземпляра x^s за j -ою ознакою до q -го кластера, p -го класу:

$$\mu_j^{p,q}(x^s) = \max_{k=1,2,\dots,k_j} \mu_{j,k}^{p,q}(x^s),$$

$$p = 1, 2, \dots, K; q = 1, 2, \dots, Q; j = 1, 2, \dots, N.$$

Крок 5.7.2.4 Знайти значення функцій належності розпізнаваного екземпляра x^s до q -го кластера, p -го класу:

$$\mu^{p,q}(x^s) = \frac{1}{N} \sum_{j=1}^N \mu_j^{p,q}(x^s) \quad \text{або} \quad \mu^{p,q}(x^s) = \max_{j=1, \dots, N} \mu_j^{p,q}(x^s),$$

$$p = 1, 2, \dots, K; q = 1, 2, \dots, Q.$$

Крок 5.7.2.5 Знайти значення функцій належності розпізнаваного екземпляра x^s до p -го класу:

$$\mu_p(x^s) = \max_{q=1, 2, \dots, Q} \mu^{p,q}(x^s), p = 1, 2, \dots, K.$$

Крок 5.7.2.6 Визначити: $y^{s*} = \arg \max_{p=1, \dots, K} \mu_p(x^s)$.

Крок 5.7.3 Розрахувати помилку для поточного розбиття екземплярів навчальної вибірки:

$$E = \sum_{s=1}^S |y^s - y^{s*}|.$$

Крок 5.7.4 Якщо $E \leq E_{max}$, тоді перейти до кроку 5.5, у протилежному випадку – скасувати об'єднання кластерів C^M та C^L , установити: $R(M,L)=R(L,M)=\text{RealMax}$, де RealMax – максимальне для розрядної сітки ЕОМ число.

Крок 5.8 Прийняти: $L=L+1$. Перейти до кроку 5.6.

Крок 5.9 Прийняти: $M=M+1$. Перейти до кроку 5.4.

Крок 5.10 Прийняти: $q = q + 1$. Перейти до кроку 5.2

Крок 6. Зупинення.

У результаті виконання кроків 1–6 для навчальної вибірки отримаємо розбиття на Q кластерів C^q .

3.2 Нейро-нечітка кластер-регресійна апроксимація

У випадку, коли вихідна змінна при побудові моделі залежності є дійсною, пропонується використовувати кластер-регресійну апроксимацію, побудову якої можна здійснювати шляхом виконання таких кроків [36, 40].

Крок 1. Ініціалізація. Задати навчальну вибірку $\langle x, y \rangle$, де $x = \{x^s\}$, $y = \{y^s\}$, $x^s = \{x_j^s\}$.

Крок 2. Дискретизувати значення цільової змінної у одним зі способів.

Спосіб 1.

Крок 2.1 Задати L – кількість інтервалів, на які розбивається діапазон значень цільової змінної y . Знайти $\min(y^s)$ та $\max(y^s)$ – мінімальне і максимальне значення цільової змінної y , відповідно.

Крок 2.2 Визначити ширину вікна дискретизації цільової змінної:

$$\Delta y = (\max(y^s) - \min(y^s)) / L.$$

Крок 2.3 Прийняти: $y^s = y^s \text{ div } \Delta y$, $s = 1, 2, \dots, S$, де $a \text{ div } b$ – операція цілочисленого ділення a на b .

Спосіб 2.

Крок 2.1 Задати δy – припустиму погрішність. Знайти $\min(y^s)$ та $\max(y^s)$ – мінімальне і максимальне значення цільової змінної y , відповідно.

Крок 2.2 Відсортувати екземпляри навчальної вибірки в порядку неспадання значень цільової змінної y .

Крок 2.3 Установити: $k = 0$, $s = 2$, $y = \min(y)$, $y^1 = 0$.

Крок 2.4 Якщо $s > S$, тоді перейти до кроку 3.

Крок 2.5 Якщо $|y^s - y^L| \leq \delta y$, тоді установити: $y^s = k$, у протилежному випадку – установити: $k = k + 1$, $y^s = k$.

Крок 2.6 Установити: $s = s + 1$. Перейти до кроку 2.4.

Крок 3. Для вибірки з дискретизованою цільовою змінною за допомогою методу 1.1 знайти $A(j,k)$, $B(j,k)$, $K(j,k)$.

Крок 4. Видалити малоінформативні ознаки і надлишкові інтервали вхідних змінних. Задати функції належності екземплярів до нечітких термів (інтервалів значень ознак) $\mu_{j,k}$, де j – номер ознаки, k – номер інтервалу значень j -ої ознаки. Як функції належності пропонується використовувати трапецієподібні функції або П-образні функції.

Крок 5. Сформувати кластери – компактні області зосередження екземплярів вибірки з однаковим номером класу шляхом задавання правил виду: (q) Якщо ... та $A(j,k) \leq x_j \leq B(j,k)$ та ..., то $K(q) = K(j,k)$, де q – номер правила (кластера), $K(q)$ – номер класу, зіставлений q -му кластеру ($q = 1, 2, \dots, Q$), Q – кількість кластерів:

$$Q = \frac{1}{2} \sum_{j=1}^N k_j \left(\sum_{\substack{p=1, \\ p \neq j}}^N k_p \right).$$

Крок 6. Задати функції належності екземплярів до кластерів:

$$\mu^q = \max_{\substack{j=1,2,\dots,N; \\ k=1,2,\dots,k_j}} w_{j,k}^q \mu_{j,k},$$

$$w_{j,k}^q = \begin{cases} 1, & K(q) = K(j,k); \\ 0, & K(q) \neq K(j,k), \end{cases}$$

$$q = 1, 2, \dots, Q.$$

Крок 7. Для екземплярів кожного кластера побудувати часткові моделі залежностей $y_q(x^s)$, використовуючи вихідні недискретизовані значення цільової змінної:

$$y_q(x^s) = \beta_0^q + \sum_{j=1}^N \beta_j^q x_j^s,$$

де β_j^q – коефіцієнт багатомірної лінійної регресійної моделі при j -й ознаці. При побудові часткових моделей можливо враховувати інформативність ознак, а також сам процес побудови моделей здійснювати шляхом послідовного нарощування числа ознак, що дозволить спростити часткові моделі [18, 34, 44].

Крок 8. Визначити метод селекції часткової моделі для кластера й одержання чіткого розрахункового значення цільової змінної:

$$y^{s*} = \sum_{q=1}^Q \alpha_q \mu^q(x^s) y_q(x^s),$$

де α_q – деякий ваговий коефіцієнт (у найпростішому випадку $\alpha_q = 1$).

Даний метод може мати нейро-нечітку інтерпретацію у вигляді чотиришарової нейро-нечіткої мережі (рис. 3.1).

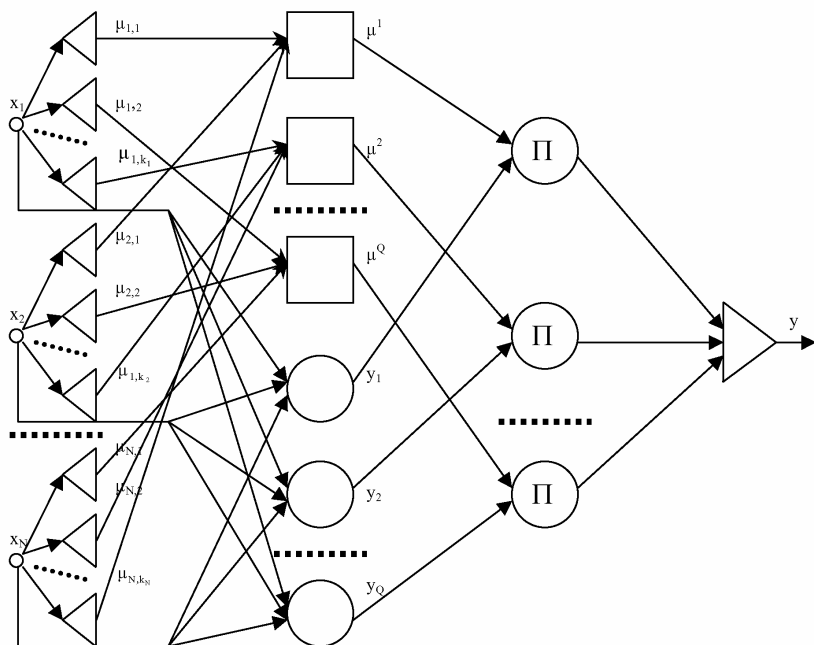


Рисунок 3.1 – Схема нейро-нечіткої мережі

На входи мережі подаються значення N ознак розпізнаваного екземпляра. Перший шар мережі містить нейрони, що визначають належності розпізнаваного екземпляра до нечітких термів (інтервалів значень ознак). Другий шар мережі містить $2Q$ нейронів: перші Q нейронів визначають належності розпізнаваного екземпляра до кластерів, другі Q нейронів реалізують часткові моделі $y_q(x^s)$. Третій шар містить Q нейронів, що виконують селекцію часткових моделей для кластерів. Четвертий (вихідний)

шар мережі містить один нейрон, що виконує об'єднання часткових моделей і дефаззифікацію цільової змінної.

Функції постсинаптичного потенціалу нейронів мережі задаються формулами:

$$\varphi^{(2,i)}(w, x) = \min_j (w_j^{(2,i)}, x_j^{(2,i)}), \quad i = 1, 2, \dots, Q,$$

$$\varphi^{(2,i)}(w^{(2,i)}, x^{(2,i)}) = \sum_{j=1}^N w_j^{(2,i)} x_j^{(2,i)} + w_0^{(2,i)}, \quad i = Q+1, Q+2, \dots, 2Q,$$

$$\varphi^{(3,i)}(w^{(3,i)}, x^{(3,i)}) = \prod_{j=i, 2i} w_j^{(3,i)} x_j^{(3,i)}, \quad i = 1, 2, \dots, Q,$$

$$\varphi^{(4,1)}(w^{(4,1)}, x^{(4,1)}) = \sum_{j=1}^Q w_j^{(4,1)} x_j^{(4,1)} + w_0^{(4,1)}.$$

Функції активації нейронів мережі задаються формулами:

$$\psi^{(2,i)}(x) = \max_j \varphi_j^{(2,i)}(w_j^{(2,i)}, x_j^{(2,i)}), \quad i=1, 2, \dots, Q; \quad \psi^{(2,i)}(x) = x, \quad i=Q+1, Q+2, \dots, 2Q;$$

$$\psi^{(3,i)}(x) = x, \quad i = 1, 2, \dots, Q; \quad \psi^{(4,1)}(x) = x.$$

Вагові коефіцієнти нейронів $w_j^{(\eta,i)}$, де j – номер входу, i – номер нейрона, η – номер шару, визначають за формулою:

$$w_j^{(\eta,i)} = \begin{cases} 1, & \eta = 2, i = 1, 2, \dots, Q, j = 1, 2, \dots, N, \\ \beta_j^q, & \eta = 2, i = Q + q, q = 1, 2, \dots, Q, j = 0, 1, \dots, N, \\ 1, & \eta = 3, i = 1, 2, \dots, Q, j = 1, 2, \dots, 2Q, \\ 0, & \eta = 4, i = 1, j = 0, \\ \alpha_j, & \eta = 4, i = 1, j = 1, 2, \dots, Q. \end{cases}$$

Після синтезу нейро-нечіткої моделі для підвищення точності можна здійснити підбір вагових коефіцієнтів єдиного нейрона останнього шару мережі, використовуючи для цього методи багатовимірної нелінійної безумовної оптимізації [14].

3.3 Метод нейро-нечіткої класифікації за узагальненою віссю

При синтезі нечітких і нейро-нечітких класифікаторів однією з найбільш складних є проблема формування нечітких термів і визначення їхніх параметрів. Для вирішення цієї проблеми використовують або експертний підхід (терми описують люди-експерти) [5, 22], або навчання (терми описують елементарними базисними функціями [22, 29], параметри яких,

як правило, налагоджують ітеративно на основі методів багатомірної оптимізації [10, 14, 24] або на основі апріорної інформації про навчальну вибірку в неітеративному режимі [36–43]).

Основним недоліком експертного підходу є його залежність від наявності та компетенції людини-експерта, що на практиці у визначених областях може взагалі бути відсутньою чи бути недостатньо компетентною.

При навчанні на основі методів багатомірної оптимізації виникає ряд проблем, таких, як вибір початкової точки пошуку в просторі керованих змінних, локальний характер пошуку (можливість потрапляння методів оптимізації до локальних екстремумів), низька швидкість навчання внаслідок ітеративності пошуку.

Процедури настроювання нечітких термів на основі використання апріорної інформації про навчальну вибірку, як правило, орієнтовані на об'єднання результатів окремих класифікацій за кожною ознакою з урахуванням їхніх індивідуальних значимостей. При цьому, у випадку складного поділу образів, можливі ситуації, коли міжкласові межі будуть погано апроксимуватися, а використовуваний набір ознак виявиться надлишковим, що може призвести до ускладнення і зниження точності моделі.

Тому актуальною є розробка методу, що дозволяє автоматично формувати нечіткі терми і налагоджувати їхні параметри, а також синтезувати нейро-нечіткі моделі якісних залежностей за прецедентами, вільні від недоліків відомих методів.

3.3.1 Постановка задачі

Нехай задані спостереження за об'єктом (прецеденти) у вигляді кортежу $\langle x, y \rangle$, що складається з множини екземплярів $x = \{x^s\}$, $x^s = \{x_j^s\}$, $s=1, 2, \dots, S$, $j=1, 2, \dots, N$, де S – кількість екземплярів у вибірці, N – кількість ознак, що характеризують екземпляри, x_j^s – значення j -ої ознаки, що характеризує s -ий екземпляр вибірки x^s , і набору значень цільової (вихідної) ознаки $y = \{y^s\}$, зіставлених кожному екземпляру вибірки, де y^s – значення цільової ознаки для екземпляра x^s .

Тоді задача синтезу моделі залежності $y(x)$ за навчальною вибіркою $\langle x, y \rangle$ на основі нейро-нечіткої класифікувальної мережі полягає у формуванні нечіткого розбиття простору ознак на терми, визначенні кількості нейроелементів у мережі, завданні їхніх дискримінантних (постсинаптичних, вагових) й активаційних функцій, визначенні топології мережі та розрахунку вагових коефіцієнтів міжнейронних зв'язків.

3.3.2. Формування узагальненої вісі

Розглянемо проекцію навчальної вибірки $\langle x, y \rangle$ на площину, утворену i -ою та j -ою ознаками. Визначимо для кожної j -ої ознаки ($j = 1, 2, \dots, N$) її мінімальне x_j^{\min} та максимальне x_j^{\max} значення. На основі знайдених значень обчислимо: $\Delta_j = (x_j^{\max} - x_j^{\min})^{-1}$ та $\alpha_j = x_j^{\min} \Delta_j$.

Тоді для комбінації двох ознак x_i та x_j можна задати перетворення на одновимірну узагальнену вісь:

$$x_{(i,j)}^s(I) = d_{(i,j)}^s + \frac{\sin \theta_{(i,j)}^s}{d_{(i,j)}^{\min}},$$

$$d_{(i,j)}^s = \sqrt{(x_i \Delta_i - \alpha_i)^2 + (x_j \Delta_j - \alpha_j)^2}, \quad d_{(i,j)}^{\min} = \min_{s=1,2,\dots,S} d_{(i,j)}^s, \quad \sin \theta_{(i,j)}^s = \frac{x_j^s \Delta_j - \alpha_j}{d_{(i,j)}^s},$$

або альтернативне йому перетворення:

$$x_{(i,j)}^{s(II)} = \frac{d_{(i,j)}^s}{\Delta_{(i,j)}^{\min}} + \frac{x_j^s \Delta_j - \alpha_j}{d_{(i,j)}^s}, \quad \Delta_{(i,j)}^{\min} = \min_{s=1,2,\dots,S} \frac{x_j^s \Delta_j - \alpha_j}{d_{(i,j)}^s}.$$

Привівши доданки до загального знаменника і, забезпечуючи умову нерівності нулю знаменника, одержимо вирази:

$$x_{(i,j)}^s(I) = \mathfrak{K}_1(x_i^s, x_j^s) = \frac{(d_{(i,j)}^s)^2 d_{(i,j)}^{\min} + (x_j^s \Delta_j - \alpha_j)}{1 + d_{(i,j)}^s d_{(i,j)}^{\min}},$$

$$x_{(i,j)}^{s(II)} = \mathfrak{K}_2(x_i^s, x_j^s) = \frac{(d_{(i,j)}^s)^2 + \Delta_{(i,j)}^{\min} (x_j^s \Delta_j - \alpha_j)}{1 + \Delta_{(i,j)}^{\min} d_{(i,j)}^s},$$

де \mathfrak{K}_z – оператор, що задає z -те перетворення.

3.3.3 Визначення параметрів розбиття за узагальненою ознакою

Розбиття навчальної множини $\langle x_s, y \rangle$, $x_s = \{x_s^s\}$, $y = \{y^s\}$, $s = 1, 2, \dots, S$, за ознакою x_s на класи вимагає визначення лівих L_k і правих R_k меж для інтервалів значень ознаки x_s , що відносяться до різних класів, де k – номер інтервалу ознаки x_s . Для кожного k -го інтервалу потрібно визначити K_k – номер класу, до якого відносяться екземпляри, що потрапили усередину k -го інтервалу ознаки.

Розбиття діапазону значень ознаки x_s на інтервали класів пропонується здійснювати шляхом виконання послідовності кроків 1–7.

Крок 1. Задати вибірку $\langle x_*, y \rangle$, а також кількість класів K , до яких можуть належати екземпляри вибірки.

Крок 2. Упорядкувати екземпляри вибірки $\langle x_*, y \rangle$ за неубуванням значення ознаки x_* .

Крок 3. Установити: $k = 1$, $L_k = x_*^1$, $R_k = x_*^1$, $K_k = y^1$, $D(x_*) = 1$, $s = 2$.

Крок 4. Якщо $s \leq S$, тоді перейти до кроку 5, у протилежному випадку – перейти до кроку 7.

Крок 5. Якщо $y^s = y^{s-1}$, тоді прийняти: $R_k = x_*^s$, $s = s+1$, перейти до кроку 4.

Крок 6. Якщо $y^s \neq y^{s-1}$, тоді прийняти: $k = k+1$, $L_k = x_*^s$, $R_k = x_*^s$, $K_k = y^s$, $D(x_*) = D(x_*) + 1$, $s = s+1$, перейти до кроку 4.

Крок 7. Зупинення.

У результаті виконання кроків 1–7 для навчальної вибірки $\langle x_*, y \rangle$ одержимо розбиття вісі x_* на діапазони значень $\langle L_k, R_k \rangle$, у кожному з яких за ознакою x_* екземпляри належать тільки до одного класу K_k . При цьому також одержимо значення $D(x_*)$ – кількість інтервалів різних класів, на які розбивається діапазон значень ознаки x_* .

Якість отриманого раніше перетворення вибірки x на узагальнену вісь x_* можна оцінити величиною $D(x_*)$, $D(x_*) \geq K$. Чим менше значення D , тим меншою кількістю різних кластерів (інтервалів) на осі x_* поданий кожен клас, і, відповідно, тим вище ефективність перетворення.

3.3.4. Формування ієрархії узагальнення ознак

Оскільки введені раніше перетворення ознак на узагальнену вісь припускають об'єднання щоразу тільки двох ознак, у той час, як у переважній більшості практичних випадків опис вибірки є багатовимірним з числом ознак більше двох, є доцільною розробка методу, що дозволяє формувати ієрархію об'єднання і перетворення ознак на узагальнену вісь. При цьому важливо виключити дублювання однієї й тієї ж ознаки при формуванні узагальненої вісі, прагнути мінімізувати кількість використовуваних ознак для спрощення розпізнавальної моделі, поєднувати тільки ті ознаки, поділ за узагальненою віссю для яких дозволяє одержати краще розбиття, ніж для кожної з ознак окремо.

Формування ієрархії узагальнення ознак з урахуванням вище викладених розумінь пропонується здійснювати шляхом виконання таких кроків.

Крок 0. Ініціалізація. Задати навчальну вибірку $\langle x, y \rangle$. Задати набір перетворень на узагальнену вісь. Установити номер поточного рівня $k = 1$.

Установити номер поточної ознаки $p = 1$. Прийняти: $x_{(k-1,i)} = x_i$, $i = 1, 2, \dots, N$, де $x_{(k,i)}$ – значення i -ої ознаки k -го рівня. Установити $N_{(k-1)} = N$, де $N_{(k)}$ – кількість ознак у k -му рівні.

Прямий прохід (знизу – нагору).

Крок 1. Визначити $D(x_{(k-1,i)})$ – кількість інтервалів різних класів, на які розбивається діапазон значень ознаки $x_{(k-1,i)}$, $i = 1, 2, \dots, N$.

Крок 2. Якщо $\exists i$, $i = 1, 2, \dots, N_{(k-1)}$: $D(x_{(k-1,i)}) = K$, тоді перейти до кроку 15, у протилежному випадку – перейти до кроку 3.

Крок 3. Установити: $i = 1$, $p = 1$.

Крок 4. Якщо $i > N_{(k-1)} - 1$, тоді перейти до кроку 12, у протилежному випадку – перейти до кроку 5.

Крок 5. Установити: $j = i + 1$.

Крок 6. Якщо $j > N_{(k-1)} - 1$, тоді перейти до кроку 12, у протилежному випадку – перейти до кроку 5.

Крок 7. Для i -ої та j -ої ознак визначити $d_{(i,j)}^{\min}$, $\Delta_{(i,j)}^{\min}$, Δ_i , Δ_j , α_i , α_j .

Установити: $d_{(k,i,j)}^{\min} = d_{(i,j)}^{\min}$, $\Delta_{(k,i,j)}^{\min} = \Delta_{(i,j)}^{\min}$, $\Delta_{(p,i)} = \Delta_i$, $\Delta_{(p,j)} = \Delta_j$, $\alpha_{(p,i)} = \alpha_i$, $\alpha_{(p,j)} = \alpha_j$. Сформулювати кандидатів до узагальненої ознаки на основі i -ої та j -ої ознак: $x_{(k,p)z} = \aleph_z(x_{(k-1,i)}, x_{(k-1,j)})$, $z = 1, 2, \dots, Z$, де \aleph_z – оператор, що поєднує ознаки $x_{(k-1,i)}$ та $x_{(k-1,j)}$ в узагальнену ознаку на основі z -го перетворення, Z – загальна кількість перетворень у заданому наборі перетворень.

Крок 8. Зробити відбір серед кандидатів перетворень в узагальнену ознаку.

Крок 8.1 Для $\forall z = 1, 2, \dots, Z$: знайти $D(x_{(k,p)z})$ – кількість інтервалів різних класів, на які розбивається діапазон значень ознаки $x_{(k,p)z}$.

Крок 8.2 Визначити номер найкращого перетворення для p -ої ознаки k -го рівня:

$$\tau_{(k,p)} = \arg \min_{z=1, 2, \dots, Z} D(x_{(k,p)z}).$$

Крок 8.3 Прийняти: $x_{(k,p)} = x_{(k,p)\tau_{(k,p)}}$, $D(x_{(k,p)}) = D(x_{(k,p)\tau_{(k,p)}})$.

Крок 9. Якщо $D(x_{(k,p)}) \leq \min(D(x_{(k-1,i)}), D(x_{(k-1,j)}))$, тоді узагальнена ознака k -го рівня не погіршує поділяючі властивості в порівнянні з одновимірними розбиттями за її складовими ознаками k -го рівня, прийняти: $f(k,p)_1 = i$, $f(k,p)_2 = j$, $p = p + 1$.

Крок 10. Установити: $j = j + 1$. Перейти до кроку 6.

Крок 11. Установити: $i = i + 1$. Перейти до кроку 4.

Крок 12. Для $\forall i = 1, 2, \dots, N_{(k-1)}$: якщо $\neg \exists x_{(k,q)}$: $x_{(k-1,i)} \in x_{(k,q)}$, $q = 1, \dots, p-1$, тоді прийняти: $x_{(k,p)} = x_{(k-1,i)}$, $f(k,p)_1 = i$, $f(k,p)_2 = i$, $\tau_{(k,p)} = 0$, $p = p + 1$. Тут $f(k,p)_1$ – відповідно, індекси першої і другої ознак (блоків попереднього рівня), що утворюють p -ту узагальнену ознаку k -го рівня.

Крок 13. Установити: $N_{(k)} = p-1$. Якщо $N_{(k-1)} = N_{(k)}$ та для $\forall i = 1, 2, \dots, N_{(k)}$: $x_{(k, i)} = x_{(k-1, i)}$, тобто k -ий та $(k-1)$ -ий рівні містять однакові набори ознак, тоді установити: $k = k-1$, перейти до кроку 15.

Крок 14. Установити: $k = k+1$. Перейти до кроку 2.

Зворотний прохід (зверху – вниз).

Крок 15. Установити:

$$I = \arg \min_{j=1,2,\dots,N_{(k)}} D(x_{(k,j)}), \quad i = \arg \min_{\substack{j=1,2,\dots,N_{(k)} \\ j \in I}} n(x_{(k,j)}),$$

де I – множина індексів ознак k -го рівня, що забезпечують найкращий поділ образів, $n(x_{(k,j)})$ – кількість первинних ознак (ознак рівня 0), що входять в узагальнену ознаку $x_{(k,j)}$.

Крок 16. Видалити всі ознаки k -го рівня, крім ознаки $x_{(k,i)}$ – установити: $x_{(k,1)} = x_{(k,i)}$, $f(k,1)_1 = f(k,i)_1$, $f(k,1)_2 = f(k,i)_2$, $\tau_{(k,1)} = \tau_{(k,i)}$, $N_{(k)} = 1$.

Крок 17. Рухаючись від передостаннього рівня до першого, видалити на кожному рівні ознаки, відсутні в ознаках наступного рівня, для чого виконати послідовність кроків 17.1–17.4.

Крок 17.1 Установити: $\ell = k$.

Крок 17.2 Якщо $\ell > 0$, тобто пройдено всі рівні, тоді перейти до кроку 17.3, у протилежному випадку – перейти до кроку 18.

Крок 17.3 Для $\forall j = N_{(\ell-1)}, N_{(\ell-1)}-1, \dots, 1$: якщо $\neg \exists x_{(\ell,q)}: x_{(\ell-1,j)} \in x_{(\ell,q)}$, $q = 1, \dots, N_{(\ell)}$, тоді установити: $x_{(\ell-1,p)} = x_{(\ell-1,p+1)}$, $f(\ell-1,p)_1 = f(\ell-1,p+1)_1$, $f(\ell-1,p)_2 = f(\ell-1,p+1)_2$, $\tau_{(\ell-1,p)} = \tau_{(\ell-1,p+1)}$, $p = j, j+1, \dots, N_{(\ell-1)}$; $N_{(\ell-1)} = N_{(\ell-1)}-1$.

Крок 17.4 Установити: $\ell = \ell-1$. Перейти до кроку 17.2.

Крок 18. Для єдиного блоку останнього k -го рівня, визначити параметри одновимірного розбиття: $D(x_{(k,1)})$ та $\{<L_i, R_i, K_i>\}$, $i = 1, 2, \dots, D(x_{(k,1)})$.

Крок 19. Зупинення.

У результаті виконання послідовності кроків 1–18 одержимо ієрархію об'єднання ознак, що містить тільки ті первинні ознаки (ознаки рівня 0) та їхні узагальнення, які є мінімально необхідними для знаходження одновимірної узагальненої осі, що забезпечує найкраще розбиття екземплярів навчальної вибірки на класи.

3.3.5 Структурно-параметричний синтез нейро-нечітких моделей

Після формування ієрархії перетворення ознак на узагальнену вісь є можливим виконати її реалізацію у вигляді нейро-нечіткої мережі (рис. 3.2). Мережа, зображена на рис. 3.2, складається з двох модулів: перший модуль являє собою реалізацію ієрархії перетворення ознак на узагальнену вісь, виконану в нейробазисі, а другий модуль являє собою нейро-нечіткий класифікатор.

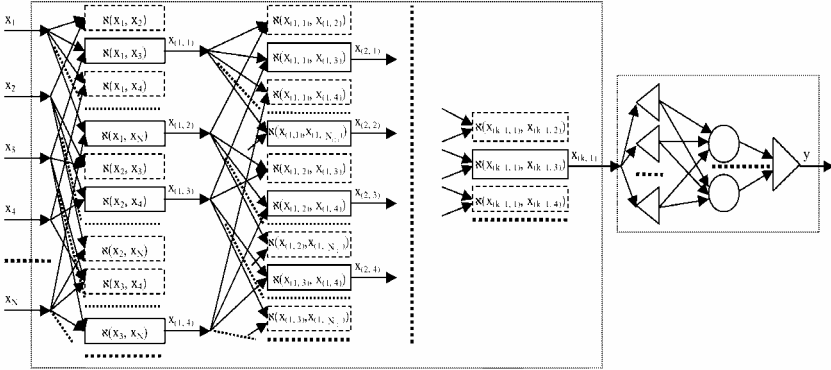


Рисунок 3.2 – Схема синтезу мережі нейро-нечіткої класифікації за узагальноною віссю

Кількість нейронів у шарах мережі визначається за формулою:

$$N_{\mu} = \begin{cases} 2N((\mu + 4) \operatorname{div} 5), \mu = 1, 3, 6, 8, \dots, M - 7, M - 5; \\ N((\mu + 4) \operatorname{div} 5), \mu = 2, 4, 5, 7, 9, 10, \dots, M - 6, M - 4, M - 3; \\ D(x_{(k,1)}), \mu = M - 2; \\ K, \mu = M - 1; \\ 1, \mu = M. \end{cases}$$

де μ – номер шару в нейро-нечіткій мережі, M – кількість шарів у мережі, $M = k+3$, N_{μ} – кількість нейронів у μ -му шарі, $a \operatorname{div} b$ – операція цілочисельного ділення a на b .

Кожен блок першого модуля виконує перетворення ознак попереднього рівня на узагальнену вісь. Границі блоків, що забезпечили поліпшення розбиття навчальної вибірки на класи і ввійшли до мережі, показані суцільною лінією, а границі блоків, що погіршили розбиття і не ввійшли до мережі, показані пунктиром.

Блоки, що відповідають перетворенню (I), пропонується реалізувати у вигляді фрагмента нейромережі прямого поширення, зображеного на рис. 3.3.

Блоки, що відповідають перетворенню (II), пропонується реалізувати у вигляді фрагменту нейромережі прямого поширення, зображеного на рис. 3.4.

Усі нейрони кожного блоку обох перетворень використовують лінійні функції активації: $\psi^{(\mu,i)}(x) = x$, $\mu = 1, 2, \dots, M-3$; $i = 1, \dots, N_{\mu}$, де i – номер нейрона в шарі, $\psi^{(\mu,i)}(x)$ – функція активації i -го нейрона μ -го шару.

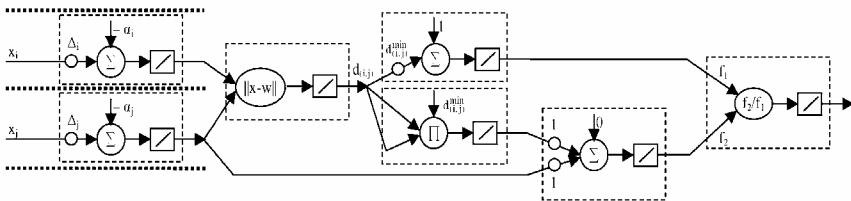


Рисунок 3.3 – Схема нейроблоку відображення двовимірних координат на узагальнену вісь (I)

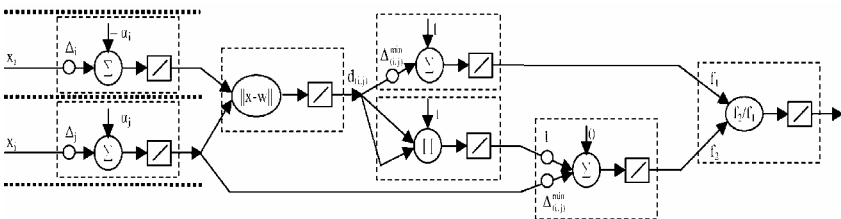


Рисунок 3.4 – Схема нейроблоку відображення двовимірних координат на узагальнену вісь (II)

Усі нейрони першого і четвертого шарів, а також непарні нейрони третього шару кожного блоку обох перетворень як дискримінантні функції використовують зважені суми:

$$\varphi^{(\mu,i)}(w_j^{(\mu,i)}, x_j^{(\mu,i)}) = \sum_{j=1}^{N_{\mu-1}} w_j^{(\mu,i)} x_j^{(\mu,i)} + w_0^{(\mu,i)},$$

$$\mu = 1, 6, \dots, M-7; i = 1, 2, \dots, N_{\mu},$$

$$\mu = 3, 8, \dots, M-5; i = 1, 3, \dots, N_{\mu-1},$$

$$\mu = 4, 9, \dots, M-4; i = 1, 2, \dots, N_{\mu},$$

де $\varphi^{(\mu,i)}(w_j^{(\mu,i)}, x_j^{(\mu,i)})$ – дискримінантна (вагова) функція i -го нейрона μ -го шару, $x_j^{(\mu,i)}$ – значення на j -му вході i -го нейрона μ -го шару, $w_j^{(\mu,i)}$ – ваговий коефіцієнт, зіставлений j -му вході i -го нейрона μ -го шару.

Усі нейрони другого шару кожного блоку обох перетворень використовують як дискримінантну функцію відстані:

$$\varphi^{(\mu,i)}(w_j^{(\mu,i)}, x_j^{(\mu,i)}) = \sqrt{\sum_{j=1}^2 (w_j^{(\mu,i)} - x_j^{(\mu,i)})^2},$$

$$\mu = 2, 7, \dots, M-6; i = 1, 2, \dots, N_{\mu},$$

Усі парні нейрони третього шару кожного блоку обох перетворень як дискримінантні використовують функції добутку:

$$\varphi^{(\mu,i)}(w^{(\mu,i)}, x^{(\mu,i)}) = w_0^{(\mu,i)} \prod_{j=1}^2 w_j^{(\mu,i)} x_j^{(\mu,i)},$$

$$\mu = 3, 8, \dots, M-5; i = 2, 4, \dots, N_\mu.$$

Нейрони п'ятого шару кожного блоку обох перетворень як дискримінантні використовують функції розподілу:

$$\varphi^{(\mu,i)}(x^{(\mu,i)}) = \frac{x_2^{(\mu,i)}}{x_1^{(\mu,i)}},$$

$$\mu = 5, 10, \dots, M-3; i = 1, 2, \dots, N_\mu.$$

Перший шар другого модуля (шар нейронів, що передують передостанньому шару мережі) складається з вузлів, що визначають належності розпізнаваного екземпляра до нечітких термів змінної узагальненої вісі верхнього рівня.

Якщо для p -го блоку ℓ -го рівня $\tau_{(\ell, p)} = 0$, то блок вважається відсутнім, а його вхідна ознака вважається вихідною (з'єднується з виходом блоку).

Як дискримінантні функції вузли першого шару другого модуля використовують трапецієподібні функції:

$$\varphi^{(M-2,i)}(x^{(M-2,i)}) = \begin{cases} 0, & x^{(M-2,i)} < 0,5(L_i + L_{i-1}), \\ \frac{x^{(M-2,i)} - 0,5(L_i + L_{i-1})}{0,5(L_i - L_{i-1})}, & 0,5(L_i + L_{i-1}) \leq x^{(M-2,i)} < L_i, \\ 1, & L_i \leq x^{(M-2,i)} \leq R_i, \\ \frac{0,5(R_{i+1} + R_i) - x^{(M-2,i)}}{0,5(R_{i+1} - R_i)}, & R_i < x^{(M-2,i)} \leq 0,5(R_{i+1} + R_i), \\ 0, & 0,5(R_{i+1} + R_i) < x^{(M-2,i)}. \end{cases}$$

де L_i, R_i – відповідно, ліва і права межі i -го інтервалу значень узагальненої вісі, $i = 1, 2, \dots, D(x_{(k,1)})$, $L_0 = L_1$, $R_{D(x_{(k,1)})+1} = R_{D(x_{(k,1)})}$.

Як функції активації нейрони даного шару використовують лінійні функції:

$$\psi^{(M-2,i)}(x) = x, \quad i = 1, 2, \dots, D(x_{(k,1)}).$$

Другий шар другого модуля (передостанній шар мережі) містить стільки нейронів, скільки класів є у навчальній вибірці. Кожен нейрон передостаннього шару мережі визначає приналежність розпізнаваного екземпляра до нечіткого терму вихідної змінної (до відповідного класу).

Як дискримінантні функції нейрони передостаннього шару мережі використовують функцію:

$$\varphi^{(\mu,i)}(w^{(\mu,i)}, x^{(\mu,i)}) = \min(w_j^{(\mu,i)}, x_j^{(\mu,i)}),$$

а як функції активації:

$$\psi^{(\mu,i)}(\varphi(w^{(\mu,i)}, x^{(\mu,i)})) = \max_j \varphi(w_j^{(\mu,i)}, x^{(\mu,i)}).$$

Єдиний нейрон третього шару другого модуля (останнього, вихідного шару мережі) здійснює дефазифікацію і видає на виході чіткий номер класу, до якого нейро-нечітка мережа віднесла розпізнаваний екземпляр.

Як дискримінантну функцію нейрон вихідного шару використовує :

$$\varphi^{(M,1)}(w^{(M,1)}, x^{(M,1)}) = \arg \max_{j=1,2,\dots,K} w_j^{(M,1)} x_j^{(M,1)}.$$

Як функцію активації – лінійну функцію: $\psi^{(M,1)}(x) = x$.

Ваги входів нейронів мережі пропонується визначати за формулою:

$$w_j^{(\mu,i)} = \begin{cases} -\alpha_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 1, 3, \dots, N_\mu - 1, j = 0, \rho = 0, 2(\mu+4), v = f(\rho, 0, 5(\hat{i}+1))_1; \\ \Delta_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 1, 3, \dots, N_\mu - 1, j = v, \rho = 0, 2(\mu+4), v = f(\rho, 0, 5(\hat{i}+1))_1; \\ 0, \mu = 1, 6, \dots, M-7, i = 1, 3, \dots, N_\mu - 1, j \neq v, j > 0, \rho = 0, 2(\mu+4), v = f(\rho, 0, 5(\hat{i}+1))_1; \\ -\alpha_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 2, 4, \dots, N_\mu, j = 0, \rho = 0, 2(\mu+4), v = f(\rho, 0, 5\hat{i})_2; \\ \Delta_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 2, 4, \dots, N_\mu, j = v, \rho = 0, 2(\mu+4), v = f(\rho, 0, 5\hat{i})_2; \\ 0, \mu = 1, 6, \dots, M-7, i = 2, 4, \dots, N_\mu, j \neq v, j > 0, \rho = 0, 2(\mu+4), v = f(\rho, 0, 5\hat{i})_2; \\ 0, \mu = 2, 7, \dots, M-6, i = 1, 2, \dots, N_\mu, j = 1, 2; \\ 1, \mu = 3, 8, \dots, M-5, i = 1, 2, \dots, N_\mu, j = 0; \\ d_{(\rho,v,\vartheta)}^{\min}, \mu = 3, 8, \dots, M-5, i = 1, 3, \dots, N_\mu - 1, \tau_{(\rho,0,5(\hat{i}+1))} = 1, j = 1, \rho = 0, 2(\mu+1), \\ v = f(\rho, 0, 5(\hat{i}+1))_1, \vartheta = f(\rho, 0, 5(\hat{i}+1))_2; \\ \Delta_{(\rho,v,\vartheta)}^{\min}, \mu = 3, 8, \dots, M-5, i = 1, 3, \dots, N_\mu - 1, \tau_{(\rho,0,5(\hat{i}+1))} = 2, j = 1, \rho = 0, 2(\mu+1), \\ v = f(\rho, 0, 5(\hat{i}+1))_1, \vartheta = f(\rho, 0, 5(\hat{i}+1))_2; \\ d_{(\rho,v,\vartheta)}^{\min}, \mu = 3, 8, \dots, M-5, i = 2, 4, \dots, N_\mu, \tau_{(\rho,0,5\hat{i})} = 1, j = 0, \rho = 0, 2(\mu+1), \\ v = f(\rho, 0, 5\hat{i})_1, \vartheta = f(\rho, 0, 5\hat{i})_2; \\ 1, \mu = 3, 8, \dots, M-5, i = 2, 4, \dots, N_\mu, \tau_{(0,2(\mu+2),0,5\hat{i})} = 2, j = 0; \\ 0, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 0; \\ 1, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 1; \\ 1, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, \tau_{(0,2(\mu+1),i)} = 1, j = 2; \\ \Delta_{(\rho,v,\vartheta)}^{\min}, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 2, \rho = 0, 2(\mu+1), \tau_{(\rho,i)} = 2, \\ v = f(\rho, i)_2, \vartheta = f(\rho, i)_2; \\ 0, \mu = M-1, i = 1, 2, \dots, N_\mu, j = 1, 2, \dots, N_{\mu-1}, K_j \neq i; \\ 1, \mu = M-1, i = 1, 2, \dots, N_\mu, j = 1, 2, \dots, N_{\mu-1}, K_j = i; \\ 1, \mu = M. \end{cases}$$

Запропонований метод дозволяє автоматично формувати ієрархічне перетворення координат екземплярів вибірки в багатовимірному просторі ознак на одномірну узагальнену вісь, виділяти нечіткі терми класів за узагальненою ознакою і налагоджувати їхні параметри, а також синтезувати нейро-нечіткі моделі якісних залежностей за прецедентами.

Розроблений метод може бути використаний при вирішенні задач технічної і біомедичної діагностики, розпізнавання образів. Застосування запропонованого методу дозволить забезпечити синтез нейро-нечітких моделей, що мають високий рівень узагальнення, а також характеризуються високою точністю і логічною прозорістю.

3.4 Нейро-нечітка кластер-регресійна апроксимація за узагальненою віссю

Для вирішення задач апроксимації кількісних залежностей за точковими даними традиційно використовуються регресійні [2, 4, 6], нейромережні [13], нечіткі та нейро-нечіткі моделі [10, 22]. Побудова таких моделей, як правило, припускає попереднє задавання користувачем структури апроксимуючої функції, а також сильно залежить від вибору початкової крапки пошуку в просторі керованих змінних і вимагає розрахунку значень похідних цільової функції навчання за керованими параметрами моделі (вагам), що у свою чергу вимагає диференційованості елементарних функцій (зокрема, функцій активації нейронів), що складають модель. Усе це істотно впливає на успішність і якість синтезу моделей, його швидкість. Крім того, традиційно використовувані на практиці методи часто призводять до побудови моделей, які складно піддаються сприйняттю й аналізу людиною, що істотно знижує ефективність і обмежує області їхнього застосування (особливо в задачах технічної і біомедичної діагностики).

Тому актуальним є створення методу синтезу моделей багатовимірних кількісних залежностей, що не залежить від вибору початкових значень ваг і не потребує задавання користувачем структури залежності, а також диференційованості цільової функції навчання. При цьому постановка задачі буде еквівалентна п. 3.3.1 та може бути конкретизована необхідністю розробки методу, що дозволяє в автоматичному режимі синтезувати структуру і налагоджувати параметри нейро-нечіткої регресійної моделі з забезпеченням її інтерпретабельності.

У методах [17, 18, 34, 36, 40, 44], загальний принцип апроксимації функції за точковими даними (спостереженнями) полягає в розбитті вхідного простору ознак на кластери – компактні області зосередження екземплярів у просторі вхідних ознак, і побудові для екземплярів кожного кластера окремої моделі залежності вихідної ознаки від набору значень вхідних ознак. На рис. 3.5 у нижній частині наведено приклад одновимірної реалізації такого підходу для апроксимації функції $y = \sin x$, графік якої зображено у центрі. Як видно з рис. 3.5, даний підхід не має досить високого рівня узагальнення, оскільки однакові періодично повторювані елементи функції апроксимуються окремими моделями, унаслідок чого виникає дубляж моделей, збільшується розмірність загальної моделі, знижуються апроксимаційні властивості й інтерпретабельність моделі.

Для усунення цих недоліків, що є властиві методам [17, 18, 34, 36, 40, 44], пропонується групувати екземпляри в кластери не за значеннями вхідних, а за упорядкованим значенням вихідних ознак екземплярів навчальної вибірки з наступним визначенням перетворення вхідних координат екземплярів у коор-

динати за узагальненою віссю та подальшою побудовою одновимірних часткових моделей залежності вихідної ознаки від координати екземпляра за узагальненою віссю (верхня частина рис. 3.5). Такі дії дозволять сформувати меншу кількість часткових моделей, забезпечити кращі апроксимаційні властивості, а також підвищити інтерпретабельність апроксимуючої моделі.

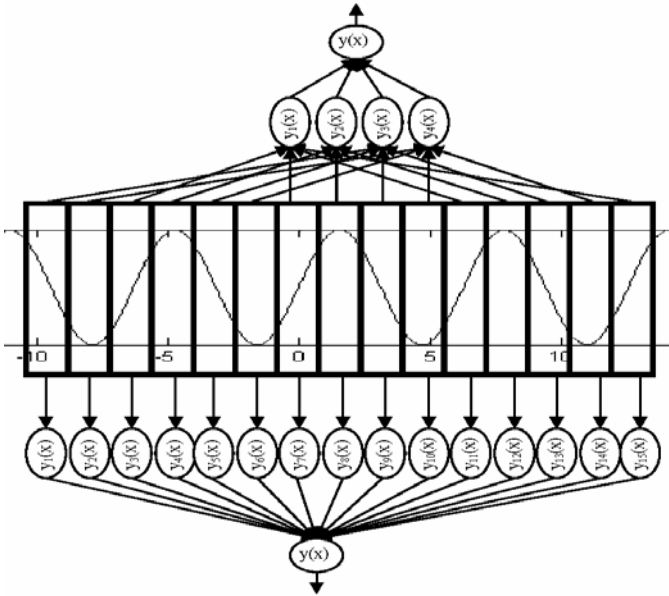


Рисунок 3.5 – Варіанти кластер-регресійної апроксимації

Формування апроксимуючої моделі на основі запропонованого методу можна здійснити шляхом виконання таких кроків.

Крок 1. Ініціалізація. Задати навчальну вибірку $\langle x, y \rangle$. Установити мінімальну S_{\min} і максимальну S_{\max} кількості екземплярів у кластері, наприклад: $S_{\min} = 5$, $S_{\max} = 30$.

Крок 2. Упорядкувати екземпляри вибірки в порядку неубування значень вихідної ознаки y . Після упорядкування номери екземплярів будуть являти собою їхні ранги.

Крок 3. Визначити межі інтервалів вихідної ознаки y .

Крок 3.1 Визначити мінімальне $y_{\min} = \min \{y^s\}$ і максимальне $y_{\max} = \max \{y^s\}$ значення вихідної ознаки, $s = 1, 2, \dots, S$. Знайти максимальну ширину інтервалу значень вихідної ознаки $\Delta y^{\max} = y_{\max} - y_{\min}$, а також мінімальну ширину інтервалу між двома значеннями вихідної ознаки

$\Delta_y^{\min} = \min |y^s - y^p|$, $s = 1, 2, \dots, S$; $p = s + 1, s + 2, \dots, S$. Визначити початкову кількість інтервалів розбиття осі вихідної ознаки: $n = \text{round}(10 \lg S)$, де round – функція округлення

Крок 3.2 Якщо $n < 1$, тоді прийняти: $n = 1$.

Крок 3.3 Якщо $n > \text{round}(\Delta_y^{\max}/\Delta_y^{\min})$, тоді прийняти: $n = \text{round}(\Delta_y^{\max}/\Delta_y^{\min})$.

Крок 3.4 Сформувати інтервали, установивши значення їхніх лівих і правих меж:

$$L_k = y_{\min} + (k-1)(\Delta_y^{\max} / n), R_k = L_k + (\Delta_y^{\max} / n), k = 1, 2, \dots, n.$$

Крок 4. Для кожного k -го інтервалу, $k = 1, 2, \dots, n$, визначити S_k – кількість екземплярів, що потрапили до нього, та середнє значення вихідної ознаки:

$$\bar{y}_k = \frac{1}{S_k} \sum_{s=1}^{S_k} y^s, L_k \leq y^s \leq R_k,$$

а також модуль коефіцієнта парної кореляції рангу екземпляра і значення його вихідної ознаки:

$$r_k = \frac{\left| \sum_{s=1}^{S_k} \{s/L_k \leq y^s \leq R_k\} - \frac{1}{S_k} \sum_{s=1}^{S_k} \{s/L_k \leq y^s \leq R_k\} \sum_{s=1}^{S_k} \{y^s - \bar{y}_k/L_k \leq y^s \leq R_k\} \right|}{\sqrt{\sum_{s=1}^{S_k} \left\{ \{s/L_k \leq y^s \leq R_k\} - \frac{1}{S_k} \sum_{s=1}^{S_k} \{s/L_k \leq y^s \leq R_k\} \right\}^2 \sum_{s=1}^{S_k} \left\{ (y^s - \bar{y}_k)^2 / L_k \leq y^s \leq R_k \right\}}$$

після чого знайти середнє значення модуля коефіцієнта парної кореляції для всіх інтервалів:

$$\bar{r} = \frac{1}{n} \sum_{k=1}^n r_k.$$

Крок 5. Виконати об'єднання і розбиття інтервалів.

Крок 5.1 Якщо існує такий k -ий інтервал, $k = 1, 2, \dots, n$, що $S_k > S_{\max}$ та $r_k < \bar{r}$, тоді розбити k -ий інтервал на дві частини на основі одного з двох описаних нижче способів, після чого перейти до кроку 5.1; у протилежному випадку – перейти до кроку 5.2.

Спосіб 1. Скорегувати номери інтервалів, зрушивши на один інтервал вправо (збільшуючи номери інтервалів на 1) починаючи з $(k+1)$ -го інтервалу, збільшити число інтервалів: $n = n + 1$. Поточний k -ий інтервал розбити на два рівних інтервали: $L_{k+1} = 0,5(L_k + R_k)$, $R_{k+1} = R_k$, $R_k = L_{k+1}$, перерахувати для k -го та $(k+1)$ -го інтервалів \bar{y}_k та \bar{y}_{k+1} , а також r_k та r_{k+1} , відповідно.

Спосіб 2. Розбити k -ий інтервал на основі попереднього способу, після чого скорегувати межу між k -им та $(k+1)$ -им інтервалами:

$$L_{k+1} = \begin{cases} R_{k+1} - \left(1 + \sigma_k \sigma_{k+1}^{-1}\right)^{-1}, & \sigma_k \geq \sigma_{k+1} > 0; \\ L_k + \left(1 + \sigma_{k+1} \sigma_k^{-1}\right)^{-1}, & 0 < \sigma_k < \sigma_{k+1}; \\ L_{k+1}, & \sigma_k = 0 \text{ або } \sigma_{k+1} = 0, \end{cases}$$

$$\sigma_k = \sum_{s=1}^S (y^s - \bar{y}_k)^2, L_k \leq y^s \leq R_k,$$

прийняти: $R_k = L_{k+1}$, перерахувати для k -го та $(k+1)$ -го інтервалів \bar{y}_k та \bar{y}_{k+1} , а також r_k та r_{k+1} , відповідно.

Крок 5.2 Якщо існує такий k -ий інтервал, $k = 1, 2, \dots, n$, що $S_k < S_{\min}$, тоді об'єднати k -ий інтервал із сусідніми інтервалами на основі одного з двох описаних нижче способів, після чого перейти до кроку 5.2; у протилежному випадку – до кроку 6.

Спосіб 1. Поточний інтервал поєднується з тим сусіднім інтервалом, найближче крайнє значення якого ближче до середнього значення вихідної ознаки в інтервалі: якщо $|R_{k-1} - \bar{y}_k| < |L_{k+1} - \bar{y}_k|$, тоді приєднати k -ий інтервал до $(k-1)$ -го інтервалу, установивши: $R_{k-1} = R_k$; у протилежному випадку приєднати k -ий інтервал до $(k+1)$ -го інтервалу, установивши: $R_k = R_{k+1}$. Скоорегувати номери інтервалів, зрушивши на один інтервал уліво (зменшуючи номери інтервалів на 1) починаючи з $(k+1)$ -го інтервалу, зменшити число інтервалів $n = n - 1$, перерахувати для k -го інтервалу \bar{y}_k та r_k .

Спосіб 2. Поточний інтервал розбивається на дві частини, кожна з яких поєднується з найближчим до неї сусіднім інтервалом: $R_{k-1} = 0,5(\bar{y}_{k-1} + \bar{y}_{k+1})$, $L_{k+1} = 0,5(\bar{y}_{k-1} + \bar{y}_{k+1})$. Скоорегувати номери інтервалів, зрушивши на один інтервал уліво (зменшуючи номери інтервалів на 1) починаючи з $(k+1)$ -го інтервалу, зменшити число інтервалів $n = n - 1$, перерахувати для k -го та $(k-1)$ -го інтервалів \bar{y}_k та \bar{y}_{k-1} , а також r_k та r_{k-1} , відповідно.

Крок 6. Сформувати вибірку $\langle x, y, K \rangle$, де $K = \{K^s\}$, K^s – мітка класу або номер інтервалу вихідної ознаки, до якого потрапило значення вихідної ознаки s -го екземпляра.

Крок 7. Для вибірки $\langle x, y, K \rangle$ визначити ієрархічну структуру перетворення координат розпізнаваного екземпляра на узагальнену вісь аналогічно п. 3.3.4. При цьому поряд з розглянутими в п. 3.3.2 перетвореннями також можна запропонувати використовувати більш прості перетворення для формування узагальненої вісі:

$$x_{(i,j)}^{s(III)} = \mathfrak{K}_3(x_i^s, x_j^s) = x_i^s + x_j^s, \quad x_{(i,j)}^{s(IV)} = \mathfrak{K}_4(x_i^s, x_j^s) = x_i^s x_j^s.$$

Для узагальненої вісі виділити інтервали, що відповідають класам вихідної ознаки аналогічно п. 3.3.3.

Крок 8. Для інтервалів узагальненої вісі побудувати часткові моделі залежностей $y_k(x_*^s) = f_{k(\zeta)}(x_*^s)$, де x_*^s – координата розпізнаваного екземпляра за узагальненою віссю, ζ – номер виду часткової моделі залежності, $k = 1, 2, \dots, D(x_*)$, шляхом виконання кроків 8.1–8.3.

Крок 8.1 Визначити параметри для нормування значень узагальненої ознаки для k -го інтервалу:

$$\bar{x}_k = \frac{1}{S_k} \sum_{s=1}^S \{x_k^s \mid L_k \leq x_k^s \leq R_k\}$$

Крок 8.2. Побудувати часткові моделі для k -го інтервалу:

$$f_{k(\zeta)}(x_k^s) = b(k, \zeta)_0 (x_k^s - \bar{x}_k)^\lambda + b(k, \zeta)_1, \lambda = 1, 2, \dots, \lambda_{\max}.$$

Значення параметрів $b(k, \zeta)_0$ та $b(k, \zeta)_1$ для моделі $f_{k(\zeta)}(x_k^s)$ пропонується визначити на основі методу найменших квадратів за формулами:

$$b(k, \zeta)_0 = \frac{\overline{y\Phi_0} - \overline{y}\overline{\Phi_0}}{\overline{\Phi_0^2} - (\overline{\Phi_0})^2}, \quad b(k, \zeta)_1 = \frac{\overline{\Phi_0^2} \overline{y} - \overline{\Phi_0} \overline{y\Phi_0}}{\overline{\Phi_0^2} - (\overline{\Phi_0})^2},$$

$$\overline{y\Phi_0} = \frac{1}{S_k} \sum_{s=1}^S \{y^s (x_k^s - \bar{x}_k)^\lambda \mid L_k \leq x_k^s \leq R_k\}, \quad \overline{y} = \frac{1}{S_k} \sum_{s=1}^S \{y^s \mid L_k \leq x_k^s \leq R_k\},$$

$$\overline{\Phi_0} = \frac{1}{S_k} \sum_{s=1}^S \{(x_k^s - \bar{x}_k)^\lambda \mid L_k \leq x_k^s \leq R_k\}, \quad \overline{\Phi_0^2} = \frac{1}{S_k} \sum_{s=1}^S \{(x_k^s - \bar{x}_k)^{2\lambda} \mid L_k \leq x_k^s \leq R_k\}.$$

Крок 8.3 Серед побудованих моделей знайти номер найбільш точної:

$$\zeta_k = \arg \min_{\zeta} E_{\zeta},$$

або найбільш простої моделі, що задовольняє вимогам точності:

$$\zeta_k = \min_{E_{\zeta} \leq \epsilon}$$

Крок 9. Зупинення.

Після формування ієрархії перетворення ознак на узагальнену вісь є можливим виконати її реалізацію у вигляді нейро-нечіткої мережі GAFNet (Generalized Axis Fuzzy Network), зображеної на рис. 3.6.

Мережа GAFNet складається з двох модулів: перший модуль являє собою реалізацію ієрархії перетворення ознак на узагальнену вісь, виконану в нейробазисі, а другий модуль являє собою нейро-нечіткий класифікатор.

Кількість нейронів у шарах мережі визначається за формулами:

– для першого блоку при використанні перетворень (I) та (II):

$$N_{\mu} = \begin{cases} 2N((\mu+4)\text{div } 5), & \mu = 1, 3, 6, 8, \dots, M-8, M-6; \\ N((\mu+4)\text{div } 5), & \mu = 2, 4, 5, 7, 9, 10, \dots, M-7, M-5, M-4, \end{cases}$$

– для першого блоку при використанні перетворень (III) та (IV):

$$N_{\mu} = 2^{(1-\mu)} N, \quad \mu = 1, 2, \dots, M-4,$$

– для другого блоку:
$$N_{\mu} = \begin{cases} 2n, & \mu = M-3; \\ n, & \mu = M-2; \\ n, & \mu = M-1; \\ 1, & \mu = M, \end{cases}$$

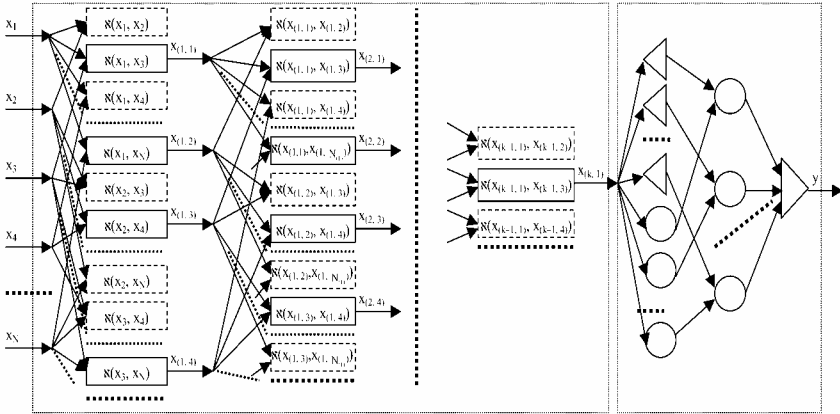


Рисунок 3.6 – Схема синтезу мережі нейро-нечіткої апроксимації за узагальненою віссю

де μ – номер шару в нейро-нечіткій мережі, M – кількість шарів у мережі, $M = k+4$, N_μ – кількість нейронів у μ -му шарі, $a \text{ div } b$ – операція цілочисельного розподілу a на b .

Кожен блок першого модуля виконує перетворення ознак попереднього рівня на узагальнену вісь. Границі блоків, що забезпечили поліпшення розбиття навчальної вибірки на класи і ввійшли до мережі, показані суцільною лінією, а границі блоків, що погіршили розбиття і не ввійшли до мережі, показані пунктиром.

Блоки, що відповідають перетворенню (I), пропонується реалізувати у вигляді фрагмента нейромережі прямого поширення, зображеного на рис. 3.3. Блоки, що відповідають перетворенню (II), пропонується реалізувати у вигляді фрагмента нейромережі прямого поширення, зображеного на рис. 3.4.

Усі нейрони кожного блоку обох перетворень використовують лінійні функції активації: $\psi^{(\mu,i)}(x) = x$, $\mu = 1, 2, \dots, M-4$; $i = 1, \dots, N_\mu$, де i – номер нейрона в шарі, $\psi^{(\mu,i)}(x)$ – функція активації i -го нейрона μ -го шару.

Усі нейрони першого і четвертого шарів, а також непарні нейрони третього шару кожного блоку обох перетворень у якості дискримінантних функцій використовують зважені суми:

$$\varphi^{(\mu,i)}(w^{(\mu,i)}, x^{(\mu,i)}) = \sum_{j=1}^{N_{\mu-1}} w_j^{(\mu,i)} x_j^{(\mu,i)} + w_0^{(\mu,i)},$$

$$\mu = 1, 6, \dots, M-8; i = 1, 2, \dots, N_\mu,$$

$$\mu = 3, 8, \dots, M-6: i = 1, 3, \dots, N_\mu-1,$$

$$\mu = 4, 9, \dots, M-5: i = 1, 2, \dots, N_\mu,$$

де $\varphi^{(\mu, i)}(w_j^{(\mu, i)}, x_j^{(\mu, i)})$ – дискримінантна (вагова) функція i -го нейрона μ -го шару, $x_j^{(\mu, i)}$ – значення на j -му вході i -го нейрона μ -го шару, $w_j^{(\mu, i)}$ – ваговий коефіцієнт, зіставлений j -му входу i -го нейрона μ -го шару.

Усі нейрони другого шару кожного блоку обох перетворень використовують як дискримінантну функцію відстані:

$$\varphi^{(\mu, i)}(w_j^{(\mu, i)}, x_j^{(\mu, i)}) = \sqrt{\sum_{j=1}^2 (w_j^{(\mu, i)} - x_j^{(\mu, i)})^2},$$

$$\mu = 2, 7, \dots, M-7: i = 1, 2, \dots, N_\mu,$$

Усі парні нейрони третього шару кожного блоку обох перетворень як дискримінантні використовують функції добутку:

$$\varphi^{(\mu, i)}(w_j^{(\mu, i)}, x_j^{(\mu, i)}) = w_0^{(\mu, i)} \prod_{j=1}^2 w_j^{(\mu, i)} x_j^{(\mu, i)},$$

$$\mu = 3, 8, \dots, M-6: i = 2, 4, \dots, N_\mu.$$

Нейрони п'ятого шару кожного блоку обох перетворень як дискримінантні використовують функції розподілу:

$$\varphi^{(\mu, i)}(x^{(\mu, i)}) = \frac{x_2^{(\mu, i)}}{x_1^{(\mu, i)}},$$

$$\mu = 5, 10, \dots, M-4: i = 1, 2, \dots, N_\mu.$$

Блоки, що відповідають перетворенням (III) та (IV), пропонується реалізувати у вигляді фрагментів нейромережі прямого поширення, зображених, відповідно, на рис. 3.7, а) та рис. 3.7, б). Ці блоки складаються усього з одного нейрона, що має лінійну функцію активації $\psi^{(\mu, i)}(x) = x$, $\mu = 1, 2, \dots, M-4; i = 1, \dots, N_\mu$.

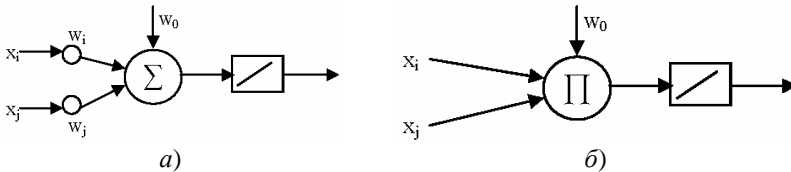


Рисунок 3.7 – Схема нейроблоку відображення двовимірних координат на узагальнену вісь: а) перетворення (III); б) перетворення (IV)

Нейрони, що відповідають третьому перетворенню, використовують у як дискримінантні функції зважену суму для $\mu = 1, 2, \dots, M-4; i = 1, \dots, N_\mu$, а четвертому перетворенню – функцію добутку без ваг зв'язків для $\mu = 1, 2, \dots, M-4; i = 1, \dots, N_\mu$.

Перша половина нейронів першого шару другого модуля (шару нейронів, що передуює передостанньому шару мережі) являє собою вузли, що визначають належності розпізнаваного екземпляра до нечітких термів змінної узагальненої осі верхнього рівня.

Якщо для p -го блоку ℓ -го рівня $\tau_{(\ell, p)} = 0$, то блок вважається відсутнім, а його вхідна ознака вважається вихідною (з'єднується з виходом блоку).

Як дискримінантні функції вузли першого шару другого модуля використовують трапецієподібні функції:

$$\varphi^{(M-3,i)}(x^{(M-3,i)}) = \begin{cases} 0, & x^{(M-3,i)} < 0,5(L_i + L_{i-1}), \\ \frac{x^{(M-3,i)} - 0,5(L_i + L_{i-1})}{0,5(L_i - L_{i-1})}, & 0,5(L_i + L_{i-1}) \leq x^{(M-3,i)} < L_i, \\ 1, & L_i \leq x^{(M-3,i)} \leq R_i, \\ \frac{0,5(R_{i+1} + R_i) - x^{(M-3,i)}}{0,5(R_{i+1} - R_i)}, & R_i < x^{(M-3,i)} \leq 0,5(R_{i+1} + R_i), \\ 0, & 0,5(R_{i+1} + R_i) < x^{(M-3,i)}, \end{cases}$$

де L_i, R_i – відповідно, ліва і права межі i -го інтервалу значень узагальненої осі, $i = 1, 2, \dots, D(x_{(k,1)})$, $L_0 = L_1$, $R_{D(x_{(k,1)})+1} = R_{D(x_{(k,1)})}$.

Як функції активації нейрони даного шару використовують лінійні функції: $\psi^{(M-3,i)}(x) = x$, $i = 1, 2, \dots, D(x_{(k,1)})$.

Нейроблок для кожної часткової i -ої моделі (рис. 3.8) реалізується у вигляді фрагмента двошарової мережі прямого поширення, що містить два нейрони, що використовують як дискримінантні функції зважену суму, а як функції активації: нейрон першого шару – функцію $\psi^{(M-3,i)}(x) = x^\lambda$, нейрон другого шару – функцію $\psi^{(M-2,i)}(x) = x$.

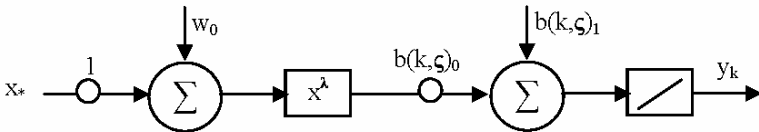


Рисунок 3.8 – Схема нейроблоку часткової моделі

Передостанній шар мережі містить стільки нейронів, скільки кластерів виділено у навчальній вибірці. Кожен нейрон передостаннього шару мережі комбінує приналежність розпізнаваного екземпляра до кластера вихідної змінної і значення вихідної змінної, оцінене відповідною частковою моделлю.

Як дискримінантні функції нейрони передостаннього шару мережі використовують зважену суму, а як функції активації – лінійні функції: $\psi^{(M-1,i)}(x) = x$.

Єдиний нейрон останнього шару другого модуля здійснює дефазифікацію і видає на виході чітке значення вихідної змінної, оцінене нейрончистою мережею для розпізнаваного екземпляра.

Як дискримінантну функцію нейрон вихідного шару використовує:

$$\varphi^{(M,1)}(x^{(M,1)}) = \max_{j=1,2,\dots,K} x_j^{(M,1)}.$$

Як функція активації використовується лінійна функція:

$$\psi^{(M,1)}(x) = x.$$

Ваги входів нейронів мережі пропонується визначати за формулами:

– для нейронів першого блоку при використанні перетворень (I) та (II):

$$w_j^{(\mu,i)} = \begin{cases} -\alpha_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 1, 3, \dots, N_\mu - 1, j = 0, \rho = 0,2(\mu+4), v = f(\rho, 0,5(i+1))_1; \\ \Delta_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 1, 3, \dots, N_\mu - 1, j = v, \rho = 0,2(\mu+4), v = f(\rho, 0,5(i+1))_1; \\ 0, \mu = 1, 6, \dots, M-7, i = 1, 3, \dots, N_\mu - 1, j \neq v, j > 0, \rho = 0,2(\mu+4), v = f(\rho, 0,5(i+1))_1; \\ -\alpha_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 2, 4, \dots, N_\mu, j = 0, \rho = 0,2(\mu+4), v = f(\rho, 0,5i)_2; \\ \Delta_{(\rho,v)}, \mu = 1, 6, \dots, M-7, i = 2, 4, \dots, N_\mu, j = v, \rho = 0,2(\mu+4), v = f(\rho, 0,5i)_2; \\ 0, \mu = 1, 6, \dots, M-7, i = 2, 4, \dots, N_\mu, j \neq v, j > 0, \rho = 0,2(\mu+4), v = f(\rho, 0,5i)_2; \\ 0, \mu = 2, 7, \dots, M-6, i = 1, 2, \dots, N_\mu, j = 1, 2; \\ 1, \mu = 3, 8, \dots, M-5, i = 1, 2, \dots, N_\mu, j = 0; \\ d_{(\rho,v,\vartheta)}^{\min}, \mu = 3, 8, \dots, M-5, i = 1, 3, \dots, N_\mu - 1, \tau_{(\rho, 0,5(i+1))} = 1, j = 1, \rho = 0,2(\mu+1), \\ v = f(\rho, 0,5(i+1))_1, \vartheta = f(\rho, 0,5(i+1))_2; \\ \Delta_{(\rho,v,\vartheta)}^{\min}, \mu = 3, 8, \dots, M-5, i = 1, 3, \dots, N_\mu - 1, \tau_{(\rho, 0,5(i+1))} = 2, j = 1, \rho = 0,2(\mu+1), \\ v = f(\rho, 0,5(i+1))_1, \vartheta = f(\rho, 0,5(i+1))_2; \\ d_{(\rho,v,\vartheta)}^{\min}, \mu = 3, 8, \dots, M-5, i = 2, 4, \dots, N_\mu, \tau_{(\rho, 0,5i)} = 1, j = 0, \rho = 0,2(\mu+1), \\ v = f(\rho, 0,5i)_1, \vartheta = f(\rho, 0,5i)_2; \\ 1, \mu = 3, 8, \dots, M-5, i = 2, 4, \dots, N_\mu, \tau_{(0,2(\mu+2), 0,5i)} = 2, j = 0; \\ 0, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 0; \\ 1, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 1; \\ 1, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, \tau_{(0,2(\mu+1), i)} = 1, j = 2; \\ \Delta_{(\rho,v,\vartheta)}^{\min}, \mu = 4, 9, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 2, \rho = 0,2(\mu+1), \tau_{(\rho,i)} = 2, \\ v = f(\rho, i)_1, \vartheta = f(\rho, i)_2; \end{cases}$$

– для нейронів першого блоку при використанні перетворень (III) та (IV):

$$w_j^{(\mu,i)} = \begin{cases} 0, \mu = 1, 6, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 0, \tau(\mu, i) = 3; \\ 1, \mu = 1, 6, \dots, M-4, i = 1, 2, \dots, N_\mu, j = 0, \tau(\mu, i) = 4; \\ 0, \mu = 1, 6, \dots, M-4, i = 1, 3, \dots, N_\mu, j > 0, j = f(\mu, i)_1 \text{ або } j = f(\mu, i)_2, \tau(\mu, i) = 3; \\ 1, \mu = 1, 6, \dots, M-4, i = 1, 3, \dots, N_\mu, j > 0, j \neq f(\mu, i)_1, j \neq f(\mu, i)_2, \tau(\mu, i) = 3; \end{cases}$$

– для нейронів другого блоку:

$$w_j^{(\mu,i)} = \begin{cases} 0, \mu = M - 1, i = 1, 2, \dots, N_\mu, j = 1, 2, \dots, 0,5N_{\mu-1}, K_j \neq i; \\ 1, \mu = M - 1, i = 1, 2, \dots, N_\mu, j = 1, 2, \dots, 0,5N_{\mu-1}, K_j = i \\ 0, \mu = M - 1, i = 1, 2, \dots, N_\mu, j = 0,5N_{\mu-1} + 1, \dots, N_{\mu-1}, K_{j/2} \neq i; \\ 1, \mu = M - 1, i = 1, 2, \dots, N_\mu, j = 0,5N_{\mu-1} + 1, \dots, N_{\mu-1}, K_{j/2} = i; \\ 1, \mu = M - 3, i = 0,5N_\mu + 1, \dots, N_\mu, j = 1; \\ -\bar{x}_s, \mu = M - 3, i = 0,5N_\mu + 1, \dots, N_\mu, j = 0; \\ \beta(i, \zeta_i)_0, \mu = M - 2, i = 1, 2, \dots, N_\mu, j = N_\mu + i; \\ 0, \mu = M - 2, i = 1, 2, \dots, N_\mu, j \neq N_\mu + i, j = 1, 2, \dots, 2N_\mu; \\ \beta(i, \zeta_i)_1, \mu = M - 2, i = 1, 2, \dots, N_\mu, j = 0. \end{cases}$$

Запропонований метод, дозволяє автоматично формувати ієрархічне перетворення координат екземплярів вибірки в багатовимірному просторі ознак на одновимірну узагальнену вісь, виділяти нечіткі терми класів за узагальненою ознакою і налагоджувати їхні параметри, а також синтезувати часткові регресійні моделі кількісних залежностей для кластерів. Розроблений метод може бути використаний при вирішенні задач прогнозування й оцінювання значень параметрів у технічній і біомедичній діагностиці. Застосування запропонованого методу дозволить забезпечити синтез нейронечітких моделей, що мають високий рівень узагальнення, а також характеризуються високою точністю й логічною прозорістю.

ЛІТЕРАТУРА ДО ЧАСТИНИ I

1. Abraham A. Neuro-Fuzzy Systems: State-of-the-Art Modeling Techniques // Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence / Eds.: J. Mira and A. Prieto. – Granada: Springer-Verlag, 2001. – P. 269–276.
2. Broomhead D.S., Lowe D. Multivariable function interpolation and adaptive networks // Complex systems. – 1988. – № 2. – P. 321–355.
3. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика. Классификация и снижение размерности. – М.: Финансы и статистика, 1989 – 607 с.
4. Андреев И. Деревья решений – CART математический аппарат // «Exponenta Pro (математика в приложениях)». – 2004. – № 3–4.
5. Андрейчиков А. В., Андрейчикова О. Н. Интеллектуальные информационные системы: Учебник. – М.: Финансы и статистика, 2004. – 424 с.
6. Аффифи А., Эйзен С. Статистический анализ: Подход с использованием ЭВМ. – М.: Мир, 1982. – 488 с.
7. Барсегян А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP: Уч. пос. – СПб: BHV, 2007. – 384 с.
8. Бодянский Е. В., Кучеренко Е. И., Михалев А. И. Нейро-фаззи сети Петри в задачах моделирования сложных систем. – Днепропетровск: Системные технологии. – 2005. – 311 с.
9. Бондарев В. Н., Аде Ф. Г. Искусственный интеллект. – Севастополь: СевНТУ, 2002. – 615 с.
10. Борисов В. В., Круглов В. В., Федулов А. С. Нечеткие модели и сети. – М.: Горячая линия-Телеком, 2007. – 284 с.
11. Брянцев И. Н. Data Mining. Теория и практика. – М.: БДЦ-Пресс, 2006. – 208 с.
12. Джонс М. Т. Программирование искусственного интеллекта в приложениях / Пер. с англ. А. И. Осипов. – М.: ДМК Пресс, 2004. – 312 с.
13. Дли М.И. Нечеткая логика и искусственные нейронные сети. – М.: Физматлит, 2003. – 225 с.
14. Дубровін В. І., Субботін С. О. Методи оптимізації та їх застосування в задачах навчання нейронних мереж: Навчальний посібник. – Запоріжжя: ЗНТУ, 2003. – 136 с.
15. Дюк В., Самойленко А. Data mining: учебный курс. – СПб.: Питер, 2001. – 368 с.
16. Зайченко Ю. П. Основы проектирования интеллектуальных систем. Навчальний посібник. – К.: Слово, 2004. – 352 с.
17. Интеллектуальные средства диагностики и прогнозирования надежности авиадвигателей: Монография / В. И. Дубровин, С. А. Субботин, А. В. Богуслаев, В. К. Яценко. – Запорожье: ОАО «Мотор-Сич», 2003. – 279 с.
18. Кирсанова Е. В., Субботин С. А. Обобщенный метод кластер-регрессионной аппроксимации в задаче моделирования показателя здоровья детей // Радиоелектроніка. Інформатика. Управління. – 2004. – № 1. – С. 62–67.

19. Кричевский М. Л. Интеллектуальные методы в менеджменте. – СПб.: Питер, 2005. – 304 с.
20. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с.
21. Ларин С. Применение ассоциативных правил для стимулирования продаж [Электронный ресурс]. – М.: Basegroup, 2003. – Режим доступа: <http://www.basegroup.ru/practice/salepromotion.htm>.
22. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2003. – 736 с.
23. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Пер. с англ. – М. – Вильямс, 2005. – 864 с.
24. Митюшкин Ю. И., Мокин Б. И., Ротштейн А. П. Soft Computing: идентификация закономерностей нечеткими базами знаний. – Винница: Универсум-Винница, 2002. – 145 с.
25. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2004. – 344 с.
26. Прикладные нечеткие системы / Асаи К., Ватада Д., Иваи С. и др. / Под ред. Т. Тэрано, К. Асаи, М. Сугено. – М.: Мир, 1993. – 368 с.
27. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е и зд.: Пер с англ. – М. – Вильямс, 2006. – 1408 с.
28. Рідкокаша А. А., Голдер К. К. Основи систем штучного інтелекту. Навчальний посібник. – Черкаси: «ВІДЛУННЯ-ПЛЮС», 2002. – 240 с.
29. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети. – Винница: УНИВЕРСУМ-Винница, 1999. – 320 с.
30. Ротштейн А. П. Медицинская диагностика на нечеткой логике. – Винница: Континент-ПРИМ, 1996. – 132 с.
31. Ротштейн О. П., Ракитянська Г. Б. Діагностика на базі нечітких відношень в умовах невизначеності. – Вінниця: Універсум-Вінниця, 2006. – 275 с.
32. Руденко О. Г., Бодянский Е. В. Основы теории искусственных нейронных сетей. – Харьков: Телетех, 2002. – 317 с.
33. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер с польск. – М.: Горячая линия-Телеком, 2004. – 452 с.
34. Субботін С. О. Алгоритми кластер-регресійної апроксимації та їх нейромережеві інтерпретації // *Радіоелектроніка. Інформатика. Управління.* – 2003. – № 1. – С. 114–121.
35. Субботин С. А. Синтез распознающих нейро-нечетких моделей с учетом информативности признаков // *Нейрокомпьютеры: разработка, применение.* – 2006. – № 10. – С. 50–56.
36. Субботин С. А. Метод синтеза нейро-нечетких аппроксиматоров // *Автоматизация и современные технологии.* – 2007. – № 11. – С. 14–18.

37. Субботин С. А. Метод формирования баз знаний для нейро-нечетких моделей // *Нейроинформатика и ее приложения: Материалы XIV Всероссийского семинара, 6–8 октября 2006 г.* / Под ред. А. Н. Горбаня, Е. М. Миркеса. Отв. За выпуск Г. М. Садовская. – Красноярск: ИВМ СО РАН, 2006. – С. 116–118.
38. Субботин С. А. Методы синтеза нейро-нечетких классификаторов для случая нескольких классов // *Информационные технологии.* – 2006. – № 11. – С. 31–36.
39. Субботин С. А. Неитеративный синтез и редукция нейро-нечетких моделей // *Искусственный интеллект.* – 2006. – № 3. – С. 323–330.
40. Субботин С. А. Нейро-нечеткая кластер-регрессионная аппроксимация // *Нейроинформатика и ее приложения: Материалы XV Всероссийского семинара / Под ред. А. Н. Горбаня, Е. М. Миркеса. Отв. За выпуск Г. М. Садовская.* – Красноярск: ИВМ СО РАН, 2007. – С. 143–146.
41. Субботин С. А. Синтез диагностических моделей авиадвигателей на основе иерархических нейро-нечетких сетей // *Вісник двигунобудування.* – 2007. – № 1. – С. 15–19.
42. Субботин С. А. Синтез нейро-нечетких моделей для выделения и распознавания объектов на сложном фоне по двумерному изображению // *Комп'ютерне моделювання та інтелектуальні системи: Збірник наукових праць / За ред. Д. М. Пізи, С. О. Субботіна.* – Запоріжжя: ЗНТУ, 2007. – С. 68–91.
43. Субботин С. А. Синтез нейро-нечетких сетей с группировкой признаков // *Программные продукты и системы.* – 2006. – № 4. – С. 3–7.
44. Субботин С. А., Кирсанова Е. В. Синтез многослойной нейросети на основе кластер-регрессионной аппроксимации в задаче моделирования показателя здоровья детей // *Нейроинформатика и ее приложения: Материалы XII Всероссийского семинара, 1–3 октября 2004 г.* / Под ред. А. Н. Горбаня, Е. М. Миркеса. Отв. за выпуск Г. М. Садовская. – Красноярск: ИВМ СО РАН, 2004. – С. 136–137.
45. Тарков М. С. *Нейрокомпьютерные системы. Учебное пособие* М.: Интуит, 2006. – 142 с.
46. Усков А. А., Кузьмин А. В. *Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика.* – М.: Горячая линия-Телеком, 2004. – 143 с.
47. Чубукова И. А. *Data Mining. Учебное пособие.* – М.: Интуит, 2006. – 382 с.
48. Шахиди А. Apriori – масштабируемый алгоритм поиска ассоциативных правил [Электронный ресурс]. – М.: Basegroup, 2002. – Режим доступа: <http://www.basegroup.ru/rules/apriori.htm>.
49. Шахиди А. Введение в анализ ассоциативных правил [Электронный ресурс]. – М.: Basegroup, 2002. – Режим доступа: <http://www.basegroup.ru/rules/intro.htm>.
50. Ярушкина Н. Г. *Основы теории нечетких и гибридных систем.* – М.: Финансы и статистика, 2004. – 320 с.
51. Яхьяева Г. Э. *Нечеткие множества и нейронные сети. Учебное пособие.* – М.: Интуит, 2006. – 316 с.

ЧАСТИНА II

ЕВОЛЮЦІЙНІ МЕТОДИ СИНТЕЗУ НЕЙРОМЕРЕЖНИХ МОДЕЛЕЙ

Побудова нейромережних моделей пов'язана з необхідністю визначення найбільш значущої комбінації ознак, з пошуком оптимальної структури НМ і настроюванням її параметрів відповідно до заданого критерію оптимальності. Однак вирішення оптимізаційних задач при побудові нейромоделей пов'язано з проблемою вибору початкової точки пошуку, неможливістю розрахунку значень похідних цільової функції внаслідок недиференційованості функцій активації окремих нейронів, значними витратами часу, обумовленими великою розмірністю навчальної вибірки й багатоекстремальністю цільової функції, що ускладнює або робить інколи неможливим використання градієнтних методів локального пошуку, що традиційно застосовуються для побудови нейронних мереж.

Тому для побудови нейромоделей доцільно використовувати методи еволюційного пошуку, які є стохастичними методами, заснованими на аналогії з природними еволюційними процесами, не висувають додаткових вимог до виду цільової функції, на кожній ітерації працюють з множиною рішень, що дозволяє в багатьох випадках більш детально в порівнянні з градієнтними методами багатовимірної нелінійної безумовної оптимізації аналізувати простір пошуку.

До еволюційних методів відносять: генетичні методи, еволюційні стратегії, генетичне програмування та еволюційне програмування.

Кожний з таких методів має велику кількість різновидів. Порівняльна характеристика методів еволюційного пошуку наведена в табл. 4.1.

Таблиця 4.1 – Порівняльний аналіз методів еволюційного пошуку

Клас методів	Тип даних хромосом	Еволюційні оператори
Генетичні методи	Бінарні, цілі або дійсні	Відбір, схрещування, мутація
Еволюційні стратегії	Дійсні	Мутація, $(\mu + \lambda)$ або (μ, λ) відбір
Генетичне програмування	Дерево подань рішень	Відбір, схрещування, мутація
Еволюційне програмування	Дійсні	Мутація, $(\mu + \lambda)$ відбір

РОЗДІЛ 4

ГЕНЕТИЧНІ МЕТОДИ

При вирішенні багатьох задач у техніці актуальною є проблема знаходження глобального оптимуму цільової функції в багатовимірному просторі керованих змінних. Традиційні методи багатовимірної оптимізації є методами локального пошуку, сильно залежать від вибору початкової точки пошуку та накладають додаткові обмеження на властивості цільової функції оптимізації [62].

Тому для вирішення оптимізаційних задач, що виникають у процесі синтезу нейромережних моделей, доцільним є використання генетичних методів [39, 45, 72], які є методами еволюційного пошуку й поєднують комп'ютерні методи моделювання генетичних процесів у природних і штучних системах.

4.1 Генетичний пошук як метод оптимізації

Оптимізаційні методи [62] використовуються для знаходження екстремальних значень функцій і відповідних їм значень незалежних змінних.

Вибір методу оптимізації пов'язаний з типом розв'язуваної оптимізаційної задачі. Задачі оптимізації можна класифікувати за такими критеріями:

- тип змінних, що оптимізуються (безперервні та дискретні);
- кількість змінних, що оптимізуються;
- вид цільової функції (лінійна, квадратична, нелінійна);
- властивості функцій (унімодальність, неперервність, гладкість, монотонність, диференційованість);
- спосіб задавання цільової функції (явне або неявне задавання);
- наявність, кількість і вид обмежень;
- вид оптимізації (глобальна або локальна);
- кількість цільових функцій, що оптимізуються (однокритеріальна або багатокритеріальна оптимізація).

Методи оптимізації в залежності від кількості керованих змінних можна розподілити на одновимірні і багатовимірні, а залежно від вигляду цільової функції – на лінійні і нелінійні.

Методи одновимірного прямого пошуку (наприклад, метод розподілу інтервалу навпіл, метод золотого перерізу, метод рівномірного пошуку [62]) засновані на обчисленні значень цільової функції та їх подальшому порівнянні в двох пробних точках, накладаючи єдину вимогу на досліджувану функцію, – унімодальність, тобто наявність єдиного оптимуму в даному інтервалі пошуку.

Одновимірний пошук з використанням поліноміальної апроксимації (наприклад, пошук оптимуму з використанням квадратичної апроксимації, метод Пауелла [62]) дозволяє врахувати відносні зміни значень цільової функції, що досягається введенням додаткових вимог до цільової функції, згідно з якими вона повинна бути неперервною і гладкою (мати неперервну похідну). Основна ідея таких методів пов'язана з можливістю апроксимації гладкої функції поліномом для подальшого його використання при оцінюванні координати точки оптимуму.

Методи одновимірної оптимізації з використанням похідних (наприклад, метод хорд, метод Ньютона-Рафсона, метод Больцано [62]) як додаткову інформацію для прискорення пошуку використовують значення похідних цільової функції в даних точках простору пошуку, накладаючи додаткове обмеження, – диференційованість цільової функції.

При *прямому багатовимірному пошуку* (наприклад, метод Нелдера-Міда, метод Хука-Дживса, метод зв'язаних напрямів Пауелла [62]) припускається, що цільова функція є унімодальною в даній області. У випадку, якщо функція є полімодальною, такі методи в більшості випадків знайдуть локальний оптимум замість глобального.

Градієнтні багатовимірні методи (метод Коші, метод Ньютона, метод Левенберга-Марквардта [62]) для визначення напряму пошуку використовують значення похідних цільової функції різних порядків.

Генетичний пошук включає групу багатовимірних, стохастичних, евристичних оптимізаційних методів [18, 19, 48, 55, 106, 107], заснованих на ідеї еволюції за допомогою природного відбору, висунутої Ч. Дарвіном в 1857 р. Методи генетичного пошуку отримані в процесі узагальнення та імітації в штучних системах таких властивостей живої природи, як природний відбір, пристосованість до змінюваних умов середовища, спадкоємність нащадками життєво важливих властивостей від батьків і т. ін.

Формально методи генетичного пошуку можуть бути описані у вигляді такої функції [45, 57, 64]:

$$GM = GM(P_0, N, L, f, \Omega, \Psi, \Theta, T),$$

де $P_0 = \{H_1^0, H_2^0, \dots, H_N^0\}$ – початкова популяція – множина рішень задачі, поданих у вигляді хромосом; $H_j^0 = \{h_{1j}^0, h_{2j}^0, \dots, h_{Lj}^0\}$ – j -та хромосома популяції P_0 – набір значень незалежних змінних, поданих у вигляді генів; h_{ij}^0 – i -ий ген j -ої хромосоми популяції P_0 – значення i -го оптимізованого параметру задачі, що входить в j -те рішення; N – кількість хромосом в популяції; L – довжина хромосом (кількість генів); f – цільова функція (фітнес-функція, функція пристосованості, функція здоров'я, функція придатності); Ω – оператор відбору; Ψ – оператор схрещування; Θ – оператор мутації; T – критерій зупинення.

З погляду штучних систем обробки інформації генетичний пошук є специфічним методом знаходження рішення задачі оптимізації. При цьому такий ітераційний пошук адаптується до особливостей цільової функції: нові хромосоми, що з'являються в процесі схрещування, тестують все більш широкі області простору пошуку і переважно розташовуються в області оптимуму. Відносно рідкісні мутації перешкоджають виродженню генофонду, що рівносильне рідкісному, але безперервному пошуку оптимуму у решті областей пошукового простору.

Таким чином, на кожній ітерації генетичного пошуку метод працює не з єдиним рішенням, а із деякою множиною рішень (сукупністю хромосом), за рахунок чого забезпечується паралельність пошуку. При цьому кожна нова множина рішень залежить лише від попередньої і, в загальному випадку, є кращою за попередню.

Оскільки генетичні методи в процесі пошуку використовують деяке кодування множини параметрів замість самих параметрів, то вони можуть ефективно застосовуватися для вирішення задач дискретної оптимізації, визначених як на числових множинах, так і на скінчених множинах довільної природи.

Для роботи генетичних методів як інформація про функцію, що оптимізується, використовуються її значення в даних точках простору пошуку і не потрібно обчислень похідних або інших характеристик. Тому дані методи можуть бути застосовані до широкого класу функцій, зокрема до тих, що не мають аналітичного опису. Таким чином, методи генетичного пошуку є достатньо гнучкими і можуть бути застосовані до широкого кола задач, в тому числі до задач, для розв'язування яких не існує загальновідомих методів [1, 12, 24, 25, 32, 40].

Генетичні методи є більш ефективним інструментом пошуку в порівнянні з класичними методами оптимізації за таких умов:

- досліджуваний простір пошуку є великим, негладким (існують точки розриву) і неунімодальним (є декілька оптимумів);
- цільова функція пошуку може мати шуми;
- задача не вимагає знаходження надточного глобального оптимуму. Тобто необхідно достатньо швидко знайти прийнятне рішення, що досить часто має місце в реальних задачах.

Таким чином, генетичний пошук може успішно використовуватися для вирішення комбінаторних задач, а також для пошуку оптимальних значень полімодальних функцій.

Генетичні методи мають такі *переваги* [23, 56, 57, 90, 92]:

- відсутність необхідності в специфічних знаннях про вирішувану задачу. Проте у випадку, якщо додаткова інформація про досліджувану систему, об'єкт або процес є відомою, то вона може бути використана в процесі пошуку;
- концептуальна простота та прозорість реалізації;
- можливість розпаралелювання;

- простота кодування вхідної і вихідної інформації. Некритичність до виду параметрів досліджуваних систем (можливість використання експертної, емпіричної, довідкової та іншої інформації про об'єкт, поданої різними типами даних);

- можливість застосування до великого кола задач без внесення серйозних змін у внутрішню структуру методу;

- можливість адаптивності параметрів генетичного пошуку до особливостей вирішуваної задачі;

- менша ймовірність попадання і зациклення в локальному оптимумі, яка досягається за рахунок використання популяційного підходу;

- можливість застосування в методі інших пошукових процедур.

До *недоліків* генетичного пошуку відносять:

- високу ітеративність;

- сильну залежність ефективності генетичного пошуку від його параметрів (розмір популяції, початкова точка пошуку, ймовірнісні характеристики генетичних операторів і т. ін.);

- епістазис – внутрішню залежність між змінними (генами), закодованими в хромосомі. Якщо гени не пов'язані один з одним, то вважається, що епістазис малий або не існує. У випадку, якщо гени є взаємозалежними, то епістазис може створювати проблеми для генетичного пошуку, спричинені тим, що при схрещуванні ланцюжки взаємозалежних генів будуть зруйновані, що призводить до появи низько пристосованих нащадків. Вирішення проблеми епістазису полягає в тому, щоб зберігати в хромосомі взаємозалежні гени, розташовуючи їх близько один до одного. При групуванні залежних генів істотно знижується ймовірність того, що вони будуть зруйновані при застосуванні схрещування;

- передчасна збіжність, пов'язана з недостатньою різноманітністю хромосом в популяції. Найпоширенішою причиною передчасної збіжності є недостатній розмір популяції. Таким чином, вирішенням такої проблеми може бути збільшення кількості хромосом в популяції.

Враховуючи особливості, переваги і недоліки оптимізаційних методів, можна надати такі *рекомендації щодо вибору пошукового методу* при вирішенні практичних задач [18, 39–41, 72]:

- якщо простір пошуку є дискретним і невеликим за розміром, то можна скористатися методом повного перебору [67], який знайде найкраще рішення. Генетичний метод, на відміну від методу повного перебору, може з більшою ймовірністю зійтися до локального оптимуму, а не до глобального. Проте генетичний пошук швидше знайде субоптимальне рішення, що знаходиться недалеко від дійсного оптимуму;

- якщо цільова функція в пошуковому просторі є гладкою і унімодальною, то будь-який градієнтний метод (наприклад, метод найшвидшого спуску [62]) буде ефективнішим, ніж генетичний пошук;

– якщо про простір пошуку відома деяка додаткова інформація (наприклад, для задачі комівояжера), то методи пошуку, що використовують апріорні відомості про пошуковий простір [16, 40, 62], часто перевершують будь-який універсальний метод, у тому числі й генетичний метод;

– при достатньо складному рельєфі цільової функції методи пошуку, що працюють із єдиним рішенням на кожній ітерації (наприклад, простий метод спуску), можуть зациклитися в локальному рішенні. Генетичні методи працюють з набором із декількох рішень [39–41, 45, 72], тому вони мають менше шансів зійтися до локального оптимуму і надійно функціонують на багатоекстремальних поверхах.

4.2 Аналогія генетичних методів з поняттями генетики

Ген – реально існуюча, незалежна одиниця спадковості, що комбінується й розщеплюється при схрещуваннях. Поняття гену як основної концепції класичної генетики було введено І. Г. Менделем з метою пояснення спостережуваної статистики спадкування.

Хромосоми – структурні елементи клітинного ядра біологічних організмів, є носіями генів у клітинному ядрі особини. У генетичних методах терміни «хромосома» та «особина» використовуються як синоніми.

Локус – місце, що займає ген у хромосомі.

Алелі – значення генів.

Схематично хромосому можна подати як прямолінійний відрізок, а локуси – як послідовні ділянки, на які цей відрізок розбитий.

Дії генів проявляються в популяціях.

Популяція – досить велике співтовариство організмів, що схрещуються між собою. Популяції характеризуються набором ланцюжків генів кожного з об'єктів, сукупність яких визначає *генофонд популяції*.

Таким чином, генетичні методи запозичили з біології понятійний апарат, ідею колективного пошуку екстремуму, способи подання генетичної інформації, способи передачі генетичної інформації в послідовності поколінь (генетичні оператори), ідею про переважне розмноження найбільш пристосованих особин [18].

4.3 Узагальнена схема роботи генетичних методів

Сутність генетичного пошуку полягає в циклічній заміні однієї популяції наступною, більш пристосованою [45, 55, 88, 93, 106, 107]. Таким чином, популяція існує не тільки в просторі, але і в часі. Часто можна вважати, що вся популяція складається в просторі і в часі з дискретних поколінь (генерацій, епох) $P_0, P_1, P_2, \dots, P_T$.

Покоління P_{t+1} – це сукупність особин, батьки яких належать поколінню P_t . Покоління P_0 є *початковою популяцією*. Процес формування покоління P_0 називається *ініціалізацією*. Кожне покоління є результатом циклу роботи генетичного методу.

Кожна хромосома (особина, точка в просторі пошуку) оцінюється мірою її пристосованості відповідно до того, наскільки є гарним відповідне їй рішення задачі. *Пристосованість* визначається як обчислена цільова функція (*фітнес-функція*) для кожної з хромосом. Правила *відбору* (*селекції*) прагнуть залишити лише ті точки-рішення, де досягається оптимум цільової функції. Найбільш пристосовані особини дістають можливість відтворювати нащадків за допомогою перехресного *схрещування* з іншими особинами популяції. Це призводить до появи нових особин, які поєднують в собі деякі характеристики, успадковані ними від батьків. Найменш пристосовані особини з меншою ймовірністю зможуть відтворити нащадків, внаслідок чого ті властивості, якими вони володіли, поступово зникатимуть з популяції в процесі еволюції.

Таким чином, з покоління в покоління, гарні характеристики розповсюджуються по всій популяції. Комбінація гарних характеристик від різних батьків іноді може призводити до появи суперпристосованого нащадка (або мутанта), чия пристосованість більша, ніж пристосованість будь-якого з його батьків. Схрещування найбільш пристосованих особин призводить до того, що досліджуються найбільш перспективні ділянки простору пошуку. Зрештою, популяція збагатиться до оптимального рішення задачі.

Після схрещування інколи відбуваються *мутації* – спонтанні зміни в генах, які випадковим чином розкидають точки по всій множині пошуку.

Процес збіжності за рахунок відбору повинен бути більш виражений порівняно з розкидом точок за рахунок мутації і інвертування, інакше збіжність до екстремумів не матиме місця. Збіжність за рахунок відбору не повинна бути занадто швидкою, інакше всі точки можуть зібратися поблизу локального екстремуму, а інший, можливо глобальний, так і не буде знайдено.

Після отримання нащадків за допомогою схрещування та мутації розмір популяції збільшується. Для подальших перетворень число хромосом поточної популяції зменшується до заданого розміру популяції.

Подальша робота генетичного методу є ітераційним процесом застосування *генетичних операторів* до особин наступного покоління. Генетичні оператори необхідні, щоб застосувати принципи спадковості і мінливості до популяції. Такі оператори володіють *властивістю ймовірності*, тобто вони не обов'язково застосовуються до всіх особин, що вносить додатковий елемент невизначеності в процес пошуку рішення. Невизначеність не має на увазі негативного чинника, а є своєрідною мірою свободи роботи генетичного методу.

Схему роботи узагальненого генетичного методу наведено на рис. 4.1.

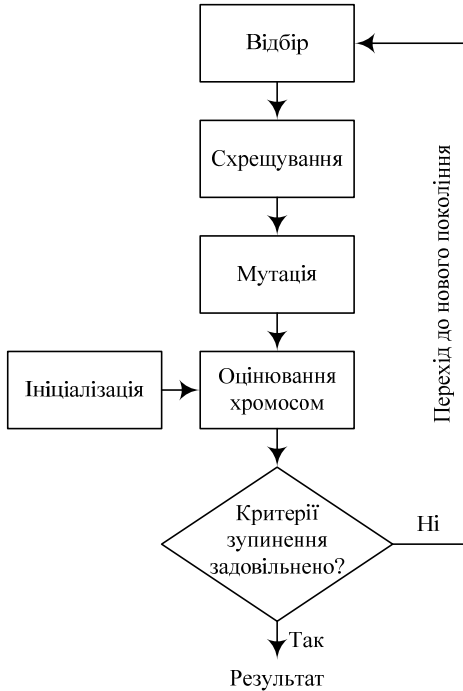


Рисунок 4.1 – Схема роботи узагальненого генетичного методу

Узагальнений метод генетичного пошуку можна записати таким чином.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Виконати ініціалізацію (initialization) початкової популяції особин: $P_t = \{H_1, H_2, \dots, H_N\}$.

Крок 2. Оцінити особини поточної популяції (evaluating) шляхом обчислення їх фітнес-функції $f(H_j)$, $j = 1, 2, \dots, N$.

Крок 3. Перевірити умови закінчення пошуку (termination criteria). Як такі умови можуть бути використані: досягнення максимально допустимого часу функціонування методу, числа ітерацій, значення функції пристосованості і т. ін. Якщо критерії закінчення пошуку задовільнено, тоді виконати перехід до кроку 12.

Крок 4. Збільшити лічильник ітерацій (часу): $t = t + 1$.

Крок 5. Вибрати частину популяції (батьківські особини) для схрещування (selection of parents) P' .

Крок 6. Сформувати батьківські пари (mating) з особин, що відібрані на попередньому кроці.

Крок 7. Схрестити (crossover) вибрані батьківські особини.

Крок 8. Застосувати оператор мутації (mutation) до особин P' .

Крок 9. Обчислити нову функцію пристосованості $f(H_j)$ особин, отриманих в результаті схрещування та мутації.

Крок 10. Сформувати нове покоління шляхом вибору особин, що вижили, виходячи з рівня їх пристосованості (replacing, selection of survivors).

Крок 11. Перейти до кроку 3.

Крок 12. Зупинення.

Таким чином, при реалізації генетичних методів необхідно:

– визначити параметри, що оптимізуються, залежно від вирішуваної задачі, вибрати спосіб кодування (подання в хромосомі) параметрів, що оптимізуються;

– задати цільову функцію;

– визначити правила ініціалізації початкової популяції;

– вибрати оператори відбору, схрещування і мутації, а також задати їх параметри;

– визначити критерії зупинення.

4.4 Класифікація методів еволюційного пошуку

В наш час запропоновано багато різних генетичних методів і в більшості випадків вони є мало схожими на узагальнений генетичний метод. З цієї причини під терміном «генетичні методи» мають на увазі достатньо широкий клас методів, часом мало схожих один на одного [1, 7, 18, 27, 28, 30, 94, 97].

Методи еволюційного пошуку можуть бути класифіковані за різними критеріям.

Укрупнену класифікаційну схему методів еволюційного пошуку наведено на рис. 4.2. У кожному з наведених на рис. 4.2 генетичних методів є можливим застосування різних генетичних операторів, тому варіанти генетичного пошуку, що розрізняються лише генетичними операторами, в схему не включено.

Нижче наведено пропонувану класифікацію еволюційних методів.

1. *За класом методів* еволюційний пошук може бути розподілений на чотири групи: генетичні алгоритми, еволюційні стратегії, генетичне програмування та еволюційне програмування.

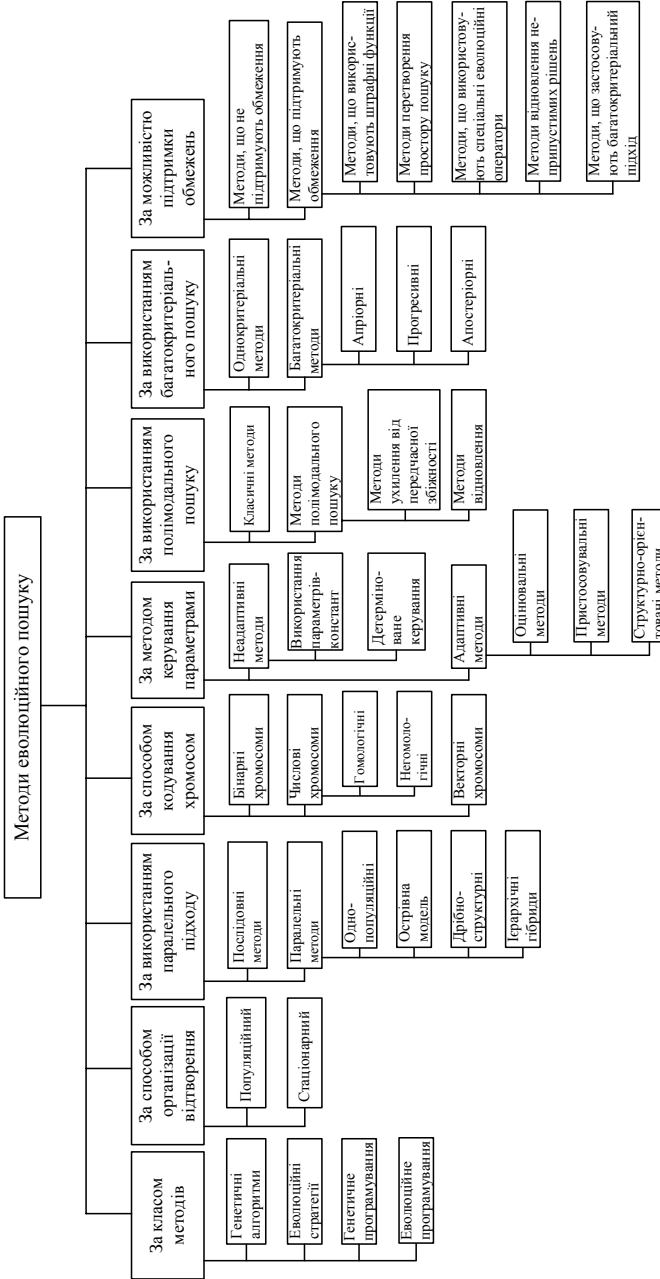


Рисунок 4.2 – Класифікація методів еволюційного пошуку

2. *За способом організації відтворення* виділяють популяційні та стаціонарні еволюційні методи.

Популяційний спосіб передбачає генерацію на кожній ітерації еволюційного пошуку всіх генотипів нової популяції P_{t+1} у відповідності до одного розподілу ймовірностей, обумовленого поточною популяцією P_t і розподілами ймовірностей операторів відбору, схрещування й мутації.

При стаціонарному способі відтворення на кожній ітерації створюється одна або дві нові особини, що заміщають у популяції особини з найменшим значенням фітнес-функції, або інші особини, обрані по деякому евристичному правилу.

3. *За використанням паралельного підходу* методи еволюційного пошуку можуть бути класифіковані на послідовні й паралельні.

Послідовні еволюційні методи використовують єдину популяцію хромосом на кожній ітерації.

У випадку застосування паралельних методів на кожній ітерації використовується декілька підпопуляцій. До паралельних методів відносяться:

- однопопуляційні методи;
- острівна модель еволюційного пошуку;
- дрібноструктурні еволюційні методи;
- ієрархічні гібриди.

4. *За способом кодування хромосом* виділяють еволюційні методи, що використовують бінарні хромосоми, числові хромосоми й векторні хромосоми. У бінарних хромосомах гени можуть приймати два значення. Числові хромосоми містять гени, що приймають будь-які значення в заданому інтервалі. При цьому в гомологічних числових хромосомах допускається наявність генів з однаковими значеннями, а в негомологічних хромосомах всі гени приймають різні значення. Векторні хромосоми складаються з генів, які являють собою вектор цілих чисел.

5. *За методом керування параметрами* еволюційного пошуку виділяють неадаптивні методи й адаптивні методи.

До неадаптивних методів керування параметрами відносяться:

- використання параметрів-констант;
- детерміноване керування параметрами.

Адаптивні методи керування можуть бути класифіковані в такий спосіб:

- оцінювальні методи;
- пристосовувальні методи;
- структурно-орієнтовані методи (розбиття популяції й метод просторової популяційної структури).

6. *За використанням полімодального пошуку* виділяють класичні методи й методи полімодального пошуку.

Методи полімодального пошуку дозволяють виділити декілька оптимумів, розташованих у різних точках простору пошуку. До методів полімодального пошуку відносяться:

- методи ухилення від передчасної збіжності (методи уповільнення генетичної збіжності й методи запобігання появи співпадаючих рішень);
- методи відновлення (методи множинного заміщення, методи перезапуску і фазові методи).

7. *За кількістю критерії пошуку пошуку* виділяють однокритеріальні та багатокритеріальні методи еволюційної оптимізації.

Методи багатокритеріальної еволюційної оптимізації дозволяють виявити оптимум сукупності цільових функцій і розподіляються на:

- апіорні методи (метод агрегувальних функцій, лексикографічне впорядковування, мінімакський метод, метод досягнення заданого значення);
- прогресивні методи;
- апостеріорні методи (популяційний підхід і пошук, заснований на підході Парето).

8. *За можливістю підтримки обмежень* виділяють методи, що не підтримують обмеження, і методи, що підтримують обмеження. До методів, що підтримують обмеження, відносяться:

- методи, що використовують штрафні функції;
- методи перетворення простору пошуку;
- методи, що використовують спеціальні еволюційні оператори;
- методи відновлення неприпустимих рішень;
- методи, що застосовують багатокритеріальний підхід.

4.5 Моделі генетичного пошуку

Окрім узагальненої схеми функціонування генетичного методу, розглянутої вище, використовують також інші технології генетичного пошуку. Вони пов'язані в основному з розпаралелюванням генетичних методів, структуризацією популяції, міграцією хромосом і популяції, застосуванням специфічних генетичних операторів і т. ін. [1, 31, 41, 58, 89]. Вибір моделі генетичного методу залежить від типу розв'язуваної задачі.

4.5.1 Канонічні моделі

Першою канонічною моделлю генетичного методу (canonical GA) є *репродуктивний план Холланда* [72]. Основні етапи процесу еволюції згідно Холланда такі.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Згенерувати початкову популяцію особин. Обчислити пристосованість особин популяції.

Крок 2. Збільшити лічильник ітерацій (часу): $t = t + 1$. Обчислити середню пристосованість особин популяції.

Крок 3. Відібрати двох батьків (хромосоми) для схрещування. При цьому відбір виконується випадково пропорційно значенню фітнес-функції.

Крок 4. Сформувати генотип нащадка: із заданою ймовірністю провести над генотипами вибраних хромосом схрещування.

Крок 5. Застосувати з певною ймовірністю оператор мутації до нащадків.

Крок 6. Обчислити пристосованість хромосом-нащадків.

Крок 7. Зберегти хромосоми-нащадки як члени нової популяції.

Крок 8. У випадку, якщо нова популяція заповнена не повністю, виконати перехід до кроку 3.

Крок 9. Якщо кількість ітерацій досягла максимально допустимого значення $t = T$, тоді виконати перехід до кроку 10, інакше – виконати перехід до кроку 2.

Крок 10. Зупинення.

Часто можна зустріти опис *простого генетичного методу* (simple GA), який вперше описано Д. Гольдбергом на основі робіт Холланда. Простий генетичний метод відрізняється від репродуктивного плану Холланда тим, що використовує або пропорційний, або турнірний відбір.

До канонічних можна віднести також *генетичний метод Л. Девіса*.

Крок 1. Ініціалізувати початкову популяцію хромосом.

Крок 2. Оцінити кожну хромосому в популяції.

Крок 3. Створити нові хромосоми за допомогою схрещування поточних хромосом. Застосувати мутацію й рекомбінацію.

Крок 4. Видалити найменш пристосовані хромосоми з популяції, щоб звільнити місце для нових хромосом.

Крок 5. Оцінити нові хромосоми і вставити їх в популяцію.

Крок 6. Якщо час вичерпано, то виконати зупинення і повернення до найкращої хромосоми, якщо ні, то виконати перехід до кроку 3.

Порівняння описаних генетичних методів Гольдберга, Холланда і Девіса показує, що в них реалізована одна основна ідея моделювання еволюції з деякими модифікаціями.

Таким чином, модель канонічних генетичних методів має такі характеристики:

- фіксований розмір популяції;
- фіксована розрядність генів;
- пропорційний, турнірний відбір або відбір за допомогою рулетки;
- особи для схрещування вибираються випадковим чином;
- одноточкове схрещування і одноточкова мутація;
- наступне покоління формується з нащадків поточного покоління без «елітизму». Нашадки займають місця своїх батьків.

4.5.2 Genitor

У моделі Genitor (Д. Уїтлі) використовується специфічна стратегія відбору [18].

Крок 1. Виконати ініціалізацію початкової популяції і оцінити пристосованість її особин.

Крок 2. Вибрати випадковим чином дві особини.

Крок 3. Схрестити обрані особини. В результаті схрещування виходить лише один нащадок, пристосованість якого оцінюється і який займає місце найменш пристосованої особини.

Крок 4. Якщо кількість ітерацій досягла максимально допустимого значення $t = T$, то виконати перехід до кроку 5, інакше – виконати перехід до кроку 2.

Крок 5. Зупинення.

Таким чином, на кожному кроці в популяції оновлюється лише одна особина.

Можна виділити такі характерні особливості моделі Genitor:

- фіксований розмір популяції;
- фіксована розрядність генів;
- особини для схрещування вибираються випадковим чином;
- обмежень на тип схрещування і мутації немає;
- в результаті схрещування особин виходить один нащадок, який займає місце найменше пристосованої особини.

4.5.3 Гібридний генетичний метод

Використання гібридного генетичного методу дозволяє об'єднати переваги генетичного пошуку з перевагами класичних методів оптимізації [41]. Відомо, що генетичні методи дозволяють знаходити гарне рішення, але знаходження оптимального рішення часто виявляється набагато важчою задачею через стохастичність принципів роботи генетичного пошуку. Тому виникла ідея використовувати генетичний метод на початковому етапі для ефективного звуження простору пошуку навколо глобального екстремуму, а потім, взявши кращу особину за початкову точку пошуку, застосувати один з класичних методів оптимізації. Можливий і інший варіант: на початковому етапі застосовується класичний пошук, а потім використовуються генетичні методи.

Основні *стратегії взаємодії класичних пошукових методів і генетичного пошуку* наведено нижче.

1. Стратегія «класичний пошук – генетичний пошук» передбачає знаходження масиву початкових рішень за допомогою традиційних пошуко-

вих методів, після чого одержаний масив початкових рішень використовується як початкова популяція при генетичному пошуку.

2. У стратегії «генетичний пошук – класичний пошук» рішення, одержане за допомогою генетичного пошуку, є початковою точкою для класичного пошукового методу.

3. Стратегія «класичний пошук – генетичний пошук – класичний пошук» передбачає послідовне виконання першої і другої стратегій.

4. Стратегія «генетичний пошук – класичний пошук – генетичний пошук» передбачає послідовне виконання другої і першої стратегій.

Характеристики гібридного методу:

- фіксований розмір популяції;
- фіксована розрядність генів;
- можливість будь-яких комбінацій стратегій відбору і формування наступного покоління;
- відсутність обмежень на тип операторів схрещування і мутації;
- генетичні методи застосовуються на початковому етапі, а потім в роботу включається класичний метод оптимізації, або навпаки.

4.5.4 Генетичний метод зі змінним часом життя особин

Генетичний метод зі змінним часом життя особин [45] передбачає формування наступного покоління з урахуванням часу життя хромосом.

Крок 1. Виконати ініціалізацію початкової популяції.

Крок 2. Обчислити значення функції пристосованості для кожної хромосоми поточного покоління.

Крок 3. Виконати схрещування і мутацію над особинами поточного покоління, внаслідок чого розмір популяції збільшується.

Крок 4. Розрахувати час життя особин за формулою:

$$TTL_j^{(t+1)} = a^{(t+1)} + b^{(t+1)} \cdot \frac{f(H_j) - f_{\min}}{f_{\max} - f_{\min}},$$

де $TTL_j^{(t+1)}$ – час життя j -ої хромосоми на $(t+1)$ ітерації; $a^{(t+1)}$ та $b^{(t+1)}$ – константи, що задаються як параметри методу, $a^{(t+1)}$ – мінімальний час життя на $(t+1)$ -й ітерації, $b^{(t+1)} \geq 0$ – коефіцієнт масштабування нормованого значення функції пристосованості; f_{\min} та f_{\max} – мінімальне та максимальне значення функції пристосованості в поточній популяції.

Змінний час життя особин дозволяє керувати генетичним пошуком при заданих параметрах $a^{(t+1)}$ та $b^{(t+1)}$. Так при $a^{(t+1)} = 1$ та $b^{(t+1)} = 0$ метод зводиться до випадку простого генетичного методу. При $a^{(t+1)} = b^{(t+1)} = 1$ метод зводиться до стратегії елітизму.

Окрім наведеного вище способу розрахунку часу життя j -ої хромосоми TTL_j , можна використовувати один із трьох способів.

$$1. \text{ Пропорційний: } TTL_j = \min \left\{ life_{\max}; life_{\max} - \eta \frac{f_j}{f_{\text{cp}}} \right\},$$

де $\eta = 0,5(life_{\max} - life_{\min})$; $life_{\max}$ – максимально можливий час життя; $life_{\min}$ – мінімально можливий час життя; f_j – значення фітнес-функції j -ої хромосоми; f_{cp} – середнє значення фітнес-функції хромосом популяції.

$$2. \text{ Лінійний: } TTL_j = life_{\max} - 2\eta \frac{f_j - f_{\min}}{f_{\max} - f_{\min}},$$

де f_{\min} – мінімальне значення фітнес-функції в популяції; f_{\max} – максимальне значення фітнес-функції в популяції.

$$3. \text{ Білінійний: } TTL_j = \begin{cases} life_{\min} + \eta \frac{f_{\max} - f_j}{f_{\max} - f_{\text{cp}}}, & \text{якщо } f_j \geq f_{\text{cp}}; \\ 0,5(life_{\min} + life_{\max}) + \eta \frac{f_{\text{cp}} - f_j}{f_{\text{cp}} - f_{\min}}, & \text{якщо } f_j < f_{\text{cp}}. \end{cases}$$

Крок 5. Сформувані наступне покоління в залежності від часу життя особин.

Порівняння особин проводиться між батьками та нащадками або між декількома поколіннями залежно від розміру архіву, в якому зберігаються особини з найбільшим часом життя. Розмір архіву може бути рівним, меншим або більшим за розмір популяції.

При фіксованому розмірі популяції видаляються хромосоми з найменшим часом життя.

При динамічному розмірі популяції видаляються хромосоми, у яких закінчився час життя. Динамічний розмір популяції визначається співвідношенням:

$$N^{(t+1)} = N^{(t)} \cdot (1 + R^{(t+1)}) - D^{(t+1)},$$

де $N^{(t+1)}$ та $N^{(t)}$ – розміри нової та поточної популяції; $R^{(t+1)}$ – рівень репродукції на $(t+1)$ -й ітерації, що є відношенням кількості одержаних на $(t+1)$ -й ітерації особин-нащадків до кількості батьків; $D^{(t+1)}$ – кількість хромосом батьків та нащадків, що видаляються на $(t+1)$ -й ітерації.

При $R^{(t+1)} = 1$, $D^{(t+1)} = N^{(t)}$ метод зводиться до випадку простого генетичного методу.

Крок 6. Перевірити умови закінчення пошуку. Якщо критерії закінчення задовільнено, перейти до кроку 7. В іншому випадку перейти до кроку 2.

Крок 7. Зупинення.

4.5.5 Мобільний генетичний метод

Мобільний генетичний метод (messy genetic method) не накладає обмежень на позиціонування генів усередині хромосоми, допускаючи їх різні перестановки, і, в результаті, не потребує попередньої інформації про найбільш оптимальне розташування бітів усередині хромосоми, що важливо при практичному застосуванні [39].

Мобільний генетичний метод має такі відмінності від традиційного генетичного пошуку:

- використання хромосом змінної довжини, які можуть бути або пере- або недовизначені по відношенню до вирішуваної задачі;
- введення правил читання або експресії генів;
- використання операторів CUT (розрізання) та SPLICE (зчеплення) замість традиційних операторів схрещування, що оперують над генами фіксованої довжини;
- конкуренція між будівельними блоками генотипу для відбору найбільш оптимальних хромосом;
- розділення еволюції на дві фази: попередня фаза та фаза процесингу.

Мобільний генетичний метод може бути поданий як послідовність кроків 1–8.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Виконати ініціалізацію початкової популяції.

Крок 2. Виконати попередню фазу для збільшення частини оптимальних елементів в популяції.

Крок 3. Перевірити умови закінчення пошуку. Якщо критерії закінчення задовільнено, тоді перейти до кроку 8.

Крок 4. Відібрати особини для схрещування.

Крок 5. Виконати схрещування за допомогою операторів CUT (розрізання) та SPLICE (зчеплення).

Крок 6. Виконати мутацію.

Крок 7. Збільшити лічильник ітерацій (часу): $t = t + 1$. Перейти до кроку 3.

Крок 8. Зупинення.

Хромосома в такому методі кодується списком пар <номер гену; значення гену>, знімаючи тим самим обмеження, пов'язані з фіксованим положенням генів всередині хромосоми. Причому одному гену може відповідати декілька значень (перевизначена або надмірна хромосома) або жодного значення (недовизначена або неповна хромосома). Таким чином, можливе існування хромосоми типу $\{<1; 5>, <2; 10>, <1; 7>, <5; 15>, <4; 29>\}$, у якої першому гену відповідають два значення (5 та 7), а значення третього гена відсутнє.

Для інтерпретації хромосом змінної довжини вводиться *процедура експресії* (читання) генів, яка використовує правило «зліва направо», згідно якого хромосома зчитується зліва направо, та ген, який розташований зліва, вважається *активним* в даній хромосомі. Наприклад, хромосома $\{<1; 5>, <2; 10>, <1; 7>, <5; 15>, <4; 29>\}$ за рахунок експресії перейде в хромосому $\{<1; 5>, <2; 10>, <5; 15>, <4; 29>\}$, оскільки друга версія першого гена не буде використана відповідно до розглянутого правила. Правило «зліва направо» було вибрано замість різних схем, які використовують поняття ефективності генів, оскільки при застосуванні таких схем гени, які одержали велику перевагу на початку еволюції (великі значення оптимальності), можуть заблокувати прояв генів, які на даній стадії є менш ефективними, але є будівельними блоками шуканого оптимуму.

Для усунення проблеми недовизначення використовується інший підхід. У деяких задачах недовизначеність не є проблемою, оскільки будь-яка структура незалежно від розміру може бути інтерпретована природним способом. У задачах параметричної оптимізації з певним числом параметрів всі змінні повинні бути подані в цільовій функції для її оптимізації.

У таких випадках мобільний генетичний метод заповнює недовизначені позиції хромосоми ділянкою найбільш конкурентноздатної хромосоми-попередниці – хромосоми, яка є оптимальною на попередньому рівні еволюції. Ідея полягає в тому, що найкраща (конкурентоспроможна) хромосома, що одержана на одному рівні еволюції та призводить до найкращої оцінки за допомогою цільової функції на цьому ж рівні, являтиме собою кращий початковий матеріал для формування оптимальних рішень в наступному поколінні. Таким чином, відновлюючи ділянки хромосоми, можна одержувати блоки, що створюють структури, які відповідають кращим рішенням. Тобто, починаючи з першої ітерації знаходиться елемент популяції, оптимальний для цієї ітерації. Він використовуватиметься як будівельний блок генотипу при формуванні оптимальних рішень в другому поколінні, заповнюючи недовизначені гени і т. ін. Таким чином, метод піднімається по «сходинах» проміжних локально-оптимальних рішень, одночасно покращуючи шукане глобальне рішення.

Цикл роботи мобільного генетичного методу складається з трьох етапів: ініціалізації, попередньої фази та фази процесингу.

Цей внутрішній цикл може бути виконаний на будь-якій ітерації, гарантуючи цим, що будівельні блоки (ділянки хромосоми) наступного покоління будуть достатньо конкурентоспроможними, щоб забезпечити еволюцію.

Ініціалізація виконується за допомогою створення популяції, яка містить по одній копії всіх генів довжини k . Цим гарантується наявність в популяції всіх необхідних будівельних блоків генотипу. Коли виконується обробка початкової популяції, то за рахунок застосування генетичних операторів CUT та

SPLICE формуються оптимальні структури хромосом. Для вибору найбільш придатного блоку необхідно оцінити пристосованість кожного члена популяції. Розмір популяції при такій схемі ініціалізації визначається як $N = 2^k \cdot (L/k)$ (оскільки розмірність хромосоми дорівнює L , то існує L/k комбінацій генів розміру k , та для кожної комбінації існує 2^k різних бінарних комбінацій алелей).

Попередня фаза призначена для збільшення частини оптимальних елементів в популяції. Виконується лише відбір, інші генетичні оператори (розрізання, зчеплення, мутація) не використовуються. Необхідність оцінки кожного елементу початкової популяції, звичайно, не є оптимальною обчислювальною процедурою. Оцінка виконується лише один раз під час попередньої фази, оскільки внаслідок незмінності хромосом значення їх оптимальності зберігається. Після цього відбувається відбір найкращих хромосом, щоб збільшити їх частку в популяції. Одночасно через певні інтервали часу кількість елементів популяції скорочується, щоб досягти рівня, який зберігатиметься постійним на наступній стадії еволюції.

Фаза процесингу. Після збагачення популяції конкурентоспроможними елементами, на попередньому етапі, проводиться обробка популяції, яка нагадує стандартний генетичний метод. Впродовж цієї фази відбір використовується спільно зі схрещуванням. Можлива також і мутація хромосом.

Для рекомбінації хромосом змінної довжини застосовуються два оператори CUT та SPLICE замість звичайного оператора схрещування з фіксованим положенням однієї або декількох точок схрещування.

Оператор CUT застосовується до хромосоми з ймовірністю $p_c = (L - 1) \cdot p_k$, де L – довжина хромосоми, p_k – задана константа.

Оператор SPLICE з фіксованою ймовірністю p_s об'єднує дві хромосоми.

Оператори CUT та SPLICE мають два граничних типи поведінки. Спочатку, коли хромосоми короткі, переважає зчеплення рядків. Пізніше, коли хромосоми стають достатньо довгими, переважає розрізання рядків.

Таким чином, при невеликій довжині хромосом ефекти руйнування незначні, при збільшенні довжини ймовірність невиконання експресії значно збільшується.

4.5.6 Паралельні та багаторівневі генетичні методи

В генетичних методах використовуються два підходи до структуризації популяції [1, 41].

Перший підхід полягає в тому, що після конструювання популяції вона випадковим або заданим чином розбивається на декілька підпопуляцій. Простий генетичний метод або будь-яка його модифікація застосовується усередині кожної підпопуляції. При цьому можливе випадкове або спрямоване переміщення хромосом між будь-якими або заданими популяціями.

При використанні другого підходу після ініціалізації популяції для кожної хромосоми визначається її просторове місцезнаходження в популяції. Далі можливе застосування розбиття популяції на підпопуляції. Вибір пар для використання операторів схрещування залежить від «близькості» рішень в просторі пошуку.

Можлива велика кількість комбінацій і модифікацій, цих підходів. Причому їх ефективність встановлюється в основному емпірично.

Паралельні генетичні методи, як правило, виконуються на багатопроекторних машинах з метою прискорення їх роботи. При цьому загальний час T для виконання однієї ітерації (покоління) паралельного генетичного методу визначається за формулою:

$$T = \frac{NT_f}{C} + \rho(C-1)T_c,$$

де N – розмір популяції; T_f – час, необхідний для обчислення значення фітнес-функції однієї хромосоми; C – кількість процесорів; ρ – параметр, що відображає вплив методу відбору або паралелізму; T_c – середній час для передачі даних між процесорами.

4.5.6.1 Однопопуляційні генетичні методи

В однопопуляційних генетичних методах (single-population master-slave GA, master-slave GA), використовують багатопроекторні системи (рис. 4.3 а). При цьому один процесор є головним (master) і керує процесом виконання генетичного пошуку (відбір, схрещування, мутація, формування нового покоління). Обчислення фітнес-функції розподіляється серед підлеглих (slaves) головному процесорів. Таким чином, головний процесор займається безпосередньо виконанням генетичних операторів в той час, як решта процесорів виконує обчислення значень фітнес-функцій хромосом, переданих їм головним процесором. Тому дана модель генетичного пошуку іноді називається моделлю «ведучий-ведений».

4.5.6.2 Острівна модель

Острівна модель (island GA, multiple population GA, multiple-deme GA, distributed GA, coarse-grained GA) використовує декілька підпопуляцій, які обмінюються особинами з деякою заданою частотою (рис. 4.3, б). Такий обмін особинами називається міграцією [18]. Окрім частоти міграції, параметрами такого виду генетичного пошуку є також кількість мігруючих особин, напрям міграції, й метод відбору особин для міграції.

Ця модель генетичного методу має такі властивості:

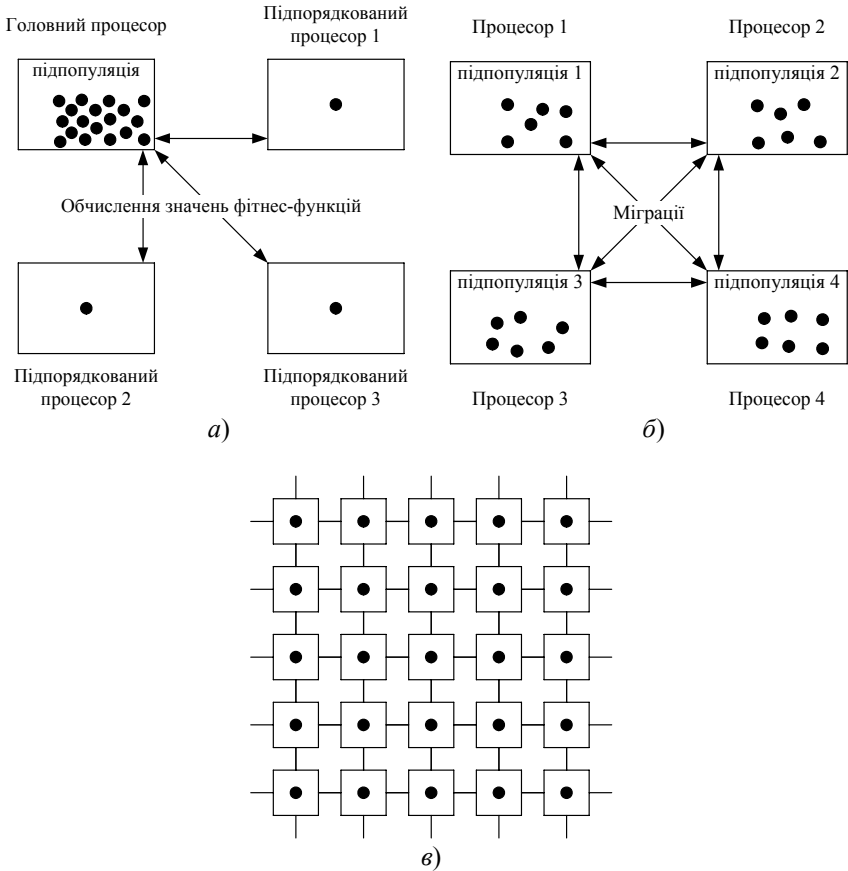


Рисунок 4.3 – Паралельні генетичні методи:
 а) модель «ведучий-ведений»; б) острівна модель; в) дифузійна модель

- наявність декількох популяцій, як правило, однакового фіксованого розміру;
- фіксована розрядність генів;
- будь-які комбінації стратегій відбору та формування наступного покоління в кожній популяції. Можливі ситуації, коли в різних популяціях використовуються різні комбінації стратегій відбору та формування нового покоління;
- обмежень на тип схрещування та мутації немає;
- випадковий обмін особинами між «островами».

Якщо міграція буде занадто активною, то особливості острівної моделі буде згладжено, й вона буде не дуже сильно відрізнятися від моделей генетичних методів без паралелізму.

Розрізняють синхронні й асинхронні паралельні генетичні методи. У синхронних методах обмін особинами при міграції виконується в один і той самий час, як правило, кожні p (задане число) ітерацій. При асинхронному генетичному пошуку кожна підпопуляція незалежно визначає, коли відправляти емігруючі особини та коли одержувати особини, що іммігрують в дану підпопуляцію з інших підпопуляцій.

Деякі методи обмінюються особинами, інші – клонують особин.

Острівні генетичні методи, як правило, швидше за однопопуляційні. Крім того, такі методи можуть виконуватися і на однопроцесорній ЕОМ.

4.5.6.3 Дрібноструктурні генетичні методи

Дрібноструктурні генетичні методи (fine-grained GA, cellular GA, diffusion-model GA) складаються з єдиної просторово структурованої популяції (рис. 4.3 в). Як правило, популяція подається у вигляді прямокутної сітки, у вузлах якої розташовані особини [1]. В ідеалі, на кожну особину виділяється окремий процесор для обчислення її фітнес-функції. Батьківському пару в даному методі можуть складати лише сусідні особини. Таким чином, характеристики кращої особини розповсюджуються по сітці не так швидко, як в інших методах, не дозволяючи тим самим деяким хромосомам домінувати на початкових ітераціях виконання генетичного пошуку.

4.5.6.4 Ієрархічні гібриди

Ієрархічні гібриди є поєднанням острівної моделі генетичного пошуку з моделлю «ведучий-ведений» або дрібноструктурною моделлю. На верхньому рівні такі моделі використовують острівну модель, на нижніх рівнях застосовується дрібноструктурна модель або модель «ведучий-ведений» [1].

4.5.6.5 Дворівневий генетичний метод

У дворівневому генетичному методі DAGA2 реалізовано ідею багаторівневої еволюції: генетичний метод використовується на двох рівнях [72]. На першому рівні DAGA2 виконує паралельний прогін генетичного методу (простий генетичний метод або будь-яка його модифікація), доки він не почне сходитися. Потім в найкращих рішеннях (хромосомах) кожної популяції визначаються найближчі один до одного об'єкти (гени), які утворюють кластери. На основі кластерів створюється нова хромосома як об'єкт для другого рівня DAGA2.

DAGA2 може діяти як паралельний генетичний метод зі схрещуванням локальної області або на основі острівної моделі генетичного методу. DAGA2 діє як генетичний метод, що адаптується, в сенсі «виживання» найпристосованішого рішення.

Архітектура DAGA2 є аналогічною архітектурі генетичного методу «вприскування». У цій архітектурі хромосоми з підпопуляцій в грубій (неточній) моделі задачі мігрують в інші підпопуляції в негрубій (точній) моделі задачі.

Архітектура DAGA2 може бути модифікована на будь-яке число рівнів залежно від пам'яті й часових обмежень на отримання результату.

4.5.7 Генетичний пошук зі зменшенням розміру популяції

В традиційних генетичних методах результуюча популяція, як правило, складається з особин, значення фітнес-функцій яких є однаковими або близькими, й при цьому тенденція до деякого кращого рішення звичайно складається за достатньо велику кількість популяцій (ітерацій) до закінчення роботи методу. Тому в генетичному методі зі зменшенням розміру популяції [75] при наблизенні до оптимального рішення знижується чисельність популяції.

Загальна ідея в генетичному методі із зменшенням розміру популяції полягає в специфічному відборі членів популяції для схрещування. Відбір проводиться, орієнтуючись на найкраще й найгірше значення фітнес-функції в популяції. При цьому враховується різниця між даними граничними значеннями, за рахунок чого досягається популяція невеликої чисельності з найкращими значеннями фітнес-функції.

Таким чином, генетичний метод із зменшенням розміру популяції можна записати таким чином.

Крок 1. Встановити початкові параметри методу – параметри, пов'язані з вирішуваною оптимізаційною задачею, а також встановити лічильник ітерацій (часу): $t = 0$.

Крок 2. Виконати генерацію початкової популяції.

Крок 3. Для кожної особини одержаної популяції розрахувати значення фітнес-функції.

Крок 4. Створити нову популяцію.

Крок 4.1. Провести відбір особин для подальшого схрещування. Відбір відбувається на основі найкращого та найгіршого значень фітнес-функції в поточній популяції, при цьому враховується різниця між даними значеннями. За рахунок цього досягається відбір найбільш пристосованих до розв'язання даної задачі особин. Особина вибирається для схрещування при виконанні такої умови:

$$e^{\frac{f_j - f_{worst}^t}{f_{best}^t - f_{worst}^t}} < \text{rand}(e^{-1}; 1), j = \overline{1; N},$$

де N – розмір поточної популяції; f_j – значення фітнес-функції j -ої особи-ни; f_{best}^t – найкраще значення фітнес-функції поточної популяції t ; f_{worst}^t – найгірше значення фітнес-функції поточної популяції t ; $\text{rand}(e^{-1}; 1)$ – випадкове число від e^{-1} до 1.

Варто відзначити, що особи, які мають найкраще значення фітнес-функції f_{best}^t , відбираються для схрещування автоматично без перевірки даної умови.

Дана умова носить такий сенс: чим далі знаходиться значення фітнес-функції i -ої особи від гіршого значення у всій популяції щодо розкиду між всіма значеннями фітнес-функції поточної популяції, тим більша ймовірність, що i -а особина буде вибрана для подальшого схрещування.

Таким чином, чим більше розкид між найкращим та найгіршим значеннями фітнес-функції, тим більше особин відбиратиметься для схрещування, а чим менше буде різниця між найгіршим та найкращим значеннями фітнес-функції популяції (коли значення фітнес-функцій наближуються до одного кращого варіанту), тим менше особин вибиратиметься для подальшого схрещування, і за рахунок цього зменшуватиметься розмір популяції.

Крок 4.2. Виконати схрещування відібраних в нову популяцію особин. Таким чином, в нову популяцію потрапляють особи, одержані в результаті виконання оператора відбору, а також їх нащадки, одержані за допомогою оператора схрещування.

Крок 5. Застосувати оператор мутації до одержаної популяції.

Крок 6. Розрахувати значення фітнес-функції для всіх особин нової популяції.

Крок 7. Збільшити лічильник ітерацій (часу): $t = t + 1$.

Крок 8. Перевірити умови закінчення пошуку (час, число ітерацій, значення фітнес-функції і т. ін.). Якщо критерії закінчення задовільнено, перейти до кроку 9, в іншому випадку перейти до кроку 4.

Крок 9. Зупинення.

4.6 Ініціалізація та запуск генетичного пошуку

4.6.1 Кодування параметрів, що оптимізуються

Будь-який організм може бути поданий своїм *фенотипом*, який фактично визначає, чим є об'єкт в реальному світі, і *генотипом*, який містить всю інформацію про об'єкт на рівні хромосомного набору [45, 72]. При цьому кожен ген, тобто елемент інформації генотипу, має своє відображення у фенотипі. Таким чином, для розв'язання задач необхідно подати кожену

ознаку (атрибут, вхідну змінну, параметр) об'єкту у певному закодованому вигляді для використання в генетичному методі. Все подальше функціонування механізмів генетичного пошуку проводиться на рівні генотипу, що дозволяє обійтися без інформації про внутрішню структуру об'єкту, що і обумовлює широке застосування генетичного пошуку в різних задачах.

За методами подання генів хромосоми можна умовно поділити на три групи.

1. *Бінарні хромосоми* – хромосоми, гени яких можуть приймати значення 0 або 1.

2. *Числові хромосоми* – гени можуть приймати значення в заданому інтервалі.

Числові хромосоми можна поділити на гомологічні і негомологічні.

Гомологічними називають хромосоми, що мають загальне походження, є морфологічно і генетично схожими, і тому не створюють неприпустимих рішень при застосуванні стандартних генетичних операторів. У гомологічних числових хромосомах кожен ген може приймати будь-які значення в заданому інтервалі. Для різних генів можуть бути задані різні інтервали. Бінарна хромосома є гомологічною числовою хромосомою, кожен ген якої може приймати цілі значення в інтервалі $[0; 1]$.

У *негомологічних* хромосомах гени можуть приймати значення в заданому інтервалі; при цьому інтервал однаковий для всіх генів, але в хромосомі не може бути двох генів з однаковим значенням. Для негомологічних хромосом застосовуються різні спеціальні генетичні оператори, що не створюють неприпустимих рішень.

3. *Векторні хромосоми* – хромосоми, гени яких є вектором цілих чисел.

Ген у векторних хромосомах має властивості негомологічної хромосоми, тобто числа у векторі можуть приймати значення в заданому інтервалі, і вектор не може містити двох однакових чисел. Проте, хоча гени у векторних хромосомах є негомологічними, самі векторні хромосоми є гомологічними.

Процес кодування параметрів, що оптимізуються, можна виконати як таку послідовність кроків.

Крок 1. Визначити налагоджувані параметри – гени.

Наведемо варіанти кодування генів в деяких задачах, що вирішуються за допомогою генетичних методів:

- оптимізація функцій: гени – незалежні змінні;
- апроксимація: гени – параметри апроксимуючих функцій;
- задача відбору інформативних ознак: гени є прапором інформативності ознак (наприклад, якщо значення гена дорівнює одиниці, то відповідна йому ознака вважається інформативною);

– настройка ваг штучної нейронної мережі: гени відповідають синаптичним вагам нейронів;

– штучне життя (artificial life): гени відповідають характеристикам особини (сила, швидкість, і т. ін.), також повинні бути незмінні гени, що позначають тип особини (наприклад, рослина або тварина);

– задача про найкоротший шлях: гени – пункти пересування. Вся хромосома цілком є маршрутом з початкової точки в кінцеву, причому таким, що не завжди існує.

Крок 2. Визначити число розрядів в кожному гені.

Крок 3. Обрати метод кодування.

Слід врахувати, що дуже велика довжина кодування прискорює процес збіжності всіх членів популяції до кращого знайденого рішення. Часто такий ефект є небажаним, оскільки при цьому велика частина простору пошуку залишається недослідженою. Передчасна збіжність може не привести до оптимального рішення, крім того, швидка збіжність до однієї області не гарантує виявлення декількох рівних екстремумів. До того ж застосування довгих кодувань зовсім не гарантує, що знайдене рішення матиме необхідну точність, оскільки цього, в принципі, не гарантує сам генетичний метод.

Тому в питанні вибору оптимальної довжини кодування потрібно досягти деякого компромісного рішення – з одного боку довжина хромосоми повинна бути достатньо великою, щоб все-таки забезпечити швидкий пошук, з іншого боку – за можливістю малою, щоб не допускати передчасної збіжності та залишити методу шанс відшукати декілька оптимальних значень.

Невдалий вибір впорядкування й кодування генів в хромосомі може викликати передчасну збіжність до локального оптимуму. Для подолання цього недоліку можна обирати спосіб кодування, ґрунтуючись на додатковій інформації про задачу.

Варто відзначити, що використання різних варіантів кодування розподіляє точки в просторі пошуку по-різному. У різновиді генетичного пошуку, що найбільш часто зустрічається, для подання генотипу об'єкту застосовуються *бітові рядки*. При цьому кожному атрибуту об'єкту у фенотипі відповідає один ген в генотипі об'єкту. *Ген* є бітовим рядком, найчастіше фіксованої довжини, який є значенням цієї ознаки.

Кількість розрядів r в гені для кодування ознаки визначається за формулою:

$$r = \text{ceil} \left(\log_2 \left(\frac{w_{\max} - w_{\min}}{\varepsilon} \right) \right),$$

де $\text{ceil}(x)$ – найближче більше або рівне x ціле число; w_{\max} та w_{\min} – максимально і мінімально можливі значення кодуваної ознаки (параметра, незалежної змінної); ε – задана похибка, що допускається при кодуванні ознаки.

Розрядність хромосоми L визначається як сума розрядностей генів. У випадку, якщо задані однакові значення w_{\max} , w_{\min} та ε для всіх p генів, розрядність хромосоми може бути обчислена за формулою $L = p \cdot r$.

Після того, як обрано параметри, їх число і розрядність, необхідно вирішити, як безпосередньо записувати дані, тобто *обрати метод кодування*. Можна використовувати *звичайне кодування* або *коди Грея*. Не дивлячись на те, що коди Грея спричиняють неминуче кодування (декодування) даних, вони дозволяють уникнути деяких проблем, що з'являються в результаті звичайного кодування.

Перевага коду Грея полягає в тому, що коли два числа відрізняються на одиницю, то й їхні двійкові коди розрізняються лише на один розряд, а при звичайному кодуванні такого ефекту немає.

Кодування (декодування) із бінарного коду в код Грея можна здійснювати таким чином.

Крок 1. Скопіювати старший розряд кодованого (декодованого) числа в старший розряд декодованого (кодованого) числа.

Крок 2. Виконати перетворення за формулами:

– з двійкового коду в код Грея: $G[i] = \text{XOR}(B[i+1], B[i])$;

– з коду Грея в двійковий: $B[i] = \text{XOR}(B[i+1], G[i])$,

де $G[i]$ – i -й розряд коду Грея, а $B[i]$ – i -й розряд бінарного коду.

Кодування ознак, яким відповідають числа з плаваючою крапкою.

Найпростіший спосіб кодування – використання бітового подання (двійкове кодування), при якому значення незалежної змінної w розраховується як:

$$w = w_{\min} + \frac{w_{\max} - w_{\min}}{2^L - 1} \sum_{i=1}^L h_{ij} \cdot 2^{L-i}.$$

де h_{ij} – значення i -го гену j -ї хромосоми, $w_{\min} \leq h_{ij} \leq w_{\max}$.

При розв'язуванні p -мірних задач хромосома містить закодовані значення p параметрів, тому наведена вище формула використовується p разів для декодування кожного з параметрів розв'язуваної задачі.

Проте такий варіант має ті ж недоліки, що і при кодуванні цілих чисел. Тому на практиці для кодування дійсних чисел, як правило, використовується код Грея, і застосовується така послідовність дій.

Крок 1. Розбити весь інтервал допустимих значень ознаки на ділянки з необхідною точністю.

Крок 2. В якості значення гена вибрати ціле число, що визначає номер інтервалу (використовуючи код Грея).

Крок 3. Як значення параметра прийняти число, що є серединою цього інтервалу.

4.6.2 Визначення цільової функції

Рішення задачі (хромосоми) оцінюються за допомогою обчислення значень цільової функції. Вигляд цільової функції визначається в першу чергу самою задачею, для розв'язання якої використовується генетичний пошук. У загальному випадку цільова функція – це «чорний ящик», організація процесів в якому не позначається на ході виконання генетичного пошуку.

У випадку якщо генетичний пошук використовується для знаходження екстремумів математичної функції, то ця функція, як правило, і використовується як цільова.

Складнішою є організація цільової функції для задач, не пов'язаних з оптимізацією математичних функцій. В такому випадку в якості цільової функції може бути використаний окремий алгоритм. Наприклад, при використанні генетичного пошуку для відбору інформативних ознак в якості цільової функції може використовуватися помилка прогнозування або класифікації по математичній моделі, що відповідає аналізованій хромосомі.

В процесі генетичного пошуку може виникнути ситуація, коли хромосоми популяції мають дуже великий розкид значень цільових функцій. Така ситуація призводить до того, що ймовірність відбору більшості хромосом для схрещування зводиться до нуля через наявність хромосоми із значенням цільової функції, що різко відрізняється від середнього.

Тому необхідно привести значення цільової функції хромосом в популяції до єдиного порядку. Для цього цільова функція O відображається (scaling) у фітнес-функцію f шляхом перерахунку її значень для кожної хромосоми.

Для обчислення фітнес-функції використовуються такі підходи:

– стандартна цільова функція: $f_c(H_j) = O(H_j) - O_{\min}$ при мінімізації функції або $f_c(H_j) = O_{\max} - O(H_j)$ для випадку максимізації, де O_{\max} – найбільше значення цільової функції в поточній популяції; O_{\min} – найменше значення цільової функції. Чим менша величина $f_c(H_j)$, тим хромосома H_j вважається більш пристосованою;

– модифікована цільова функція: $f_m(H_j) = \frac{1}{1 + f_c(H_j)}$.

Значення такої цільової функції лежить в інтервалі $[0; 1]$;

– нормалізована цільова функція: $f_n(H_j) = \frac{f_m(H_j)}{\sum_{j=1}^N f_m(H_j)}$,

де N – розмір поточної популяції.

Також для обчислення фітнес-функції можуть бути використані такі формули:

$$f(H_j) = aO(H_j) + b \text{ або}$$

$$f(H_j) = O(H_j) - (O_{\text{сер.}} - c\sigma) \text{ або} \\ f(H_j) = O^k(H_j),$$

де a, b, c і k – константи, що залежать від вирішуваної задачі; $O_{\text{сер.}}$ – середнє значення цільової функції на поточній ітерації; σ – середньоквадратичне відхилення значень цільової функції для поточної популяції. Проте використання таких формул може привести до негативного наслідку у вигляді появи хромосом з від’ємними значеннями фітнес-функції. Для запобігання появі хромосом з від’ємними значеннями фітнес-функції, всім таким хромосомам ставиться у відповідність нульове значення фітнес-функції.

Можна відзначити, що обчислення фітнес-функції є одним з найбільш важливих і ресурсоємних етапів генетичного пошуку. Тому *фітнес-функція* (цільова функція) *задачі повинна відповідати таким вимогам:*

- адекватність вирішуваній задачі;
- відсутність ділянок типу «плато» (тобто не повинна мати однакові значення в деякій області), оскільки така фітнес-функція не містить інформації про напрям пошуку;
- фітнес-функція не повинна бути занадто «гострою» (проблема «вужького горла»), оскільки генетичний пошук може не знайти глобального екстремуму, розташованого на такій ділянці;
- використання мінімуму ресурсів, оскільки обчислення фітнес-функції є найбільш частою процедурою генетичного пошуку, і тому фітнес-функція має істотний вплив на швидкість його роботи;
- мінімум кількості локальних оптимумів.

4.6.3 Ініціалізація

Стандартні генетичні методи починають свою роботу з ініціалізації [39], тобто формування *початкової популяції* P_0 – скінченного набору допустимих рішень задачі: $P_0 = \{H_1, H_2, \dots, H_N\}$; де N – розмір популяції; $H_j = \{h_{1j}, h_{2j}, \dots, h_{Lj}\}$ – j -та хромосома, що складається з L генів; h_{ij} – значення i -го гену j -ої хромосоми, $w_{\min, i} \leq h_{ij} \leq w_{\max, i}$; $w_{\min, i}$ та $w_{\max, i}$ – мінімальне й максимальне значення i -го параметру у вирішуваній за допомогою генетичного методу задачі.

Хромосоми початкової популяції можуть бути вибрані випадковим чином або введені користувачем. Вибір початкової популяції не має значення для збіжності процесу в асимптотиці, проте, формування гарної початкової популяції (наприклад, з множини локальних оптимумів) може помітно скоротити час досягнення глобального оптимуму. Таким чином, за наявності необхідної інформації задавання початкової популяції користувачем є переважним.

Найчастіше розмір початкової популяції вибирається в інтервалі 20–100 особин.

Стратегії ініціалізації початкової популяції (рис. 4.4) використовуються такі:

- випадкове формування початкової популяції;
- рівномірне формування початкової популяції (сіткова ініціалізація).
При такій стратегії генерується множина рішень, рівно віддалених одне від одного;

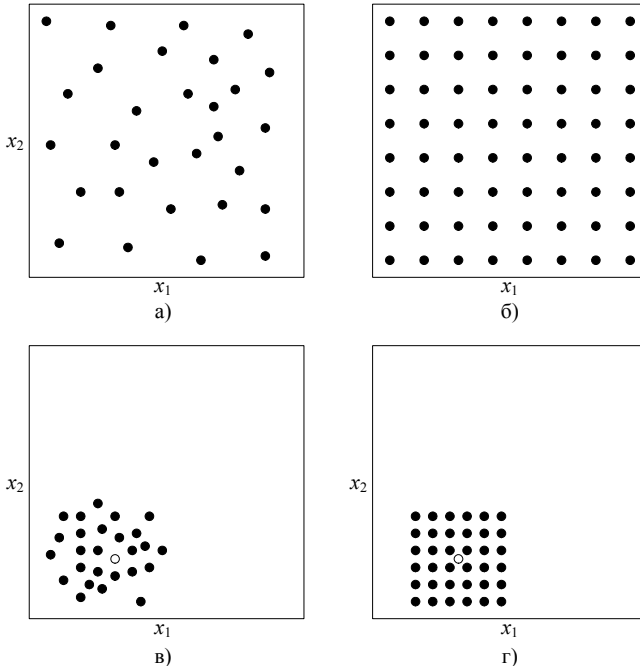


Рисунок 4.4 – Стратегії формування початкової популяції:

- а) випадкова ініціалізація;
- б) сіткова ініціалізація;
- в) випадкова ініціалізація, заснована на знаннях;
- г) сіткова ініціалізація, заснована на знаннях

– формування початкової популяції, засноване на знаннях. Така стратегія застосовується в тих випадках, коли є припущення, що деяка точка простору пошуку розташовується поблизу глобального оптимуму цільової функції;

– сіткова ініціалізація, заснована на знаннях, – об'єднує основні положення попередніх двох стратегій.

4.7 Генетичні оператори

Основними генетичними операторами (рис. 4.5) є:

- вибір;
- схрещування;
- мутація.

4.7.1 Відбір

Оператор відбору (вибору, селекції) є механізмом, на основі якого відбувається вибір хромосом для застосування до них того або іншого еволюційного оператора, а також для формування нової популяції [18, 55, 107]. При різних моделях еволюційного пошуку оператор відбору може бути використаний для таких цілей:

- вибір хромосом-батьків для схрещування;
- формування пар хромосом для схрещування;
- вибір хромосом для формування нового покоління з хромосом-батьків старої популяції і хромосом-нащадків, одержаних в результаті схрещування і мутації.

Як правило, оператор відбору використовується для вибору хромосом-батьків для схрещування. При цьому хромосоми вибираються на основі значення цільової функції так, щоб з ненульовою ймовірністю будь-який елемент популяції міг би бути вибраний як один з батьків для схрещування.

Найбільш поширеними є такі оператори відбору:

- пропорційний відбір (пропорційно-імовірнісний);
- відбір ранжируванням;
- турнірний відбір;
- відбір з використанням порогу.

4.7.1.1 Пропорційний відбір

Даний вид відбору виконується у такій послідовності кроків.

Крок 1. Обчислити значення фітнес-функції кожної особи f_j .

Крок 2. Знайти середнє значення фітнес-функції $f_{\text{ср}}$ популяції як середнє арифметичне значень фітнес-функцій всіх особин:

$$f_{\text{ср}} = \frac{1}{N} \sum_{j=1}^N f_j .$$

Крок 3. Для кожної особи обчислити відношення:

$$P_s(j) = \frac{f_j}{f_{\text{ср}}} .$$

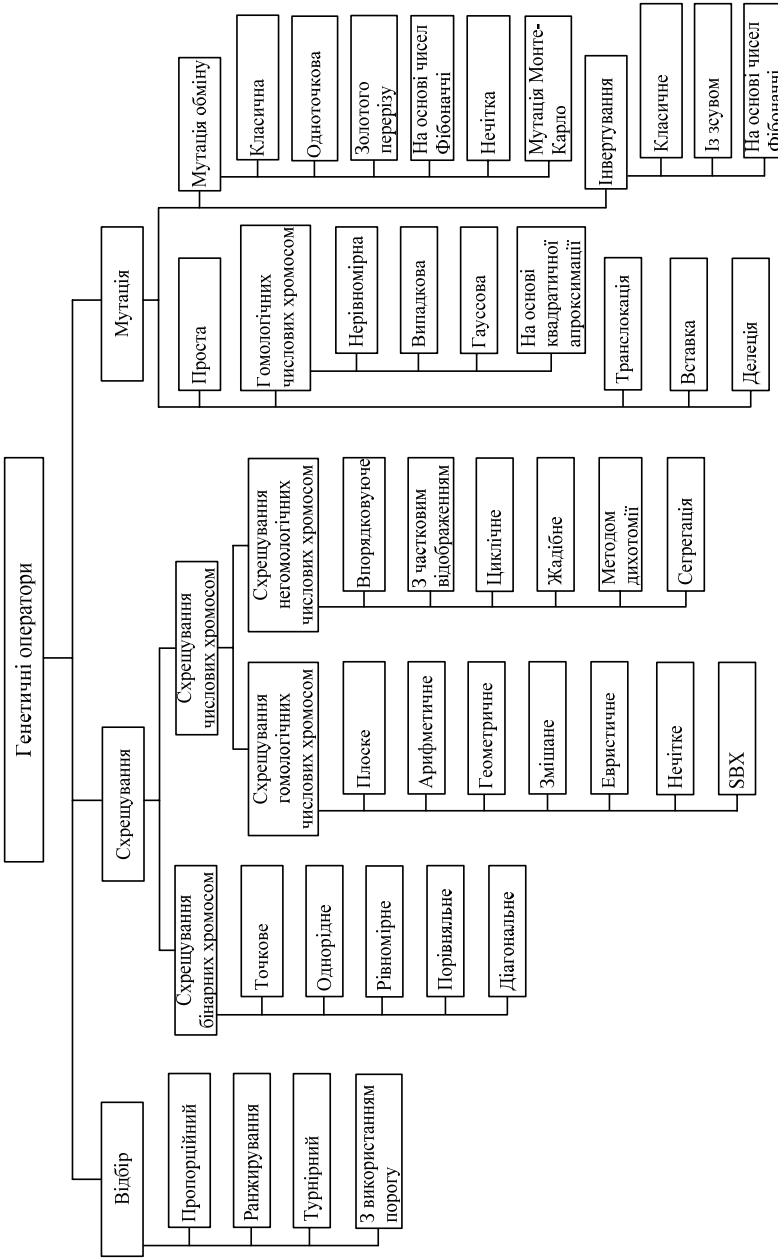


Рисунок 4.5 – Класифікація еволюційних операторів

Крок 4. Залежно від величини $P_s(j)$ сформувати масив особин, допущених до схрещування.

Формування масиву допущених до схрещування особин (крок 4) можна здійснити двома способами [72].

Перший спосіб (стохастичний залишковий відбір): якщо $P_s(j) > 1$, тоді особина вважається добре пристосованою і допускається до схрещування.

Наприклад, якщо $P_s(j) = 2,36$, то дана особина має подвійний шанс на схрещування і матиме ймовірність, що дорівнює 0,36, третього схрещування. Якщо ж пристосованість дорівнює 0,54, то особина візьме участь в єдиному схрещуванні з ймовірністю 0,54.

Реалізується це таким чином. Нехай є масив всіх особин популяції і додатковий масив для особин, допущених до схрещування. Для кожної особини популяції визначається значення описаного вище відношення. Далі, записуються рядки в проміжний масив згідно такого правила: необхідно узяти цілу частину відношення $P_s(j)$ і рівно стільки разів записати дану особину в допоміжний масив, після цього за допомогою випадкової величини визначити, чи буде вноситься дана особина ще раз: якщо випадкова величина більше дробової частини відношення, то буде, в іншому випадку – не буде. Надалі, особини для схрещування вибираються лише з проміжного масиву випадковим чином.

Для вже розглянутих прикладів з числами 2,36 і 0,54 ситуація виглядатиме таким чином: перша хромосома запишеться в проміжний масив двічі і з ймовірністю 0,36 запишеться втретє. Друга ж хромосома має ймовірність, що дорівнює 0,54 того, що вона взагалі буде присутня в допоміжному масиві.

Другий спосіб полягає в тому, що після знаходження відношення $P_s(j)$ відбувається відбір (із заміщенням) всіх N особин для подальшої генетичної обробки згідно величини $P_s(j)$.

Простий пропорційний відбір – *рулетка* – відбирає особини за допомогою N запусків рулетки. Колесо рулетки містить по одному сектору для кожного члена популяції. Розмір j -го сектора пропорційний відповідній величині $P_s(j)$. Особина одержує нащадків, якщо випадково згенероване число в межах від 0 до 2π потрапляє в сектор, що відповідає цій особині. При такому відборі члени популяції з вищою пристосованістю з більшою ймовірністю частіше вибиратимуться, ніж особини з низькою пристосованістю.

При реалізації відбору рулеткою доцільно замінити колесо рулетки інтервалом $[0; 1]$ у зв'язку з тим, що в такому разі немає необхідності обчислювати ширину кожного сектора – в цьому випадку кожній особині зіставляється напівінтервал $[x_{j-1}; x_j)$, де $x_{j-1} - x_j = P_s(j)$, а $x_0 = 0$ (при цьому $x_N = 1$). Допускається до схрещування особина з номером j , де $j: x_{\text{rnd}} \in [x_{j-1}; x_j)$, а число x_{rnd} генерується кожного разу випадковою функцією з рівномірним розподілом щільності ймовірності на відрізьку $[0; 1]$.

Схема рулетки може давати дуже великі відхилення між одержаною кількістю нащадків даної особини і очікуваною. Одержане число нащадків наближається до очікуваного лише в популяціях дуже великих розмірів.

4.7.1.2 Відбір ранжируванням

Відбір ранжируванням [18] виконується за чотири кроки.

Крок 1. Обчислити пристосованість кожної особини f_j .

Крок 2. Відсортувати (ранжувати) популяцію за збільшенням пристосованості особин.

Крок 3. Для кожної особини обчислити величину $P_s(j)$. Для цього використовувати один з двох видів ранжирування:

$$\text{а) лінійне ранжирування: } P_s(j) = \frac{1}{N} \left(\eta_{\max} - (\eta_{\max} - \eta_{\min}) \frac{j-1}{N-1} \right),$$

де $\eta_{\max} \in [1; 2]$, $\eta_{\min} = 2 - \eta_{\max}$;

$$\text{б) рівномірне ранжирування: } P_s(j) = \begin{cases} \frac{1}{\mu}, & \text{якщо } 1 \leq j \leq \mu; \\ 0, & \text{якщо } \mu < j \leq N, \end{cases}$$

де μ – деяке фіксоване число перших членів популяції.

Крок 4. Залежно від величини $P_s(j)$ відібрати певну частину особин для схрещування.

4.7.1.3 Турнірний відбір

Турнірний відбір реалізує k турнірів, щоб вибрати k особин [39]. Кожен турнір складається з двох етапів.

Етап 1. Вибір t елементів з популяції.

Етап 2. Вибір кращої особини серед особин, відібраних на попередньому етапі.

Розмір групи особин, відібраних для турніру, часто дорівнює двом. В цьому випадку говорять про *двійковий (парний) турнір*. Взагалі ж t називається *чисельністю турніру*.

Турнірний відбір має певні переваги перед пропорційним відбором, оскільки не втрачає своєї вибірковості у випадку, коли в ході еволюції всі елементи популяції стають приблизно рівними по значенню цільової функції.

4.7.1.4 Відбір з використанням порогу

Відбір з використанням порогу (відбір усіканням) [45] виконується таким чином.

Крок 1. Обчислити пристосованість кожної особини f_j .

Крок 2. Відсортувати популяцію за збільшенням пристосованості особин.

Крок 3. Задати поріг $tr \in [0; 1]$. Поріг визначає, яка частка особин, починаючи з найпершої (самої пристосованої), братиме участь в схрещуванні. В принципі, поріг можна задати і числом, більшим за 1, тоді він буде просто дорівнювати числу особин з поточної популяції, допущених до схрещування.

Крок 4. Серед особин, що потрапили під значення порогу, випадковим чином N разів вибрати саму везучу і записати її в проміжний масив, з якого потім вибираються особини безпосередньо для схрещування.

Через те, що в цій стратегії використовується відсортована популяція, час її роботи може бути великим для популяцій великого розміру і залежати також від методу сортування.

4.7.2 Схрещування

У теорії еволюції важливу роль відіграє те, яким чином ознаки батьків передаються нащадкам. У генетичних методах за передачу ознак батьків нащадкам відповідає оператор схрещування (кросинговер, кросовер, рекомбінація). Цей оператор моделює процес схрещування особин і визначає передачу ознак батьків нащадкам [39, 68, 69, 71, 72, 101].

Метою оператора схрещування є породження з наявної множини рішень нового, в якому кожна хромосома буде нащадком деяких двох елементів попередньої популяції, тобто нести в собі частково інформацію кожного батька. Допускається ситуація, коли обидва батька подані одним і тим же елементом популяції.

4.7.2.1 Вибір батьківської пари

Вибираючи кожного разу для схрещування найбільш пристосовані особини, можна з певним ступенем впевненості стверджувати, що нащадки будуть або не набагато гірші, ніж батьки, або кращі за них [72].

Існує декілька способів вибору батьківської пари.

Випадковий вибір батьківської пари («панміксія») – це найпростіший підхід, коли обидві особини, які складуть батьківську пару, випадковим чином вибираються зі всієї популяції, причому будь-яка особина може стати членом декількох пар.

Крок 1. Для вибору пари батьків задати ймовірність схрещування P_c .

Крок 2. Довільним чином пронумерувати всіх представників поточної популяції.

Крок 3. Вибрати першого батька. Для цього, починаючи з першої, перебрати всі хромосоми популяція до тих пір, поки випадково вибране чи-

сло з інтервалу $[0; 1]$ не буде меншим за P_c . Коли це відбудеться для одного з елементів популяції, цей елемент стане першим батьком.

Крок 4. Продовжити перегляд популяції, починаючи з наступного після першого батька рішення. Перегляд продовжувати до тих пір, поки знову випадково обране число не буде меншим, ніж P_c . Хромосома, для якої виконується така умова, буде другим батьком.

Описаним способом складаються пари до тих пір, поки не вибереться потрібна кількість пар батьків.

Конкретне значення P_c залежить від вирішуваної задачі, і в загальному випадку лежить в інтервалі $[0,6; 0,99]$.

Не зважаючи на простоту, такий підхід універсальний для розв'язання різних класів задач. Проте, він достатньо критичний до чисельності популяції, та його ефективність знижується із зростанням чисельності популяції.

Інший метод випадкового вибору батьківської пари може бути поданий у вигляді такої послідовності кроків.

Крок 1. Розбити популяцію випадковим чином на два масиви (підпопуляції) одного розміру.

Крок 2. Відсортувати кожну підпопуляцію.

Крок 3. Сформувані пари для схрещування з особин, що мають однаковий ранг (номер) в підпопуляціях.

Крок 4. Допустити до схрещування пари, для яких число, що випадково згенероване в інтервалі $[0; 1]$, перевищуватиме задану ймовірність схрещування.

Селективний спосіб формування батьківської пари полягає в тому, що «батьками» можуть стати лише ті особини, значення пристосованості яких не менше середнього значення пристосованості по популяції, при рівній імовірності таких кандидатів скласти батьківську пару.

Такий підхід забезпечує більш швидку збіжність генетичного пошуку. Проте через швидку збіжність селективний вибір батьківської пари не підходить тоді, коли ставиться задача визначення декількох екстремумів, оскільки для таких задач метод, як правило, швидко сходиться до одного з рішень.

Крім того, для деякого класу задач зі складним ландшафтом фітнес-функції швидка збіжність може перетворитися на передчасну збіжність до квазіоптимального рішення. Цей недолік може бути частково компенсований використанням відповідного механізму відбору, який би сповільнював дуже швидку збіжність методу.

Інші два способи формування батьківської пари – це *інбридинг* і *аутбридинг*. Обидва ці методи побудовані на формуванні пари на основі близької і далекої «спорідненості» відповідно. Під «спорідненістю» тут розуміється відстань між членами популяції як в сенсі Евклідової (геометричної)

відстані особин в просторі параметрів (для фенотипів), так і в сенсі відстані Хеммінга між хромосомними наборами особин (для генотипів).

Евклідова відстань $d_E(H_j; H_k)$ між j -ою і k -ою особинами популяції визначається за формулою:

$$d_E(H_j; H_k) = \sqrt{\sum_{i=1}^p (x_i^{(j)} - x_i^{(k)})^2},$$

де p – кількість параметрів (генів) особини; $x_i^{(j)}$ – i -ий параметр в незакодованому вигляді j -ої особини.

Відстань Хеммінга $d_H(H_j; H_k)$ між j -ою і k -ою особинами популяції визначається як кількість неспівпадаючих бітів в однакових позиціях j -ої і k -ої хромосом:

$$d_H(H_j; H_k) = \sum_{i=1}^L |h_{ij} - h_{ik}|.$$

Інбридинг складається з двох етапів. На першому етапі випадковим чином обирається перший член пари, на другому етапі вибирається другий батько. При цьому другою батьківською особиною з більшою ймовірністю буде максимально близька до першої особина.

Один з варіантів процедури інбридингу може бути реалізований таким чином.

Крок 1. Вибрати випадковим чином першого батька.

Крок 2. Вибрати з поточної популяції випадковим чином групу із C хромосом ($C = 1\%–15\%$ від розміру популяції).

Крок 3. Розрахувати Евклідову відстань від хромосоми, отриманої на першому кроці, до кожної з C відібраних на другому кроці хромосом.

Крок 4. Як другого батька вибрати найближчу до першого батька хромосому.

Аутбридинг формує батьківські пари з максимально далеких особин.

Використання генетичних інбридингу і аутбридингу є ефективним для багатоекстремальних задач. Проте, ці два способи по-різному впливають на поведінку генетичного методу.

Інбридинг можна охарактеризувати властивістю концентрації пошуку в локальних вузлах, що фактично призводить до розбиття популяції на окремі локальні групи навколо підозрілих на екстремум ділянок ландшафту фітнес-функції.

Аутбридинг, навпаки, спрямований на попередження збіжності методу до вже знайдених рішень, примушуючи метод проглядати нові, недосліджені області.

4.7.2.2 Оператори схрещування для бінарних хромосом

До бінарних хромосом можуть бути застосовані такі оператори схрещування [12]:

- точкове;
- однорідне;
- рівномірне;
- порівняльне;
- діагональне.

Оператори схрещування, розроблені для бінарних хромосом, також можуть застосовуватися для гомологічних числових і векторних хромосом.

4.7.2.2.1 Точкове схрещування

Точкове схрещування [72] виконується таким чином.

Крок 1. Випадково вибрати n точок розриву, внаслідок чого одержати розбиття початкових хромосом на $(n + 1)$ частин різної довжини.

Крок 2. Обміняти у початкових хромосом ділянки з парними номерами, а ділянки з непарними залишити без змін.

Класичним варіантом такого схрещування є *одноточкове схрещування*, при якому:

- випадковим чином визначається точка всередині хромосоми. Ця точка називається *точкою розриву (точкою схрещування, crossover point)*;
- у вибраній точці обидві хромосоми діляться на дві частини і обмінуються ними. В результаті утворюються два нащадки.

Даний тип схрещування називається *одноточковим*, оскільки при його застосуванні батьківські хромосоми розрізаються лише в одній випадковій точці.

При *двоточковому схрещуванні* в хромосомі випадково вибираються вже дві точки схрещування. Ліву точку вважають першою, а праву – другою. Перший нащадок формується з частин першого батька, розташованих лівіше першої точки схрещування і правіше другої точки, і частини другого батька, розташованої між першою і другою точками схрещування. Другий нащадок отримується з лівої і правої частин другого батька і центральної частини першого батька.

4.7.2.2.2 Однорідне схрещування

Однорідне схрещування (uniform crossover) генерує нащадка шляхом випадкової передачі йому генетичної інформації від батьків [45]. Генерація нащадка виконується таким чином.

Крок 1. Встановити лічильник бітів (генів) нащадка: $j = 1$.

Крок 2. Визначити хромосому з батьківської пари, яка передасть значення свого j -го гена нащадку: $n = \text{round}(\text{rand}[0; 1])$, де $\text{rand}[0; 1]$ – число, що випадково згенероване в інтервалі $[0; 1]$; $\text{round}(A)$ – округлене значення числа A .

Крок 3. Виконати: $h_{jo} = h_{jn}$, де h_{jo} – значення j -го гена нащадка, h_{jn} – значення j -го гена n -го батька.

Крок 4. Виконати: $j = j + 1$.

Крок 5. Якщо $j > L$, де L – довжина хромосоми, тоді виконати перехід до кроку 6. Інакше перейти до кроку 2.

Крок 6. Зупинення.

4.7.2.2.3 Рівномірне схрещування

Рівномірне схрещування [68] зручно застосовувати у тому випадку, коли вже одержані індивіди з гарними спадковими ознаками, і їх необхідно закріпити в поточній популяції. Рівномірне схрещування виконується як послідовність кроків 1–3.

Крок 1. Випадковим чином задати маску схрещування, що є рядком з нулів і одиниць, довжиною, що дорівнює довжині хромосом.

Крок 2. Сформувати першого нащадка.

Одиниця на конкретній позиції в масці схрещування означає, що елемент, який стоїть на тому ж місці в першого батька, необхідно помістити на це місце в першого нащадка. Нуль на цій позиції в масці схрещування означає, що елемент, який стоїть на тому ж місці другого батька, необхідно помістити на це місце першого нащадка.

Крок 3. Сформувати другого нащадка.

Якщо першого батька вважати другим, а другого – першим, то аналогічно кроку 2 можна одержати другу хромосому-нащадка.

Варто відзначити, що маска схрещування може бути одна для всіх хромосом, або своя для кожної пари батьків.

4.7.2.2.4 Порівняльне схрещування

При порівняльному схрещуванні [12] в хромосомах батьків порівнюються всі біти. Біти, що співпали (0 і 0 або 1 і 1), залишаються без зміни, а ті, що не співпали визначаються через генератор випадкових чисел.

4.7.2.2.5 Діагональне схрещування

При діагональному схрещуванні [72] для схрещування R батьків випадковим чином вибирається $(R - 1)$ однакових точок схрещування в кож-

ному з них. R нащадків формують шляхом комбінації відповідних елементів батьків по діагоналі.

Крок 1. Вибрати випадковим чином R батьківських хромосом для схрещування.

Крок 2. Вибрати випадковим чином $(R - 1)$ точок схрещування в хромосомах, розділивши тим самим кожну з них на R сегментів.

Крок 3. Встановити: $r = 1$.

Крок 4. Сформувати r -го нащадка.

Крок 4.1 Встановити: $k = 1$.

Крок 4.2. Сформувати k -ий сегмент r -го нащадка, узявши g -ий сегмент з k -ої хромосоми-батька, де

$$g = \begin{cases} k + r - 1, & \text{якщо } k + r - 1 \leq R; \\ k + r - 1 - R, & \text{якщо } k + r - 1 > R. \end{cases}$$

Крок 4.3. Виконати: $k = k + 1$.

Крок 4.4. Якщо $k \leq R$, тоді виконати перехід до кроку 4.2.

Крок 5. Виконати: $r = r + 1$.

Крок 6. Якщо $r \leq R$, тоді виконати перехід до кроку 4.

Крок 7. Зупинення.

Варто відзначити, що різні типи схрещування володіють загальною *позитивною властивістю*: вони контролюють баланс між подальшим використанням вже знайдених гарних підобластей простору пошуку і дослідженням нових підобластей. Це досягається за рахунок неруйнування загальних блоків всередині хромосом-батьків і одночасного дослідження нових областей в результаті обміну частинами хромосом.

Сумісне використання операторів відбору і схрещування призводить до того, що області простору, які володіють кращою в середньому оптимальністю, містять більше членів популяції, ніж інші. Таким чином, еволюція популяції прямує до областей, що містять оптимум з більшою ймовірністю. Тому, оператор схрещування є найбільш критичним з усіх генетичних операторів з погляду отримання глобальних результатів.

У невеликих популяціях краще застосовувати більш руйнівні варіанти схрещування (багатоточкове і однорідне), а у великих гарно працює двоточкове.

При застосуванні оператора схрещування, як і при застосуванні мутації, слід передбачити метод відкидання тих особин, чії хромосоми не задовольняють області допустимих значень (якщо це необхідно).

4.7.2.3 Оператори схрещування для гомологічних числових хромосом

До гомологічних числових хромосом крім вище розглянутих можуть бути застосовані такі оператори схрещування [18]:

– плоске;

- арифметичне;
- геометричне;
- змішане;
- евристичне;
- нечітке;
- SBX-схрещування.

Також існують оператори простого (simple crossover) і дискретного (discrete crossover) схрещування, які аналогічні операторам одноточкового і однорідного схрещування, що використовуються для схрещування бінарних хромосом.

4.7.2.3.1 Плоске схрещування

При плоскому схрещуванні (flat crossover) дві хромосоми $H_1 = \{h_{11}, h_{12}, \dots, h_{1L}\}$ та $H_2 = \{h_{21}, h_{22}, \dots, h_{2L}\}$, вибрані для схрещування, генерують одного нащадка H_{Π} [39]. При цьому i -ий ген $h_{\Pi i}$ хромосоми-нащадка H_{Π} визначається за формулою:

$$h_{\Pi i} = \text{rand}[h_{1i}; h_{2i}],$$

де $\text{rand}[h_{1i}; h_{2i}]$ – число, що випадково згенероване, в інтервалі $[h_{1i}; h_{2i}]$.

4.7.2.3.2 Арифметичне схрещування

Арифметичне схрещування (arithmetical crossover) є найбільш вдалим для пошуку оптимуму функції багатьох дійсних змінних.

При використанні арифметичного схрещування (рис. 4.6) на основі значень генів хромосом-батьків H_1 та H_2 створюються два нащадки $H_{1\Pi}$ й $H_{2\Pi}$, значення i -их генів $h_{1\Pi i}$ та $h_{2\Pi i}$ яких розраховуються за формулами [69]:

$$h_{1\Pi i} = k \cdot h_{1i} + (1 - k)h_{2i},$$

$$h_{2\Pi i} = k \cdot h_{2i} + (1 - k)h_{1i},$$

де $k \in [0; 1]$ – деякий дійсний коефіцієнт, який може залежати від $d(H_1; H_2)$ – відстані між хромосомами.

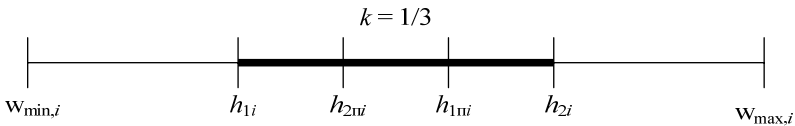


Рисунок 4.6 – Геометрична інтерпретація арифметичного схрещування

Залежно від способу завдання коефіцієнта k розрізняють:

– *рівномірне арифметичне схрещування*, при якому коефіцієнт k задається користувачем на етапі ініціалізації генетичного пошуку і є постійним впродовж функціонування генетичного методу;

– *нерівномірне арифметичне схрещування* – коефіцієнт k генерується випадковим чином на кожній ітерації або змінюється залежно від номера поточної ітерації.

Якщо початкова функція є неперервною, то ця модель працює гарно, а якщо вона ще і є гладкою, тоді – відмінно.

Арифметичне схрещування може бути модифіковано таким чином. Дві хромосоми-батьки H_1 та H_2 породжують чотирьох нащадків $H_{1n} - H_{4n}$, значення i -их генів $h_{1ni} - h_{4ni}$ яких розраховуються за формулами:

$$\begin{aligned} h_{1ni} &= \frac{h_{1i} + h_{2i}}{2}, \\ h_{2ni} &= w_{\max,i}(1-k) + k \max(h_{1i}, h_{2i}), \\ h_{3ni} &= w_{\min,i}(1-k) + k \min(h_{1i}, h_{2i}), \\ h_{4ni} &= \frac{(w_{\max,i} + w_{\min,i})(1-k) + (h_{1i} + h_{2i})k}{2}, \end{aligned}$$

де $w_{\min,i}$ та $w_{\max,i}$ – мінімальне й максимальне значення i -го параметра у вирішуваній за допомогою генетичного методу задачі; $k \in [0; 1]$.

В якості нащадку вибирається хромосома з найкращим значенням фітнес-функції.

Лінійне схрещування (linear crossover) є іншою модифікацією арифметичного схрещування. При використанні лінійного схрещування двоє хромосом-батьків H_1 та H_2 формують трьох нащадків $H_{1n} - H_{3n}$, значення i -их генів $h_{1ni} - h_{3ni}$ яких розраховуються за формулами:

$$\begin{aligned} h_{1ni} &= \frac{h_{1i} + h_{2i}}{2}, \\ h_{2ni} &= 1,5h_{1i} - 0,5h_{2i}, \\ h_{3ni} &= -0,5h_{1i} + 1,5h_{2i}. \end{aligned}$$

Модифікацією нерівномірного арифметичного схрещування є *розширене лінійчате схрещування* (extended line crossover), при якому значення i -го гена h_{ni} хромосоми-нащадка H_n визначається за формулою:

$$h_{ni} = h_{1i} + k \cdot (h_{1i} - h_{2i}).$$

При цьому інтервал можливих значень генів нащадка розширюється за рахунок збільшення діапазону можливих значень коефіцієнта k . Так, якщо $k \in [-0,25; 1,25]$, то інтервал можливих значень гена хромосоми-нащадка збільшується таким чином (рис. 4.7).

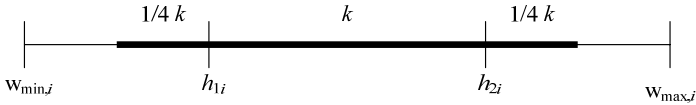


Рисунок 4.7 – Геометрична інтерпретація розширеного лінійчатого схрещування

4.7.2.3.3 Геометричне схрещування

При геометричному схрещуванні (geometrical crossover) створюються два нащадки $H_{1п}$ та $H_{2п}$. При цьому значення i -их генів h_{1pi} й h_{2pi} хромосом-нащадків $H_{1п}$ та $H_{2п}$, відповідно, визначаються за формулами [12]:

$$\begin{aligned} h_{1pi} &= h_{1i}^k \cdot h_{2i}^{1-k}, \\ h_{2pi} &= h_{2i}^k \cdot h_{1i}^{1-k}, \end{aligned}$$

де $k \in [0; 1]$ – дійсний коефіцієнт, що задається користувачем на етапі ініціалізації генетичного пошуку.

4.7.2.3.4 Змішане схрещування

При змішаному схрещуванні (blend, BLX-alpha crossover) генерується один нащадок $H_{п}$ (рис. 4.8). При цьому i -ий ген h_{pi} хромосоми-нащадка $H_{п}$ визначається за формулою [45]:

$$h_{pi} = \text{rand}[c_{\min} - I \cdot a; c_{\max} + I \cdot a],$$

де $c_{\min} = \min(h_{1i}; h_{2i})$; $c_{\max} = \max(h_{1i}; h_{2i})$; $I = c_{\max} - c_{\min}$; $a \in [0; 1]$ – дійсний коефіцієнт, що задається користувачем на етапі ініціалізації генетичного пошуку.

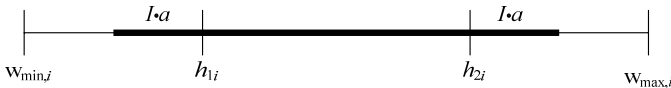


Рисунок 4.8 – Геометрична інтерпретація змішаного схрещування

Окремим випадком змішаного схрещування (при $a = 0$) є плоске схрещування.

4.7.2.3.5 Евристичне схрещування

При евристичному схрещуванні (heuristic crossover) створюється нащадок $H_{п}$ [41], розташований на деякій відстані від батька з кращим значенням фітнес-функції H_1 в напрямі від батька з гіршим значенням фітнес-функції H_2 (рис. 4.9).



Рисунок 4.9 – Геометрична інтерпретація евристичного схрещування

При використанні евристичного схрещування значення i -го гена h_{pi} хромосоми-нащадка H_{pi} визначається за формулою:

$$h_{pi} = k \cdot (h_{1i} - h_{2i}) + h_{1i},$$

де $k \in [0; 1]$ – дійсний коефіцієнт, що задається користувачем на етапі ініціалізації генетичного пошуку.

4.7.2.3.6 Нечітке схрещування

При нечіткому схрещуванні (fuzzy recombination, FR-d crossover) створюються два нащадки H_{1pi} і H_{2pi} (рис. 4.10).

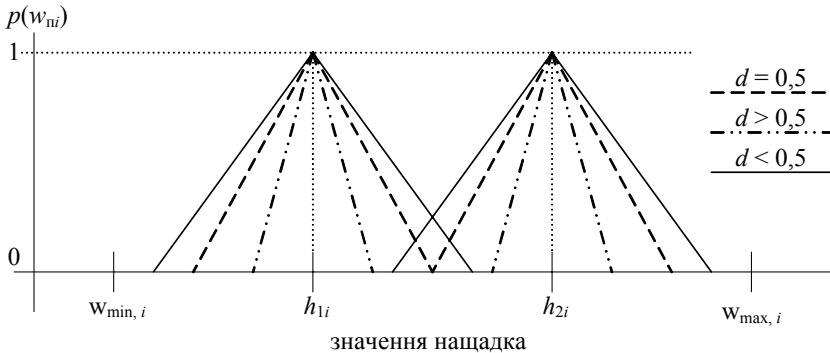


Рисунок 4.10 – Геометрична інтерпретація нечіткого схрещування

При цьому ймовірність $p(w_{pi})$ того, що i -ий ген h_{pi} нащадка прийме значення w_{pi} визначається розподілом [18]:

$$p(w_{pi}) \in \{\mu_1(w), \mu_2(w)\},$$

де $\mu_1(w)$ і $\mu_2(w)$ – розподіли ймовірностей трикутної форми (трикутні нечіткі функції належності):

$$\mu_1(w) = \begin{cases} 0, & \text{якщо } w \leq a_1; \\ \frac{w - a_1}{b_1 - a_1}, & \text{якщо } a_1 < w \leq b_1; \\ \frac{c_1 - w}{c_1 - b_1}, & \text{якщо } b_1 < w \leq c_1; \\ 0, & \text{якщо } w \geq c_1, \end{cases} \quad \mu_2(w) = \begin{cases} 0, & \text{якщо } w \leq a_2; \\ \frac{w - a_2}{b_2 - a_2}, & \text{якщо } a_2 < w \leq b_2; \\ \frac{c_2 - w}{c_2 - b_2}, & \text{якщо } b_2 < w \leq c_2; \\ 0, & \text{якщо } w \geq c_2, \end{cases}$$

де $a_1 = h_{1i} - d \cdot I$; $b_1 = h_{1i}$; $c_1 = h_{1i} + d \cdot I$; $a_2 = h_{2i} - d \cdot I$; $b_2 = h_{2i}$; $c_2 = h_{2i} + d \cdot I$; $I = |h_{1i} - h_{2i}|$.

Параметр $d \in [0; 1]$ задається на етапі ініціалізації генетичного пошуку і визначає ступінь перекриття трикутних функцій належності. Як правило, задається $d = 0,5$.

4.7.2.3.7 SBX-схрещування

SBX-схрещування (simulated binary crossover) імітує принципи роботи двійкового оператора схрещування [39].

SBX-схрещування виконується в такій послідовності кроків.

Крок 1. Згенерувати випадкове число $u = \text{rand}[0; 1]$.

Крок 2. Обчислити параметр β :
$$\beta = \begin{cases} (2u)^{\frac{1}{n+1}}, & \text{якщо } u \leq 0,5; \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{n+1}}, & \text{якщо } u > 0,5, \end{cases}$$

де n – параметр, що задається користувачем.

Крок 3. Одержати нащадків H_{1n} та H_{2n} з батьків H_1 й H_2 . При цьому значення i -их генів h_{1ni} та h_{2ni} хромосом-нащадків H_{1n} й H_{2n} , відповідно, визначаються за формулами:

$$h_{1ni} = 0,5 \cdot ((1 + \beta)h_{1i} + (1 - \beta)h_{2i}),$$

$$h_{2ni} = 0,5 \cdot ((1 - \beta)h_{1i} + (1 + \beta)h_{2i}).$$

На рис. 4.11 наведено геометричну інтерпретацію роботи SBX-схрещування при схрещуванні двох генів хромосом. З рисунка видно, що збільшення параметра n призводить до збільшення ймовірності появи нащадка в околиці батька і навпаки.

4.7.2.4 Оператори схрещування для негомологічних числових хромосом

Комбінаторні задачі оперують з дискретними структурами, незначні зміни яких часто викликають стрибкоподібну зміну фітнес-функції [40, 88].

Описані вище оператори схрещування не підходять при вирішенні комбінаторних задач (задач з підстановками). Це пов'язано з тим, що при

застосуванні розглянутих раніше видів схрещування можуть формуватися неприпустимі хромосоми, у яких гени мають значення, що повторюються. У зв'язку з цим були запропоновані альтернативні види оператора схрещування.

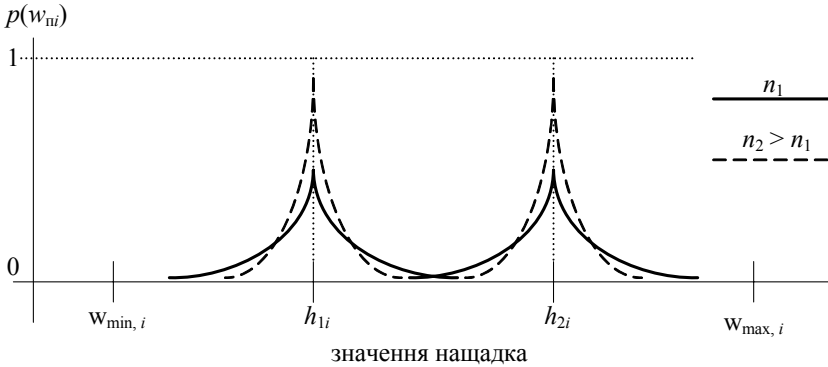


Рисунок 4.11 – Геометрична інтерпретація SBX-схрещування

4.7.2.4.1 Впорядковувачий оператор схрещування

Впорядковувачий оператор схрещування (order crossover, OX) був запропонований Д. Девісом в 1985 році для негомологічних числових хромосом [69]. Схрещування може проводитися по одній або по двох точках. Точки схрещування вибираються випадково.

При *одноточковому* *впорядковувачому* *схрещуванні* в хромосому першого нащадка копіюється хромосома першого батька, а потім гени нащадка, що розташовані правіше точки схрещування, перевпорядковуються в послідовність, яка відповідає другому батькові. При цьому другий батько переглядається від початку до кінця, зліва направо, і елементи, яких не вистачає в нащадка, додаються, починаючи з точки схрещування, по порядку.

Крок 1. Випадковим чином вибрати точку схрещування.

Крок 2. Скопіювати в хромосому першого нащадка сегмент хромосоми першого батька, що розташований ліворуч від точки схрещування.

Крок 3. Решта генів в нащадка копіюється з другого батька у впорядкованому вигляді зліва направо, виключаючи елементи, що вже увійшли до нащадка.

Для створення другого нащадка застосовується зворотний порядок дій.

При *двоточковому* *впорядковувачому* схрещуванні змінюється частина хромосоми, яка знаходиться між точками схрещування.

Також застосовується *позиційно* *впорядковуваче* *схрещування*, яке виконується в такій послідовності.

Крок 1. Випадковим чином вибрати деяку кількість позицій (генів) в першій хромосомі-батькові.

Крок 2. Скопіювати в хромосому першого нащадка вибрані на попередньому кроці гени першої хромосоми-батька. При цьому копіювання генів відбувається в ті ж позиції, на яких вони розташовувалися в першій хромосомі-батькові.

Крок 3. Решта генів нащадка копіюється з другого батька у впорядкованому вигляді зліва направо, виключаючи елементи, що вже увійшли до нащадка.

Для створення другого нащадка застосовується аналогічна послідовність дій за винятком того, що перший батько тепер вважається другим, а другий – першим.

В даному випадку кожна хромосома є впорядкованою послідовністю чисел, внаслідок чого застосування стандартних операторів схрещування неминуче призвело б до утворення неприпустимих рішень.

4.7.2.4.2 Схрещування з частковим відображенням

Схрещування з частковим відображенням (partially mapped crossover, PMX-схрещування), запропоноване Д. Голдбергом і Р. Лінглом, полягає в прямому відображенні частини батьківської інформації в дитині [12, 69]. PMX-схрещування можна одержати шляхом композиції початкової хромосоми і декількох транспозицій.

Сам метод полягає в наступному. Нехай P_1 та P_2 – негомологічні числові хромосоми (підстановки), що беруть участь в схрещуванні.

Крок 1. Випадковим чином вибрати ділянку, яка повинна бути відображена в нових хромосомах. Для цього необхідно згенерувати два числа m_1 та m_2 , що є лівою і правою межами ділянки, що відображається.

Крок 2. Обчислити розмір вибраної ділянки: $k = m_2 - m_1 + 1$.

Крок 3. Визначити транспозиції t_1, t_2, \dots, t_k .

Позначимо через (a, b) транспозицію, яка здійснює перестановку елементів a та b , а інші залишає на своїх місцях. Розглянемо такі транспозиції:

$$t_1 = (m_1, P_1^{-1}(P_2(m_1)));$$

$$t_2 = (m_1 + 1, (P_1 \circ t_1)^{-1}(P_2(m_1 + 1)));$$

.....

$$t_k = (m_2, (P_1 \circ t_1 \circ t_2 \circ \dots \circ t_{k-1})^{-1}(P_2(m_2))),$$

де $P_2(m_1)$ – значення m_1 -го гену хромосоми P_2 ; $P_1^{-1}(a)$ – номер гену хромосоми P_1 зі значенням a ; $P_1 \circ (a, b)$ – хромосома, що одержується з хромосоми P_1 в результаті обміну значень a -го та b -го генів.

Крок 4. Одержати рішення C_1 , що є першим нащадком:

$$C_1 = P_1 \circ t_1 \circ t_2 \circ \dots \circ t_k.$$

Крок 5. Рішення C_2 одержати за допомогою заміни P_1 на P_2 в попередньому виразі (крок 4) і при визначенні транспозицій t_i .

У двоточковому схрещуванні з частковим відображенням обмін проводиться між центральними частинами хромосом.

4.7.2.4.3 Циклічне схрещування

При циклічному схрещуванні (cycle crossover) [18] негомологічних числових хромосом P_1 та P_2 нове рішення C_1 отримується шляхом композиції початкової хромосоми P_2 і деякої підстановки (s_0, s_1, \dots, s_k) , що є циклом довжини $(k + 1)$, тобто $C_1 = P_2 \circ (s_0, s_1, \dots, s_k) = P_2 \circ (s_0, s_1) \circ (s_1, s_2) \circ (s_2, s_3) \circ \dots \circ (s_{k-1}, s_k)$, при цьому:

$$s_1 = P_2^{-1}(P_1(s_0));$$

$$s_2 = P_2^{-1}(P_1(s_1));$$

.....

$$s_k = P_2^{-1}(P_1(s_{k-1})).$$

де s_0 – випадково вибране ціле число від 1 до $(L - 1)$ і $s_{k+1} = P_2^{-1}(P_1(s_k)) = s_0$. Рішення C_2 отримується аналогічним шляхом за допомогою заміни P_2 на P_1 в попередніх виразах.

4.7.2.4.4 Жадібний оператор схрещування

Жадібний оператор схрещування (поглинаюче схрещування, greedy crossover) був запропонований в 1985 році Д. Грефенстеттом в співавторстві з іншими вченими для розв'язання задачі комівояжера [41, 72]. Це евристичний оператор схрещування, орієнтований на використання знань про об'єкт.

Ідея побудови «жадібного» алгоритму полягає в наступному. На кожному кроці послідовно вибираються кращі елементи з множини тих, що є, тобто рішення, що поліпшують цільову функцію, причому таким чином, щоб не порушувати діючих обмежень. Генерація нащадків відбувається за рахунок вибору кращих ділянок батьківських хромосом і їх подальшого сполучення.

Схема роботи жадібного оператора схрещування може змінюватися залежно від характеру вирішуваних задач.

Послідовність виконання жадібного оператора схрещування наведена нижче.

Крок 1. Обчислити значення цільової функції у відібраних для схрещування хромосом: $f(H_1)$ та $f(H_2)$, $H_1 = \{h_{11}, h_{12}, \dots, h_{1L}\}$, $H_2 = \{h_{21}, h_{22}, \dots, h_{2L}\}$, $i, h_{vi} = \overline{1, L}$, $v = \{1, 2\}$.

Крок 2. Встановити $j = 1$. Випадковим чином вибрати початкову точку для генерації хромосоми-нащадка: $p_j = \text{rand}[1, L]$.

Крок 3. Встановити $\text{temp} = p_j$; $j = j + 1$.

Крок 4. Визначити наступну точку хромосоми-нащадка: $p_j = \min(f(h_{1k_1}), f(h_{2k_2}))$, де $k_1 = H_1^{-1}(\text{temp}) + 1$; $k_2 = H_2^{-1}(\text{temp}) + 1$; $H_v^{-1}(\text{temp})$ – номер гена хромосоми H_v , що дорівнює значенню temp .

Крок 5. У випадку, якщо хромосома-нащадок складена повністю ($j = L$), перейти до кроку 8.

Крок 6. Виконати перевірку на передчасне замикання циклу: $p_j = p_d$, $d = \overline{1, j - 1}$. В разі передчасного замикання циклу збільшити шлях за рахунок включення гена, вибраного випадковим чином з числа ще не включених.

Крок 7. Виконати перехід до кроку 3.

Крок 8. Зупинення.

Практика показує, що застосування жадібного оператора схрещування підвищує швидкість збіжності рішення, але в той же час це сприяє зменшенню різноманітності популяції, що веде до її швидкого виродження, а також зниження можливостей виходу з локального оптимуму.

4.7.2.4.5 Схрещування методом дихотомії

Схрещування методом дихотомії [72] реалізується за рахунок механізму перебору точок розриву.

Крок 1. Розділити хромосоми-батьки довжини L навпіл (при нечіткому розмірі в будь-яку частину береться більше ціле), визначивши точку розриву.

Крок 2. За правилами одноточкового схрещування одержати дві нові хромосоми-нащадки.

Крок 3. Кожну половину хромосоми-нащадка знову розділити навпіл і процес розрахунку продовжити за початковою схемою: $1C_1 \cup 2C_3 \cup 1C_2 \cup 2C_4$; а для другої хромосоми-нащадка: $2C_1 \cup 1C_3 \cup 2C_2 \cup 1C_4$, де bC_i – i -та частина b -ої хромосоми розміром $(1/2^j)$ після j -го розбиття. Процес продовжувати до тих пір, доки не буде одержана задана кількість хромосом-нащадків або метод дихотомії завершиться.

При отриманні неприпустимих хромосом з генами, що повторюються, останні змінюються на відсутні гени з хромосом-батьків.

Крок 4. Зупинення.

4.7.2.4.6 Оператор сегрегації

В результаті застосування оператора сегрегації R батьківських хромосом породжують одного нащадка [45].

При застосуванні оператора сегрегації повинні бути відкинуті рішення, що повторюються, або рішення, що містять однакові елементи. Даний оператор можна реалізувати різними способами залежно від методу вибору генів з хромосом.

Крок 1. Вибрати випадковим чином R батьківських хромосом для схрещування.

Крок 2. Встановити: $k = 1$.

Крок 3. Сформувати k -ий елемент нащадка, узявши k -ий елемент (один або декілька генів) з k -ої хромосоми-батька.

У випадку, якщо даний елемент вже є в нащадка, взяти наступний елемент батька. Якщо ж і цей елемент вже включений в нащадка, повторювати дану процедуру до тих пір, доки не буде знайдений елемент в k -ої хромосоми-батька, який ще не включений в нащадка.

Крок 4. Виконати: $k = k + 1$.

Крок 5. Якщо $k \leq R$, виконати перехід до кроку 3.

Крок 6. Зупинення.

4.7.2.4.7 Інші оператори схрещування

Окрім жадібного оператора схрещування, гарні результати дають також інші оператори схрещування, орієнтовані на знання [39, 41, 72]. До них можна віднести *оператор схрещування, заснований на принципі золотого перерізу*. У такому операторі точка схрещування, на відміну від стандартних операторів схрещування, вибирається не випадковим чином, а на основі пропорції так званого, «золотого перерізу», що виражається числом $\tau = (-1 \pm \sqrt{5})/2 \approx 0,61803$, яке визначається з рівняння $1 - \tau = \tau^2$. Таким чином, точка розриву D для схрещування хромосом довжини L визначається за формулою $D = \text{Ціле}(\tau L)$.

Укрупнений *нечіткий метод побудови оператора сегрегації методу золотого перерізу* виконується в такій послідовності.

Крок 1. У заданій популяції хромосом вибрати R батьківських хромосом довжини L для схрещування.

Крок 2. Визначити точку розриву для реалізації оператора сегрегації методу золотого перерізу як найближче ціле $0,618L$ з обох кінців хромосом, розділивши тим самим кожен з хромосом на 3 сегменти.

Крок 3. За правилами побудови оператора сегрегації в кожній хромосомі випадковим або спрямованим чином вибрати один з трьох сегментів.

Крок 4. Вибрані сегменти з'єднати в хромосому з видаленням генів, що повторюються. Далі процес повторити аналогічно.

Крок 5. Метод закінчує роботу, коли проаналізовані всі сегменти або досягнуті інші критерії зупинення.

Точки розриву в операторах схрещування можуть також визначатися на основі чисел Фібоначчі: $\varphi_0 = \varphi_1 = 1$, $\varphi_k = \varphi_{k-1} + \varphi_{k-2}$ для $k = 2, 3, \dots$, тобто в числовій послідовності 1, 2, 3, 5, 8, 13, 21, ...

4.7.3 Мутація

Оператор мутації полягає в зміні генів у випадково вибраних позиціях [11, 48, 69]. На відміну від операторів відбору і схрещування, які використовуються для поліпшення структури хромосом, метою оператора мутації є *диверсифікація*, тобто підвищення різноманітності пошуку і введення нових хромосом в популяцію для того, щоб більш повно досліджувати простір пошуку. Оскільки число членів популяції P , як правило, вибирається значно меншим в порівнянні із загальним числом (2^L) можливих хромосом в просторі пошуку, то через це вдається досліджувати лише частину пошукового простору. Отже, мутація ініціює різноманітність в популяції, дозволяючи проглядати більше точок в просторі пошуку і долати таким чином локальні екстремуми в ході пошуку.

В ході мутації з ненульовою ймовірністю чергове рішення може перейти в будь-яке інше рішення.

Вибір хромосом для мутації відбувається таким чином.

Крок 1. Довільно нумеруються всі хромосоми H_j поточної популяції.

Крок 2. Починаючи з першої хромосоми, переглядається вся популяція, при цьому кожній хромосомі H_j ставляться у відповідність випадкові числа x_j з інтервалу $[0; 1)$.

Крок 3. Якщо число x_j виявляється меншим за ймовірність мутації P_m , то поточна хромосома H_j піддається мутації.

Серед рекомендацій по вибору ймовірності мутації нерідко можна зустріти варіанти $1/L$ або $1/N$, де L – довжина хромосоми, N – розмір популяції. Ймовірність мутації значно менше ймовірності схрещування і рідко перевищує 1%.

Необхідно відзначити, що оператор мутації є основним пошуковим оператором і існують методи, що не використовують інших операторів окрім мутації.

4.7.3.1 Проста мутація

Проста мутація використовується для бінарних, гомологічних числових і векторних хромосом. Суть її полягає у внутрішньогенній мутації. При цьому в хромосомі випадковим чином вибирається ген, а потім проводиться його випадкова зміна [69].

Алгоритм простої мутації для бінарних і гомологічних числових хромосом може складатися з таких кроків.

Крок 1. Скопіювати батьківську хромосому в хромосому-нащадка.

Крок 2. Вибрати випадковим чином ген, що мутує.

Крок 3. У заданому інтервалі допустимих значень гена вибрати нове значення гена, що не дорівнює поточному.

Для векторних хромосом проста мутація проводиться шляхом внесення змін в порядок елементів усередині обраного гена.

Крок 1. Скопіювати батьківську хромосому в хромосому-нащадка.

Крок 2. Вибрати випадковим чином ген для мутації.

Крок 3. Вибрати випадковим чином точку мутації усередині гена, що мутує.

Крок 4. Провести обмін значеннями між розрядами гена, що мутує, які знаходяться безпосередньо ліворуч і праворуч від точки мутації.

Відомий також модифікований оператор простої мутації, що називається *точковою мутацією*. При такій мутації в хромосомі мутує не один, а декілька генів із заданою ймовірністю. Послідовність виконання даного оператора наведено нижче.

Крок 1. Скопіювати батьківську хромосому в хромосому-нащадка.

Крок 2. Встановити: $i = 1$.

Крок 3. Випадковим чином згенерувати число x_i з інтервалу $[0; 1)$.

Крок 4. Якщо число x_i виявляється менше ймовірності мутації гена P_{mg} , то виконати мутацію гена h_i .

Крок 5. Встановити: $i = i + 1$.

Крок 6. Якщо $i < (L + 1)$, де L – довжина хромосоми, то виконати перехід до кроку 3.

Крок 7. Зупинення.

В наш час відома велика кількість різних варіантів операторів мутації. Більшість з них використовують комбінації таких ідей:

– так само як і схрещування, мутація може проводитися не лише по одній випадковій точці. Можна обирати деяку кількість точок в хромосомі для зміни значень генів, причому їх число також може бути випадковим;

– піддається мутації відразу деяка група генів, що йдуть підряд;

– сумісне застосування випадкової мутації і часткової перебудови рішення алгоритмами локального пошуку. Застосування локального спуску дозволяє генетичному методу зосередитися лише на локальному оптимумі. Множина локальних оптимумів може опинитися експоненціально великою і на перший погляд здається, що такий варіант методу не матиме великих переваг. Проте експериментальні дослідження розподілу локальних оптимумів [18, 41] свідчать про високу концентрацію їх в безпосередній близькості від глобального оптимуму. Це спостереження відоме як

гіпотеза про існування «великої долини» для задач на мінімум або «центрального гірського масиву» для задач на максимум [18, 45].

4.7.3.2 Мутація гомологічних числових хромосом

Такі види мутацій полягають в зміні вибраного для мутації гена h_{ij} (або всієї хромосоми H_j) на деяку величину Δh_{ij} , розраховану за певними методами [88]:

$$h_{ij}' = h_{ij} + \Delta h_{ij},$$

де h_{ij} – ген до мутації; h_{ij}' – ген після мутації.

1. *Нерівномірна (non-uniform) мутація* до вибраного для мутації i -го гена h_{ij} хромосоми H_j застосовується за формулою:

$$h_{ij}' = \begin{cases} h_{ij} + \Delta(t, w_{\max,i} - h_{ij}), & \text{якщо } r \leq 0,5; \\ h_{ij} - \Delta(t, h_{ij} - w_{\min,i}), & \text{якщо } r > 0,5, \end{cases}$$

де $\Delta(t, y) = y \cdot \left(1 - r^{(1-t/T)^k}\right)$; $r = \text{rand}[0; 1]$ – випадково згенероване число

в інтервалі $[0; 1]$; t – номер поточної ітерації; T – максимальна кількість ітерацій; k – параметр, що визначає ступінь однорідності (рівномірності); $w_{\min,i}$ та $w_{\max,i}$ – мінімальне і максимальне значення i -го параметра у вирішуваній за допомогою генетичного методу задачі.

Крім того, нерівномірна мутація i -го гена j -ої хромосоми h_{ji} може бути виконана за формулою:

$$h_{ij}' = \begin{cases} h_{ij} + \Delta(t, w_{\max,i} - h_{ij}), & \text{якщо } f(H_j + \Delta(t, w_{\max,i} - h_{ij})) \geq f(H_j - \Delta(t, h_{ij} - w_{\min,i})); \\ h_{ij} - \Delta(t, h_{ij} - w_{\min,i}), & \text{якщо } f(H_j + \Delta(t, w_{\max,i} - h_{ij})) < f(H_j - \Delta(t, h_{ij} - w_{\min,i})), \end{cases}$$

де $\Delta(t, y) = y \cdot w_i(t)$; $w_i(t)$ – коефіцієнт, що залежить від відношення t/T . Наприклад, коефіцієнт $w_i(t)$ може бути заданий такою формулою:

$w_i(t) = r \left(1 - \frac{t}{T}\right)^{k_i}$, де $r = \text{rand}[0; 1]$ – число, що згенероване випадковим

чином в інтервалі $[0; 1]$; $k_i > 0$ – параметр, що задається користувачем.

2. *Випадкова мутація* вибраного гена h_{ij} полягає в зміні його значення на величину Δh_{ij} , розраховану за формулою:

$$\Delta h_{ij} = \text{rand}[w_{\min,i} \cdot r \cdot q(t); w_{\max,i} \cdot r \cdot q(t)], \quad q(t) = \frac{\ln T - \ln t}{\ln T},$$

де $\text{rand}[a; b]$ – випадкове число в інтервалі $[a; b]$; $r = \text{rand}[0; 1]$.

3. *Гауссова (нормальна) мутація* до вибраного для мутації i -го гена j -ої хромосоми h_{ij} застосовується за формулою:

$$h_{ij}' = h_{ij} + \varepsilon_i,$$

де ε_i – випадкове число, одержане за нормальним законом розподілу:

$$\varepsilon_i = N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(h_{ij}-\mu)^2}{2\sigma^2}},$$

де μ – математичне сподівання, як правило $\mu = 0$; σ – середньоквадратичне відхилення. Параметр σ може залежати від номера ітерації. В такому випадку він може бути розрахований за однією з формул:

$$\sigma_i(t) = \frac{T-t}{T} \cdot \frac{w_{\max,i} - w_{\min,i}}{3} \quad \text{або} \quad \sigma_i(t) = \sqrt{1/(1+t)}.$$

Число ε_i може бути додано до одного гена. Можливий варіант додавання випадкового вектора до всієї хромосоми.

Недоліком Гаусової мутації є те, що для ряду цільових функцій в разі розташування оптимуму осторонь від концентрації основної частини популяції існує велика ймовірність передчасної збіжності популяції до рішення, розташованого в області зосередження особин поточної популяції.

Тому з метою зменшення ймовірності даної події замість нормально-го розподілу для розрахунку значення параметра ε_i доцільно використовувати розподіл Коші:

$$\varepsilon_i = \frac{1}{\pi} \frac{a}{a^2 + h_{ij}^2},$$

де a – параметр розподілу Коші.

При використанні закону Коші більш ймовірним є отримання великих значень параметру ε_i , що відповідає більшому відхиленню нащадків від батьківських особин. Ця обставина дозволяє зменшити ймовірність передчасної збіжності, проте, в свою чергу, заважає більш ретельному дослідженню околу батьківської особини, що шкідливо на завершальних стадіях пошуку.

4. Мутація вибраного гена h_{ij} на основі квадратичної апроксимації.

Крок 1. Обчислити значення фітнес-функції при $h_{ij} + \Delta h_{ij}$ та при $h_{ij} - \Delta h_{ij}$: $f(h_{ij} + \Delta h_{ij})$ й $f(h_{ij} - \Delta h_{ij})$. Значення Δh_{ij} можуть бути обчислені за формулами знаходження Δ та ε , аналогічними нерівномірній і Гаусовій мутації.

Крок 2. Апроксимувати точки h_{ij} , $h_{ij} + \Delta h_{ij}$ та $h_{ij} - \Delta h_{ij}$ в параболу.

Крок 3. Знайти мінімальне значення одержаної кривої $f_{n, \min}$ і значення точки в просторі ознак, яка відповідає мініимальному значенню параболи $h_{ij, \min}$.

Крок 4. Присвоїти: $h_{ij} = h_{ij, \min}$.

Важливо відзначити, що описані оператори мутації можуть застосовуватися і для бінарних хромосом, заздалегідь перетворених до реальних числових значень з погляду вирішуваної задачі. Після застосування описаних вище операторів мутації до таких хромосом, їх необхідно знову перетворити до бінарного вигляду, застосувавши використовуваний метод кодування.

4.7.3.3 Мутація обміну

Мутація обміну [41, 48, 72] використовується для бінарних і числових негомологічних хромосом.

1. При *класичній мутації обміну* в хромосомі випадковим чином вибираються два гени, які міняються місцями.

Крок 1. Створити хромосому-нащадок як копію батьківської хромосоми $H = \{h_1, h_2, \dots, h_L\}$.

Крок 2. Вибрати два числа y_1 і y_2 випадковим чином з множини $Y = \{0, 1, 2, \dots, L + 1\}$, причому $y_1 \neq y_2$.

Крок 3. Сформувати нову хромосому H шляхом обміну елементів, що стоять на позиціях y_1 і y_2 .

Таким чином, після застосування класичної мутації обміну одержуємо хромосому H' :

$$H' = \{h_1, h_2, \dots, h_{y_1-1}, h_{y_2}, h_{y_1+1}, \dots, h_{y_2-1}, h_{y_1}, h_{y_2+1}, \dots, h_L\}.$$

2. *Одноточкова мутація обміну*. У даному операторі, на відміну від попереднього, обмінюються місцями лише сусідні гени, і точка мутації вибирається між двома генами.

Крок 1. Створити хромосому-нащадок як копію батьківської хромосоми $H = \{h_1, h_2, \dots, h_L\}$.

Крок 2. Вибрати точку мутації випадковим чином з множини $Y = \{1, 2, \dots, L - 1\}$.

Крок 3. Сформувати нову хромосому H шляхом обміну елементів, що стоять на позиціях y і $y + 1$.

Таким чином, після застосування одноточкової мутації обміну одержуємо хромосому H' :

$$H' = \{h_1, h_2, \dots, h_{y+1}, h_y, \dots, h_L\}.$$

3. *Мутація золотого перерізу*. У даному операторі вибір точки мутації здійснюється на основі правила «золотого перерізу», тобто точка мутації хромосом довжини L визначається за формулою: $D = \text{Ціле}(\tau \cdot L)$, де $\tau = (-1 \pm \sqrt{5})/2 \approx 0,61803$. В результаті застосування оператора мутації золотого перерізу хромосома $H = \{h_1, h_2, \dots, h_D, h_{D+1}, \dots, h_L\}$ перетвориться в хромосому $H = \{h_1, h_2, \dots, h_{D+1}, h_D, \dots, h_L\}$.

4. *Мутація на основі чисел Фібоначчі*. При використанні даного оператора гени для мутації вибираються на основі чисел Фібоначчі: $\varphi_0 = \varphi_1 = 1$, $\varphi_k = \varphi_{k-1} + \varphi_{k-2}$ для $k = 2, 3, \dots$. Тобто для мутації вибираються гени, позиції яких відповідають числам в числовій послідовності 1, 2, 3, 5, 8, 13, 21, ...

Крок 1. Створити хромосому-нащадок як копію батьківської хромосоми.

Крок 2. Вибрати гени для мутації на основі чисел Фібоначчі.

Крок 3. Виконати обмін числових значень між генами, що мутують, по колу із зсувом праворуч або вліворуч.

Необхідно відзначити, що застосування подібних операторів мутації до гомологічних числових і векторних хромосом може приводити до утворення неприпустимих рішень.

Другий спосіб мутації на основі чисел Фібоначчі може бути виконаний в такій послідовності кроків.

Крок 1. В заданій популяції хромосом вибрати батьківську хромосому довжини L з найменшим значенням цільової функції.

Крок 2. У вибраній хромосомі визначити точку розриву для реалізації оператора мутації методу Фібоначчі. Вона відповідає третьому числу ряду Фібоначчі. Встановити: $i = 0$.

Крок 3. За правилами побудови оператора простої мутації одержати нову хромосому-нащадок.

Крок 4. Обчислити значення цільової функції нащадка. Якщо знайдено глобальний оптимум, тоді перейти до кроку 9.

Крок 5. Встановити: $i = i + 1$.

Крок 6. Як точку мутації вибрати $(3 + i)$ -й член ряду Фібоначчі.

Крок 7. Якщо $(3 + i) \leq L$, виконати перехід до кроку 3.

Крок 8. В якості хромосоми-нащадка вибрати хромосому з оптимальним (кращим) значенням цільової функції.

Крок 9. Зупинення.

5. *Нечітка мутація на основі методу дихотомії.*

Крок 1. Розділити батьківську хромосому довжини L навпіл (при нечіткому розмірі в будь-яку частину береться більше ціле), визначивши точку розриву.

Крок 2. За правилами односточкової мутації одержати нову хромосому-нащадка.

Крок 3. Кожну половину хромосоми-нащадка знову розділити навпіл і процес розрахунку продовжувати по початковій схемі до тих пір, поки не буде одержана задана кількість хромосом-нащадків або оператор мутації методу дихотомії завершиться.

Крок 4. Зупинення.

6. У інженерних задачах, де необхідно виконувати аналіз моделей з великим числом елементів (близько 10^4), використовують *оператор мутації Монте-Карло* (мутація на основі статистичних випробувань). Цей оператор мутації полягає в єдиному обміні двох випадково вибраних генів або ансамблів з декількох генів.

Окрім цього, при розбитті моделей великих розмірностей на частини доцільно використовувати оператор мутації, що зберігає симетрію. Такий оператор полягає в обміні випадково вибраного сегменту генів з аналогічним сегментом, розташованим симетрично йому.

4.7.3.4 Інвертування

Оператор інвертування (інверсії) змінює порядок проходження генів в ділянках хромосом [12, 88]. Мета перепорядкування полягає в тому, щоб спробувати знайти порядок генів, який має кращий еволюційний потенціал. Перепорядкування також значно розширює область пошуку. Таким чином, генетичний метод намагається знаходити гарну множину значень генів, а також одночасно намагається знаходити гарне впорядкування генів. Це набагато важча задача для розв'язання. Найчастіше використовуються такі види інвертування.

1. *Класичне інвертування*: вибираються 2 точки, між якими відбувається перерозподіл гена.

Крок 1. Створити хромосому-нащадок як копію батьківської хромосоми $H = \{h_1, h_2, \dots, h_L\}$.

Крок 2. Вибрати два числа y_1 та y_2 випадковим чином з множини $Y = \{0, 1, 2, \dots, (L+1)\}$, причому $y_1 < y_2$.

Крок 3. Сформувати нову хромосому з H , інвертуючи сегмент, який лежить праворуч від позиції y_1 , і ліворуч від y_2 у хромосомі H .

Таким чином, після застосування класичного інвертування одержуємо хромосому H' :

$$H' = \{h_1, h_2, \dots, h_{y_1-1}, h_{y_1}, h_{y_2-1}, h_{y_2-2}, \dots, h_{y_1+1}, h_{y_2}, h_{y_2+1}, \dots, h_L\}.$$

2. *Інвертування із зсувом*: випадковим чином вибирається елемент і розміщується на випадково вибрану позицію, зсуваючи решту елементів циклічно праворуч.

Іншою формою такого виду інвертування є випадковий вибір підрядка і перенесення його на випадково вибрану позицію із зсувом елементів, що залишилися.

Як правило, спочатку задається ймовірність інвертування P_{in} . Конкретне значення P_{in} залежить від вирішуваної задачі і в загальному випадку знаходиться в інтервалі $[0,001; 0,01]$.

3. *Інвертування з використанням методу Фібоначчі*.

Крок 1. У заданій популяції хромосом вибрати батьківську хромосому довжини L .

Крок 2. Визначити точку розриву для реалізації оператора інвертування методу Фібоначчі. Вона відповідає третьому числу ряду Фібоначчі. Встановити: $i = 0$.

Крок 3. За правилами побудови оператора інвертування одержати нову хромосому-нащадка, інвертуючи праву частину від точки оператора інвертування з використанням методу Фібоначчі.

Крок 4. Обчислити значення цільової функції нащадка.

Крок 5. Встановити: $i = i + 1$.

Крок 6. Як точку інвертування вибрати $(3 + i)$ -й член ряду Фібоначчі.

Крок 7. Якщо $(3 + i) \leq L$, виконати перехід до кроку 3.

Крок 8. В якості хромосоми-нащадка вибрати хромосому з оптимальним значенням цільової функції.

Крок 9. Зупинення.

4. *Інвертування на основі методу золотого перерізу* є аналогічним оператору інвертування на основі чисел Фібоначчі. Для оператора інвертування методу золотого перерізу перша точка розриву визначається на відстані цілої частини $0,618L$ від будь-якого краю хромосоми. Потім елементи частини хромосоми, що розташовані між точкою розриву і правим кінцем хромосоми, інвертуються. Друга точка розриву в новій хромосомі визначається як найближче ціле з виразу $(L - 0,618L) \cdot 0,618$. Далі процес триває аналогічно до закінчення можливості розбиття хромосоми.

4.7.3.5 Транслокація, вставка і делеція

Застосування операторів транслокації, вставки і делеції [39, 72] може привести до зміни довжини хромосом, внаслідок чого ці оператори мутації використовуються дуже рідко, хоча в деяких класах комбінаторних задач вони можуть привести до непоганих результатів.

1. Транслокація.

Крок 1. Випадковим чином вибрати дві хромосоми.

Крок 2. На кожній хромосомі визначити точку зрізу.

Крок 3. Обміняти хромосоми ділянками, обмеженими точками зрізу.

2. Вставка, на відміну від транслокації, є унарним оператором.

Крок 1. Визначити точку (одну або декілька точок) вставки.

Крок 2. Проаналізувати гени хромосоми для визначення альтернативних вставок.

Крок 3. Провести пробну вставку генів (праворуч від точки вставки або між двома точками оператора вставки) і обчислити нове значення фітнес-функції хромосоми. Виконати дану операцію кілька разів.

Крок 4. Як результат оператора вставки прийняти хромосому, одержану на кроці 3, при якій досягається оптимальне значення фітнес-функції.

3. Делеція (видалення).

Крок 1. Створити хромосому-нащадок як копію батьківської хромосоми $H = \{h_1, h_2, \dots, h_L\}$.

Крок 2. На вибраній хромосомі визначити точку зрізу.

Крок 3. Видалити праву або ліву (від точки зрізу) ділянку з хромосоми.

Оператори вставки і видалення змінюють розмір хромосоми. Якщо це неприпустимо при вирішенні конкретної задачі, то ці два оператори необхідно застосовувати спільно.

4.7.4 Формування нового покоління

Після схрещування і мутації необхідно створити нову популяцію [18, 19, 45]. Види операторів формування нового покоління (репродукції, редукції) практично співпадають з видами операторів відбору батьків, що передбачають формування проміжного масиву особин, допущених до схрещування.

Необхідно визначити, які з нових особин увійдуть до наступного покоління, а які – ні. Для цього застосовують один з двох способів.

Перший спосіб полягає в тому, що *нові особини (нащадки) займають місця своїх батьків*. Після чого настає наступний етап, в якому нащадки оцінюються, відбираються, дають потомство і поступаються місцем своїм нащадкам.

Недоліком даного способу є можливість втрати найбільш пристосованої особини попереднього покоління. Одним із способів вирішення даної проблеми може бути використання *принципу елітизму*, який полягає в тому, що хромосоми з найбільшою пристосованістю гарантовано переходять в нову популяцію. Кількість елітних особин k_e , які гарантовано перейдуть в наступну популяцію, може бути обчислена за формулою:

$$k_e = (1 - S_0) \cdot N,$$

де S_0 – ступінь оновлення популяції, що знаходиться в діапазоні $[0,95; 1,0]$; N – розмір популяції.

Використання принципу елітизму зазвичай дозволяє прискорити збіжність генетичного пошуку. Недолік використання даної стратегії полягає в тому, що підвищується ймовірність попадання методу в локальний оптимум.

Другий спосіб передбачає створення проміжної популяції, яка включає як батьків, так і їх нащадків. Члени цієї популяції оцінюються, а потім з них вибираються N *найкращих, які й увійдуть до наступного покоління*. Другий варіант є більш оптимальним, але він вимагає сортування масиву розміром $2N$.

Другий варіант формування нового покоління можна реалізувати за допомогою *принципу витіснення*, який носить двокритеріальний характер: включення особин в нове покоління залежить не лише від величини їх фітнес-функції, але і від того, чи існує вже у формованій популяції наступного покоління особина з аналогічним хромосомним набором. З усіх особин з однаковими генотипами перевага спочатку віддається тим, чия пристосованість вище. Таким чином, досягаються дві мети: по-перше, не втрачаються кращі знайдені рішення, що мають різні хромосомні набори, а по-друге, в популяції постійно підтримується достатня генетична різноманітність. Витіснення в даному випадку формує нову популяцію скоріше з далеко розташованих особин, замість особин, що групуються біля поточного знайденого рішення. Цей метод особливо добре себе показав при вирішенні багатоекстремальних задач, при цьому крім визначення глобальних екстремумів з'являється можливість виділити і ті локальні оптимуми, значення яких близькі до глобальних.

4.7.5 Критерії зупинення

Одним з важливих елементів генетичного пошуку є критерії зупинення [45, 48, 107]. Очевидно, що еволюція – нескінченний процес, в ході якого пристосованість особин поступово підвищується. Примусово зупинивши цей процес через достатньо довгий час після його початку і вибравши найбільш пристосовану особину в поточному поколінні, можна одержати не абсолютно точне, але близьке до оптимального рішення.

Як правило, як критерій зупинення застосовується *обмеження на максимальну кількість ітерацій* функціонування методу (тобто обмеження на кількість поколінь). Кількість популяцій може бути будь-якою, але зазвичай встановлюється 50–100 популяцій. Якщо в якості критерію зупинення обирається максимальна кількість ітерацій, то задається кількість ітерацій T , внаслідок чого цикл генетичного пошуку виконується T разів.

Зупинення роботи генетичного методу може відбутися також у випадку, якщо *популяція вироджується*, тобто, якщо практично немає різноманітності в генах особин популяції. Виродження популяції називають передчасною збіжністю.

Якщо в якості критерію зупинення вибирається виродження популяції, то задаються кількість ітерацій $t_{\text{п}}$ та поріг коефіцієнта поліпшення значень фітнес-функції кращої хромосоми $\rho_{\text{п}}$.

Починаючи з циклу $(t_{\text{п}} + 1)$, обчислюються значення цільової функції для кожної хромосоми поточної t -ої популяції, і вибирається з цих значень найкраще значення цільової функції f_{best_t} .

З попередніх $(t - 1)$ поколінь вибирається найкраще значення цільової функції $f_{\text{best}_{t-1}}$.

Після чого обчислюється коефіцієнт поліпшення ρ за формулою:

$$\rho = \frac{f_{\text{best}_t} - f_{\text{best}_{t-1}}}{f_{\text{best}_{t-1}}}.$$

Якщо значення ρ виявляється меншим, ніж $\rho_{\text{п}}$, то критерій зупинення вважається досягнутим, в іншому випадку виконується наступний цикл генетичного пошуку.

Прийняття значення цільової функції $f_{\text{п}}$ також може використовуватися як критерій зупинення. Якщо в процесі функціонування генетичного методу значення цільової функції f деякої особини досягло значення $f_{\text{п}}$ з певною наперед заданою точністю ϵ , то метод зупиняється. При цьому рішенням задачі є набуте значення цільової функції $f_{\text{п}}$.

4.8 Теоретичний аналіз генетичного пошуку

Дотепер ще не розроблені повні теорії, які дозволяють аналізувати і прогнозувати динаміку поведінки генетичного пошуку. Найбільш відомими є гіпотеза про будівельні блоки і теорема схем [18, 23, 29], що дають якісну картину роботи генетичних методів. Проте їх дуже важко застосувати для кількісного аналізу динаміки роботи методів генетичного пошуку і вибору його параметрів, оскільки будівельні блоки, що є компонентами оптимального рішення, в реальних задачах спочатку є невідомими. Також робилися спроби аналізу генетичного пошуку за допомогою ланцюгів Маркова, принципів статистичної механіки і т. п., проте вони розглядають лише окремі випадки генетичного пошуку. Тому теоретичний аналіз різних генетичних методів проводиться індивідуально залежно від особливостей конкретного методу і вирішуваної за його допомогою задачі.

4.8.1 Основні поняття

Генетичний пошук, як метод оптимізації, має внутрішній паралелізм, який полягає в тому, що різні окремі суттєві комбінації генів, які називають *схемами* (*шимами*, *схематами* або *шаблонами*), відшукуються паралельним чином одночасно для всіх комбінацій [29]. Чим меншою є комбінація, тим легше вона може бути знайдена.

Іншими словами, хоча зовні здається, що генетичний метод обробляє хромосоми, але насправді при цьому неявно відбувається обробка схем, які являють собою шаблони подібності між хромосомами. Генетичний пошук практично не може займатися повним перебором всіх точок в просторі пошуку. Проте він може проводити вибірку значного числа гіперплощин в областях пошуку з високою пристосованістю. Кожна така гіперплощина відповідає множині схожих хромосом з високою пристосованістю.

Схема – це хромосома довжини L , що складається із знаків алфавіту $\{0; 1; *\}$, де $\{*\}$ – невизначений символ. Кожна схема визначає множину всіх бінарних хромосом довжини L , що мають у відповідних позиціях або 0, або 1, залежно від того, який біт знаходиться у відповідній позиції самої схеми.

Основною властивістю, яка використовується при аналізі генетичного пошуку, є те, що схема описує множину точок (хромосом) в просторі рішень, які мають певну схожість між собою. В той же час одна хромосома може належати різним схемам. Кожна хромосома в популяції є представником 2^L схем. Для популяції хромосом з N членів можна скласти не більше, ніж $N \cdot 2^L$ схем. Кількість різних бінарних схем для хромосом довжини L дорівнює 3^L .

Таким чином, при оцінюванні генетичного пошуку переходять від розгляду конкретного рядка-хромосоми до розгляду набору хромосом, тобто до розгляду схем. У загальному випадку пошук оптимального рішення відбувається в L -мірному просторі, де L – довжина хромосоми. Схема є гіперплощиною в цьому просторі.

У схем виділяють дві властивості: порядок і визначена довжина. *Порядок схеми $O(H)$* – це число визначених бітів (0 або 1) в схемі. *Визначена довжина $L(H)$* – відстань між крайніми визначеними бітами в схемі.

Пристосованість (якість) схеми визначається як середня пристосованість хромосом, які її містять.

4.8.2 Аналіз впливу генетичних операторів

Для теоретичного аналізу функціонування методів генетичного пошуку необхідно досліджувати індивідуальний та сумісний вплив генетичних операторів на схеми, що містяться всередині популяції хромосом [12, 23]. Для цього необхідно проаналізувати вплив генетичних операторів відбору, схрещування і мутації.

4.8.2.1 Вплив оператора відбору

Нехай на певному часовому інтервалі t з'являється m представників деякої схеми H , що міститься в популяції $P(t)$, тобто $m = m(H, t)$. При виконанні пропорційного відбору хромосома вибирається пропорційно своїй значущості, що визначається як значення фітнес-функції.

Таким чином, кількість представників схеми H у момент часу $(t + 1)$ буде дорівнювати:

$$m(H, t + 1) = m(H, t) \cdot N \cdot \frac{f(H, t)}{\sum_{j=1}^N f(H_j)} = m(H, t) \frac{f(H, t)}{f_{\text{cp.}}(t)},$$

де $f(H, t)$ – середня пристосованість хромосом-представників схеми H у момент часу t ; $f_{\text{cp.}}(t)$ – середня пристосованість всіх хромосом популяції у момент часу t .

З наведеної формули виходить, що схема з середнім значенням пристосованості, яке перевищує середнє значення пристосованості по популяції, на наступній ітерації збільшить число своїх представників, в той час, як схема з меншим середнім значенням пристосованості зменшить кількість своїх представників в новому поколінні. Таким чином, кількість високо пристосованих схем зростає, а решта схем вмирає [18].

З метою отримання математичних закономірностей темпів зміни кількості представників певних схем припустимо, що кількість представників деякої схеми H змінюється за формулою cf , де c – константа, що характеризує темп зміни кількості представників схем. В такому разі попередній вираз можна записати таким чином:

$$m(H, t + 1) = (1 + c) \cdot m(H, t).$$

Припускаючи, що c – постійна величина, отримаємо:

$$m(H, t) = (1 + c)^t \cdot m(H, 0).$$

Таким чином, зміна кількості представників різних схем відбувається за деяким показниковим законом.

4.8.2.2 Вплив оператора схрещування

В результаті схрещування деякі схеми руйнуються, а інші виживають. Одержувана схема вважається *зруйнованою*, якщо вона не відповідає жодному з нащадків [39].

Наприклад, розглянемо дві схеми 00^{**} і $0^{**}0$. В результаті одноточкового схрещування кожна з вказаних схем може бути зруйнована лише тоді, коли точка розриву потрапить між двома нулями. Проте дана подія з більшою ймовірністю відбудеться для другої схеми. Таким чином, ймовірність руйнування таких схем пропорційна відстані між нулями.

Отже, якщо точка розриву одноточкового оператора схрещування вибирається випадково серед $(L - 1)$ можливих позицій, тоді схема H знищується з ймовірністю $P_d(H) = L(H) / (L - 1)$, а виживає з ймовірністю $P_s(H) = 1 - P_d(H) = 1 - L(H) / (L - 1)$.

Одноточкове схрещування характеризується тим, що воно більше руйнує схеми з великою визначеною довжиною, ніж з короткою. Двоточкове схрещування руйнує схеми з великою визначеною довжиною з меншою ймовірністю, ніж одноточкове. І взагалі, n -точкове схрещування з меншою ймовірністю руйнує схеми при парному n , ніж при непарному.

Рівномірне схрещування більш руйнівне для схем, ніж одноточкове і двоточкове, але ймовірність руйнування схеми в даному випадку не залежить від довжини схеми. Також встановлено, що рівномірне схрещування успішніше конструює представників нових якісніших схем з менш якісних, ніж одноточкове і двоточкове схрещування.

Проте проблему вибору оператора схрещування треба вирішувати кожного разу стосовно конкретної задачі.

4.8.2.3 Вплив оператора мутації

Для того, щоб схема не була зруйнована внаслідок застосування оператора мутації, необхідно, щоб всі значущі позиції даної схеми залишилися незмінними [45].

Кожна алель схеми окремо не зміниться з ймовірністю $(1 - P_m)$. Враховуючи, що кожна мутація є незалежною, можна зробити висновок про те, що схема виживає, якщо кожна з $O(H)$ значущих позицій не змінюється. Помноживши вираз $(1 - P_m)$ сам на себе $O(H)$ разів, одержимо, що ймовірність виживання схем після мутації складе $(1 - P_m)^{O(H)}$.

Для невеликих значень $P_m \ll 1$ можна обчислити наближене значення ймовірності виживання схем за формулою: $1 - O(H) \cdot P_m$.

4.8.3 Теорема схем

Математично механізм роботи генетичного пошуку можна виразити за допомогою теореми схем [29, 48]. Ця теорема, сформульована Д. Холландом, доводить, що застосування генетичних операторів дозволяє формувати нову популяцію з кращими показниками, ніж у поточної. Це забезпечується експоненційним зростанням корисних схем від ітерації до ітерації.

Нехай є популяція двійкових рядків (хромосом) довжини L . Ймовірність схрещування дорівнює P_c . Тип схрещування: однокриве. Визначена довжина схеми H дорівнює $L(H)$, а її пристосованість $f(H)$. Частина рядків, що відповідають схемі H в поточному поколінні t , дорівнює $m(H, t)$.

Необхідно оцінити, яка частина рядків, що відповідають схемі H , буде присутня в популяції в наступному поколінні – $m(H, t+1)$.

Очевидно, що якщо точка розриву не потрапляє всередину вже наявної схеми, то схема не буде зруйнована. Тобто, якщо $P_c \cdot L(H)/(L - 1)$ – ймовірність того, що точка розриву потрапить всередину схеми, то

$1 - P_c \frac{L(H)}{L - 1} \left(1 - \frac{m(H, t) f(H, t)}{N f_{cp.}(t)} \right)$ – ймовірність того, що схрещування не

зруйнує схему.

Згідно стратегії відбору канонічного генетичного методу шанси особи взяти участь в схрещуванні обчислюються відповідно до відношення $f/f_{cp.}$, де f – значення пристосованості даної особи, $f_{cp.}$ – середня пристосованість. Таким чином, ймовірність того, що хромосома, яка відповідає схемі H , братиме участь в схрещуванні дорівнює $m(H, t) \cdot f(H, t) / f_{cp.}(t)$. Беручи до уваги ймовірність руйнування схеми після схрещування, одержуємо:

$$m(H, t+1) \geq m(H, t) \frac{f(H, t)}{f_{cp.}(t)} \left[1 - P_c \frac{L(H)}{L-1} \left(1 - \frac{m(H, t)}{N} \frac{f(H, t)}{f_{cp.}(t)} \right) \right].$$

Одним з недоліків даної теореми є те, що в ній не враховується вплив мутації на створення і руйнування схем. Якщо вважати, що ймовірність мутації дорівнює P_m , а порядок схеми H дорівнює $O(H)$, то ймовірність того, що мутація не зруйнує схему, дорівнює $(1 - P_m)^{O(H)}$. Тобто, якщо розряд, що мутує, не потрапляє ні на одну фіксовану позицію усередині схеми, то вона не змінюється. З урахуванням цього виправлена теорема схем виглядає таким чином:

$$m(H, t+1) \geq m(H, t) \frac{f(H, t)}{f_{cp.}(t)} \left[1 - P_c \frac{L(H)}{L-1} \left(1 - \frac{m(H, t)}{N} \frac{f(H, t)}{f_{cp.}(t)} \right) \right] (1 - P_m)^{O(H)}.$$

Для базової структури генетичного пошуку можливо безпосереднє застосування класичного варіанту теореми схем. Проте, зміна в структурі генетичних методів, внесення додаткових генетичних операторів, введення адаптаційних механізмів, перехід від послідовного до паралельного пошуку і т. ін. приводять до необхідності внесення доповнень в базу теорему.

Крім того, в теоремі схем практично не враховується та обставина, що схрещування і мутація можуть не лише руйнувати схему, але і створювати її з інших схем. Тому в теоремі схем присутній знак нерівності.

Ще одним недоліком теореми схем є те, що вона дозволяє розрахувати частину схем в популяції лише для наступного покоління. Тобто спроба оцінити за допомогою теореми схем число хромосом, що відповідають даній схемі через декілька поколінь, до успіху не приведе. Так виходить, зокрема, через імовірнісний характер стратегії відбору.

Проте, не дивлячись на наведені аргументи, теорема схем є першою серйозною успішною спробою зрозуміти, як і чому працюють генетичні методи. Сама ж теорема схем є на сьогодні найпоширенішим інструментом аналізу генетичних методів.

4.8.4 Будівельні блоки

З теореми схем виходить, що короткі (по відношенню до схрещування), низького порядку (по відношенню до мутації) і, що мають оптимальність, вище за середню, схеми, збільшують своє представництво в послідовності поколінь експоненційно [29, 72].

Будівельні блоки – це схеми, що мають високу пристосованість, низький порядок і коротку визначену довжину.

Після процедури відбору залишаються лише хромосоми з більш високою пристосованістю. Отже, хромосоми, які є прикладами схем з високою присто-

сованістю, вибираються частіше. Холланд показав, що, тоді як генетичний метод явним чином обробляє N хромосом в кожному поколінні, в той же час неявно обробляються порядку N^2 таких коротких схем низького порядку і з високою пристосованістю (корисних схем). Це явище названо *неявним паралелізмом*. Для вирішення реальних задач присутність неявного паралелізму означає, що велика популяція має більше можливостей виявити глобальний оптимум швидше, ніж популяція з меншим числом особин.

4.9 Програмні засоби генетичної оптимізації

Одним з найбільш потужних засобів для виконання генетичної оптимізації є пакет Matlab. Оптимізацію на основі генетичного пошуку дозволяє виконувати бібліотека Genetic Algorithm and Direct Search Toolbox, що містить такі компоненти для генетичної оптимізації:

- функції, що реалізують процес виконання генетичного пошуку та основні генетичні оператори;
- візуальний інтерфейсний модуль для роботи з генетичними методами;
- демонстраційні приклади.

Для використання генетичних методів, у середовищі Matlab передбачена функція `ga`, формат виклику якої наведено нижче:

```
[x, Fmin] = ga(@fitnessfun, n, options)
```

Функція `ga` знаходить мінімум `Fmin` функції `fitnessfun`, що має `n` параметрів, а також вектор `x`, при якому досягається мінімум цільової функції.

Параметри роботи функції `ga` задаються в змінній `options`, що являє собою структуру, у якій можуть бути задані: спосіб подання інформації в хромосомі, використовувані генетичні оператори відбору, схрещування й мутації, критерії зупинення й інші параметри генетичного пошуку. За допомогою структури `options` також можна вибрати графіки, які будуть відображати основні результати генетичного пошуку в процесі оптимізації цільової функції.

Для зміни значень параметрів функції `ga` передбачена функція `gaoptimset`, а для одержання поточних параметрів – функція `gaoptimget`.

Візуальний інтерфейсний модуль для роботи з генетичними методами Genetic Algorithm Tool (GAT) дозволяє більш зручними візуальними засобами виконувати оптимізацію функцій за допомогою методів генетичного пошуку.

Запуск GAT виконується за командою `gatool`. Головна форма GAT зображена на рис. 4.12. GAT містить два загальносистемні пункти меню: File і Help. Меню File дозволяє імпортувати й експортувати в робочу об-

ласті параметри генетичного пошуку. За допомогою меню Help можна одержати довідку по роботі з модулем Genetic Algorithm Tool.

Поля Fitness function і Number of variables призначені для вказівки цільової функції й кількості її параметрів, відповідно.

В області Plots вибираються графіки, що ілюструють роботу генетичного методу, а також вказується період відображення результатів (поле Plot interval):

- best fitness – графік залежності кращого й середнього значень фітнес-функції від номеру ітерації еволюційного пошуку;
- expectation – очікувана кількість нащадків для кожного значення фітнес-функції на поточній ітерації;
- score diversity – гістограма, що відображає кількість хромосом для кожного значення фітнес-функції, отриманого в поточній популяції;
- stopping – досягнення критеріїв зупинення на поточній ітерації;
- best individual – значення керованих змінних для кращого члена поточної популяції;
- genealogy – генеалогія індивідів: червоними лініями відображені нащадки, отримані в результаті мутації, синіми – в результаті схрещування, чорними – елітні особини;
- scores – гістограма, що містить значення фітнес-функції для кожної хромосоми поточної популяції;
- distance – графік залежності середньої відстані між хромосомами популяції від номера ітерації;
- range – графік залежності діапазону зміни значень фітнес-функції від номера ітерації;
- selection – гістограма, що відображає кількість нащадків для кожної хромосоми поточної популяції;
- custom function – функція виведення результатів еволюційного пошуку, визначена користувачем.

В області Run Solver відображаються досягнуті результати генетичної оптимізації. Запуск генетичного методу відбувається шляхом натискання кнопки Start. При необхідності пошук можна призупинити (кнопка Pause) або завершити (кнопка Stop). У полі Status and Results відображаються результати роботи генетичного методу (досягнуте значення фітнес-функції, а також причина зупинення генетичного пошуку), а в таблиці Final Point – значення вектора керованих змінних, при яких цільова функція досягає мінімуму

Область Options призначена для завдання параметрів роботи генетичного методу, що відповідають структурі options.

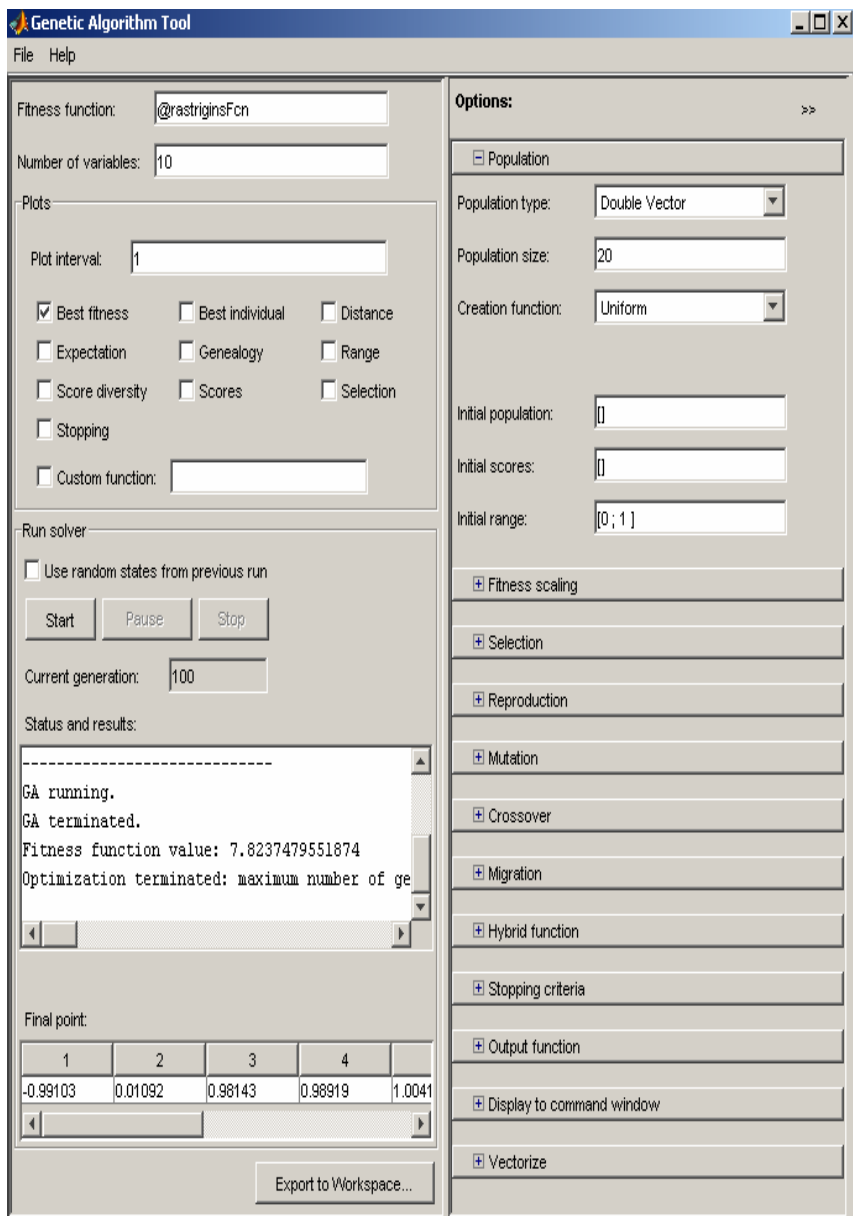


Рисунок 4.12 – Головна форма GAT

РОЗДІЛ 5 ЕВОЛЮЦІЙНІ СТРАТЕГІЇ. ГЕНЕТИЧНЕ ТА ЕВОЛЮЦІЙНЕ ПРОГРАМУВАННЯ

5.1 Еволюційні стратегії

Еволюційні стратегії орієнтовані на оптимізацію неперервних функцій з використанням операторів схрещування [18].

В той час, як генетичний метод моделює еволюцію на рівні геномів (хромосом), еволюційні стратегії спрямовані на еволюцію фенотипів. Оскільки еволюційні стратегії створювались спеціально для чисельної оптимізації, в них фенотипи подаються дійсними векторами.

5.1.1 ($\mu + \lambda$) та (μ, λ) еволюційні стратегії

Оригінальна (у первинному варіанті) еволюційна стратегія була двоелементною схемою, що складалася з батька і нащадка. У базовому методі батько, мутуючи, створює нащадка, і одна з двох особин з кращим значенням фітнес-функції переходить в популяцію наступного покоління. Цей метод був узагальнений пізніше на так звані ($\mu + \lambda$) і (μ, λ) еволюційні стратегії. Параметри μ і λ означають число батьків і нащадків, відповідно.

У еволюційній ($\mu + \lambda$) стратегії μ батьків породжують λ нащадків. При цьому батьки і нащадки беруть участь у загальному селективному відборі, внаслідок чого μ кращих особин переходять в наступну популяцію.

Еволюційна (μ, λ) стратегія передбачає, що μ батьків, породивши λ нащадків, гинуть. У наступну популяцію переходять μ особин з множини нащадків, причому в даному способі повинна виконуватися умова $\lambda > \mu$. Такий підхід може бути застосовний до задач з оптимумом, що змінюється, і з зашумленими даними.

Коефіцієнт $\eta = \lambda/\mu$ зазвичай більше або дорівнює семи. Чим він більше, тим більше шансів, що кожен батько згенерує, щонайменше, одного нащадка, кращого, ніж він сам. Проте відмінності між ($\mu + \lambda$) та (μ, λ) еволюційними стратегіями стають меншими, якщо η є достатньо великим.

Канонічна еволюційна стратегія може бути подана такою послідовністю кроків.

Крок 1. Встановити: $t = 0$.

Крок 2. Ініціалізувати популяцію P_t μ випадковими індивідами.

Крок 3. Випадковим чином відібрати особини з P_t для генерації λ нащадків.

Крок 4. Виконати оператор мутації над нащадками.

Крок 5. Обчислити фітнес-функції нащадків.

Крок 6. Вибрати кращі μ особин, базуючись на значеннях фітнес-функції, і створити популяцію P_{t+1} .

Крок 7. Встановити: $t = t + 1$.

Крок 8. Виконати перевірку критеріїв зупинення пошуку. В разі їх невиконання, перейти до кроку 3.

Крок 9. Зупинення.

Порівняльний аналіз еволюційних стратегій і генетичних методів показує, що головна їх відмінність в тому, що тоді як в генетичному методі вибираються особини для рекомбінації пропорційно їх фітнес-функції і замінюються особини з попередньої популяції, в еволюційних стратегіях діють навпаки. В цьому випадку особини для репродукції вибираються з рівними ймовірностями, а формування наступної популяції базується на значеннях фітнес-функцій.

5.1.2 Оператор мутації

Мутація, яка часто в еволюційних стратегіях є єдиним еволюційним оператором, полягає в зміні кожного елементу вектора (хромосоми) на величину, що має нормальний розподіл, середньоквадратичне відхилення $\sigma(t)$ якого адаптується у часі [18, 41].

5.1.2.1 Правило «20% успіху»

У стратегії (1 + 1) так зване правило «20% успіху» (1/5 успіху) використовується для адаптації середньоквадратичного відхилення. При цьому таке правило може застосовуватися на кожній ітерації або через кожні A ітерацій. Періодично під час пошуку одержують частоту успіху, тобто відношення кількості успіхів до всього числа спроб (мутацій). Під успішністю виконання еволюційного оператора над певною хромосомою мають на увазі появу нової хромосоми, фітнес-функція якої краща в порівнянні з фітнес-функцією батьківської хромосоми. Якщо відношення більше 1/5, то середньоквадратичне відхилення зростає, інакше – спадає:

$$\sigma(t) = \begin{cases} c\sigma(t - A), & \text{якщо } k < 1/5; \\ \sigma(t - A), & \text{якщо } k = 1/5; \\ \frac{\sigma(t - A)}{c}, & \text{якщо } k > 1/5, \end{cases}$$

де A – кількість ітерацій, впродовж яких не відбувається зміна середньоквадратичного відхилення; $\sigma(t)$ та $\sigma(t - A)$ – середньоквадратичне відхилення нормального розподілу на t -ій та $(t - A)$ -ій ітераціях, відповідно;

c – параметр швидкості зміни кроку середньоквадратичного відхилення, значення якого вибирають в інтервалі $0,817 \leq c \leq 1$; k – коефіцієнт успішності виконання мутації від $(t - A - 1)$ -ої до $(t - 1)$ -ої ітерації:

$$k = \sum_{l=t-A-1}^{t-1} \left(\frac{1}{B_l} \sum_{j=1}^{B_l} u(H_j^l) \right),$$

де B_l – кількість виконаних мутацій на l -ій ітерації; $u(H_j^l)$ – функція, що характеризує успішність виконання мутації на l -ій ітерації для j -ої хромосоми. При мінімізації фітнес-функції f значення $u(H_j^l)$ розраховується за формулою:

$$u(H_j^l) = \begin{cases} 1, & \text{якщо } f(H_j^l) < f(H_j^{l-1}); \\ 0, & \text{якщо } f(H_j^l) \geq f(H_j^{l-1}). \end{cases}$$

При використанні $(\mu + \lambda)$ або (μ, λ) еволюційних стратегій, в яких μ хромосом-батьків породжують λ нащадків, функцію успішності u_t виконання мутації на t -ій ітерації розраховують за формулою:

$$u_t = \begin{cases} 1, & \text{якщо } \frac{\sum_{j=1}^{\mu} f(H_j^t)}{\mu} < \frac{\sum_{j=1}^{\lambda} f(H_j^{t-1})}{\lambda}; \\ 0, & \text{якщо } \frac{\sum_{j=1}^{\mu} f(H_j^t)}{\mu} \geq \frac{\sum_{j=1}^{\lambda} f(H_j^{t-1})}{\lambda}. \end{cases}$$

Коефіцієнт успішності виконання мутації від $(t - A - 1)$ -ої до $(t - 1)$ -ої ітерації при використанні $(\mu + \lambda)$ або (μ, λ) еволюційних стратегій розраховується за формулою:

$$k = \sum_{l=t-A-1}^{t-1} u_l.$$

5.1.2.2 Мутація в багатоелементних еволюційних стратегіях

Якщо використовують багатоелементну стратегію для еволюції популяції з μ батьків та λ нащадків, кожна особина складається з двох дійсних векторів. Перший вектор містить змінні значення, а другий – відповідні середньоквадратичні відхилення, що використовуються в операторі мутації.

Більш прийнятною за правило «20% успіху» при виконанні мутації в багатоелементних стратегіях є зміна середньоквадратичних відхилень з використанням логнормального розподілу:

$$\sigma_j(t) = \sigma_j(t-A) \cdot e^{-\frac{t^2}{2\gamma^2}},$$

де $\sigma_j(t)$ та $\sigma_j(t-A)$ – середньоквадратичні відхилення j -ої хромосоми на t -ій та $(t-A)$ -ій ітераціях, відповідно; γ – коефіцієнт, що характеризує ступінь зміни значень вектора середньоквадратичних відхилень.

Збіжність еволюційних стратегій чутлива до вибору коефіцієнта γ та початкових значень вектора середньоквадратичних відхилень. Універсального методу їх отримання незалежно від цільової функції поки не існує. Рекомендується приймати:

$$\gamma^2 = \frac{C}{\sqrt{\sum_{j=1}^N \sigma_j^2}},$$

де C залежить від μ та λ . Приймають $C = 1$ для еволюційної стратегії (10,100).

Для ініціалізації вектора середньоквадратичних відхилень початковими значеннями використовується рівність:

$$\sigma_j = \frac{R_j}{\sqrt{\sum_{j=1}^N \sigma_j^2}},$$

де константа R_j – максимальний ранг невизначеності відповідної змінної.

5.1.3 Метод імітації відпалу

Метод імітації відпалу (simulated annealing) є імовірнісним варіантом локального спуску з можливістю переходу до гірших рішень [18]. Основними параметрами методу є: початкова температура, коефіцієнт зниження температури, довжина температурного інтервалу.

Метод імітації відпалу з нульовою температурою можна розглядати як варіант (1 + 1) еволюційної стратегії, що дозволяє віднести його до методів еволюційного пошуку. Відмінність полягає в тому, що замість вибору нового рішення з околу в (1 + 1) еволюційній стратегії використовується оператор мутації.

Даний метод моделює природний процес відновлення. *Відновлення* – це фізичний процес, який полягає в нагріванні та подальшому контрольованому охолодженні субстанції. В результаті виходить міцна кристалічна структура, яка відрізняється від структури з дефектами, що утворюється при швидкому неконтрольованому охолодженні. *Структура* тут є ко-

дованим рішенням, а *температура* використовується для того, щоб вказати, як і коли прийматимуться нові рішення.

Властивості структури залежать від коефіцієнта охолодження після того, як субстанцію було нагріто до точки плавлення. Якщо структура охолоджувалася повільно, будуть сформовані великі кристали, що дуже корисно для будови субстанції. Якщо субстанція охолоджується стрибкоподібно, утворюється слабка структура.

Послідовність виконання методу імітації відпалу наведено нижче.

Крок 1. Створити *початкове рішення*. Для більшості задач початкове рішення буде випадковим. На найпершому кроці воно присвоюється *поточному рішенню* (current solution). Інша модифікація методу полягає в тому, щоб завантажити в якості початкового рішення такий розв'язок, який вже знайдений за допомогою попереднього запуску методу імітації відпалу або іншого пошукового методу. Це надає методу базу, на підставі якої виконується пошук оптимального рішення задачі.

Крок 2. Оцінити рішення.

Крок 2.1. Декодувати поточне рішення. При цьому закодоване рішення може просто складатися з набору змінних. Вони будуть декодовані з існуючого рішення.

Крок 2.2. Виконати необхідну дію, що дозволяє оцінити ефективність поточного рішення для даної задачі.

Крок 3. Знайти за допомогою випадкового пошуку нове рішення.

Крок 3.1. Скопіювати поточне рішення в *робоче рішення* (working solution).

Крок 3.2. Випадковим чином виконати модифікацію робочого рішення. Спосіб модифікації робочого рішення залежить від того, яким чином воно подається (кодується).

Крок 3.3. Після виконання пошуку робочого рішення його необхідно оцінити описаним раніше способом (крок 2).

Крок 4. Визначити критерій допуску.

На цьому етапі методу є два рішення:

- оригінальне (поточне) рішення;
- знайдене (робоче) рішення.

З кожним рішенням пов'язана певна енергія, що являє собою його ефективність (припустимо, що чим нижча енергія, тим рішення є більш ефективним).

Крок 4.1. Порівняти робоче рішення з поточним рішенням. Якщо робоче рішення має меншу енергію, ніж поточне рішення (тобто є переважнішим), то робоче рішення копіюється в поточне рішення і відбувається перехід до етапу зниження температури (крок 6).

Крок 4.2. Якщо робоче рішення є гіршим, ніж поточне рішення, то необхідно визначити критерій допуску, щоб з'ясувати, що слід зробити з поточним робочим рішенням. Ймовірність допуску розраховується за формулою:

$$P(\delta E) = e^{-\delta E / T},$$

де $\delta E = E_{ws} - E_{cs}$ – різниця енергії робочого рішення E_{ws} і енергії поточно-го рішення E_{cs} ; T – температура.

Значення цієї формули полягає в наступному: при високій температурі (понад 60 °C) погане рішення приймається частіше, ніж відкидається. Якщо енергія менша, ймовірність прийняття рішення вище. При зниженні температури ймовірність прийняття гіршого рішення також знижується. При цьому вищий рівень енергії також сприяє зменшенню ймовірності прийняття гіршого рішення. При високих температурах імітоване відновлення дозволяє приймати гірші рішення для того, щоб провести повніший пошук рішень. При зниженні температури діапазон пошуку також зменшується, доки не досягається рівність при температурі 0°.

Крок 5. Повтор. При одній температурі виконується декілька (задана кількість) ітерацій (кроки 3 та 4). Після завершення ітерацій температуру необхідно знизити (крок 6).

Крок 6. Виконати зниження температури. Існує багато варіантів зниження температури. Наприклад, для зниження температури може бути використана функція:

$$T_{i+1} = \alpha T_i, \quad \alpha < 1,$$

де T_{i+1} , T_i – температура на $(i + 1)$ -ій та i -ій ітераціях, відповідно; α – коефіцієнт, що визначає ступінь зниження температури. Можна використовувати і інші стратегії зниження температури, включаючи лінійні і нелінійні функції.

Крок 7. Виконати перевірку критеріїв закінчення пошуку. Процес продовжується, доки температура не досягне нуля.

Крок 8. Зупинення.

Важливо відзначити, що однією з переваг даного методу є використання техніки випадкового пошуку, що дозволяє застосовувати його до знаходження глобального оптимуму, виключивши попадання в локальний оптимум, що є дуже важливим при розв'язанні реальних задач.

5.2 Генетичне програмування

Генетичне програмування застосовує еволюційний пошук для оптимізації комп'ютерних програм [12, 18].

Як правило, методи генетичного програмування використовуються для пошуку функціональної залежності, що найповніше відображає досліджуваний об'єкт, процес або явище. При цьому як фітнес-функція може бути використаний критерій, що характеризує ефективність застосування одержаної функціональної залежності або моделі.

5.2.1 Основні поняття

У генетичному програмуванні гіпотези про вигляд залежності цільової функції від незалежних змінних формулюються у вигляді програм на деякій внутрішній мові програмування. Процес пошуку оптимальної програми відбувається за допомогою еволюційного підходу, при якому особини популяції подаються у вигляді комп'ютерних програм.

Коли метод генетичного програмування знаходить програму, що задовільно виражає шукану залежність, він починає вносити в неї невеликі модифікації і відбирає серед побудованих дочірніх програм ті, які підвищують точність. Таким чином, відбувається синтез декількох генетичних ліній програм, які конкурують між собою за точністю виразу шуканої залежності.

При реалізації методів генетичного програмування розробляють спеціальний модуль, який переводить знайдені залежності з внутрішньої мови системи на зрозумілу користувачу мову (математичні формули, моделі, таблиці і т. ін.).

5.2.2 Етапи генетичного програмування

Генетичне програмування використовує дерево подань рішень, над яким виконуються генетичні оператори схрещування і мутації для отримання нових рішень шляхом виконання певних дій над сегментами дерев, що являють собою фрагменти комп'ютерних програм.

Генетичне програмування виконується в такій послідовності кроків.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$.

Крок 2. Ініціалізувати початкову популяцію P_0 випадково згенерованими рішеннями.

Крок 3. Оцінити (обчислити значення цільової функції) нові програми в популяції.

Крок 4. Вибрати програми P_{temp} , які згенерують нащадків.

Крок 5. Застосувати генетичні оператори схрещування і мутації до програм з P_{temp} для формування наступного покоління P_{t+1} .

Крок 6. Встановити: $t = t + 1$.

Крок 7. Перевірити критерії зупинення. В разі їх невиконання, перейти до кроку 3.

Крок 8. Зупинення.

5.2.3 Оператори генетичного програмування

Методи генетичного програмування використовують генетичні оператори відбору, схрещування і мутації.

Оператори відбору, які розроблені для генетичних методів, можуть бути також застосовні і в методах генетичного програмування.

Основною особливістю реалізації генетичних операторів схрещування і мутації при використанні генетичного програмування є підтримка синтаксичної цілісності одержуваних програм.

5.2.3.1 Схрещування

Оператор схрещування у генетичному програмуванні виконується так.

Крок 1. Вибрати випадковим чином на кожному з батьківських дерев одну або декілька точок (вузлів дерева подань рішень).

Точки перетину вибираються випадковим чином для кожного батька окремо, що призводить до виникнення декількох особливостей:

– при виборі точок перетину необхідно враховувати те, що одержані в результаті схрещування нащадки повинні зберегти синтаксичну структуру. Тому при реалізації складних мов можуть виникати проблеми з вибором сумісних точок перетину;

– як наслідок вибору різних точок перетину, в генетичному програмуванні однакові батьки можуть породжувати різних нащадків.

Крок 2. Виконати генерацію нащадків. Для цього необхідно обміняти у початкових програм фрагменти дерев відповідно до вибраних на попередньому кроці точок розриву.

5.2.3.2 Мутація

У генетичному програмуванні використовуються такі оператори мутації.

1. *Мутація із збільшенням* (grow mutation).

Крок 1. Випадковим чином вибрати вузол дерева подань програм для виконання мутації.

Крок 2. У вибраному вузлі побудувати нову гілку програми відповідно до семантики використовуваної мови.

2. *Мутація із стисненням* (shrink mutation).

Крок 1. Випадковим чином вибрати вузол дерева подань програм для виконання мутації.

Крок 2. Видалити вибраний вузол так, щоб збереглася синтаксична правильність одержуваної програми.

3. *Мутація із заміною* (switch mutation).

Крок 1. Вибрати випадковим чином дві гілки дерева подань програм для виконання мутації.

Крок 2. Обміняти вибрані гілки місцями. При цьому необхідно забезпечити синтаксичну правильність одержуваної програми.

4. *Циклічна мутація* (cycle mutation).

Крок 1. Вибрати випадковим чином вузол дерева подань програм для виконання мутації.

Крок 2. Випадковим чином замінити вибраний вузол на вузол того ж типу.

5.3 Еволюційне програмування

Еволюційне програмування орієнтоване на оптимізацію неперервних функцій без використання оператора схрещування. Інший напрям еволюційного програмування пов'язаний з пошуком залежності цільових параметрів від інших змінних у формі функцій якогось певного вигляду [18, 41, 55]. Наприклад, в одному з найбільш вдалих методів цього типу, методі групового врахування аргументів, залежність шукають у формі поліномів.

Від генетичних методів еволюційне програмування відрізняється не лише відсутністю схрещування, але і поданням особин популяції. Оскільки тут немає необхідності у використанні хромосом, то особина часто є реальним рішенням без додаткового кодування. Наприклад, в еволюційному програмуванні в якості особин дуже часто використовуються скінченні автомати. Відповідно оператори відбору і мутації застосовуються безпосередньо до них. При оптимізації функцій в якості особин популяції використовуються безпосередньо вектори дійсних чисел.

5.3.1 Основні поняття

Еволюційне програмування у вигляді еволюції логічних автоматів, що вирішує задачі прогнозування, діагностики, розпізнавання і класифікації образів, а також задачі керування об'єктом з невідомим характером, було запропоновано Л. Фогелем, А. Оуенсом і М. Уолшем.

Логічні (скінченні) автомати – це моделі, що описують засобами формальної логіки, можливі переходи досліджуваної системи з деякого початкового стану в завершальний. Зручною формою подання скінченних автоматів є орієнтовані графи.

Еволюційна програма реалізує моделювання процесів природної еволюції моделей-автоматів, причому в кожний момент часу зберігається особина, яка найкращим чином може справитися з даною задачею. Батьківський організм оцінюється в залежності від здатності приймати необхідне рішення на основі наявних даних. Цей організм піддається мутації і породжує нащадка, якому ставиться та ж задача і який оцінюється тим же чином. Автомат, який демонструє найкращу здатність виконувати необхідні функції, зберігається і породжує нащадків, які переходять в наступне покоління. Процес завершується, коли одержана достатньо гарна програма або вичерпані ресурси часу. Всякий раз, коли надходить нова

інформація, відбувається еволюційний пошук логічної структури, що забезпечує отримання найбільш прийняттого рішення.

5.3.2 Етапи еволюційного програмування

Узагальнена послідовність виконання еволюційного програмування включає такі кроки.

Крок 1. Виконати ініціалізацію параметрів методу.

При використанні в якості особин скінчених автоматів необхідно сформулювати вхідний словник, множину вхідних і вихідних станів, набір можливих станів, умови переходів зі стану в стан, фітнес-функцію для характеристики моделей, що генеруються.

Крок 2. Випадковим чином згенерувати початкову популяцію особин.

Крок 3. Виконати тестування особин початкової популяції шляхом розв'язання поставленої задачі (на вхід моделі подається задана вибірка даних) і оцінювання отриманих результатів на основі обраної фітнес-функції.

Крок 4. Відібрати рішення з кращими значеннями фітнес-функції.

Крок 5. На основі випадкового застосування оператора мутації до особин-батьків згенерувати нащадків, які перейдуть в нову популяцію.

Крок 6. Виконати тестування нащадків шляхом розв'язання поставленої задачі і оцінювання отриманих результатів.

Крок 7. Відібрати найбільш перспективних нащадків.

Крок 8. Перевірити умови закінчення процесу еволюції, якими можуть бути: досягнення оптимального значення фітнес-функції або досягнення граничних значень, що обмежують тривалість процесу. Якщо умови закінчення пошуку задовільнено, тоді виконати перехід до кроку 9, в іншому випадку виконати повернення до кроку 5, де особини останньої згенерованої популяції, виступають в якості батьків.

Крок 9. Зупинення.

5.3.3 Оператор мутації

У класичному еволюційному програмуванні мутація ґрунтується на випадкових числах з гаусовим розподілом. Нащадок генерується шляхом додавання випадкового числа до батька. Однією з найважливіших особливостей еволюційного програмування є використання параметрів, що самоадаптуються. Як такі параметри може використовуватися дисперсія гаусового розподілу. При цьому дисперсія подається як частина особини і є предметом еволюції.

В еволюційному програмуванні при використанні як особин скінчених автоматів застосовуються такі способи реалізації оператора мутації:

– зміна завершального стану;

- зміна умови переходу з одного стану в інший;
- додавання нового стану;
- видалення стану;
- зміна початкового стану.

5.3.4 Метод групового врахування аргументів

Різновидом еволюційного програмування є метод групового врахування аргументів [66, 107], що дозволяє ефективно оцінювати динамічні характеристики досліджуваного процесу шляхом послідовних уточнень результатів статистичної підгонки спостережень.

Основна ідея методу групового врахування аргументів полягає в тому, що на кожній ітерації повний опис об'єкту $y = f(x_1, x_2, \dots, x_N)$ замінюється декількома рядами часткових описів.

Крок 1. Обчислити множину часткових наближень до досліджуваного процесу в виду:

$$y_k^{(1)} = g(x_{k_1}, x_{k_2}),$$

де $y_k^{(1)}$ – k -те часткове наближення процесу в першому ряді (на першій ітерації); x_{k_1}, x_{k_2} – пара аргументів, для якої будується наближення; $k_1 = 1, 2, \dots, (N-1)$; $k_2 = (k_1 + 1), (k_1 + 2), \dots, N$; $k = 1, 2, \dots, K_1$; $K_1 = C_N^2 = N(N-1)/2$ – загальна кількість функцій $y_k^{(1)}$, що синтезуються.

З K_1 синтезованих функцій виду $y_k^{(1)}$ обирається F_1 найкращих за певним критерієм.

Крок 2. Обчислити множину часткових наближень до досліджуваного процесу в виду:

$$y_k^{(2)} = g(y_{k_1}^{(1)}, y_{k_2}^{(1)}),$$

де $y_k^{(2)}$ – k -е часткове наближення процесу в другому ряді; $y_{k_1}^{(1)}, y_{k_2}^{(1)}$ – пара аргументів, для якої будується наближення; $k_1 = 1, 2, \dots, (F_1 - 1)$; $k_2 = (k_1 + 1), (k_1 + 2), \dots, F_1$; $k = 1, 2, \dots, K_2$; $K_2 = C_{F_1}^2 = F_1(F_1 - 1)/2$ – загальна кількість функцій $y_k^{(2)}$, що синтезуються.

З K_2 синтезованих функцій виду $y_k^{(2)}$ обирається F_2 найкращих за тим же критерієм, що і в першому ряді.

.....
Крок r . Обчислити множину часткових наближень до досліджуваного процесу в виду:

$$y_k^{(r)} = g(y_{k_1}^{(r-1)}, y_{k_2}^{(r-1)}),$$

де $y_k^{(r)}$ – k -е часткове наближення процесу в r -му ряді; $y_{k_1}^{(r-1)}, y_{k_2}^{(r-1)}$ – пара аргументів, для якої будується наближення; $k_1 = 1, 2, \dots, (F_{r-1} - 1)$; $k_2 = (k_1 + 1), (k_1 + 2), \dots, F_{r-1}$; $k = 1, 2, \dots, K_r$; $K_r = C_{F_{r-1}}^2 = F_{r-1}(F_{r-1} - 1) / 2$ – загальна кількість функцій $y_k^{(r)}$, що синтезуються.

З K_r синтезованих функцій виду $y_k^{(r)}$ обирається F_r найкращих за тим же критерієм, що і в попередньому ряді.

Різні методи групового врахування аргументів відрізняються один від одного виглядом опорної функції g . Як правило, в методах групового врахування аргументів в якості опорної функції g використовуються квадратичні поліноми. Проте як базисні моделі можуть використовуватися не тільки поліноми, але й нелінійні, імовірнісні функції.

Як критерії, за якими можуть відбиратися функції виду $y_k^{(r)}$ для переходу на наступну ітерацію можуть виступати:

– критерій регулярності (середньоквадратична помилка, коефіцієнт кореляції);

– критерій незміщеності (навчальна вибірка розбивається на дві частини: у першу частину потрапляють екземпляри з парними рангами, в другу – з непарними, ранг визначається як номер у відсортованій послідовності значень вихідних параметрів).

Як критерій зупинення вибирається досягнення заданого значення середньоквадратичної помилки.

Метод групового врахування аргументів застосовується у різних областях для аналізу даних і відшукування знань, прогнозування й моделювання систем, оптимізації й розпізнавання образів, оскільки він дозволяє автоматично знаходити взаємозалежності в даних, вибрати оптимальну структуру моделі або мережі, і збільшувати точність існуючих методів.

Цей підхід самоорганізації моделей принципово відрізняється від звичайно використовуваних дедуктивних методів, оскільки він заснований на індуктивних принципах – знаходження кращого рішення засновано на переборі всіляких варіантів.

За допомогою перебору різних рішень підхід індуктивного моделювання намагається мінімізувати роль передумов автора про результати моделювання. Комп'ютер сам знаходить структуру моделі й закони, що діють в об'єкті. Він може бути використаний при створенні штучного інтелекту як порадник для вирішення конфліктів й при прийнятті рішень.

Таким чином, особливостями методу групового врахування аргументів є:

– знаходження оптимальної складності структури моделі, адекватної рівню завад у вибірці даних. Для вирішення реальних проблем із зашумленими або короткими даними, спрощені прогнозуючі моделі виявляються більш точними;

– при синтезі нейромережних моделей кількість шарів і нейронів у схованих шарах, структура моделі й інші оптимальні параметри нейромереж знаходяться автоматично;

– гарантується знаходження найбільш точної або незміщеної моделі, оскільки метод не пропускає найкращого рішення під час перебору всіх варіантів (у заданому класі функцій);

– будь-які нелінійні функції або ознаки, які можуть мати вплив на вихідну змінну, використовуються як вхідні аргументи;

– метод дозволяє автоматично знаходити в даних взаємозв'язки, що легко інтерпретуються, і вибирає ефективні вхідні змінні;

– переборні алгоритми, що відносяться до методу групового врахування аргументів, досить просто можуть бути програмно реалізовані;

– метод використовує інформацію безпосередньо з вибірки даних і мінімізує вплив апріорних припущень автора про результати моделювання;

– дає можливість відшукання незміщеної фізичної моделі об'єкту (закоону або кластеризації) – однієї й тієї ж для всіх майбутніх вибірок.

РОЗДІЛ 6

КЕРУВАННЯ ПАРАМЕТРАМИ ЕВОЛЮЦІЙНОГО ПОШУКУ

Як наголошувалося раніше, одним з недоліків еволюційних методів є сильна залежність ефективності їх функціонування від параметрів пошуку (розмір популяції, імовірнісні характеристики еволюційних операторів, параметри кодування незалежних змінних і т. ін.).

Проте, не існує оптимальних значень параметрів еволюційного пошуку, що дозволяють ефективно вирішувати всі можливі задачі. Таким чином, зі всіх можливих значень параметрів необхідно вибрати оптимальні з урахуванням особливостей вирішуваної задачі і вхідних даних.

Тому сильна чутливість еволюційних методів до значень параметрів пошуку спричинила необхідність розробки різних методів керування параметрами [8, 11, 14, 15] еволюційного пошуку (рис. 6.1).



Рисунок 6.1 – Методи керування параметрами еволюційного пошуку

6.1 Неадаптивні методи керування параметрами

Такі методи не використовують механізми адаптації параметрів еволюційного пошуку до особливостей цільової функції і не передбачають зміни значень параметрів в процесі пошуку в залежності від різноманітності хромосом в популяції.

6.1.1 Використання параметрів-констант

При використанні такого підходу параметри еволюційного пошуку встановлюються на етапі ініціалізації і є постійними впродовж всього процесу пошуку. Як правило, вибирають значення ймовірності мутації: $1/L$ або $0,001 - 0,01$, ймовірність схрещування: $0,7 - 1,0$, розмір популяції – більше 50 хромосом.

Для підвищення ефективності еволюційного пошуку при розв'язанні конкретних оптимізаційних задач застосовуються методи *настроювання параметрів*. При використанні таких методів підбирається комбінація параметрів еволюційного пошуку, яка була б оптимальною для досліджуваного класу задач оптимізації. При цьому відбувається пошук оптимальних значень параметрів до початку роботи еволюційного методу, після чого одержана комбінація параметрів застосовується без зміни впродовж всього оптимізаційного процесу. Як правило, для настроювання параметрів еволюційного пошуку експериментально тестуються різні варіанти комбінації значень параметрів, потім вибирається комбінація параметрів, що забезпечує найкращі результати оптимізації.

Недоліками такого підходу є такі:

- складність повторного використання настроєних параметрів при розв'язанні задач, що частково або повністю відрізняються від тих, які були використані при настроюванні параметрів;
- різна ефективність одержаного набору параметрів на різних ітераціях еволюційного пошуку.

6.1.2 Детерміноване керування параметрами

До детермінованих методів відносяться методи, в яких застосовується стратегія зміни параметра відповідно до певного закону без зворотного зв'язку з оптимізаційним процесом.

При детермінованому керуванні параметрами спочатку задається функція зміни параметрів еволюційного пошуку, яка, як правило, залежить від номера ітерації. Наприклад, ймовірність мутації хромосоми на t -ій ітерації може бути розрахована за однією з формул:

$$p_m(t) = \alpha \cdot \frac{e^{-\beta t}}{N\sqrt{L}} \quad \text{або} \quad p_m(t) = \begin{cases} \left(2 + \frac{L-2}{t_p} \cdot t\right)^{-1}, & \text{якщо } 0 \leq t \leq t_p; \\ 1/L, & \text{якщо } t_p < t, \end{cases}$$

де t_p – порогове значення номера ітерації, після якого ймовірність мутації функціонально змінюється в процесі пошуку.

Зміна параметрів еволюційного пошуку в залежності від номера ітерації також може бути виконана за допомогою застосування операторів нерівномірної, випадкової або Гаусової мутації, розглянутих раніше.

Істотним недоліком такого підходу є те, що функція зміни параметрів еволюційного пошуку повністю залежить від номера ітерації і не залежить від інших характеристик процесу пошуку (досягнуте значення фітнес-функції, різноманітність в популяції і т. п.).

6.2 Адаптивні методи

Адаптивні методи управління змінюють параметри еволюційного пошуку в залежності від результатів, досягнутих на попередніх ітераціях.

6.2.1 Оцінювальні методи

Оцінювальні методи (measure-based methods) – методи, що змінюють параметри еволюційного пошуку за допомогою деяких заданих наперед правил, наприклад, у вигляді «якщо – то». Основні умови зміни параметрів пошуку використовують такі показники, що вимірюються в процесі пошуку:

а) різноманітність хромосом, що розраховується як середня відстань Хеммінга в популяції. При цьому середня відстань може бути розрахована за допомогою одного з таких способів:

- відстань від кожного індивіда до всіх інших;
- відстань від кожного індивіда до середнього значення хромосоми в популяції;
- відстань від кожного індивіда до кращої хромосоми в популяції;
- відстань від кожного індивіда до найближчого його сусіда в популяції;

б) різноманітність значень фітнес-функції;

в) різниця між кращим (гіршим) значенням фітнес-функції і середнім значенням.

6.2.2 Пристосовувальні методи

При використанні пристосовувальних методів хромосома окрім основних змінних, які настроюються в задачі, містить також параметри еволюційного пошуку.

Пристосовувальні методи часто застосовуються при використанні Гаусової мутації в еволюційних стратегіях. При цьому хромосома містить пару значень (w , σ). Можливі три випадки:

– хромосома подається у вигляді $(w_1, w_2, \dots, w_L, \sigma)$. В такому разі мутація відбувається таким чином:

$$\sigma_{t+1} = \sigma_t \exp(\tau_0 N(0,1)),$$

$$w_{j,t+1} = w_{j,t} + \sigma_{t+1} N_j(0,1),$$

де $w_{j,t}$ та $w_{j,t+1}$ – значення j -го гена на t -ій і $(t+1)$ -ій ітераціях, відповідно; σ_{t+1} й σ_t – коефіцієнти, що визначають розмах зміни значень генів при використанні мутації на t -ій і $(t+1)$ -ій ітераціях, відповідно; τ_0 – параметр, що керує рівнем адаптації;

– хромосома подається у вигляді $(w_1, \sigma_1, w_2, \sigma_2, \dots, w_L, \sigma_L)$. При цьому для мутації використовується формула:

$$\sigma_{j,t+1} = \sigma_{j,t} \exp(\tau N(0,1) + \tau_j N_j(0,1)),$$

$$w_{j,t+1} = w_{j,t} + \sigma_{t+1} N_j(0,1),$$

де τ_j – параметр, що керує рівнем адаптації для j -ої змінної залежно від $N_j(0,1)$; τ – параметр, що керує рівнем адаптації для всієї популяції;

– хромосома містить трійку значень (w, α, σ) , де w та σ – вектори, аналогічні використовуваним в попередньому підході; α – матриця кутів обертання, що відповідають кореляції між j -ою та k -ою змінними w_j і w_k . При використанні такого підходу мутація виконується за формулами:

$$\sigma_{j,t+1} = \sigma_{j,t} \exp(\tau N(0,1) + \tau_j N_j(0,1)),$$

$$\alpha_{jk,t+1} = \beta N_{jk}(0,1),$$

$$w_{t+1} = w_t + N(0, C(\sigma_{t+1}, \alpha_{t+1})),$$

де α_{jk} – кути обертання, що відповідають кореляції між j -ою та k -ою змінними w_j та w_k ; β – параметр, що керує рівнем адаптації кутів обертання ($\beta = 0,0873 \approx 5^0$); $N(0, C(\sigma_{t+1}, \alpha_{t+1}))$ – кореляційний вектор мутації.

6.2.3 Структурно-орієнтовані методи

Структурно-орієнтовані методи (population-structure-based methods) керують параметрами еволюційного пошуку в залежності від структури популяції.

6.2.3.1 Розбиття популяції

Метод розбиття популяції на декілька окремих підпопуляцій передбачає настроювання параметрів всіх підпопуляцій, ґрунтуючись на значеннях параметрів підпопуляції, в якій на поточній ітерації досягнуто кращих результатів.

Крок 1. Виконати ініціалізацію методу.

Крок 1.1. Сформувати таблицю основних параметрів пошуку. При цьому в таблицю записується k записів, кожному з яких відповідають певні значення параметрів пошуку.

Крок 1.2. Згенерувати k підпопуляцій.

Крок 1.3. Встановити лічильник ітерацій (часу): $t = 1$.

Крок 2. Запустити еволюційний пошук для кожної підпопуляції.

Крок 2.1. Оцінити нові хромосоми в кожній підпопуляції.

Крок 2.2. Виконати оператори відбору, схрещування і мутації.

Крок 2.3. Виконати: $t = t + 1$.

Крок 2.4. Якщо відношення t/t_a – ціле (де t_a – період адаптації, тобто число ітерацій, після яких відбувається перенастроювання параметрів еволюційного пошуку для кожної підпопуляції), тоді виконати кроки 2.4.1–2.4.3.

Крок 2.4.1. Вибрати кращу (за значенням фітнес-функції) хромосому в кожній підпопуляції.

Крок 2.4.2. Вибрати кращу хромосому з числа одержаних на кроці 2.4.1 хромосом.

Крок 2.4.3. Перенастроїти параметри еволюційного пошуку для кожної підпопуляції залежно від параметрів підпопуляції, в якій на поточній ітерації досягнуто кращих результатів (одержано кращу хромосому). При цьому значення параметрів пошуку збільшуються (зменшуються) для підпопуляцій, у яких значення параметра, що настраюється, було менше (більше), ніж в кращій підпопуляції. Зміна параметрів пошуку відбувається шляхом вибору наступного (попереднього) значення параметра, що настраюється, з таблиці параметрів, згенерованої на кроці 1.1.

Крок 3. Виконати перевірку критеріїв закінчення пошуку. В разі незадоволення таких критеріїв, виконати перехід до кроку 2.

Крок 4. Зупинення.

6.2.3.2 Метод просторової популяційної структури

Метод просторової популяційної структури (spatial control) відображає місце розташування кожної хромосоми на відповідну множину значень параметрів пошуку (рис. 6.2). Такий підхід застосовується в так званому територіальному генетичному методі (terrain-Based Genetic Algorithm), при використанні якого формується сітка параметрів еволюційного пошуку. При цьому кожній чарунці згенерованої сітки, відповідають певні значення параметрів еволюційного пошуку (наприклад, ймовірність мутації і кількість точок схрещування). Кількість чарунок в сітці дорівнює кількості хромосом в популяції, оскільки кожній хромосомі відповідає певна чарунка.

Керування параметрами еволюційного пошуку за допомогою методу просторової популяційної структури реалізовано також в територіальній мозаїчній моделі (terrain-based patchwork model) еволюційного пошуку.

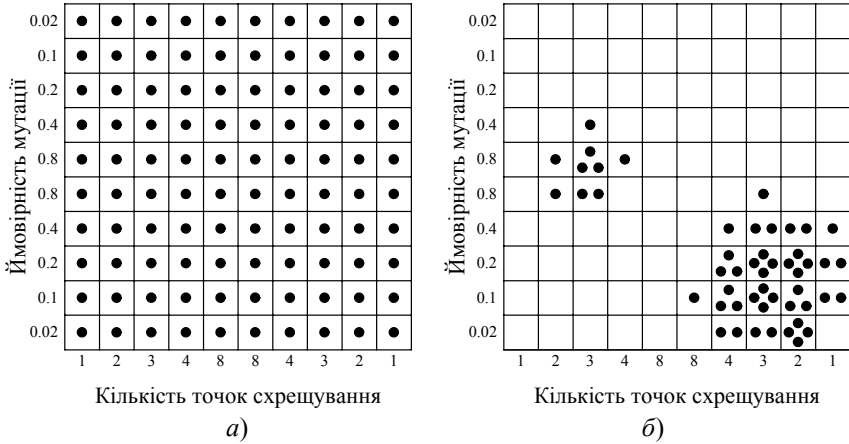


Рисунок 6.2 – Керування параметрами за допомогою методів просторової популяційної структури:

- a)* територіальний генетичний метод (the TBGA);
б) територіальна мозаїчна модель (the TBPM)

Така модель аналогічна моделі штучного життя (artificial life). У ній хромосоми подаються у вигляді автономних агентів, які виконують певні дії (переміщення, схрещування і т. ін.) в залежності від мотиваційної мережі, на вході якої подається стан навколишнього середовища і внутрішній стан агента.

РОЗДІЛ 7

ПОЛІМОДАЛЬНА ЕВОЛЮЦІЙНА ОПТИМІЗАЦІЯ

Більшість реальних задач є полімодальними, тобто цільова функція таких задач містить множину оптимумів, розташованих в різних точках простору пошуку. Таким чином, для розв'язання полімодальних задач необхідно передбачити засоби, що не допускають зациклення в локальному оптимумі.

Класичні методи еволюційного пошуку мають такий недолік, як передчасна збіжність до локального оптимуму. З метою усунення передчасної збіжності для розв'язання задач полімодальної оптимізації розроблено методи полімодального еволюційного пошуку (рис. 7.1).

Методи полімодального еволюційного пошуку [13, 42, 104] переслідують дві мети:

- пошук глобального оптимуму, уникаючи попадання в локальні;
- пошук декількох оптимумів (або субоптимальних рішень), розташованих в різних точках простору пошуку. Пошук декількох альтернативних субоптимальних рішень актуальний у випадках, коли для прийняття остаточного висновку про можливість використання знайдених рішень на практиці необхідно виконати їх експертну оцінку, ґрунтуючись на додаткових критеріях, що не враховуються у фітнес-функції при еволюційному пошуку.

7.1 Методи ухилення від передчасної збіжності

Методи ухилення (avoid strategies) запобігають передчасній збіжності (premature convergence) до локального оптимуму. Такі методи зменшують генетичну збіжність з метою забезпечення різноманітності популяції і недопущення зациклення в локальному оптимумі.

7.1.1 Уповільнення генетичної збіжності

7.1.1.1 Методи заміщення

При використанні методів заміщення (replacement methods) кожна нова хромосома перевіряється на те, чи повинна вона замінити іншу хромосому в популяції.

Для заміщення старих особин новими використовується процедура ущільнення (crowding), яка на кожній ітерації еволюційного пошуку виконується в такій послідовності кроків.

Крок 1. Згенерувати множину нащадків G (generation gap).

Крок 2. Встановити лічильник перевірених нащадків: $g = 1$.

Крок 3. Для хромосоми-нащадка G_g вибрати невелику кількість (1–4) (crowding factor) хромосом C з поточної популяції.



Рисунок 7.1 – Класифікація методів полімодального еволюційного пошуку

Крок 4. Знайти хромосому S – найбільш схожу на G_g хромосому з множини C . При цьому схожість хромосом може бути визначена як відстань Хеммінга між ними.

Крок 5. У випадку, якщо значення фітнес-функції хромосоми G_g краще, ніж у знайденої на попередньому кроці хромосоми S , тоді замінити хромосому S на G_g .

Крок 6. Виконати: $g = g + 1$.

Крок 7. Якщо $g > |G|$, тоді виконати перехід до кроку 8, інакше – перейти до кроку 3.

Крок 8. Зупинення.

Детерміноване ущільнення (deterministic crowding) передбачає розрахунок відстані між батьками і нащадками. Хромосома-нащадок заміщує найбільш близьку до неї хромосому з поточного покоління лише у випадку, якщо нащадок має краще значення фітнес-функції.

Крок 1. Розрахувати відстань Хеммінга між парою батьків і їх нащадками.

Крок 2. Створити пари, що складаються з батька і найближчого до нього нащадка.

Крок 3. Порівняти фітнес-функції в двох одержаних парах.

Крок 4. Додати в наступну популяцію по одній особині з пари з кращим значенням фітнес-функції.

Крок 5. Якщо всі нащадки проаналізовані, тоді виконати перехід до кроку 6. Інакше – перехід до кроку 1.

Крок 6. Зупинення.

Імовірнісне ущільнення (probabilistic crowding) використовує ймовірність заміщення батька H_j нащадком H_j^t , яка розраховується за формулою:

$$p(H_j^t, H_j) = \frac{f(H_j^t)}{f(H_j^t) + f(H_j)}$$

7.1.1.2 Просторові методи

До просторових методів (spatial population topology) відносять: клітинні еволюційні методи, мозаїчну модель та релігійні еволюційні методи.

7.1.1.2.1 Клітинні еволюційні методи

Клітинні еволюційні методи (cellular evolutionary algorithms) або дифузійна модель (diffusion model) використовують просторову сітку, в якій кожна чарунка містить одну хромосому. При цьому хромосома може взаємодіяти виключно з найближчими сусідами, що дозволяє значно уповільнити процес передчасної збіжності і заикнення в локальному оптимумі.

7.1.1.2.2 Мозаїчна модель

Мозаїчна модель (patchwork model) є гібридом еволюційних методів, що поєднують в собі ідеї дифузійної моделі, мультиагентних систем і оптимальної моделі еволюційного пошуку. Мозаїчна модель складається з сіткового простору (grid world) і певної кількості взаємодіючих агентів.

На кожній ітерації агент може виконувати деякі дії (переміщення, схрещування, прийняття їжі і т. ін.), що визначаються мотиваційною мережею (motivation network). Процес прийняття рішень заснований на вхідній інформації, що одержується від сенсорів, і її інтерпретації за допомогою мотиваційної мережі (рис. 7.2). Сенсори визначають різну інформацію в нормованому вигляді (від 0 до 1), таку як краще значення фітнес-функції в популяції, щільність популяції (зовнішні сенсори), власне значення фітнес-функції агента (внутрішні сенсори).

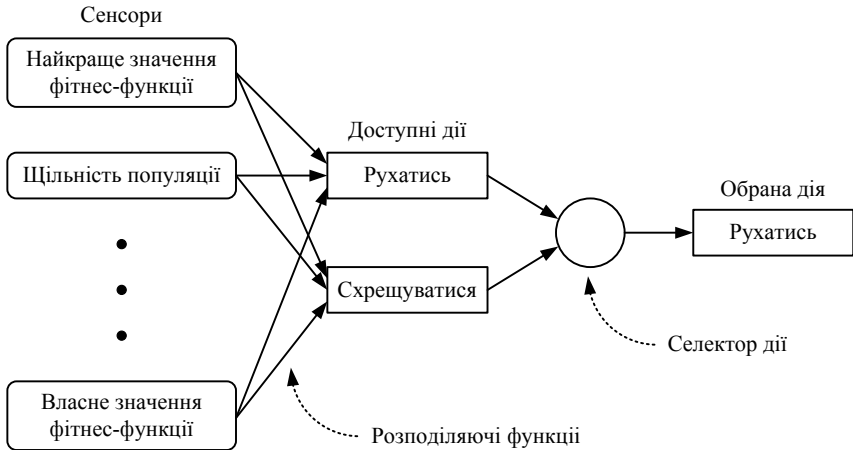


Рисунок 7.2 – Мотиваційна мережа

Для кожної можливої i -ої дії мотиваційна мережа містить набір функцій (mapping functions), які перетворюють значення сенсорів в значення рівня мотивації M_i для кожної дії.

$$M_i = \sum_{j=1}^S w_{ij} \text{map}_{ij}(s_j),$$

де w_{ij} – вага для j -го сенсора і i -ої дії, $w_{ij} \in [0; 1]$; map_{ij} – функція перетворення значення s_j , одержаного з j -го сенсора для i -ої дії; S – кількість сенсорів.

7.1.1.2.3 Релігійні еволюційні методи

Релігійні еволюційні методи (religion-based evolutionary algorithms) розширюють дифузійну і мозаїчну моделі шляхом додавання концепції релігійності (підпопуляцій) і порядку залучення (conversion scheme) особин в свої підпопуляції. Агенти, використовувані в такому методі, можуть виконувати три типи дій:

- перехід в порожню сусідню чарунку;
- залучення сусідніх особин в свою підпопуляцію (засновано на імовірнісному підході і можливо лише у випадку, якщо поточна особина має краще значення фітнес-функції в порівнянні з особоною, що залучається);
- генерація нащадка шляхом схрещування з сусідом, що належить до тієї ж підпопуляції.

7.1.2 Запобігання появі співпадаючих рішень

До методів, що запобігають появі співпадаючих рішень (prevent overlap of solutions), відносять:

- методи відбору;
- методи розбиття простору пошуку.

7.1.2.1 Методи відбору

Методи відбору (selection-based approaches) використовують спеціальні фітнес-функції або модифіковані оператори відбору для запобігання передчасної збіжності шляхом відбору хромосом, що підвищують різноманітність в популяції.

7.1.2.1.1 Методи розділення

Методи розділення (sharing) використовують ідею зміни фітнес-функції (raw fitness) для збільшення тиску відбору (selection pressure) дуже схожих особин, таким чином підвищуючи генетичну різноманітність.

Один з таких методів передбачає погіршення значення фітнес-функції хромосом, близько розташованих одна до одної. Основна мета такого підходу полягає в запобіганні збіжності до єдиного оптимуму. Нове значення фітнес-функції (shared fitness) розраховується за формулою:

$$f_{new}(H_j) = \frac{f(H_j)}{\sum_{k=1}^N sh(j,k)},$$

де $sh(j, k)$ – функція розділення, для розрахунку якої використовується вираз:

$$sh(j, k) = \begin{cases} 1 - \left(\frac{d(H_j; H_k)}{\sigma_{sh}} \right)^{\alpha_{sh}}, & \text{якщо } d(H_j; H_k) < \sigma_{sh}; \\ 0, & \text{якщо } d(H_j; H_k) \geq \sigma_{sh}, \end{cases}$$

де $d(H_j, H_k)$ – відстань між хромосомами H_j та H_k ; σ_{sh} – порогова відстань між хромосомами (якщо $d(H_j, H_k) < \sigma_{sh}$, то вважається, що хромосоми розташовані близько одна до одної); α_{sh} – коефіцієнт, що визначає ступінь погіршення значення фітнес-функції, як правило, $\alpha_{sh} = 1$.

7.1.2.1.2 Методи керування різноманітністю в популяції

Методи керування різноманітністю в популяції (diversity control-oriented evolutionary algorithms) використовують нетрадиційний механізм відбору (CPSS – cross-generational probabilistic survival selection), заснований не лише на значенні фітнес-функції хромосоми, але і на відмінності кожної хромосоми популяції від кращої хромосоми на поточній ітерації.

Крок 1. Розбити поточну популяцію на пари хромосом.

Крок 2. Одержати по одному новому нащадку для кожної пари шляхом застосування операторів схрещування та/або мутації.

Крок 3. Об'єднати нові і старі хромосоми в проміжну популяцію.

Крок 4. Відсортувати хромосоми проміжної популяції за значенням фітнес-функції.

Крок 5. Видалити з проміжної популяції однакові хромосоми.

Крок 6. Перемістити визначену користувачем на етапі ініціалізації кількість k_c елітних особин в нову популяцію.

Крок 7. Послідовно, починаючи з хромосоми з кращим значенням фітнес-функції, відібрати для переміщення $(N - k_c)$ хромосом в нову популяцію. При цьому ймовірність відбору p_s розраховується за формулою:

$$p_s = \left(\frac{(1 - c) \cdot d_H(H_j, H_{opt})}{L} + c \right)^\alpha,$$

де H_{opt} – значення найкращої на поточній ітерації хромосоми; $d_H(H_j, H_{opt})$ – відстань Хеммінга між хромосомами H_j та H_{opt} ; c та α – коефіцієнти, що визначають ступінь залежності ймовірності відбору хромосоми H_j від її близькості до оптимальної хромосоми H_{opt} .

При цьому на кроці 7 можливо відбирати не строго $(N - k_c)$ хромосом, а лише ті хромосоми, значення ймовірності відбору p_s яких вище за порогове, внаслідок чого кількість хромосом на кожній ітерації буде різною.

7.1.2.2 Методи розбиття простору пошуку

Методи розбиття простору пошуку (search space division) використовують множину підпопуляцій та прагнуть досягти мінімального перекриття простору пошуку різними підпопуляціями. Такі методи перетворюють задачу на декілька простіших задач, що розв'язуються одночасно за допомогою використання різних підпопуляцій. В ідеальному випадку кожна підпопуляція покриває свою область простору пошуку, що відповідає певному оптимуму у вирішуваній задачі.

7.1.2.2.1 Еволюційні методи, що розгалужуються

Еволюційні методи, що розгалужуються (forking evolutionary algorithms), призначені для пошуку єдиного глобального оптимуму, уникаючи попадання в локальний оптимум. Популяція складається з батьківської популяції і змінної кількості популяцій-нащадків. При цьому популяції-нащадки формуються у випадках, якщо спостерігається надмірність батьківської популяції (зменшується різноманітність в популяції). Генерація популяції-нащадка проводиться так, щоб не допустити перетину (перекриття) батьківської популяції і популяції-нащадка. Для оцінки різноманітності популяції використовуються метрики у вигляді відстані Хеммінга (генетична різноманітність) і Евклідової відстані (фенотипна відстань).

Таким чином, головна (батьківська) популяція спеціалізується на пошуку нових областей, в яких можуть бути розташовані оптимуми, а популяції-нащадки призначені для пошуку оптимуму в таких областях.

7.1.2.2.2 Методи рівноваги, що зміщується

Методи рівноваги, що зміщується (shifting balance evolutionary algorithms), використовують аналогічний попередньому методу підхід. При цьому створюється головна популяція (core population) і деяка кількість колоній (colony populations). Головна популяція шукає глобальний оптимум, а колонії призначені для пошуку в областях простору пошуку, що не покриваються особинами головної популяції.

Кращі хромосоми колоній переходять в головну популяцію через певне число ітерацій. Таким чином, головна популяція містить кращі рішення, розташовані в різних точках простору пошуку, а колонії призначені для пошуку нових потенційних рішень (точок оптимуму).

Використовується спеціальний оператор відбору, що віддаляє колонії від головної популяції у випадку, якщо вони починають перетинатися. При цьому в кожній колонії при формуванні нового покоління відбира-

ється певна кількість хромосом в залежності від ступеня їх близькості до особин головної популяції.

Ймовірність відбору хромосом в колонії P для переходу в наступне покоління залежить від:

– величини різноманітності (diversity) хромосом в колонії:

$$\text{diversity}(P) = \frac{1}{L \cdot N(N-1)} \sum_{j=1}^N \sum_{k=1}^N d_H(H_j, H_k);$$

– коефіцієнта C , що визначає рівень близькості колонії A до головної популяції M :

$$C(A, M) = \frac{1}{N_A} \sum_{j=1}^{N_A} Wdist(H_j, M) = \frac{1}{N_A N_M} \sum_{j=1}^{N_A} \sum_{k=1}^{N_M} \delta(H_j, H_k),$$

де N_A – розмір популяції колонії A ; N_M – розмір головної популяції M ; $Wdist(H_j, M)$ – функція, що визначає кількість хромосом в головній популяції M , які розташовані ближче до M в порівнянні з H_j ; $\delta(H_j, H_k)$ – функція, яка визначає, чи ближче до M розташована хромосома H_j з колонії A , ніж хромосома H_k з головної популяції M :

$$\delta(H_j, H_k) = \begin{cases} 1, & \text{якщо } dist(H_j, M) < dist(H_k, M); \\ 0, & \text{якщо } dist(H_j, M) \geq dist(H_k, M), \end{cases}$$

де $dist(H_j, M)$ – функція, що визначає відстань від хромосоми H_j до головної популяції M :

$$dist(H_j, M) = \frac{1}{L \cdot N_M} \sum_{k=1}^{N_M} d(H_j, H_k),$$

де $d_H(H_j, H_k)$ – відстань Хеммінга між хромосомами H_j і H_k .

7.1.2.2.3 Багатонаціональні еволюційні методи

Багатонаціональні еволюційні методи (multinational evolutionary algorithms) об'єднують в собі ідеї самоприспособлення, адаптації до задачі, розбиття простору пошуку, використання підпопуляцій (рис. 7.3).

При використанні багатонаціональних еволюційних методів відбувається автоматичне розбиття популяції на декілька націй (підпопуляцій), кожна з яких відповідає певному локальному оптимуму. Нація характеризується елементами:

- популяція особин;
- уряд – набір кращих за значенням фітнес-функції особин нації;
- показник політики (policy), що є точкою в просторі пошуку, яка характеризує оптимум в нації. Даний показник розраховується на основі представників уряду.

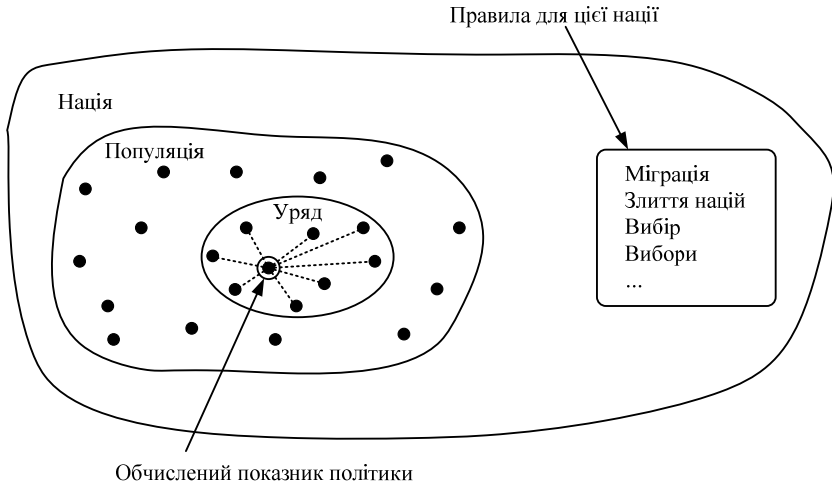


Рисунок 7.3 – Елементи нації
в багатонаціональному еволюційному методі

7.2 Методи відновлення

Методи відновлення (repair strategy) засновані на відновленні різноманітності в популяції і включають такі групи методів:

- методи множинного заміщення;
- методи перезапущу і фазові методи.

7.2.1 Методи множинного заміщення

У методах множинного заміщення (mass extinction) різноманітність в популяції досягається шляхом заміщення частини популяції. Такі методи відрізняються між собою способом відбору хромосом для заміщення, кількістю хромосом для заміщення і способом генерації нових рішень.

7.2.1.1 Методи випадкової міграції

Методи випадкової міграції (random immigrants evolutionary algorithms) використовують просту схему заміщення, що передбачає заміщення в кожній популяції певної частини (5–10% від загальної кількості хромосом в популяції) хромосом і створення нових особин шляхом застосування генератора випадкових чисел.

7.2.1.2 Еволюційне програмування із заміщенням особин

Еволюційне програмування із заміщенням особин (extinction evolutionary programming) передбачає генерацію на кожній t -ій ітерації стрес-фактору: $\eta(t) = U(0; 0,96)$, що визначає ступінь оновлення популяції. Після цього значення фітнес-функції хромосом відображаються на інтервал $[\alpha; 1]$ за формулою (при вирішенні задачі мінімізації):

$$f'(H_j) = \alpha + (1 - \alpha) \frac{f(H_j) - f(H_{\max})}{f(H_{\min}) - f(H_{\max})},$$

де α – параметр, що настроюється.

Заміщуються хромосоми, для яких виконується нерівність:

$$f(H_j) < \eta(t).$$

Нові хромосоми генеруються шляхом мутації особин, що вижили. У випадку, якщо наведена вище нерівність виконується для всіх хромосом, тоді генерація нових рішень відбувається шляхом мутації гіршої хромосоми і декількох інших випадково вибраних хромосом.

7.2.1.3 Критичні еволюційні методи, що пристосовуються

Критичні еволюційні методи, що пристосовуються (self-organized criticality evolutionary algorithms) керують рівнем заміщення особин на кожній ітерації. Для керування рівнем заміщення хромосом використовується один з адаптивних методів керування параметрами еволюційного пошуку.

Нові хромосоми генеруються за допомогою мутації хромосом, що вижили.

7.2.2 Методи перезапуску і фазові методи

Основна ідея методів перезапуску і фазових методів (restart and phase-based techniques) полягає у визначенні передчасної збіжності, після чого запускаються процедури, які підвищують різноманітність в популяції. У таких методах оптимізація відбувається за певну кількість фаз, що послідовно виконуються. При цьому використовується принцип елітизму.

7.2.2.1 СНС-метод

СНС-метод (Cross-population selection, Heterogeneous recombination and Cataclysmic mutation – Перехресний відбір, Гетерогенне схрещування, Катастрофічна мутація), запропонований Л. Ешелменом, досить швидко сходиться через те, що в ньому немає традиційних мутацій, використовуються популяції невеликого розміру, і відбір особин в наступне покоління проводиться як між батьківськими особинами, так і між їх нащадками.

Такий метод зменшує генетичну збіжність шляхом застосування модифікованого оператора однорідного схрещування, що називається половинним однорідним схрещуванням або HUX-схрещування (half uniform crossover). Даний оператор генерує двох нащадків шляхом обміну половиною різних бітів хромосом-батьків (non-matching), внаслідок чого генерується два нащадки з максимальною відстанню Хеммінга між батьками і нащадками.

Далі застосовується процедура, що обмежує кількість нащадків, згенерованих схожими індивідуумами (incest prevention). В результаті застосування такої процедури батьківську пару можуть утворювати лише ті хромосоми, відстань Хеммінга між якими перевищує порогове значення. Як правило, порогове значення відстані Хеммінга встановлюється $L/4$, де L – довжина хромосоми. Поріг зменшується у випадку, якщо жоден нащадок не може бути сформований внаслідок незадовільної відстані між батьками.

Крім того, для зменшення генетичної збіжності використовується стратегія перезапуску, коли виявляється, що в популяції відсутня різноманітність. Перезапуск відбувається у випадках, якщо порогове значення відстані Хеммінга зменшилося до нуля, а нові нащадки не можуть бути одержані внаслідок одноманітності хромосом в популяції. Після перезапуску в нову фазу еволюційного пошуку потрапляє краща хромосома, одержана на попередній фазі, а інші особини початкової популяції нової фази формуються шляхом сильної мутації (мутує близько 35 % бітів) особин, одержаних на попередній фазі.

При цьому відбір і схрещування використовуються в процесі еволюційного пошуку, а мутація застосовується лише при перезапуску методу.

Таким чином, СНС-метод має такі властивості:

- фіксований розмір популяції;
- фіксована розрядність генів;
- перезапуск методу після знаходження рішення;
- невелика популяція;
- особини для схрещування розбиваються на пари і схрещуються за умови суттєвих відмінностей;
- відбір в наступне покоління проводиться між батьківськими особинами і нащадками;
- використовується половинне однорідне схрещування (HUX);
- макромутація при перезапуску.

7.2.2.2 Еволюційний метод з керуванням різноманітністю в популяції

Еволюційний метод з керуванням різноманітністю в популяції (diversity-guided evolutionary algorithm) використовує критерій різноманітності diversity(P) популяції при еволюційному пошуку:

$$\text{diversity}(P) = \frac{1}{N} \sum_{j=1}^N \sqrt{\sum_{k=1}^R (w_{jk} - \overline{w_k})^2},$$

де w_{jk} – значення k -го параметра j -ої хромосоми; $\overline{w_k}$ – середнє значення k -го параметра в популяції; R – розмірність задачі.

У випадках, якщо критерій різноманітності $\text{diversity}(P)$ менше порогового, тоді застосовуються оператори відбору і схрещування, що підвищують різноманітність в популяції, внаслідок чого відбувається фаза дослідження простору пошуку (exploration). Коли різноманітність в популяції досягла достатнього рівня, відбувається фаза пошуку рішення (exploitation). Таким чином, по черзі залежно від різноманітності в популяції виконуються фаза дослідження простору пошуку і фаза пошуку рішення (рис. 7.4).

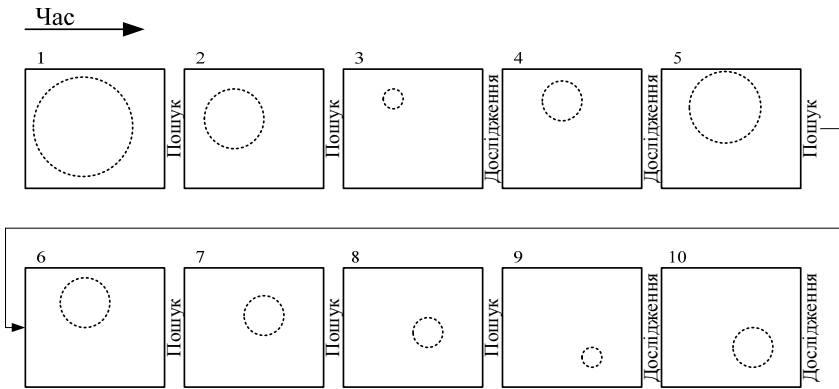


Рисунок 7.4 – Фази еволюційного методу з керуванням різноманітністю популяції

В такому методі застосовується модифікований оператор мутації, який досить швидко збільшує відстань до точки, що є центром популяції. Використання такого оператора дозволяє методу не знаходитися у фазі дослідження простору пошуку впродовж достатньо великої кількості ітерацій. Такий оператор використовує значення центральної точки популяції для розрахунку напрямку мутації. Після цього хромосоми мутують за допомогою застосування Гаусової мутації, для якої середнє значення встановлюється на певній відстані від центральної точки.

РОЗДІЛ 8

БАГАТОКРИТЕРІАЛЬНИЙ ЕВОЛЮЦІЙНИЙ ПОШУК З ОБМЕЖЕННЯМИ

8.1 Постановка задачі та класифікація методів багатокритеріальної оптимізації

Задача багатокритеріальної оптимізації [2, 22, 37] полягає в пошуку точки оптимуму сукупності функцій $F(x) = \{f_1(x), f_2(x), \dots, f_k(x)\}$, де K – кількість цільових функцій.

Задача багатокритеріальної оптимізації з обмеженнями полягає в мінімізації (максимізації) сукупності функцій $F(x) = \{f_1(x), f_2(x), \dots, f_k(x)\}$, при t обмеженнях-нерівностях виду $g_i(x) \leq 0$ та p обмеженнях-рівностях виду $h_i(x) = 0$.

Оптимальним рішенням задачі багатокритеріальної оптимізації є вектор керуючих змінних x^* , при якому всі функції, що оптимізуються $f_1(x), f_2(x), \dots, f_k(x)$ досягають свого оптимального значення.

Основні способи розв'язання задач багатокритеріальної оптимізації за допомогою методів еволюційного пошуку [4, 21, 26] можна класифікувати таким чином (рис. 8.1):

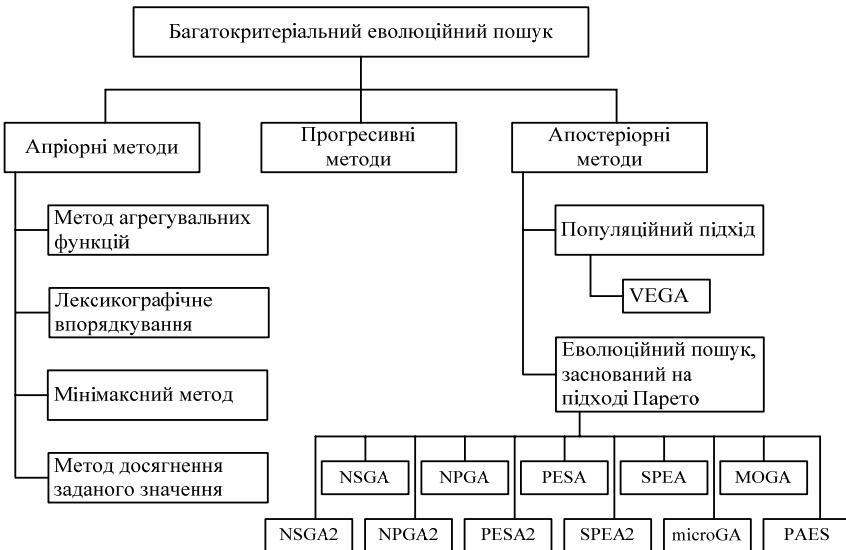


Рисунок 8.1 – Методи багатокритеріального еволюційного пошуку

– *апріорні методи* передбачають попередню оцінку значущості цільових функцій до запуску еволюційного пошуку. При цьому дослідник, як правило, сам задає значущість кожної цільової функції. Для задавання значущості цільових функцій можуть використовуватися вагові коефіцієнти;

– *прогресивні методи* протягом еволюційного пошуку на деяких (або всіх) ітераціях запитують додаткову інформацію про прийнятність того або іншого рішення. У зв'язку з великим навантаженням на дослідника такі методи, як правило, не використовуються;

– *апостеріорні методи* знаходять множину оптимальних рішень, після чого дослідник вибирає кращий з них. До апостеріорних методів відносяться популяційний підхід і пошук, заснований на підході Парето.

8.2 Апріорні методи багатокритеріальної оптимізації

При використанні апріорних методів [2] формується одна цільова функція на основі всієї сукупності функцій $F(x)$. Потім проводиться еволюційний пошук оптимального одержаної функції, який вважається рішенням багатокритеріальної задачі оптимізації.

Виділяють такі апріорні методи:

- метод агрегувальних функцій;
- лексикографічне впорядкування;
- мінімаксий метод;
- метод досягнення заданого значення.

8.2.1 Метод агрегувальних функцій

Метод агрегувальних функцій (вагових коефіцієнтів) набув ширшого поширення в порівнянні з іншими апріорними методами. Такий метод передбачає використання вагових коефіцієнтів для кожної цільової функції [22]. При цьому складається одна агрегувальна цільова функція за формулою:

$$f(x) = \sum_{k=1}^K w_k f_k(x),$$

де w_k – ваговий коефіцієнт k -ої цільової функції, $\sum_{k=1}^K w_k = 1$.

Ваги w_k можуть бути встановлені до запуску еволюційного пошуку (задані як параметри еволюційного методу) або змінюватися залежно від хромосоми H_j та номеру поточної ітерації t .

При оптимізації двохкритеріальної функції вага w_1 агрегувальної функції виду $F = w_1 f_1 + w_2 f_2$ може бути розрахована за допомогою одного з таких способів:

– $w_1 = \text{rand}(0; 1)$ – вага w_1 генерується як випадкове число з інтервалу $(0; 1)$;

– $w_1 = \text{rand}(0; \lambda) / \lambda$ – використовується при застосуванні (μ, λ) еволюційних стратегій, де λ – кількість нащадків, породжуваних μ батьками;

– $w_1 = |\sin(2\pi t / P)|$, де P – періодичність (у ітераціях) – застосовується при необхідності періодичної зміни вагів впродовж еволюційного пошуку.

Вага w_2 розраховується з умови $w_1 + w_2 = 1$: $w_2 = 1 - w_1$.

Істотний недолік агрегаційного підходу полягає в складності вибору агрегуювальної функції, внаслідок чого часто замість глобального оптимуму рішенням є локальний, далеко розташований від глобального. Крім того, даний метод знаходить прийнятний результат при оптимізації невеликого набору функцій.

Перевага застосування агрегуювальних функцій полягає в можливості використання традиційних (не багатокритеріальних) методів для оптимізації одержаної агрегуювальної функції.

8.2.2 Лексикографічне впорядкування

При застосуванні лексикографічне впорядкування (lexicographic ordering) відбувається ранжирування цільових функцій без використання вагових коефіцієнтів. Оптимальне рішення знаходиться шляхом оптимізації цільових функцій, починаючи з найбільш важливої. Проте при великій кількості цільових функцій такий підхід складний в застосуванні, оскільки пошук відбувається по перших найбільш важливих функціях.

8.2.3 Мінімаксний метод

Мінімаксний метод [4] передбачає перетворення багатокритеріальної задачі оптимізації в задачу оптимізації однієї функції за формулою:

$$f(x) = \max_{k \in 1, m} w_i \frac{|f_k^0 - f_k(x)|}{|f_k^0|},$$

де f_k^0 – окремо отриманий мінімум k -ої цільової функції; w_i – ваговий коефіцієнт k -ої цільової функції; $f(x)$ – узагальнена цільова функція для мінімізації.

8.2.4 Метод досягнення заданого значення

При використанні методу досягнення заданого значення [26] відбувається перетворення багатокритеріальної задачі оптимізації в задачу оптимізації однієї функції за формулою:

$$f(x) = \|(F(x) - G)W^{-1}\|,$$

де $F(x)$ – вектор-рядок цільових функцій; G – вектор необхідних значень цільових функцій; W – вектор-стовбець вагових коефіцієнтів, що визначають значущість цільових функцій.

8.3 Популяційний підхід

Популяційний підхід [2] передбачає при вирішенні задачі багатокритеріальної оптимізації з K цільовими функціями формування K підпопуляцій з подальшим пошуком оптимуму єдиної цільової функції в кожній з підпопуляцій. При цьому k -й підпопуляції відповідає k -та цільова функція.

Такий підхід використовується у векторному генетичному методі *VEGA* (vector evaluated genetic algorithm).

Крок 1. Сформувати початкову популяцію з N хромосом.

Крок 2. Випадковим чином розбити популяцію на K підпопуляцій розміром N/K кожна, де K – кількість цільових функцій вирішуваної задачі. При цьому кожна підпопуляція відповідає певній цільовій функції.

Крок 3. Розрахувати значення цільових функцій для хромосом кожної підпопуляції. При цьому для хромосоми H_j з k -ої підпопуляції обчислюється значення цільової функції $f_k(H_j)$.

Крок 4. Відібрати з кожної підпопуляції декілька кращих особин.

Крок 5. Застосувати оператори схрещування і мутації для хромосом об'єднаної популяції, одержаної на кроці 4.

Крок 6. Якщо критерії закінчення пошуку незадовільнено, виконати перехід до кроку 2.

Крок 7. Зупинення.

Основна проблема векторного генетичного методу в тому, що метод відбору хромосом полягає в їх оцінюванні лише за значенням однієї з множини цільових функцій. В результаті цього хромосома, при якій досягається оптимальне рішення для всієї сукупності функцій, може бути відкинута, оскільки для деяких з них вона є далеко не оптимальною.

8.4 Багатокритеріальний еволюційний пошук, заснований на підході Парето

Досягти оптимальності всіх функцій, що оптимізуються, при розв'язанні практичних задач, як правило, неможливо, оскільки значення вектора x^* , при якому одна з цільових функцій є оптимальною, може не приводити до оптимальності іншої цільової функції. Тому при розв'язанні задач багатокритеріальної оптимізації використовують підхід, що полягає в пошуку так званих «Парето оптимальних» рішень [4]. При такому під-

ході знаходиться не істинно оптимальне рішення, а множина субоптимальних рішень.

Вектор x^* – *Парето оптимальний*, якщо не існує іншого x , при якому для всіх цільових функцій виконується умова $f(x) \leq f(x^*)$ та хоча б для однієї з них: $f(x) < f(x^*)$.

Основним недоліком пошуку Парето оптимальних рішень (хромосом) є те, що такий підхід призводить не до єдиного рішення, а до множини рішень, що називається *Парето оптимальною множиною*. Вектори (хромосоми) x^* , що складають Парето оптимальну множину, також називаються *недомінантними*. Множина обчислених значень цільових функцій при недомінантних рішеннях, називається *фронтом Парето*.

8.4.1 Основні поняття

Методи багатокритеріального еволюційного пошуку, засновані на підході Парето, побудовані на виявленні *множини Парето* [2], тобто на визначенні множини недомінантних хромосом. Таких хромосом, як правило, є декілька.

При цьому вважається, що хромосома H_1 домінує над H_2 , якщо для всіх цільових функцій виконується умова $f_i(H_1) \leq f_i(H_2)$ і хоча б для однієї з них: $f_i(H_1) < f_i(H_2)$.

На рис. 8.2 наведено *фронт Парето* (значення цільових функцій для хромосом з множини Парето) для двокритеріальної задачі при мінімізації функції f_1 і максимізації функції f_2 .

Порівняти недомінантні хромосоми між собою за наведеною умовою не можна, оскільки одна з них краще за інші при їх оцінюванні за допомогою однієї функції, тоді як інша хромосома краща за всіх інших при їх порівнянні на основі іншої функції, що видно з рисунка.

Відсутність можливості порівняння недомінантних хромосом призводить до того, що дані методи шукають не одну оптимальну хромосому, а їх множину. А поліпшення рішення на кожній ітерації полягає в зсуві фронту Парето у бік оптимуму множини цільових функцій.

Багатокритеріальний еволюційний пошук, заснований на підході Парето, може бути виконаний в такій послідовності кроків.

Крок 1. Ініціалізувати початкову популяцію P_0 випадково згенерованими рішеннями.

Крок 2. Оцінити кожне рішення в популяції.

Крок 3. Вибрати хромосоми P_{temp} , які згенерують нащадків (Pareto selection mechanism).

Крок 4. Застосувати генетичні оператори схрещування і мутації до хромосом з P_{temp} .

Крок 5. Оцінити нові рішення в популяції.

Крок 6. Вибрати кращі рішення в популяції для формування наступного покоління P_{t+1} (survivor selection mechanism and retention of non-dominated solutions).

Крок 7. Перевірити критерії зупинення. В разі їх невиконання, перейти до кроку 3.

Крок 8. Зупинення.

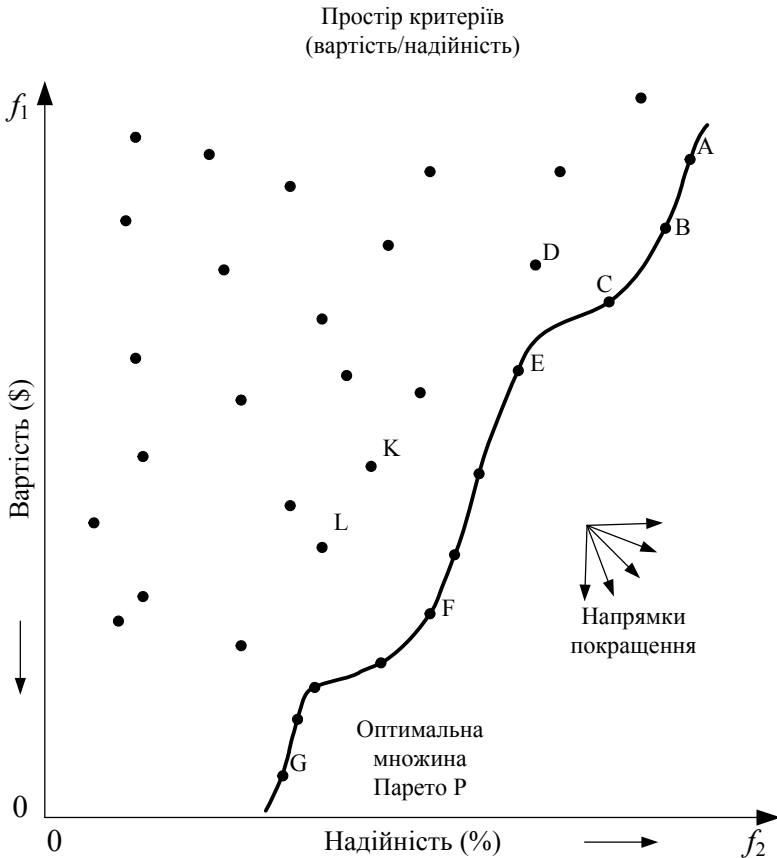


Рисунок 8.2 – Фронт Парето для двоцільової задачі

Таким чином, основними компонентами багатокритеріального еволюційного пошуку, заснованого на підході Парето, є:

- ініціалізація;
- процедура оцінювання хромосом;
- еволюційні оператори відбору, схрещування і мутації;
- критерії зупинення.

Блоки ініціалізації, виконання еволюційних операторів і перевірки критеріїв зупинення аналогічні тим, що використовуються при однокритеріальній еволюційній оптимізації.

Багатокритеріальність оптимізації призводить до необхідності використання спеціальних процедур оцінювання хромосом. При цьому оцінювання виконується, як правило, в два етапи:

- безпосереднє оцінювання хромосом за принципом недомінантності;
- зміна значень фітнес-функцій хромосом, близько розташованих одна від одної (ущільнення множини Парето), спричинена необхідністю підтримки різноманітності в популяції.

8.4.2 Оцінювання хромосом

При багатокритеріальному еволюційному пошуку, заснованому на підході Парето [2, 4], використовується один з двох методів оцінювання хромосом (рис. 8.3):

- сортування за принципом недомінантності;
- ранжирування за принципом недомінантності.

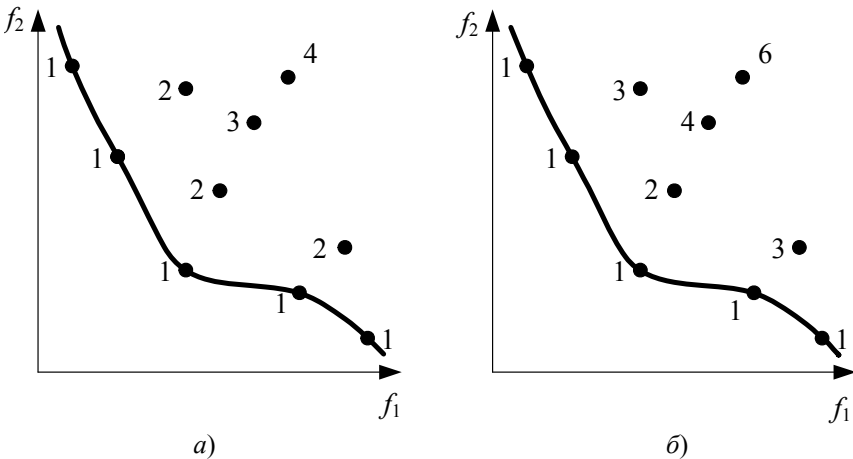


Рисунок 8.3 – Сортування (а) і ранжирування (б) за недомінантністю

Сортування за принципом недомінантності (*nondominated sorting*) передбачає групування хромосом за їх розташуванням у фронтах Парето. При цьому значення фітнес-функції хромосом, що називаються тут фіктивними значеннями фітнес-функції (*dummy fitness*), відповідають номеру їх групи.

Ранжирування за принципом недомінантності (*nondominated ranking, Pareto ranking*) розраховує ранг хромосоми як кількість хромосом, які домінують над нею, збільшену на одиницю.

8.4.3 Методи ущільнення множини Парето

При багатокритеріальному еволюційному пошуку, заснованому на підході Парето, часто виникають ситуації, коли велика кількість хромосом задовольняє умовам недомінантності, що призводить до збільшення хромосом в множині Парето.

Переповнювання такої множини призводить до різкого збільшення машинних витрат на еволюційний пошук і зниження різноманітності в популяції (рис. 8.4).

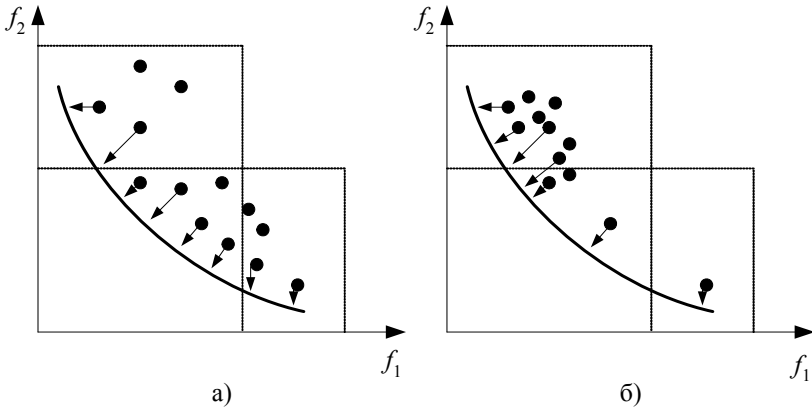


Рисунок 8.4 – Приклад наявності (а) та відсутності (б) різноманітності в популяції

Для запобігання переповнюванню множини Парето використовують різні методи її ущільнення [4] (рис. 8.5).

Метод розподілення фітнес-функції (*fitness sharing*, використовується в NSGA і MOGA) знижує значення фітнес-функції хромосоми від f_j до $f_{j,sh}$ в залежності від насиченості її кластера в просторі цільових функцій.

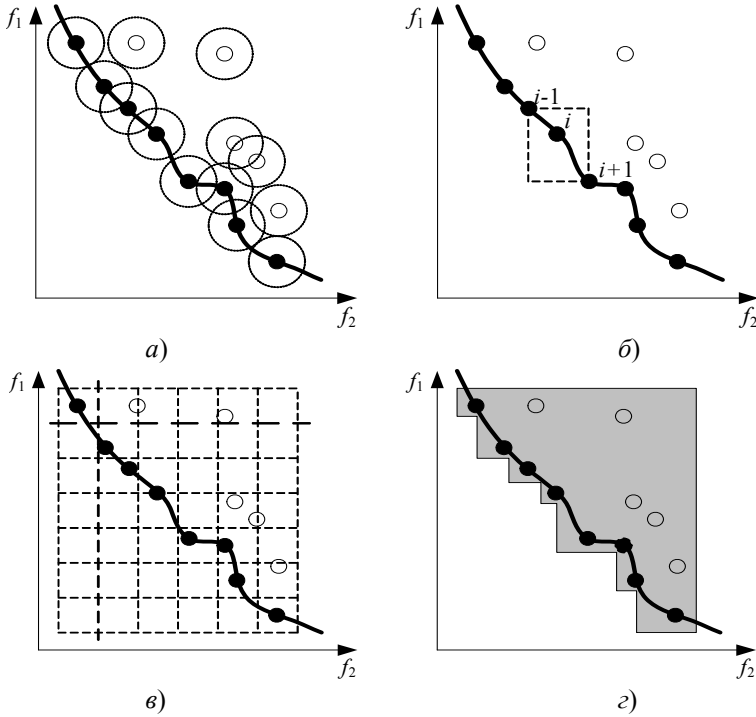


Рисунок 8.5 – Методи ущільнення множини Парето:

- а) розподілення фітнес-функції (fitness sharing);
 б) NSGA2-ущільнення (crowding);
 в) сіткове розбиття простору цільових функцій (grid-based niching);
 г) архівування недомінантних хромосом (ϵ -dominance archiving)

Під насиченістю (*niche counts* m_j) тут розуміється функція, що залежить від кількості розташованих в одному кластері з оцінюваною хромосою особин та ступеню їх близькості до хромосоми H_j . При цьому величина насиченості для хромосоми визначається в залежності від близькості до неї інших хромосом, що потрапили в її кластер (гіперсфера радіусу σ_{sh}), – чим ближче розташовуються інші хромосоми до неї, тим більше величина насиченості.

Крок 1. Задати радіус кластерів σ_{sh} .

Крок 2. Для кожної хромосоми H_j обчислити величину m_j :

$$m_j = \sum_{k=1}^N sh(j, k),$$

$$\text{де } sh(j, k) = \begin{cases} 1 - \left(\frac{d(H_j; H_k)}{\sigma_{sh}} \right)^{\alpha_{sh}}, & \text{якщо } d(H_j; H_k) < \sigma_{sh}; \\ 0, & \text{якщо } d(H_j; H_k) \geq \sigma_{sh}, \end{cases}$$

$d(H_j; H_k)$ – Хеммінгова (або геометрична) відстань між j -ою і k -ою хромосомами.

$$\text{Крок 3. Обчислити } f_{j,sh} = \frac{1}{m_j} f_j.$$

Метод ущільнення (crowding), використовується в NSGA2) ранжирує хромосоми шляхом вимірювання відстані до їх найближчих непомічених сусідів по кожній цільовій функції.

Метод сіткового розбиття простору цільових функцій (grid-based niching), використовується в PAES, PESA, PESA2) групує хромосоми по чарунках в сітці. Якщо в одній чарунці опиняться декілька хромосом, то ймовірність відбору кожної з них знижується пропорційно кількості хромосом в чарунці.

Метод архівування непомічених хромосом (ϵ -dominance archiving) – непомічені хромосоми в архіві утворюють область з межами, що визначаються параметром ϵ . Таким чином, нові непомічені рішення (хромосоми), що знаходяться близько (на відстані, менше ϵ) не увійдуть до архіву.

8.4.4 Методи багатокритеріального еволюційного пошуку

8.4.4.1 Генетичний метод з сортуванням за принципом непоміченості

При використанні генетичного методу з сортуванням за принципом непоміченості (nondominated sorting genetic algorithm, NSGA) спочатку формується група хромосом з найкращими значеннями фітнес-функції (хромосоми, для яких немає домінуючих над ними інших хромосом), потім друга група з тих, що залишилися і т. ін. [26]. Таким чином, хромосоми класифікуються за значенням їх фітнес-функції. При цьому хромосоми, що потрапили в першу групу (множина Парето), мають велику ймовірність бути відібраними для схрещування. Для ущільнення хромосом в групах використовується метод розподілення фітнес-функції.

NSGA може бути виконаний у такій послідовності кроків.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$.

Крок 2. Сформувані початкову популяцію з N хромосом.

Крок 3. Обчислити значення цільових функцій нових хромосом.

Крок 4. Виконати сортування за принципом непоміченості (nondominated sorting).

Крок 4.1. Встановити лічильник кількості фронтів (груп хромосом): $b = 1$.

Крок 4.2. Визначити хромосоми з числа ще не класифікованих, для яких немає домінуючих над ними інших хромосом (недомінантні хромосоми).

Крок 4.3. Привласнити хромосомам, виявленим на кроці 4.2, значення фітнес-функції, що дорівнює номеру поточної групи, $-b$.

Крок 4.4. Виконати процедуру розподілення фітнес-функції (fitness sharing), погіршивши значення хромосом залежно від близькості їх взаємного розташування.

Крок 4.5. У випадку, якщо всі хромосоми в популяції згруповано, виконати перехід до кроку 5.

Крок 4.6. Встановити: $b = b + 1$.

Крок 4.6. Перейти до кроку 4.1.

Крок 5. В залежності від значення фітнес-функції відібрати частину популяції для переходу в нове покоління.

Крок 6. Встановити: $t = t + 1$.

Крок 7. Виконати оператор схрещування.

Крок 8. Застосувати оператор мутації.

Крок 9. Перевірити умови закінчення пошуку. Якщо критерії закінчення незадовольнено, перейти до кроку 3.

Крок 10. Зупинення.

8.4.4.2 Генетичний метод з перевіркою на недомінантність в групі хромосом

Генетичний метод з перевіркою на недомінантність в групі хромосом (niched-Pareto genetic algorithm, NPGA) використовує турнірний відбір у такій послідовності кроків [2].

Крок 1. Відібрати випадковим чином дві хромосоми з популяції.

Крок 2. Відібрати випадковим чином підмножину хромосом (близько 10% від розміру популяції) з популяції.

Крок 3. У випадку, якщо одна з відібраних на першому кроці хромосома є домінуючою перед одержаною на другому кроці множиною хромосом (тобто є хоча б одна хромосома, що домінує над нею), а друга – недомінантною, то вибирається недомінантна хромосома. У випадку, якщо обидві хромосоми домінують або недомінантні, обирається одна з них в залежності від величини насиченості (niche count) її кластеру, розрахованої за допомогою процедури розподілення фітнес-функції (fitness sharing).

Оскільки даний метод не використовує сортування за принципом недомінантності до всієї популяції (на відміну від NSGA і MOGA), то він працює швидше. Але для ефективної роботи методу необхідно правильно вибрати розмір турніру.

8.4.4.3 Багатокритеріальний еволюційний метод

Багатокритеріальний еволюційний метод (multi-objective genetic algorithm, MOGA) використовує ранжирування за принципом недомінантності (nondominated ranking) [4]. При цьому ранг кожної хромосоми залежить від кількості хромосом, які над нею домінують. Недомінантним хромосомам привласнюється ранг $R_j = 1$, іншим: $R_j = 1 + d_j$, де d_j – кількість хромосом, що домінують над хромосоною H_j .

Значення фітнес-функції $f(H_j)$ привласнюються хромосомам залежно від їхнього рангу R_j , використовуючи при цьому деяку функцію, що перетворює значення рангу в значення фітнес-функції.

Для ущільнення хромосом використовується метод розподілення фітнес-функції (fitness sharing). Для отримання нових рішень застосовуються оператори двоточкового схрещування та простої мутації.

Наступні методи відносяться до так званого другого покоління багатокритеріальних еволюційних методів. Вони використовують принцип елітизму (класичний або за допомогою застосування додаткових масивів або архівів, в яких зберігаються значення хромосом кращих особин, одержаних на попередніх ітераціях еволюційного пошуку).

8.4.4.4 Еволюційний метод з архівуванням недомінантних хромосом

Еволюційний метод з архівуванням недомінантних хромосом (strength Pareto evolutionary algorithm, SPEA) використовує архів недомінантних (кращих) хромосом [18], знайдених на кожній ітерації еволюційного пошуку. Для кожної хромосоми в архіві розраховується її потужність (strength) в залежності від її домінування над хромосомами внутрішньої популяції.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$.

Крок 2. Сформувати початкове покоління внутрішньої популяції з N хромосом.

Крок 3. Визначити хромосоми внутрішньої популяції, для яких немає домінуючих над ними інших хромосом (недомінантні хромосоми).

Крок 4. Додати недомінантні хромосоми внутрішньої популяції в архів (зовнішню популяцію).

Крок 5. Обчислити значення фітнес-функції для всіх хромосом H зовнішньої і внутрішньої популяцій.

Крок 5.1. Розрахувати потужність (strength) для всіх хромосом H_j зовнішньої популяції: $R_j = 1 + d_j$, де d_j – кількість хромосом внутрішньої популяції, над якими домінує хромосома H_j .

Крок 5.2. Розрахувати ранг (dummy fitness) для всіх хромосом H_j внутрішньої популяції: $R_j = 1 + d_j$, де d_j – кількість хромосом зовнішньої популяції, які домінують над хромосоною H_j .

Крок 6. Перевірити умови закінчення пошуку. Якщо критерії закінчення задовільнено, перейти до кроку 7.

Крок 7. У випадку, якщо кількість хромосом зовнішньої популяції перевищує максимально допустиме, тоді видалити необхідну кількість хромосом пропорційно зниженим за допомогою процедури, аналогічної розподіленню фітнес-функції, значенням фітнес-функцій (потужності).

Крок 8. Сформувати об'єднану популяцію з внутрішньої і зовнішньої популяцій.

Крок 9. За допомогою парного турнірного відбору залежно від значення фітнес-функції відібрати частину об'єднаної популяції для переходу в нове покоління внутрішньої популяції.

Крок 10. Встановити: $t = t + 1$.

Крок 11. Виконати оператор схрещування для хромосом внутрішньої популяції.

Крок 12. Застосувати оператор мутації до хромосом внутрішньої популяції.

Крок 13. Виконати перехід до кроку 3.

Крок 14. Зупинення.

На основі SPEA був розроблений метод *SPEA2*, основні відмінності якого від SPEA полягають у наступному:

- значення фітнес-функції кожної хромосоми H_i розраховується пропорційно кількості хромосом, які домінують над хромосомою H_j , і кількості хромосом, над якими вона домінує;
- використання техніки найближчого сусіда для підвищення ефективності пошуку;
- використання методу скорочення розміру архіву, що гарантує зберігання граничних рішень.

8.4.4.5 Еволюційна стратегія з архівуванням недомінантних хромосом

Еволюційна стратегія з архівуванням недомінантних хромосом (Pareto archived evolution strategy, PAES) використовує (1 + 1) еволюційну стратегію (один батько породжує за допомогою мутації одного нащадка) в поєднанні з архівом, в якому зберігаються деякі недомінантні хромосоми, одержані на попередніх ітераціях. Одержані в результаті мутації хромосоми порівнюються з хромосомами в архіві, після чого приймаються рішення про їх входження в наступну популяцію [2].

Для ущільнення архіву використовується метод сіткового розбиття простору цільових функцій (grid-based niching).

8.4.4.6 Генетичний метод з сортуванням за принципом недомінантності 2

Генетичний метод з сортуванням за принципом недомінантності 2 (nondominated sorting genetic algorithm 2, NSGA2) ефективніше застосовує елітизм, використовує числовий спосіб подання хромосом, не використовує зовнішню пам'ять у вигляді архіву. Натомість механізм елітизму полягає в поєднанні кращих батьків з кращими нащадками [41].

Крок 1. Встановити лічильник ітерацій: $t = 0$. Випадковим чином сформувати початкову популяцію P_0 з N хромосом.

Крок 2. Відсортувати популяцію за принципом недомінантності хромосом. При цьому кожній хромосомі відповідає свій рівень недомінування (dummy fitness).

Крок 3. Застосувати оператори парного турнірного відбору, SBX-схрещування і мутації для отримання популяції Q_t .

Крок 4. Сформувати об'єднану популяцію $R_t = P_t \cup Q_t$, що дозволить надалі порівняти батьків з нащадками.

Крок 5. Встановити: $t = t + 1$.

Крок 6. Виконати ущільнення об'єднаної популяції R_t за допомогою застосування процедури ущільнення (crowding).

Крок 7. Відібрати хромосоми з популяції R_{t-1} у популяцію P_t , ґрунтуючись на принципі недомінування.

Крок 8. Якщо критерії зупинення незадовільнено, виконати перехід до кроку 3.

Крок 9. Зупинення.

8.4.4.7 Генетичний метод з перевіркою на недомінантність в групі хромосом 2

Генетичний метод з перевіркою на недомінантність в групі хромосом 2 (niched-Pareto genetic algorithm 2, NPGA2) використовує ранжирування за принципом недомінантності із застосуванням турнірного відбору [2]. Як і в методі NSGA2, метод NPGA2 не використовує зовнішню пам'ять у вигляді архіву і застосовує аналогічний механізм елітизму.

Розрахунок величини насиченості кластеру (niche counts) для хромосом відбувається з використанням хромосом вже частково сформованої нової популяції. Така процедура називається розподілення фітнес-функції, що безперервно модернізується.

8.4.4.8 Еволюційний метод з пошуком огинаючої в просторі цільових функцій

Еволюційний метод з пошуком огинаючої в просторі цільових функцій (Pareto envelope-based selection algorithm, PESA) передбачає використання невеликої внутрішньої (internal) популяції і зовнішньої (external, secondary, допоміжної) популяції, що складається з великої кількості хромосом [4]. Для відбору хромосом та керування розмаїтстю в популяції використовує метод сіткового розбиття простору цільових функцій.

У зовнішню популяцію (архів) заносяться хромосоми, що є домінуючими по відношенню до хромосом як внутрішньої, так і зовнішньої популяцій. Після додавання нових хромосом у зовнішню популяцію хромосоми, що раніше містилися в ній, перевіряються на недомінуючість, після чого хромосоми, що не є домінуючими, видаляються. Таким чином, зовнішня популяція являє собою множину Парето, знайдену на поточній ітерації, що відповідає фронту Парето, який, по суті, є огинаючою поверхнею у просторі цільових функцій.

У PESA2 використовується регіональний відбір, заснований на розташуванні в сітці (region-based selection). У даному методі відбирається спочатку чарунка сітки з множиною хромосом, розташованих в ній, а потім одна з цих хромосом.

8.4.4.9 Мікрогенетичний метод

Мікро генетичний метод (micro genetic algorithm, microGA) – метод еволюційного пошуку з малим розміром популяції і процедурою реініціалізації [2].

Мікро генетичний метод може бути виконаний у такій послідовності кроків.

Крок 1. Випадковим чином згенерувати початкову популяцію (малого розміру, наприклад, у чотири хромосоми).

Крок 2. Розбити початкову популяцію на групу, що заміщується і групу, що не заміщується (не змінюється в поточній ітерації).

Крок 3. Провести еволюційний пошук над всією популяцією.

Крок 3.1. Застосувати парний турнірний відбір, заснований на недомінуючості хромосом.

Крок 3.2. Виконати двоточкове схрещування.

Крок 3.3. Виконати просту мутацію.

Крок 4. Сформуванню нове покоління, замінивши особини групи, що заміщається, на кращі особини, утворені в результаті застосування операторів схрещування і мутації.

Крок 5. Провести формування (модифікацію) архіву, що містить хромосоми фронту Парето.

Крок 5.1. Відібрати дві кращі хромосоми з поточної популяції за принципом недомінантності.

Крок 5.2. Порівняти відібрані хромосоми з хромосомами в архіві. В разі недомінантності занести хромосоми в архів.

Крок 5.3. Перевірити архів на наявність домінувальних особин. В разі наявності таких, видалити їх.

Крок 6. Якщо умови закінчення пошуку незадовільнено, тоді виконати перехід до кроку 2.

Крок 7. Зупинення.

Таким чином, в мікрогенетичному методі використовуються три форми елітизму:

- збереження недомінантних хромосом, знайдених у внутрішньому циклі еволюційного методу;
- наявність періодично оновлюваної пам'яті, що заміщується (архіву);
- заміна популяції номінальними рішеннями, знайденими в ході виконання внутрішнього циклу еволюційного методу.

8.4.4.10 Порівняння методів багатокритеріального еволюційного пошуку, заснованого на підході Парето

Для порівняльного аналізу багатокритеріальних еволюційних методів пропонується використовувати такі критерії:

- спосіб кодування хромосом;
- метод оцінювання хромосом;
- метод ущільнення множини Парето;
- оператор відбору;
- оператор схрещування;
- оператор мутації;
- стратегія елітизму.

Результати порівняльного аналізу наведено в табл. 8.1.

8.5 Критерії оптимальності

Оцінку ефективності роботи методів багатокритеріальної еволюційної оптимізації можна проводити за допомогою таких факторів:

- мінімізація відстані між знайденим оптимальним фронтом Парето і реальним фронтом Парето;
- рівномірність розподілу хромосом у фронті Парето. Скупчення рішень поблизу однієї точки вважається негативним чинником;
- максимізація кількості хромосом у фронті Парето (кількості хромосом, що задовольняють умовам Парето).

Таблиця 8.1 – Порівняння методів багатокритеріального еволюційного пошуку, заснованого на підході Парето

№	Назва методу	Спосіб кодування хромосом	Метод оцінювання хромосом	Метод уцілювання множини Парето	Генетичні оператори			Стратегія елітизму
					оператор відбору	оператор схрещування	оператор мутації	
1	NSGA	бінарні	сортування за принципом недомінантності	розподілення фітнес-функції	буть-який	буть-який	буть-який	не використовується
2	NPGA	бінарні	перевірка на домінувальність перед обраною множиною хромосом	розподілення фітнес-функції	турнірний	буть-який	буть-який	не використовується
3	MOGA	бінарні	ранжирування за принципом недомінантності	розподілення фітнес-функції	турнірний	двоточковий	проста мутація	не використовується
4	SPEA	бінарні	ранжирування за принципом недомінантності	розподілення фітнес-функції	турнірний	буть-який	буть-який	зовнішній архів
5	SPEA2	бінарні	ранжирування за принципом недомінантності	розподілення фітнес-функції	турнірний	буть-який	буть-який	зовнішній архів
6	PAES	бінарні	порівняння із хромосомами в архіві	сіткове розбиття простору цільових функцій	не використовується	не використовується	буть-який	зовнішній архів
7	NSGA2	числові	сортування за принципом недомінантності	уцілювання (stowding)	турнірний	SBX-схрещування	буть-який	класичний елітизм
8	NPGA2	бінарні	ранжирування за принципом недомінантності	розподілення фітнес-функції	турнірний	буть-який	буть-який	класичний елітизм
9	PESA	бінарні	сортування за принципом недомінантності	сіткове розбиття простору цільових функцій	турнірний	буть-який	буть-який	зовнішній архів
10	PESA2	бінарні	сортування за принципом недомінантності	сіткове розбиття простору цільових функцій	регіональний турнірний	буть-який	буть-який	зовнішній архів
11	microGA	бінарні	сортування за принципом недомінантності	сіткове розбиття простору цільових функцій	парний турнірний	двоточковий	проста мутація	зовнішній архів

Як критерії використовуються [2, 4, 26]:

– коефіцієнт помилковості одержаного фронту Парето: $K_{er} = K / Pr$, де K – кількість хромосом з одержаного фронту Парето, що не входять в реальний фронт Парето; Pr – загальна кількість хромосом одержаного фронту Парето. Перевагою такого коефіцієнта є простота розрахунку, а недоліком – відсутність відомостей про дійсний фронт Парето при розв’язанні реальних задач;

– узагальнена відстань D – показує, наскільки віддалені хромосоми в знайденому фронті Парето від хромосом дійсно оптимального фронту Парето:

$$D = \frac{1}{Pr} \sqrt{\sum_{p=1}^{Pr} d_p^2},$$

де Pr – загальна кількість хромосом одержаного фронту Парето; d_p – геометрична відстань між p -ою хромосомою в знайденому фронті Парето і найближчою до неї хромосомою в реальному фронті Парето;

– коефіцієнт розрядженості (spacing) S – використовується для оцінювання різноманітності хромосом у фронті Парето:

$$S = \sqrt{\frac{1}{Pr-1} \sum_{p=1}^{Pr} (\bar{d} - d_p)^2},$$

де $d_p = \min_j (|f_1^p(x) - f_1^j(x)| + |f_2^p(x) - f_2^j(x)|)$, $i, j = 1, 2, \dots, Pr$; \bar{d} – середнє значення всіх d_p . Якщо $S = 0$, то всі хромосоми в одержаному фронті Парето рівновіддалені одна від одної (розподілені рівномірно по фронті);

– розтяженність (spread) SP – оцінює розтяженність фронту Парето:

$$SP = \frac{\sum_{k=1}^K d_k^e + \sum_{p=1}^{Pr} |\bar{d} - d_p|}{\sum_{k=1}^K d_k^e + |Pr - 1| \bar{d}},$$

де d_k^e – відстань між екстремальним рішенням у фронті Парето за k -ою фітнес-функцією і екстремальним рішенням за k -ою фітнес-функцією з числа всіх хромосом на поточній ітерації;

– C -метрика (C -metric, set convergence) використовується для порівняння двох фронтів Парето A і B між собою:

$$C(A, B) = K_{AB} / |B|, \quad C(B, A) = K_{BA} / |A|,$$

де $|A|$ та $|B|$ – потужності множин A й B (кількість хромосом у фронтах Парето A та B , відповідно); K_{AB} – кількість хромосом у фронті Парето B , над якими домінує хоча б одна хромосома з фронту Парето A . Якщо $B(A, B) = 0$, тоді жодна з хромосом фронту Парето B є недомінуювальною хромосомами фронту A . Якщо $C(A, B) = 1$, тоді над всіма хромосомами фронту Парето B домінує принаймні одна хромосома фронту Парето A .

8.6 Підтримка обмежень

Більшість практичних оптимізаційних задач містять різні обмеження: лінійні, нелінійні і їх комбінації. Обмеження визначають топологію простору допустимих рішень поставленої задачі. Урахування обмежень дозволяє значною мірою полегшити процес пошуку рішення за рахунок скорочення простору пошуку.

Тому було розроблено методи підтримки обмежень при використанні методів еволюційного пошуку. До таких методів відносять такі [2, 4]:

- методи, що використовують штрафні функції;
- перетворення простору пошуку;
- використання спеціальних еволюційних операторів;
- відновлення неприпустимих рішень;
- застосування багатокритеріального підходу.

8.6.1 Використання штрафних функцій

Методи, що використовують штрафні функції, перетворюють задачі умовної еволюційної оптимізації в послідовність задач безумовної оптимізації шляхом застосування допоміжних функцій.

Використання штрафних функцій передбачає збільшення (при вирішенні задачі на мінімум) значень фітнес-функцій неприпустимих хромосом залежно від ступеня їх неприпустимості, яка визначається кількістю невиконуваних обмежень і рівнем їх нездійсненності. Таким чином, пошук області допустимих рішень здійснюється не випадковим чином, а спрямовано, прискорюючи при цьому весь процес еволюційного пошуку.

Як штрафні функції можуть бути використані такі [4]:

– *статичні штрафні функції* – однакові (постійні) на всіх ітераціях еволюційного пошуку:

$$f^*(x) = f(x) + \sum_{i=1}^m k_i |g_i(x)|^\beta,$$

де $f(x)$ – дійсне значення фітнес-функції; $f^*(x)$ – значення фітнес-функції з урахуванням штрафу; k_i – ваговий коефіцієнт i -го обмеження $g_i(x)$, в разі виконання обмеження $k_i = 0$; β – коефіцієнт, що задається користувачем, як правило β вибирають рівним одиниці або двом;

– *динамічні штрафні функції* – змінюють значення фітнес-функцій неприпустимих хромосом на невелику величину на ранніх ітераціях пошуку, поступово збільшуючи вплив обмежень:

$$f^*(x) = f(x) + C_t \sum_{i=1}^m |g_i(x)|^\beta,$$

де $g_i(x)$ – відповідає значенню i -го обмеження у випадку його невиконання, і нулю – при виконанні обмеження; C_t – коефіцієнт впливу неприпустимих обмежень на цільову функцію на t -ій ітерації;

– *адаптивні штрафні функції* – використовують штрафні функції в залежності від значень фітнес-функцій кращих хромосом:

$$f^*(x) = f(x) + \left| opt_t - opt_{f,t} \right| \sum_{i=1}^m \left(\frac{g_i(x)}{q_{i,t}} \right)^B,$$

де opt_t – оптимальне значення фітнес-функцій на t -ій ітерації; $opt_{f,t}$ – оптимальне значення фітнес-функцій для допустимих хромосом на t -ій ітерації; $q_{i,t}$ – штрафна функція, що використовується на t -ій ітерації.

Важливо враховувати можливість ситуації, коли значення фітнес-функції з врахуванням штрафу неприпустимої хромосоми буде краще фітнес-функції допустимих хромосом, що може призвести до помилкового оптимуму. Таким чином, необхідно передбачити процедуру, що зменшує (при розв'язанні задачі на мінімум) значення фітнес-функції допустимих хромосом так, щоб не було неприпустимих хромосом з кращим значенням фітнес-функції в порівнянні з допустимими хромосомами.

Переваги такого підходу полягають в можливості використання класичних методів еволюційного пошуку для розв'язання задачі з обмеженнями. Недоліком використання штрафних функцій є необхідність вибору коефіцієнтів штрафної функції, що часто є нездійсненною задачею. Вибір відповідних коефіцієнтів штрафних функцій залежить від співвідношення розмірів допустимої області пошуку і області визначення функції, топологічних особливостей допустимого простору пошуку, типу цільової функції, числа змінних, числа і типу обмежень.

До методів, що використовують штрафні функції, відноситься *при власнення максимально можливих значень* (наприклад, нескінченність) *фітнес-функціям хромосом, що не задовольняють обмеженням*. Як недоліки такого підходу можна виділити:

- складність знаходження області допустимих значень хромосом. Даний недолік особливо виявляється на початкових ітераціях еволюційного пошуку і значно збільшує час пошуку;

- відсутність будь-якого впливу неприпустимих хромосом на пошук;

- можливість випадку, коли еволюційний метод не потрапить в область допустимих значень.

8.6.2 Перетворення простору пошуку

Перетворення простору пошуку направлене на гарантування допустимих рішень [18]. Відображення (mapping) реального пошукового простору в інший з врахуванням обмежень на значення змінних досягається шляхом

кодування хромосом так, щоб була відсутня можливість появи неприпустимих хромосом. При цьому обмеження задаються, як правило, у вигляді граничних значень змінних або інших лінійних залежностей. Такий підхід передбачає також використання операторів схрещування і мутації, що виробляють мінімальну кількість (у ідеальному випадку – відсутність) неприпустимих хромосом. Проблема застосування такого підходу полягає в складності синтезу і реалізації процедур кодування / декодування.

Цей і попередній підходи взаємодоповнювані і використовуються, як правило, спільно.

8.6.3 Використання спеціальних еволюційних операторів

У даній групі методів замість класичних операторів еволюційного пошуку застосовуються їх модифікації, що дозволяють одержувати в процесі пошуку лише допустимі рішення задачі оптимізації з обмеженнями [2]. Методи, які використовують штрафні функції, незважаючи на можливість отримання неприпустимого рішення, перевершують за якістю і швидкістю оптимізації методи даної групи. Такий ефект пов'язаний з тим, що при пошуку по всьому можливому простору пошуку використовується не лише інформація з області, заданої обмеженнями, але і додаткова інформація, що сприяє збільшенню збіжності до оптимуму.

8.6.4 Відновлення неприпустимих рішень

Відновлення полягає в приведенні за певним правилом неприпустимих хромосом до допустимого вигляду шляхом зміни значень деяких генів хромосоми так, щоб вона стала задовольняти обмеженням. Такий підхід є мало ефективним, оскільки вимагає додаткових обчислень значень фітнес-функцій і перевірки обмежень відновлюваних хромосом [4].

8.6.5 Застосування багатокритеріального підходу

Застосування багатокритеріального підходу передбачає перетворення задачі оптимізації K цільових функцій з m -обмеженнями до багатокритеріальної оптимізації $(m + K)$ функцій [2].

8.6.5.1 Метод багатокритеріального генетичного пошуку з підтримкою обмежень

Метод багатокритеріального генетичного пошуку з підтримкою обмежень (constrained optimization by multi-objective genetic algorithm, COMOGA) використовує ранжирування хромосом в залежності від кільк-

кості порушень обмежень [18]. При формуванні нової популяції використовується два види відбору: перший – на основі ранжирування за кількістю порушень обмежень, другий – на основі значенні фітнес-функції. При цьому задається параметр P_{cost} , що визначає частину хромосом, що відбираються за допомогою ранжирування за значенням фітнес-функції.

8.6.5.2 Векторний генетичний метод

Використання векторного генетичного методу (vector evaluated genetic algorithm, VEGA) для розв'язання задач з обмеженнями передбачає до основних підпопуляцій додавання підпопуляцій, що відповідають не цільовим функціям, а обмеженням [4]. При цьому цільову функцію для підпопуляції, відповідної k -му обмеженню, розраховують таким чином:

$$f_k(H_j) = \begin{cases} g_k(H_j), & \text{якщо } g_k(H_j) \text{ не виконується;} \\ v, & \text{якщо } v \neq 0; \\ f(H_j), & \text{якщо } g_k(H_j) \text{ виконується,} \end{cases}$$

де v – кількість невиконаних обмежень.

8.6.5.3 Лінійний пошук і домінування Парето

При використанні лінійного пошуку і домінування Парето (line search and Pareto dominance) задача оптимізації однієї цільової функції з m -обмеженнями виду $g_k(H_j) \leq 0$ зводиться до оптимізації двох функцій, першою з яких є цільова функція задачі, а друга – функція вигляду [4]:

$$f_2(H_j) = \sum_{k=1}^m (\max\{0; g_k(H_j)\})^\beta,$$

де $\beta \in [1; 2]$.

Після цього знаходиться множина Парето для одержаної двокритеріальної задачі. З множини Парето обираються дві хромосоми, на основі яких проводиться лінійний пошук у напрямі вектора

$$d = \frac{(H_i - H_j)}{|H_i - H_j|}$$

таким чином, щоб хромосома на цьому векторі була одночасно кращою, ніж H_i та H_j . При цьому лінійний пошук виступає в ролі оператора схрещування. Мутація виконується аналогічно, за винятком того, що вектор d проектується на вісь хромосоми H_j .

Недоліками даного методу є необхідність підтримки різноманітності рішень в популяціях і великий обсяг необхідних ресурсів для виконання пошуку.

8.6.5.4 Мінімаксна стратегія

Мінімаксна стратегія [2, 41] використовує відбір за правилами:

– якщо обидві хромосоми H_i та H_j є прийнятними (задовольняють обмеженням), то вибирається хромосома з кращим значенням фітнес-функції;

– якщо одна з хромосом не задовольняє обмеженням, а інша задовольняє, то обирається та хромосома, яка задовольняє обмеженням;

– якщо обидві хромосоми не задовольняють обмеженням, то обирається та з них, для якої виконується більша кількість обмежень.

Основний недолік такого підходу полягає в тому, що еволюційний пошук сконцентрований на підтримці обмежень, що призводить до переважно випадкового пошуку оптимуму цільових функцій в знайденому допустимому просторі керованих змінних.

Таким чином, еволюційні методи багатокритеріальної оптимізації дозволяють знаходити оптимальні значення керованих змінних для декількох цільових функцій.

РОЗДІЛ 9

ЕВОЛЮЦІЙНИЙ СИНТЕЗ НЕЙРОМЕРЕЖНИХ МОДЕЛЕЙ

Застосування градієнтних методів локального пошуку, традиційно використовуваних для побудови нейромоделей, у ряді випадків (багатоекстремальність цільової функції, недиференційованість функцій активації нейроелементів та ін.) є неприйнятним або неможливим [49, 51, 53, 62, 67].

Тому доцільним є використання методів еволюційного пошуку [18, 39, 86], які не вимагають обчислення значень похідних цільових функцій, що дозволяє ефективно застосовувати їх для вирішення задач побудови нейромоделей.

У наш час методи еволюційної оптимізації використовуються для вирішення різних завдань, пов'язаних із синтезом нейромереж: відбір ознак, настроювання ваг, вибір оптимальної архітектури мережі, адаптація навчального правила, ініціалізація значень вагових коефіцієнтів, витяг правил з побудованої мережі, пошук оптимальних значень параметрів досліджуваної системи за наявною нейромоделлю і т. п. [9, 10, 33, 34, 38, 46, 49, 79, 98].

9.1 Загальна постановка задачі синтезу нейромережних моделей

У загальному випадку нейронна мережа являє собою набір адаптивних елементів (штучних нейронів), що мають непусту множину входів і зв'язаних між собою синаптичними з'єднаннями [61, 63, 85, 91, 105].

Базовим елементом нейронної мережі є формальний нейрон, що має декілька входів і один вихід. Кожному i -му входу ставиться у відповідність ваговий коефіцієнт (синаптична вага) w_i , що відповідає його значущості. Дискримінантна функція (вагова функція) нейрона перетворює зважені входи й подає їх на функцію активації (передатна функція), результат обчислення якої є виходом нейрона. Таким чином, формальний нейрон реалізує скалярну функцію векторного аргументу.

Як дискримінантні функції використовуються зважена сума, зважений добуток, функція відстані, а як функції активації – лінійна, гранична, сигмоїдна, функція Гаусса, гіперболічний тангенс та ін. Як правило, застосовуються зважена сума (як дискримінантна функція) і сигмоїдна функція (як функція активації).

Математична модель штучного нейрона описується співвідношенням:

$$y = \psi(\varphi(w, x)),$$

де y – значення сигналу на виході нейрона; ψ – функція активації; φ – дискримінантна функція; w – вектор, що містить значення вагових коефіцієнтів і значення зсуву (граничне значення); x – вектор вхідних аргументів.

Таким чином, основними параметрами формальних нейронів є:

- кількість входів L ;
- значення вагових коефіцієнтів w_1, w_2, \dots, w_L ;
- значення зсуву (порога) w_0 ;
- використовується дискримінантна функція;
- вид і параметри функції активації.

Як правило, кількість входів, дискримінантна функція й функція активації визначаються специфікою задачі. Тому параметрами формального нейрона, що налагоджують, є значення вагових коефіцієнтів і значення зсуву.

Всі моделі штучних нейронних мереж являють собою сукупність нейронів, пов'язаних між собою певним чином. Основними розходженнями нейромережних моделей є способи зв'язку нейронів між собою, а також механізми й напрямки поширення сигналів по мережі.

В залежності від способів зв'язку нейронів і напряму поширення сигналів по мережі розрізняють: нейронні мережі прямого поширення й нейронні мережі зворотного поширення (із прямими зворотними зв'язками, з непрямыми зворотними зв'язками, повнозв'язні).

При вирішенні реальних практичних задач прогнозування, класифікації, керування складними об'єктами, процесами й системами, як правило, досить застосування нейромережних моделей прямого поширення, що мають досить просту структуру та є інтерпретабельними.

Багат шарова нейронна мережа прямого поширення складається з формальних нейронів. При цьому зовнішній сигнал керування змінних x надходить тільки на входи нейронів першого шару, виходи нейронів μ -го шару надходять тільки на входи $(\mu + 1)$ -го шару, а виходи нейронів останнього M -го шару є виходами мережі.

Параметрами такої мережі є:

- кількість шарів мережі M ;
- кількість нейронів на кожному шарі N_1, N_2, \dots, N_M ;
- матриця зв'язків між нейронами мережі;
- параметри формальних нейронів, що утворюють мережу.

У силу теорем Колмогорова-Арнольда та Хехт-Нільсена [105] трьох шарів достатньо для одержання ефективного рішення за допомогою нейронних мереж прямого поширення. Тому при пошуку оптимальної структури неромережної моделі актуальним є настроювання таких параметрів, як кількість нейронів на кожному шарі, матриця зв'язків між нейронами мережі, а також вибір параметрів нейронів, що утворюють мережу.

Таким чином, нейромережна модель може бути подана у вигляді багатовимірної функції f , що залежить від структури й параметрів моделі:

$$y = f(w, x),$$

де w – множина значень вагових коефіцієнтів і зсувів нейромережі; x – вектор аргументів, що належать простору ознак.

Нехай задано початкову вибірку вихідних даних у вигляді:

$$\langle X = \{X_1, X_2, \dots, X_L\} = \{X_i\}, Y = \{y_1, y_2, \dots, y_m\} = \{y_p\} \rangle,$$

де X – набір значень ознак, що характеризують досліджуваний об'єкт або процес; Y – масив значень вихідного параметра в заданій вибірці; $X_i = \{x_{ip}\}$ – i -та ознака у вибірці, $i = 1, 2, \dots, L$; x_{ip} – значення i -ої ознаки для p -го екземпляра вибірки, $p = 1, 2, \dots, m$; y_p – значення прогнозованого параметра для p -го екземпляра; L – загальна кількість ознак у вихідному наборі; m – кількість екземплярів вибірки.

Тоді завдання синтезу нейромережної моделі [70, 91, 105] за навчальною вибіркою полягає у побудові нейромережі виду:

$$\text{HM} = \text{HM}(C, W, B, DF, TF),$$

для якої:

$$\xi(\text{HM}, X, Y) \rightarrow \min,$$

де C – матриця, що визначає наявність синаптичних зв'язків між елементами мережі (рецепторами, нейронами); $W = W(C)$ – матриця вагових коефіцієнтів, що відповідають присутнім у мережі HM зв'язкам; $B = B(C)$ – вектор зсувів нейронів мережі; $DF = DF(C)$ – вектор дискримінантних функцій нейроелементів; $TF = TF(C)$ – вектор функцій активації нейронів мережі; $\xi(\text{HM}, X, Y)$ – критерій, що визначає ефективність використання нейромережної моделі HM для апроксимації залежності між набором вхідних параметрів X і відповідним йому вектором значень вихідного параметру Y .

Як правило, за критерій оптимальності нейромоделі беруть середньоквадратичну помилку:

$$\xi = \sum_{p=1}^m \left(y_p - y(\text{HM}, Z_p) \right)^2,$$

де Z_p – набір значень ознак для p -го екземпляру; $y(\text{HM}, Z_p)$ – значення виходу нейромоделі HM, обчислене для набору значень Z_p .

Для побудови ефективної нейромережної моделі, що найкращим чином дозволяє апроксимувати досліджувану залежність, необхідно виконати структурний і параметричний синтез такої моделі на основі найбільш інформативного набору ознак.

Процес побудови ефективних нейромережних моделей на основі відомої вибірки вихідних даних може бути поданий у вигляді послідовності етапів [61, 73, 105]:

- вибір системи інформативних ознак;
- структурний синтез;
- параметричний синтез;
- оптимізація побудованої нейромоделі.

Ці етапи наведені відповідно до хронології їхнього виконання, однак результати, отримані на деяких пізніших етапах, можуть приводити до необхідності повернення до попереднього етапу.

9.1.1 Вибір системи інформативних ознак

Масиви даних великого розміру, характеризуються надлишковими й неінформативними ознаками, які ускладнюють процес синтезу нейромоделі, знижують її інтерпретабельність і приводять до її надмірності, що надалі збільшує час прогнозування або класифікації за синтезованою моделлю [3, 5, 6, 16, 17, 20, 47].

Крім того, принципово неможливо включити в модель всю множину вхідних змінних і зв'язків між ними, що визначають і формують вихідну змінну. У розгляд зазвичай вводиться кінцеве число найбільш важливих, домінуючих вхідних змінних, для яких є засоби знімання інформації в процесі функціонування об'єкту. Саме для цієї невеликої кількості вимірюваних вхідних змінних і будується модель зв'язку з вихідною змінною [16, 43, 52, 59, 82].

Тому побудова нейромережних моделей тісно пов'язана з вирішенням завдання вибору набору вхідних змінних, що мають найбільшу інформацію про даний конкретний вихід.

Постановка завдання відбору інформативних ознак може бути подана одним з таких способів [5, 16, 35, 44, 50].

1. Ідеалізована постановка: виділити комбінацію ознак X^* з вихідного масиву даних, при якій досягається мінімум заданого критерію оцінювання набору ознак:

$$J(X^*) = \min_{X \in XS} J(Xe),$$

де Xe – елемент множини XS ; $J(Xe)$ – критерій оцінювання значущості набору ознак Xe ; XS – множина всіх можливих комбінацій ознак, отримана з вихідного набору ознак X .

2. Класична постановка: відібрати з множини вихідних L ознак комбінацію, що складається не більш, ніж з L_0 ознак ($L_0 < L$), при якій досягається оптимум заданого критерію:

$$J(X^*) = \min_{X \in XS, |Xe| \leq L_0} J(Xe),$$

де $|Xe|$ – кількість елементів у множині Xe .

3. Знайти набір ознак мінімального розміру, що забезпечує досягнення заданого значення критерію оцінювання значущості набору ознак:

$$|X^*| = \min_{X \in XS, J(Xe) < \varepsilon} |Xe|,$$

де ε – задане значення критерію оцінювання набору ознак J .

Результатом виконання процедури відбору ознак є оптимальний набір ознак X^* , що має достатню інформативність. Інформативність ознаки (набору ознак) – це величина, що відбиває ступінь взаємозв'язку ознаки

(набору ознак) із прогнозованим параметром. Інформативність комбінації ознак дорівнює сумі інформативності окремих ознак тільки при їхній незалежності. Якщо ознаки залежні одна від одної, то інформативність набору не виражається через інформативність окремих ознак.

Таким чином, отриманий у результаті відбору ознак оптимальний набір X^* , маючи достатню інформативність, найбільш повно відбиває досліджуваний об'єкт або процес. При цьому з вихідного набору X виключаються:

- незначущі ознаки – ознаки, що не впливають на вихідний параметр;
- надлишкові ознаки – ознаки, значення яких залежать від інших ознак. Такі ознаки не приводять до поліпшення якості прогнозування за синтезованою нейромоделлю.

Виділення найбільш значущої комбінації інформативних ознак є досить важким і ресурсомістким завданням, оскільки воно пов'язано з необхідністю комбінаторного перебору. У наш час запропоновано різні методи виділення набору ознак, серед яких найбільше поширення отримали [16, 36, 59, 60, 67, 84]:

- метод повного перебору (exhaustive search);
- пошук у глибину (depth-first search);
- пошук в ширину (breadth-first search);
- метод гілок та границь (branch and bound method) або скорочений пошук у глибину;
- метод групового врахування аргументів або скорочений пошук в ширину;
- метод послідовного додавання ознак (forward selection);
- метод послідовного видалення ознак (backward selection);
- метод почергового додавання й видалення ознак (combined selection);
- ранжирування ознак;
- кластеризація ознак (unsupervised learning for feature selection);
- випадковий пошук з адаптацією (adaptive stochastic search);
- еволюційний пошук (evolutionary search).

Застосування методів повного й скороченого перебору [5, 67] вимагає оцінки значної кількості комбінацій ознак, складених з вихідної сукупності ознак, що унеможливило використання такого підходу при великій кількості ознак у вихідному наборі, оскільки вимагає величезних обчислювальних витрат.

Методи евристичного пошуку [16, 59] недостатньо ефективні через неоптимальність жадібної стратегії пошуку, що послідовно додає або видаляє по одній ознаці, у результаті чого одержуваний набір ознак містить надлишкові ознаки, що корелюють із іншими ознаками в наборі. Крім того, при відборі ознак з масиву даних, що характеризується великою розмірністю, евристичний пошук також вимагає значних витрат на оцінку наборів ознак.

Оскільки при вирішенні задач оцінювання й класифікації, як правило, доводиться мати справу із системою статистично залежних ознак, інформативність набору яких не виражається через інформативність окремих ознак [52, 67], використання ранжирування ознак за обчисленою індивідуальною оцінкою, як правило, є недоцільним.

Для пошуку комбінації інформативних ознак в умовах їхньої взаємної залежності одна від одної представляється доцільним вибрати методи еволюційного пошуку [23, 24, 80, 83, 95, 99], оскільки вони є більш пристосованими для знаходження нових рішень за рахунок об'єднання кращих рішень, отриманих на різних ітераціях, мають можливості для виходу з локальних оптимумів. Однак відносно повільна збіжність і залежність від початкових умов еволюційного пошуку приводять до необхідності розробки нових еволюційних методів відбору інформативних ознак, вільних від зазначених недоліків.

9.1.2 Структурний синтез

Етап структурного синтезу нейромоделей є найважливішим завданням при побудові нейромереж [88, 105], оскільки на цьому етапі формується топологія зв'язків, обираються функції активації нейронів, що надалі визначають принцип функціонування мережі і її ефективність для вирішення досліджуваного завдання. Так, нейромережі, що мають невелику кількість нейронів і лінійні функції активації, як правило, через свої обмежені апроксимаційні здатності не дозволяють вирішувати реальні практичні завдання. У той же час вибір надлишкової кількості нейронів у мережі приводить до проблеми перенавчання й втрати апроксимаційних властивостей нейромоделі [49, 67]. Оскільки для вирішення більшості практичних завдань розпізнавання й оцінювання можуть бути використані нейромережі прямого поширення з нейроелементами, що використовують зважену суму як дискримінантну функцію, вибір виду вагових функцій при структурному синтезі нейромоделей, як правило, не здійснюють [49, 51].

Головною метою дослідника на етапі структурного синтезу є визначення загального виду структури шуканого зв'язку, поданої у вигляді нейромережної моделі, вихідного параметра й вектора керуючих змінних.

Завдання структурного синтезу нейромоделі полягає у пошуку структури моделі виду $HM = HM(C, TF)$, для якої $\xi(HM, X, Y) \rightarrow \min$, де $C = C(L, A)$ – матриця, що визначає наявність синаптичних зв'язків між елементами мережі (рецепторами, нейронами); A – максимально припустима кількість нейронів у мережі.

У наш час не існує стандартних методів, які утворювали б строгу теоретичну базу для вирішення завдання структурного синтезу нейромоде-

лей. Будучи вузловим у процесі синтезу нейромоделей реальних об'єктів, процесів або систем, цей етап не має строгих і закінчених математичних рекомендацій з його реалізації. Тому його реалізація вимагає спільної роботи фахівця у відповідній предметній області й аналітика, спрямованої на найглибше проникнення у фізичний механізм досліджуваного зв'язку.

У ряді випадків структуру нейромоделі можна вибрати апріорі, виходячи з фізичних, хімічних і емпіричних міркувань про характер взаємодії досліджуваного об'єкту із зовнішнім середовищем.

Однак для складних об'єктів, процесів і систем загальних апріорних міркувань може виявитися недостатньо.

У цьому випадку завдання вибору структури зводиться до пошуку математичними методами оптимізації за апостеріорним даними з використанням наявної додаткової інформації. Використання таких методів припускає формування певних гіпотез про топологію мережі, які, як правило, перевіряють на основі критеріїв, що базуються на таких ідеях:

- досягнення компромісу між складністю моделі й точністю її оцінювання;
- пошук моделі, найбільш стійкої до варіювання складу вибіркового даних, на основі яких вона оцінюється.

Існуючі методи автоматичного пошуку оптимальної структури нейромережних моделей, як правило, використовують жадібну стратегію пошуку [49, 70, 105]. Так конструктивні методи починають пошук з мінімально можливої архітектури мережі (нейромережа із мінімальною кількістю шарів, нейронів і міжнейронних зв'язків) і послідовно на кожній ітерації додають нові шари, нейрони й міжнейронні зв'язки. При використанні деструктивних методів на початковій ітерації оцінюється ефективність нейромоделі, що містить максимально припустиму кількість шарів, нейронів і міжнейронних зв'язків, потім у процесі пошуку структура такої моделі скорочується до найбільш прийнятної.

Однак такі методи внаслідок застосування жадібної стратегії досліджують незначну частину простору всіх можливих структур нейромоделей і схильні до влучення в локальні оптимуми. Отже, виникає необхідність розробки нових методів структурного синтезу нейромоделей, що дозволяють більш ефективно досліджувати простір пошуку при виборі оптимальної структури мережі.

9.1.3 Параметричний синтез

На етапі параметричного синтезу відбувається навчання нейромережної моделі, тобто підбираються такі значення вагових коефіцієнтів мережі, а іноді й параметрів функцій активації, при яких мережа найбільш ефективним чином дозволяє вирішувати поставлене завдання [61, 91, 105].

Задача параметричного синтезу нейромоделі заданої структури полягає у пошуку такого набору значень вагових коефіцієнтів і зсувів, при якому досягається мінімум критерію помилки нейромоделі. Навчання нейронної мережі полягає в зміні значень синаптичних ваг у результаті послідовного пред'явлення екземплярів навчальної вибірки.

Розрізняють дві концепції навчання нейромереж: навчання із учителем і навчання без учителя.

Навчання без учителя припускає використання для настроювання вагових коефіцієнтів мережі тільки значень вхідних ознак екземплярів навчальної вибірки. При цьому реальні значення вихідного параметра невідомі, а процес навчання полягає в об'єднанні близько розташованих екземплярів у кластери. Таким чином, екземпляри, розташовані досить близько один до одного в просторі ознак, будуть віднесені до одного класу при їхній класифікації за допомогою налагодженої мережі.

При навчанні із учителем навчальна вибірка містить значення вхідних ознак, що характеризують досліджуваний об'єкт або процес, і значення вихідного параметра. При такому виді навчання мінімізується сума квадратів різниці між модельними й реальними виходами досліджуваного об'єкту, процесу або системи.

У наш час відомі різні методи навчання багатшарових нейронних мереж прямого поширення, більшість із яких засновані на правилі Хебба. Відповідно до цього правила у випадку одночасної активації двох нейронів вага їхнього зв'язку зростає, у результаті чого часто використовувані зв'язки в нейромережній моделі підсилюються.

Істотним недоліком такого підходу є відсутність механізму зменшення значень вагових коефіцієнтів, у результаті чого відбувається значне збільшення значень вагових коефіцієнтів з ростом кількості екземплярів, використовуваних для навчання, що приводить до некоректної роботи мережі.

Іншим методом навчання нейромережних моделей є дельта-правило, що може бути використане для настроювання одношарових мереж прямого поширення й полягає в зміні значень вагових коефіцієнтів мережі пропорційно різниці між реальним виходом і виходом мережі.

Більшість методів, що використовуються в наш час для навчання нейронних мереж прямого поширення, є градієнтними. Використання градієнтних методів навчання засновано на мінімізації середньоквадратичної помилки, при цьому для корекції значень синаптичних ваг w_{ij} нейронної мережі використовують формулу:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \left. \frac{\partial \xi}{\partial w_{ij}} \right|_{w_{ij}=w_{ij}(t)},$$

де $w_{ij}(t+1)$ та $w_{ij}(t)$ – значення вагового коефіцієнта між i -им та j -им нейронами на t -ій та $(t+1)$ -ій ітераціях, відповідно.

Таким чином, у загальному випадку навчання нейронної мережі зводиться до завдання багатовимірної нелінійної оптимізації, розв'язуваної, як правило, за допомогою градієнтних методів. Для обчислення значень часток похідних може бути використаний метод зворотного поширення помилки, що, однак, може бути застосований тільки для настроювання ваг нейромережних моделей з диференційованими функціями активації.

У розглянутих вище детермінованих методах навчання використовуються чітка послідовність дій, виконувана за певними правилами.

Поряд з детермінованими методами також можуть бути використані стохастичні методи настроювання вагових коефіцієнтів нейромережної моделі, які виконують параметричний синтез на основі дій, що підкоряються деякому випадковому закону. При використанні таких методів на початкових ітераціях виконують корекції значень вагових коефіцієнтів на великі величини, поступово зменшуючи їх у процесі пошуку. До стохастичних методів відносять: метод випадкового пошуку, метод імітації відпалу, а також методи еволюційного пошуку. Значною перевагою стохастичних методів навчання є відсутність необхідності обчислення значень похідних цільової функції. Однак, оскільки такі методи використовують імовірнісний підхід, не враховуючи при цьому інформації про поверхню функції помилки, рішення (набір параметрів нейромоделі), що забезпечує прийнятну точність, може бути не знайдено.

Таким чином, з метою усунення зазначених недоліків існуючих методів параметричного синтезу нейромоделей, доцільно розробити ефективні методи навчання нейронних мереж прямого поширення, що дозволяють виконувати параметричний синтез, не висуваючи при цьому різних вимог до функцій активації нейроелементів.

9.1.4 Оптимізація побудованої нейромоделі

На можливість використання нейромережних моделей на практиці істотно впливають складність побудованої нейромережі й швидкість обчислення значення цільового параметра за набором даних, що не входить у навчальну вибірку [70, 91, 92].

Тому актуальним є спрощення структури синтезованої нейронної мережі. Завдання оптимізації побудованої нейромережної моделі вигляду $HC(C, W, B, DF, TF)$, полягає в пошуку таких нових значень $C' \subseteq C, W', B', DF', TF'$, при яких досягаються оптимальні значення заданих критеріїв оптимальності $\xi_1, \xi_2, \dots, \xi_K$, що враховують основні характеристики нейромоделі, де K – кількість цільових критеріїв.

У наш час існує два основних підходи до оптимізації структури синтезованої нейромережної моделі [91, 105]:

- методи видалення зв'язків;
- методи видалення нейронів.

Методи видалення зв'язків зменшують кількість елементів матриці ваг нейромережі, спрощуючи в такий спосіб її структуру. При використанні таких методів видаляються зв'язки, що мають найменші значення синаптичних ваг, або зв'язки, що мають найменший вплив на ефективність функціонування нейромережної моделі. Виділяють такі методи видалення зв'язків: метод обнуління найменших ваг, метод оптимального руйнування нейромережі, метод оптимальної перебудови мережі.

Методи другої групи спрощують структуру нейромоделі шляхом виключення з її нейронів, видалення яких не приводить до значного збільшення помилки мережі. Таким чином, при використанні методів видалення нейронів додатково необхідно оцінювати їхню значущість шляхом обчислення помилки спрощеної моделі. До таких методів відносять: метод видалення нейронів з урахуванням їх важливості й метод видалення нейронів з використанням вартісної функції.

Таким чином, існуючі підходи спрощення нейромоделей передбачають послідовне видалення нейронів або синаптичних з'єднань у мережі [70, 91, 105], використовуючи при цьому жадібну стратегію. Як правило, застосування жадібної стратегії приводить до дослідження незначної частини простору пошуку, оскільки припускає послідовне видалення елементів мережі (нейронів або зв'язків між ними), не оцінюючи при цьому ефективності виключення наборів, що складаються з декількох елементів.

Тому необхідно розробити нові методи спрощення нейромережних моделей, що здатні виконувати пошук у всьому просторі можливих структур нейронних мереж прямого поширення.

9.2 Еволюційний відбір інформативних ознак

Як показано вище, для синтезу ефективних нейромережних моделей, що володіють високими апроксимаційними якостями й не є надлишковими, необхідно виділити з вихідного набору даних найбільш значущу комбінацію ознак. Для пошуку комбінації інформативних ознак в умовах їхньої взаємної залежності одна від одної ефективно може бути використаний еволюційний пошук.

З метою використання еволюційного пошуку для відбору ознак необхідно визначити:

- спосіб подання інформації в хромосомі;
- фітнес-функцію, за допомогою якої виконується оцінювання ефективності хромосом для розв'язуваної задачі.

9.2.1 Кодування хромосом для виділення найбільш значимого набору ознак

Для еволюційного відбору інформативних ознак можуть бути використані такі методи подання інформації про значимості ознак у хромосомі:

– дійсне кодування, при якому кожна ознака подається вагою її значущості стосовно вихідного параметру, що дозволяє оцінити кожен ознаку з огляду її відносної важливості в знайденому інформативному наборі. Такий спосіб кодування хромосоми приводить до скривлення простору ознак, що у свою чергу викликає необхідність розробки способу врахування важливості ознак у фітнес-функції;

– використання хромосом змінної довжини, у яких кожний ген подається у вигляді пари (номер гену; значення гену). Хромосоми можуть бути різної довжини, недовизначені або перевизначені, аналогічно мобільному генетичному методу. Значення гену визначає значущість (вага) відповідної ознаки. При розрахунку інформативності набору ознак, що відповідає оцінюваній хромосомі, враховуються тільки ті ознаки, які подані в хромосомі відповідними генами. Використання дійсних значень генів хромосоми при такому способі подання хромосоми приводить до недоліків, аналогічним попередньому способу кодування;

– застосування негомологічних числових хромосом. При використанні такого способу подання інформації про значущість ознак хромосома містить L_0 генів, які є цілими числами з інтервалу $[1; L]$, де L_0 – максимально припустима кількість ознак в інформативному наборі; L – кількість ознак у вихідному наборі даних. При цьому поява генів однієї хромосоми з однаковими значеннями є неприпустимою, що викликає необхідність розробки спеціальних еволюційних операторів, що не допускають можливості появи некоректних рішень;

– бінарне подання, при якому для відбору інформативних ознак з вихідного масиву, що містить L ознак, за допомогою методів еволюційного пошуку рішення (хромосома) подаються бітовим рядком розміру L . Якщо ген хромосоми приймає одиничне значення, тоді відповідна йому ознака вважається інформативною і враховується при оцінюванні набору ознак, що відповідає хромосомі. У протилежному випадку, коли ген приймає нульове значення, ознака вважається неінформативною і не використовується при оцінці комбінації ознак.

Найбільш ефективним способом подання інформації про значущість ознак у хромосомі є бінарне кодування. Перевага такого подання полягає у тому, що класичні еволюційні оператори схрещування й мутації можуть бути застосовані для відбору ознак без яких-небудь змін. Крім того, бінарне кодування вимагає значно менше ресурсів пам'яті в порівнянні з попередніми способами подання хромосом.

9.2.2 Фітнес-функція

При пошуку найбільш значущої комбінації ознак на основі методів еволюційної оптимізації як фітнес-функції хромосом використовуються критерії, які дозволяють оцінити інформативність набору ознак, що відповідає оцінюваній хромосомі.

Як такі критерії використовуються такі [16, 52, 59, 65]:

- показники ефективності класифікації або прогнозування за моделями, синтезованими на основі оцінюваних комбінацій ознак;
- фільтруючі критерії.

Критерії, що відносяться до першої групи, оцінюють набір ознак за допомогою помилки прогнозування або класифікації за моделлю, побудованою на основі ознак з аналізованого набору. Як синтезовані моделі можуть використовуватися регресійні, нечіткі, нейромережні, нейронечіткі й інші.

У випадку відбору ознак при вирішенні завдання прогнозування як критерії оцінювання інформативності можуть бути використані: середньоквадратична помилка, сума квадратів відхилень, середня абсолютна помилка, сума значень абсолютних відхилень, максимальне абсолютне відхилення, середня відносна помилка, сума відносних відхилень, максимальне відносне відхилення. При відборі ознак для класифікації використовуються імовірність прийняття помилкових рішень і критерій Фішера.

Використання помилок синтезованих моделей для оцінювання інформативності набору ознак є досить ресурсомісткою процедурою, оскільки синтез моделей на основі оцінюваної комбінації ознак займає значно більший час у порівнянні з оцінюванням ознак шляхом застосування фільтруючих критеріїв оцінювання спільного впливу ознак.

Як правило, використання таких критеріїв приводить до кращих результатів у порівнянні з фільтруючими критеріями, оскільки вони орієнтовані на пошук інформативної комбінації ознак для конкретної моделі, що надалі буде застосовуватися на практиці.

Однак це приводить до зменшення гнучкості результатів у вигляді набору інформативних ознак. І, у випадку прийняття рішення про зміну типу моделі, що використовується для опису досліджуваного об'єкту або процесу, необхідно буде запускати метод для повторного пошуку комбінації інформативних ознак, що відповідає новій моделі.

На практиці часто виникають ситуації, коли вихідний набір даних містить надзвичайно великі обсяги інформації. Це приводить до значних обчислювальних і часових витрат на побудову моделі, що описує досліджуваний об'єкт або процес. У результаті побудова моделей і використання їхніх помилок для оцінювання хромосом є неприйнятним.

У таких випадках використовують фільтруючі критерії, які припускають виключення неінформативних ознак з вихідного набору до побудови моделі, що описує досліджуваний об'єкт.

Однією з переваг таких методів оцінювання інформативності є те, що вони не мають потреби в повторному запуску при необхідності синтезу нової моделі по вже відібраних ознаках.

Фільтри є обчислювально більш простими в порівнянні з іншими критеріями й ефективно можуть застосовуватися для відбору інформативних ознак з масивів даних дуже великого розміру.

Однак у результаті використання фільтруючих критеріїв можуть бути отримані такі комбінації ознак, на основі яких не вдасться побудувати модель, що забезпечує необхідну точність. Це викликано тим, що такі критерії безпосередньо не пов'язані з математичною моделлю, що буде використовуватися для опису досліджуваного об'єкта, процесу або системи.

До фільтруючих критеріїв, використовуваних для оцінювання ознак, можуть бути віднесені [5, 52, 60, 67]:

- критерії оцінювання індивідуальної інформативності (коефіцієнт парної кореляції, коефіцієнт кореляції знаків, коефіцієнт кореляції Фехнера, дисперсійне відношення, коефіцієнт зв'язку, інформаційний критерій, теоретико-інформаційний критерій, ентропія ознаки, критерій, заснований на імовірнісному підході, критерій, заснований на статистичному підході);

- критерії оцінювання спільного впливу набору ознак (множинний коефіцієнт кореляції, коефіцієнт кореляції Пірсона, множинне дисперсійне відношення, множинний коефіцієнт зв'язку, інформаційний критерій, ентропія набору ознак, критерій, заснований на статистичному підході).

Очевидно, що при відборі інформативних ознак за допомогою методів еволюційної оптимізації доцільно застосовувати критерії оцінювання спільного впливу ознак.

Розглянуті вище критерії оцінювання інформативності комбінації ознак не враховують кількість відібраних ознак. Тому як фітнес-функцію пропонується використовувати вираз, що мінімізує кількість відібраних ознак і критерій оцінювання інформативності набору ознак:

$$f(H_j) = \left(1 + \frac{1}{L} \sum_{i=1}^L h_{ij} \right) I_j,$$

де I_j – критерій оцінювання спільного впливу набору ознак, що відповідає оцінюваній хромосомі H_j .

Запропонований критерій дозволить забезпечити ефективне оцінювання хромосом з урахуванням інформативності оцінюваної комбінації й кількості ознак, що містяться в ній.

9.2.3 Послідовність еволюційного пошуку інформативного набору ознак

Для вирішення задачі відбору інформативних ознак еволюційний пошук виконується в такій послідовності кроків.

Крок 1. Виконати ініціалізацію початкової популяції. Для цього згенерувати початкові комбінації інформативних ознак у вигляді бінарних хромосом H_j ($j = 1, 2, \dots, N$; де N – кількість згенерованих хромосом) розмірності L (кількість ознак).

Крок 2. Оцінити хромосоми поточної популяції шляхом обчислення фітнес-функції, яка враховує значущість набору ознак, що відповідає оцінюваній хромосомі.

Крок 3. Перевірити умови закінчення пошуку. Як такі умови можуть бути використані: досягнення максимально припустимого часу пошуку, кількості ітерацій, прийнятного значення критерію оцінювання інформативності набору ознак. Якщо критерії закінчення пошуку задовільнено, тоді перейти до кроку 7.

Крок 4. Відібрати найбільш пристосовані особини.

Крок 5. Виконати еволюційні оператори схрещування й мутації над особинами, отриманими на попередньому кроці.

Крок 6. Створити нове покоління хромосом з елітних особин і особин-нащадків, отриманих у результаті застосування схрещування й мутації. Виконати перехід до кроку 2.

Крок 7. Зупинення.

Однак, як відзначено вище, відносно повільна збіжність і залежність від початкових умов класичного еволюційного пошуку приводять до необхідності розробки нових еволюційних методів відбору інформативних ознак, вільних від зазначених недоліків.

Для розробки ефективних еволюційних методів виділення найбільш значущої комбінації ознак пропонується використовувати додаткову інформацію про значущість ознак, що дозволить прискорити пошук за рахунок наближення початкової точки пошуку до оптимальної шляхом генерації на етапі ініціалізації хромосом, що володіють більшою інформативністю в порівнянні з рішеннями, отриманими випадковим чином.

9.2.4 Керування параметрами еволюційного пошуку

Одним з недоліків відомих еволюційних методів є сильна залежність ефективності їхнього функціонування (досягнутого значення фітнес-функції, часу знаходження оптимуму, кількості обчислень фітнес-функції) від параметрів пошуку (розміру популяції, імовірнісних характеристик еволюційної оптимізації і т. п.).

Відомо [8, 14], що не існує оптимальних значень параметрів еволюційного пошуку, що дозволяють ефективно вирішувати всі можливі завдання. Тому при вирішенні практичних завдань, в тому числі завдання відбору інформативних ознак, з усіх можливих значень параметрів необхідно вибрати найбільш оптимальні з урахуванням особливостей розв'язуваного завдання й вхідних даних.

Таким чином, сильна чутливість еволюційних методів до значень параметрів пошуку викликала необхідність розробки різних методів керування параметрами еволюційного пошуку. Однак існуючі підходи [8, 11, 14, 15] до керування параметрами не враховують показників, що характеризують структуру поточної популяції хромосом і ступінь її поліпшення в порівнянні з попередньою популяцією.

З метою усунення недоліків існуючих методів розроблено метод керування параметрами еволюційного пошуку, у якому пропонується аналізувати наступні показники:

- критерії однорідності популяції;
- критерії еволюційної адаптації.

Оцінювання однорідності популяції на різних ітераціях дозволяє робити висновки про рівномірність покриття простору пошуку особинами поточної популяції. Для оцінювання однорідності популяції пропонується використовувати наступні критерії.

1. Коефіцієнт однорідності генотипу t -ої популяції:

$$\lambda_g = \frac{1}{N} \sum_{j=1}^N \frac{1}{1 + D_g(H_j, \bar{H})},$$

де \bar{H} – хромосома, гени якої розраховуються як середні значення генів хромосом поточної популяції; $D_g(H_j, \bar{H})$ – відстань між хромосомами H_j і \bar{H} .

Відстань між хромосомами $D_g(H_j, \bar{H})$ пропонується обчислювати за однією з формул [59, 67]:

- відстань Евкліда:

$$D_g(H_j, \bar{H}) = \sqrt{\sum_{i=1}^L (h_{ij} - \bar{h}_i)^2},$$

де $\bar{h}_i = \frac{1}{N} \sum_{j=1}^N h_{ij}$ – середнє значення i -го гена в поточній популяції;

- відстань Хемінга:

$$D_g(H_j, \bar{H}) = \sum_{i=1}^L |h_{ij} - \bar{h}_i|.$$

2. Коефіцієнт однорідності генотипу-фенотипу t -ої популяції:

$$\lambda_{gf} = \frac{1}{N} \sum_{j=1}^N \frac{1}{1 + D_{gf}(H_j, \bar{H})}.$$

Відстань $D_{gf}(H_j, \bar{H})$ між хромосомами H_j і \bar{H} може бути обчислено різними способами в залежності від розв'язуваного за допомогою еволюційного пошуку завдання. Для завдання відбору інформативних ознак відстань $D_{gf}(H_j, \bar{H})$ пропонується розраховувати за формулою:

$$D_{gf}(H_j, \bar{H}) = \sqrt{\sum_{i=1}^L I_i^2 (h_{ij} - \bar{h}_i)^2},$$

де I_i – значення індивідуальної інформативності i -ої ознаки [5, 52, 60, 67], що відповідає i -му гену хромосоми H_j .

Розроблені критерії однорідності популяції дозволяють оцінювати розмаїтість хромосом у поточному поколінні еволюційного пошуку.

Для усунення неоднорідності популяції пропонується застосовувати оператор інверсії, що змінює порядок розташування генів у хромосомах. Використання такого оператора для популяцій, у яких відсутня розмаїтість хромосом, дозволяє значно розширити область пошуку й збільшити ймовірність знаходження таких сполучень генів, при яких досягаються більш оптимальні в порівнянні зі знайденими значення функції пристосованості.

Оцінювання еволюційної адаптації дозволяє визначити стійкість пошуку, тобто здатність еволюційного методу поліпшувати середні значення цільової функції і не попадати в локальні оптимуми.

Еволюційну адаптацію пропонується оцінювати за допомогою наступних критеріїв.

1. Критерій стабільності t -ої популяції C_t : $C_t = \frac{N_{n,t-1}}{N_t}$,

де $N_{n,t-1}$ – кількість особин поточної t -ої популяції, що перейшли в неї з $(t-1)$ -ої популяції; N_t – загальна кількість особин t -ої популяції.

2. Критерій оцінювання швидкості зміни фітнес-функції $K_{\Delta f}$:

$$K_{\Delta f} = \left| \overline{f_t} - \overline{f_{t-1}} \right|,$$

де $\overline{f_t} = \frac{1}{N_t} \sum_{j=1}^{N_t} f_j$ та $\overline{f_{t-1}} = \frac{1}{N_{t-1}} \sum_{j=1}^{N_{t-1}} f_j$ – середні значення фітнес-функції

на t -ій та $(t-1)$ -ій ітераціях, відповідно.

У випадку появи популяцій з неприйнятними значеннями критеріїв еволюційної адаптації пропонується збільшити ймовірність мутації хромосом,

підвищуючи тим самим імовірність знаходження областей простору пошуку, що відповідають більш оптимальним значенням фітнес-функції.

Таким чином, в основному циклі еволюційного пошуку пропонується використовувати процедуру керування параметрами, що складається з наступних кроків.

Крок 1. Обчислити значення обраного критерію оцінювання однорідності поточної популяції λ_t .

Крок 2. Якщо виконується умова: $\lambda_t > \lambda_{\Pi}$, де λ_{Π} – граничне значення критерію оцінювання однорідності популяції, тоді вважається, що більшість особин поточної популяції розташовуються поблизу однієї або декількох точок простору пошуку, утворюючи неоднорідну популяцію. У випадку невиконання такої умови перейти до кроку 4.

Крок 3. Застосувати з імовірністю p_m ($p_m \in [0,01; 0,5]$) оператор інверсії до близько розташованих особин поточної популяції, виконавши циклічну заміну генів обраних для інверсії хромосом.

Крок 4. Обчислити значення критеріїв еволюційної адаптації C_t і $K_{\Delta,t}$.

Крок 5. Якщо виконується одна з наступних умов: $C_t > C_{\Pi}$ або $K_{\Delta,t} < K_{\Delta,\Pi}$, де λ_{Π} та $K_{\Delta,\Pi}$ – граничні значення критерію стабільності популяції й критерію оцінювання швидкості зміни фітнес-функції, відповідно, тоді виконати перехід до кроку 6. У іншому випадку перейти до виконання кроку 7.

Крок 6. Збільшити ймовірність мутації хромосом на поточній ітерації за правилом:

$$p_m(t) = \rho \cdot p_m(0),$$

де ρ – коефіцієнт збільшення ймовірності мутації, $1 < \rho < (p_m(0))^{-1}$; $p_m(0)$ – ймовірність мутації, задана користувачем на етапі ініціалізації параметрів еволюційної оптимізації.

Крок 7. Зупинення процедури керування параметрами еволюційного пошуку.

Розроблений метод керування параметрами еволюційного пошуку з оцінкою однорідності популяції й еволюційної адаптації дозволяє підвищити ефективність пошуку шляхом використання одержуваної в процесі пошуку інформації про структуру популяції хромосом та її ефективності для вирішення розв'язуваного завдання.

9.2.5 Еволюційний метод з фіксацією частини простору пошуку

Істотним недоліком практичного використання класичного еволюційного пошуку для відбору ознак є значні часові витрати. При відборі інформативних ознак за допомогою еволюційного підходу основна частина часу витрачається на оцінювання фітнес-функції хромосом.

Тому зменшення часових витрат на обчислення значень фітнес-функції хромосом дозволить значно скоротити час, необхідний на пошук найбільш інформативної комбінації ознак.

Для підвищення ефективності еволюційних методів при виділенні найбільш значимого набору ознак з масивів даних, що характеризуються великою кількістю ознак, доцільно на етапі ініціалізації аналізувати індивідуальну інформативність ознак шляхом застосування традиційних методів оцінювання інформативності, після чого виключати з подальшого розгляду малозначущі ознаки, зменшуючи тим самим простір пошуку [36, 83].

Теорема. Зменшення простору пошуку в φ разів при використанні еволюційних методів для відбору інформативних ознак скорочує час, що витрачається на пошук, не менш, ніж в φ^n разів, $n > 1$.

Доведення. Нехай $\mu \in [0; 1)$ – коефіцієнт зменшення простору пошуку, L – кількість ознак у вихідному наборі даних, N – кількість особин у популяції, T – максимальна кількість ітерацій роботи еволюційного методу.

Тоді на етапі ініціалізації класичного еволюційного пошуку випадковим чином генерується N хромосом, що складаються з L генів кожна. При цьому середня кількість одиничних генів (ознак, що вважаються інформативними) для кожної хромосоми складе $K1_{\text{клас}} = L/2$.

У випадку скорочення простору пошуку шляхом виключення з подальшого розгляду $(1 - \mu) \cdot L$ малозначущих ознак, середня кількість одиничних генів $K1_c$ у хромосомі зменшується пропорційно коефіцієнту зменшення простору пошуку μ : $K1_c = \mu L/2$.

Час виконання еволюційного пошуку t може бути оцінений виходячи з формули:

$$O(t) = N \cdot T \cdot O(f),$$

де $O(f)$ – оцінка часу, необхідного для обчислення значення фітнес-функції однієї хромосоми.

Незалежного від використовуюваного критерію для оцінювання хромосом при еволюційному відборі інформативних ознак оцінка $O(f)$ пропорційна середньому значенню кількості одиничних генів у хромосомі, тобто:

$$O(f) \sim O(K1),$$

де оцінка $O(K1)$ відображає залежність часу, необхідного для оцінювання однієї хромосоми, від середньої кількості одиничних генів у хромосомі. Як показано раніше, як оцінки хромосоми при відборі ознак можуть використовуватися помилки моделей, відповідних оцінюваній хромосомі, або інші критерії, що характеризують спільний вплив комбінації ознак на вихідний параметр.

Час, необхідний для оцінювання хромосоми за допомогою помилки моделі, може бути визначений як сума часу синтезу моделі t_c і часу розрахунку помилки побудованої моделі t_{po} . У випадку використання найпростішої моделі у вигляді багатовимірної лінійної регресії для розрахунку її

параметрів буде необхідний час $t_c = O(K1^3)$. Очевидно, що при використанні більш складних моделей час на їхній синтез буде більшим, ніж $O(K1^3)$. Час, необхідний для розрахунку помилки моделі t_{po} , як правило, значно менший часу синтезу t_c , тому часом t_{po} можна знехатити. Отже, у випадку використання помилок прогнозування або класифікації відповідних моделей при оцінюванні хромосом, час, необхідний для обчислення значення фітнес-функції однієї хромосоми може бути оцінений в такий спосіб: $O(f) = O(K1^n)$, $n \geq 3$.

При оцінюванні хромосом за допомогою критеріїв, що не потребують побудови моделі, як правило, виконують розрахунок індивідуальної значущості кожної ознаки стосовно вихідного параметра й інших ознак в оцінюваному наборі. Тому в цьому випадку $O(f) = O(K1 + (K1/2) \cdot (K1 - 1))$ або $O(f) = O(K1^2)$.

Очевидно, що в обох випадках $O(\mu K1) = O(\mu) \cdot O(K1)$ або $O((\mu K1)^n) = O(\mu^n) \cdot O(K1^n)$.

Таким чином, оцінка часу $O(t)$, необхідного для еволюційного відбору інформативних ознак, може бути здійснена за формулою:

$$O(t) = N \cdot T \cdot O(K1^n).$$

Коефіцієнт зменшення часу γ , необхідного для виконання еволюційного відбору інформативних ознак, може бути розрахований у такий спосіб:

$$\gamma = \frac{O(t)_c}{O(t)_{\text{клас.}}} = \frac{N \cdot T \cdot O(K1_c)}{N \cdot T \cdot O(K1_{\text{клас.}})} = \frac{O((\mu L / 2)^n)}{O((L / 2)^n)} = \frac{O(\mu^n) O((L / 2)^n)}{O((L / 2)^n)} = O(\mu^n),$$

де $O(t)_c$ та $O(t)_{\text{клас.}}$ – оцінка часу, необхідного для виконання відбору інформативних ознак за допомогою класичного еволюційного пошуку та еволюційного методу з фіксацією частини простору пошуку, відповідно.

Отже, при скороченні простору пошуку в $\varphi = 1 / \mu$ раз час, необхідний на еволюційний пошук найбільш інформативної комбінації ознак, зменшується не менш, ніж в μ^n раз, де показник ступеню $n > 1$ – залежить від методу оцінювання хромосом. Теорему доведено. ν

Важливо відзначити вплив еволюційних операторів схрещування й мутації, що формують нові рішення, на зміну середньої кількості одиничних генів у хромосомі. За допомогою зазначених операторів у процесі еволюційного пошуку хромосоми сходяться до оптимального рішення. При цьому середня кількість одиничних генів у хромосомі прямує до кількості ознак у максимально значущій комбінації, розмір якої може бути більшим, ніж $L/2$. Зменшення простору пошуку не приводить до збільшення кількості ознак в оптимальному наборі, оскільки на етапі ініціалізації виключаються малозначущі ознаки. Тому середня кількість одиничних генів у хромосомі в результаті застосування схрещування й мутації в еволюційному методі зі скороченням простору пошуку може збільшуватися, але не більш, ніж при використанні класичного еволюційного пошуку.

На основі наведених вище міркувань розроблено еволюційний метод з фіксацією частини простору пошуку [36].

Крок 1. Оцінити індивідуальну інформативність I_i кожної ознаки з вихідного набору даних.

Для оцінки індивідуального впливу i -ої ознаки на значення вихідного параметру доцільно використовувати такі критерії:

- для ознак, що мають дійсні значення, – коефіцієнт парної кореляції [52];
- для дискретних ознак – коефіцієнт кореляції Фехнера або коефіцієнт кореляції знаків [67].

Крок 2. Видалити з вихідного набору малоінформативні ознаки, скоротивши в такий спосіб простір пошуку.

Крок 2.1. Розрахувати середнє значення оцінок індивідуальної інформативності ознак I_c за формулою:

$$I_c = \frac{1}{L} \sum_{i=1}^L I_i .$$

Крок 2.2. Визначити коефіцієнт зменшення простору пошуку μ , використовуючи вираз:

$$\mu = \frac{1}{L} \sum_{i=1}^L \beta_i ,$$

$$\text{де } \beta_i = \begin{cases} 1, & \text{якщо } I_i < I_c; \\ 0, & \text{якщо } I_i \geq I_c. \end{cases}$$

При цьому величина $\sum_{i=1}^L \beta_i$ визначає кількість ознак, що володіють індивідуальною інформативністю, нижчою за середню.

Крок 2.3. Упорядкувати ознаки за зростанням індивідуальної інформативності.

Крок 2.4. Виключити з вихідного набору перші $\alpha\mu L$ ознак, що мають незначну інформативність. При цьому α – коефіцієнт, що задається користувачем і визначає ступінь зменшення простору пошуку, $0 < \alpha < 1/\mu$.

Крок 3. Виконати еволюційний пошук з фіксацією частини простору пошуку, при якому найбільш значуща комбінація ознак шукається серед $(1 - \alpha\mu)L$ залишених після виконання попереднього кроку ознак.

Крок 3.1. Встановити лічильник ітерацій (часу): $t = 0$. Ініціалізувати початкову популяцію N хромосомами розмірністю $(1 - \alpha\mu)L$.

Крок 3.2. Обчислити значення фітнес-функції кожної хромосоми початкової популяції шляхом оцінювання комбінації ознак, що вважаються інформативними для оцінюваної хромосоми.

Крок 3.3. Виконати перевірку умов закінчення пошуку. У випадку, якщо такі умови виконуються, тоді перейти до кроку 4. В іншому випадку виконати перехід до кроку 3.4.

Крок 3.4. Збільшити лічильник ітерацій (часу): $t = t + 1$.

Крок 3.5. Вибрати частину хромосом популяції для схрещування.

Крок 3.6. Сформувати батьківські пари.

Крок 3.7. Схрестити обрані батьківські особини.

Крок 3.8. Виконати оператор мутації.

Крок 3.9. Обчислити значення фітнес-функції нових особин у популяції.

Крок 3.10. Сформувати нове покоління хромосом.

Крок 3.11. Перейти до кроку 3.3.

Крок 4. Зупинення.

Таким чином, зменшення простору пошуку за рахунок виключення з розгляду малозначущих ознак підвищує ймовірність знаходження більш інформативної комбінації ознак і скорочує час, необхідний на пошук.

9.2.6 Методи еволюційного пошуку з використанням апріорної інформації про значущість ознак

Запропонований еволюційний метод з фіксацією частини простору пошуку дозволяє ефективно виділити комбінацію інформативних ознак з наборів даних, що характеризуються великою кількістю ознак. При цьому недоліком такого методу є те, що на етапі ініціалізації можуть бути виключені ознаки, що володіють незначною індивідуальною інформативністю, але в сполученні з іншими ознаками утворюють оптимальний набір. Тому для усунення наведеного недоліку пропонується використовувати апріорну інформацію про значущість ознак у всіх еволюційних операторах, які генерують нові рішення.

9.2.6.1 Еволюційний пошук з групуванням ознак на основі індивідуальних оцінок значущості ознак

У методі еволюційного пошуку з групуванням ознак [95] пропонується аналізувати інформативність ознак на етапі ініціалізації параметрів шляхом застосування неітеративних методів оцінювання індивідуальної інформативності ознак [52, 60, 67]. При цьому показники індивідуальної інформативності ознак і тісноти зв'язку ознак між собою використовуються не тільки на етапі ініціалізації, але й на всіх наступних етапах еволюційної оптимізації.

У розробленому методі еволюційного пошуку з групуванням ознак виконані модифікації основних еволюційних операторів, що породжують нові рішення таким чином, щоб при генерації нових хромосом враховувалася індивідуальна інформативність ознак, розрахована на етапі ініціалізації.

Ініціалізація (генерація хромосом початкової популяції) може бути виконана в такий спосіб.

Крок 1. Установити лічильник згенерованих хромосом: $j = 0$.

Крок 2. Розрахувати ймовірність P_i включення i -ої ознаки в хромосому.

Крок 2.1. Визначити показники індивідуальної інформативності ознак I_i і тісноти зв'язку ознак d_{ik} , $i, k = \overline{1, L}$, де L – кількість ознак.

Крок 2.1.1. Розрахувати I_i – оцінку індивідуальної інформативності i -ої ознаки, обумовлену тісністю зв'язку i -ої ознаки навчальної вибірки із прогнозованим параметром u .

Крок 2.1.2. Розрахувати d_{ik} – оцінку тісноти зв'язку i -ої й k -ої ознак. Для обчислення оцінки d_{ik} необхідно визначити індивідуальну інформативність i -ої ознаки стосовно k -ої ознаки.

Крок 2.2. Виконати групування ознак.

Крок 2.2.1. Встановити: $q_i = 0$, $i = \overline{1, L}$, $N_g = 0$, де q_i – прапорець обробки i -ої ознаки ($q_i = 1$, якщо ознака оброблена, і $q_i = 0$ – в іншому випадку), N_g – кількість груп ознак.

Крок 2.2.2. Визначити центр групи ознак за формулою:

$$c = \arg \max_{q_i=0, i=\overline{1, L}} I_i.$$

Установити: $g_c = 1$, $g_c = c$, $N_g = N_g + 1$, де g_i – номер ознаки, що є центром групи для i -ої ознаки.

Крок 2.2.3. Обчислити d_{cp} – середнє значення тісноти зв'язку ознак d_{ip} :

$$d_{cp} = \frac{2}{L(L-1)} \sum_{i=1}^L \sum_{k=i+1}^L d_{ik}.$$

Крок 2.2.4. Визначити ознаки, для яких виконуються умови: $d_{ic} \geq d_{cp}$ та $q_i = 0$.

Крок 2.2.5. Для всіх знайдених на попередньому кроці ознак встановити $g_i = c$ та $q_i = 1$.

Крок 2.2.6. Якщо є хоча б одна ознака, для якої $q_i = 0$, тоді перейти до кроку 2.2.2.

Крок 2.3. Розрахувати ймовірність включення i -ої ознаки в хромосому за формулою:

$$P_i = I_i(1 - I_{g_i} + I_i).$$

Крок 3. Якщо $j > N$, де N – кількість хромосом початкової популяції, тоді перейти до кроку 7.

Крок 4. Згенерувати випадкове число $\text{rand} \in [0; 1]$.

Крок 5. Ініціалізувати j -ту хромосому за формулою:

$$h_{ij} = \begin{cases} 1, & \text{якщо } \text{rand} \geq P_i; \\ 0, & \text{якщо } \text{rand} < P_i, \end{cases}$$

де h_{ij} – i -ий ген j -ої хромосоми; P_i – імовірність включення i -ої ознаки в хромосому, що враховує її інформативність, а також інформативність її центру групи.

Крок 6. Встановити: $j = j + 1$. Перейти до виконання кроку 3.

Крок 7. Зупинення.

В операторі відбору крім врахування інформативності оцінюваного набору ознак, доцільно також враховувати середню індивідуальну інформативність включених у хромосому ознак. Для цього пропонується використовувати таку формулу для розрахунку фітнес-функції:

$$f'(H_j) = \frac{f(H_j) \sum_{i=1}^L h_{ij}}{1 + \sum_{i=1}^L I_i h_{ij}},$$

де $f(H_j)$ – значення критерію, що враховує кількість відібраних ознак і інформативність набору ознак, що відповідає хромосомі H_j .

В операторі рівномірного схрещування апріорна інформація про значущість ознак може бути використана в такий спосіб. Спочатку визначити граничне значення інформативності ознак. Потім створити маску схрещування, установивши одиничні значення для тих ознак, оцінки інформативності яких вище порога, і нульові значення для ознак, оцінки інформативності яких рівні або нижче порога. З метою збільшення розмаїтості популяції після формування маски схрещування можна випадковим чином змінити в ній деякі розряди.

В операторі мутації апріорні відомості про рівень інформативності ознак можна використовувати за допомогою посилення ймовірності мутації ознак, що мають низькі оцінки індивідуальної інформативності й ослаблення ймовірності мутації ознак з високими оцінками індивідуальної інформативності. Це дозволить закріпити ознаки, що найбільш сильно впливають на цільовий параметр, і сконцентрувати пошук на переборі сполучень ознак з меншою інформативністю.

Імовірність мутації пропонується розраховувати за однією з таких формул: – імовірнісну мутацію:

$$P_{Mi} = \alpha (1 - P_i),$$

де P_{Mi} – імовірність мутації i -го гену в хромосомі; α – коефіцієнт ступеню мутації, $\alpha \in [0; 1]$;

– радикальну мутацію:

$$P_{Mi} = \begin{cases} 0, & \text{якщо } g_i = i; \\ \alpha, & \text{якщо } g_i \neq i. \end{cases}$$

Розроблений еволюційний метод з групуванням ознак на основі індивідуальних оцінок значущості ознак підвищує ефективність пошуку за

допомогою наближення початкової точки пошуку до оптимальної й використання показників індивідуальної інформативності ознак на всіх етапах еволюційної оптимізації.

9.2.6.2 Еволюційний метод з кластеризацією ознак

Класичні методи еволюційного пошуку при відборі ознак не враховують близькості розташування ознак у просторі екземплярів, в результаті чого нові комбінації ознак (хромосоми), що отримані шляхом застосування еволюційних операторів ініціалізації, схрещування й мутації, можуть містити в собі ознаки, які містять однакову інформацію про досліджуваний об'єкт, процес або систему. Очевидно, що набори ознак, які відповідають таким хромосомам, є малоінформативними або надлишковими.

У розробленому методі еволюційного пошуку з кластеризацією ознак пропонується групувати схожі ознаки за допомогою методів кластеризації [65], які дозволяють розбивати вибірку на групи компактно розташованих ознак у просторі екземплярів (кластери, факторні групи) і виділяти в кожному кластері по одній найбільш типовій ознаці.

При формуванні нових хромосом в результаті застосування еволюційних операторів ініціалізації, схрещування й мутації пропонується розраховувати ймовірність включення ознаки в хромосому, що залежить від розташування ознаки в кластері (відстані від неї до центру кластера), індивідуальної інформативності ознаки, а також індивідуальної інформативності центру її кластеру.

Еволюційний пошуку з кластеризацією ознак пропонується виконувати як таку послідовність кроків.

Крок 1. Згрупувати ознаки вихідної вибірки даних у кластери.

Крок 1.1. Для кожної ознаки X_i розрахувати Евклідову відстань від неї до всіх інших ознак у вибірці. Евклідова відстань $d(X_a; X_b)$ між ознаками X_a і X_b обчислюється за формулою:

$$d_E(X_a; X_b) = \sqrt{\sum_{p=1}^m (x_{pa} - x_{pb})^2},$$

де x_{pa} та x_{pb} – значення a -ої й b -ої ознак p -го екземпляру навчальної вибірки, відповідно.

Крок 1.2. На основі розрахованих раніше відстаней між екземплярами, використовуючи методи кластера-аналізу [65], сформувані групи ознак, компактно розташованих у просторі екземплярів. Виділити ознаки, що є центрами кластерів.

Крок 1.3. Для кожної ознаки X_i обчислити ймовірність її включення в хромосому.

Крок 1.3.1. Розрахувати значення індивідуальної інформативності I_i ознаки X_i .

Крок 1.3.2. Визначити ймовірність P_i включення i -ої ознаки в хромосому:

$$P_i = I_i + \frac{d_E(X_i; X_{c,i})}{d_{E \max, c}} (I_i - I_c),$$

де $d(X_i; X_{c,i})$ – відстань від ознаки X_i до центру її кластера; $d_{\max, c}$ – максимальна відстань у кластері, в якому розташована i -та ознака; I_c – інформативність ознаки, що є центром кластеру, в якому розташована ознака X_i .

Крок 2. Встановити лічильник ітерацій (часу): $t = 0$.

Крок 3. Ініціалізувати початкову популяцію з N хромосом.

Крок 3.1. Встановити лічильник сформованих хромосом: $j = 1$.

Крок 3.2. Сформувати j -ту хромосому H_j .

Крок 3.2.1. Встановити лічильник визначених генів: $i = 1$.

Крок 3.2.2. Згенерувати випадкове число: $r = \text{rand}[0; 1]$.

Крок 3.2.3. Якщо $P_i > r$, тоді i -му гену j -ої хромосоми привласнити одиничне значення: $h_{ij} = 1$, у противному випадку: $h_{ij} = 0$.

Крок 3.2.4. Якщо j -та хромосома сформована повністю ($i = L$), тоді виконати перехід до кроку 3.3.

Крок 3.2.5. Встановити: $i = i + 1$.

Крок 3.2.6. Перейти до кроку 3.2.2.

Крок 3.3. Якщо сформовані всі хромосоми ($j = N$), тоді виконати перехід до виконання кроку 4.

Крок 3.4. Встановити: $j = j + 1$.

Крок 3.5. Перейти до кроку 3.2.

Крок 4. Обчислити значення фітнес-функції $f'(H_j)$ хромосом поточної популяції:

$$f'(H_j) = \frac{f(H_j) \sum_{i=1}^L h_{ij}}{\left(1 + \sum_{i=1}^L I_i h_{ij}\right) \left(1 + \sum_{i=1}^L P_i h_{ij}\right)}.$$

Крок 5. Виконати перевірку критеріїв зупинення. Якщо критерії завершення пошуку задовільнено, тоді виконати перехід до кроку 11.

Крок 6. Збільшити лічильник ітерацій: $t = t + 1$.

Крок 7. Вибрати хромосоми для схрещування й мутації.

Крок 8. Виконати оператор рівномірного схрещування. При цьому в масі схрещування встановити одиничні значення для генів, яким відповідають ознаки з імовірністю включення в хромосому, вищою за середню, іншим генам привласнити нульові значення.

Крок 9. Виконати оператор точкової мутації. Імовірність мутації P_{Mi} i -го гена в мутуючій хромосомі пропонується розраховувати за формулою, що використовується в методі еволюційного пошуку з групуванням ознак на основі індивідуальних оцінок значущості ознак.

Крок 10. Сформувати нове покоління. Виконати перехід до кроку 4.

Крок 11. Зупинення.

У запропонованому методі еволюційного пошуку з кластеризацією ознак враховується близькість розташування ознак у просторі екземплярів, що дозволяє формувати нові рішення з ознак, розташованих, як правило, у різних групах, збільшуючи ймовірність відшукування комбінації ознак, що володіє максимальною інформативністю.

9.2.6.3 Острівна модель еволюційного пошуку з урахуванням апіорної інформації про значущість ознак

Пропонована острівна модель еволюційного пошуку для відбору інформативних ознак на основі апіорної інформації про їхню значущість передбачає використання двох островів, на одному з яких іде розвиток популяції з урахуванням інформації про значущість ознак за допомогою використання розробленого еволюційного методу з групуванням ознак. На другому острові використовується еволюційний пошук з кластеризацією ознак. При цьому через задану як параметр методу t_m кількість ітерацій (інтервал міграцій) відбуваються міграції особин з одного острова на іншій.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Ініціалізувати популяцію кожного острова з використанням апіорної інформації про значущість ознак у послідовностях, що використовуються у методах, описаних вище.

Крок 2. Виконати перевірку умов закінчення пошуку. У випадку задоволення таких умов, виконати перехід до кроку 7.

Крок 3. Встановити: $t = t + 1$. Сформувати нові покоління хромосом на кожному острові.

Крок 4. Якщо (Залишок $(t / t_m) = 0$), де t_m – інтервал міграцій, тоді провести міграцію особин.

Крок 5. Провести еволюційний пошук на поточній популяції в кожному з островів, застосувавши при цьому модифіковані оператори відбору, схрещування й мутації, запропоновані в раніше описаних методах.

Крок 6. Виконати перехід до кроку 2.

Крок 7. Зупинення.

Таким чином, розроблена острівна модель еволюційного пошуку, на відміну від класичного острівного еволюційного методу [1], використовує апіорну інформацію про значущість ознак і дозволяє організувати паралельний пошук по двох різних стратегіях, внаслідок чого підвищується ймовірність знаходження глобального оптимуму.

9.3 Параметричний синтез нейромереж

У випадку, якщо структура нейромережної моделі визначена, тоді після відбору інформативних ознак виконується параметричний синтез нейромоделі.

Вище сказано, що при настроюванні значень вагових коефіцієнтів нейромоделей заданої архітектури, як правило, застосовуються градієнтні методи, які, однак, пов'язані з необхідністю обчислення значень цільової функції й не можуть бути застосовані для пошуку оптимальних значень синаптичних ваг у нейромережах, що містить нейрони з недиференційованими функціями активації. Крім того, такі методи схильні до влучення в локальні оптимуми й, отже, не дозволяють знайти глобальний оптимум полімодальних функцій.

Методи еволюційної оптимізації, на відміну від градієнтних методів, є методами глобального пошуку й не використовують значення похідних цільової функції в процесі пошуку. Таким чином, при параметричному синтезі нейромоделей у випадках, коли градієнтні методи не можуть бути застосовані по різних причинах, доцільним є застосування еволюційних методів для пошуку оптимальних значень вагових коефіцієнтів нейромережних моделей.

Незважаючи на те, що еволюційна оптимізація може виконуватися довше в порівнянні із градієнтними методами, вона, у загальному випадку, є значно менш чутливою до початкових параметрів навчання. Методи еволюційного пошуку завжди намагаються знайти глобальний оптимум, у той час, як градієнтні методи, як правило, знаходять локальний оптимум, розташований в околі початкової точки пошуку.

Для застосування еволюційного пошуку до параметричного синтезу нейромереж необхідно: визначити цільову функцію й вибрати спосіб подання значень ваг у хромосомі.

При виборі фітнес-функції для параметричного синтезу нейромоделей враховують два фактори:

- помилка між реальним і модельним виходом мережі;
- складність нейромоделі.

Як правило, як цільова функція при параметричному синтезі нейромоделей використовується функція помилки, що обчислюється як середня різниця між реальним і модельним виходом мережі [91, 105].

Хромосома при параметричному синтезі (рис. 9.1) складається з K генів, що містять значення ваг і зсувів всіх нейронів мережі [49]. При цьому для подання значень вагових коефіцієнтів у хромосомах застосовується дійсне кодування.

Розмір хромосоми визначається за формулою:

$$K = N_1(L + 1) + \sum_{\mu=1}^M N_{\mu}(N_{\mu-1} + 1),$$

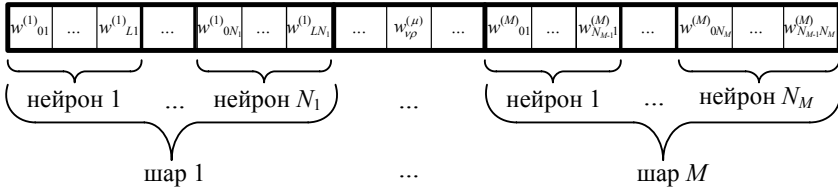


Рисунок 9.1 – Схематичне подання хромосоми при параметричному синтезі нейромоделей

де N_{μ} – кількість нейронів на μ -ому шарі; L – кількість ознак у навчальній вибірці; M – кількість шарів нейромережі.

Еволюційна оптимізація синаптичних ваг нейромереж може бути виконана в такій послідовності.

Крок 1. Виконати ініціалізацію початкової популяції хромосомами, що містять інформацію про значення вагових коефіцієнтів мережі заданої структури.

Крок 2. Оцінити хромосоми поточної популяції.

Крок 2.1. Декодувати кожну хромосому популяції в набір вагових коефіцієнтів нейронної мережі.

Крок 2.2. Побудувати нейромережі, що відповідають оцінюваним хромосомам.

Крок 2.3. Обчислити значення фітнес-функції оцінюваних хромосом, що враховує помилку й складність мережі.

Крок 3. Перевірити критерії закінчення пошуку (досягнення прийнятної помилки синтезованої нейромережної моделі, перевищення максимально допустимої кількості ітерацій, перевищення допустимого часу функціонування методу). У випадку, якщо критерії зупинення задоволено, виконати перехід до кроку 7.

Крок 4. Виходячи зі значення фітнес-функції, вибрати особини для генерації нових рішень.

Крок 5. Застосувати оператори схрещування й мутації для хромосом, відібраних на попередньому кроці.

Крок 6. Сформувати нове покоління з елітних хромосом і хромосом-нащадків, отриманих шляхом застосування схрещування й мутації. Перейти до виконання кроку 2.

Крок 7. Зупинення.

Таким чином, навчання нейромоделей, засноване на еволюційному підході, не має потреби в обчисленні градієнта цільової функції й дозволяє знайти значення глобальних оптимумів синаптичних ваг для багатовимірних, полімодальних і недиференційованих цільових функцій.

Ще однією перевагою еволюційного пошуку при параметричному синтезі нейромоделей є можливість застосування одного методу синтезу до побудови різних моделей нейромереж (прямого поширення, рекурентних і ін.).

Однак при використанні класичних методів еволюційної оптимізації [18, 39] хромосоми початкової популяції формуються випадковим чином, тобто початкова популяція хромосом являє собою набір нейронних мереж зі значеннями вагових коефіцієнтів і зсувів, згенерованими випадково. Випадкове створення хромосом початкової популяції при параметричному синтезі нейромоделей приводить до зменшення ефективності процесу еволюційної оптимізації при пошуку оптимальних значень матриці ваг і збільшенню часу, необхідного для пошуку.

У розробленому еволюційному методі параметричного синтезу нейромоделей [74] пропонується враховувати апріорну інформацію про значущість ознак в операторах ініціалізації й мутації, що використовуються при еволюційному пошуку.

Створення нейромереж при ініціалізації хромосом початкової популяції пропонується здійснювати за аналогією з методом Нгуена-Відроу [105], що передбачає обчислення початкових значень вагових коефіцієнтів і зсувів нейронів у шарі таким чином, щоб рівномірно розподілити активну область визначення функції активації кожного нейрона по простору вхідних змінних.

Під активною областю визначення функції активації нейрона мається на увазі обмежена область значень вхідних параметрів, у якій спостерігається істотна зміна значень функції активації.

Для логістичної сигмоїдної функції використовується інтервал $[-4; 4]$ як активна область визначення, при цьому функція приймає значення в інтервалі $(0,018; 0,982)$, що становить 96,4 % від всієї області значень. Для тангенційної сигмоїдної і радіально-базисної функцій як активну область визначення запропоновано інтервал $[-2; 2]$, у якому зазначені функції приймають значення в інтервалах $(-0,964; 0,964)$ і $(0,0183; 1]$, відповідно.

Для порогових і лінійних функцій активації активні області не визначені, внаслідок чого метод Нгуена-Відроу для нейронів, що мають такі функції активації, припускає випадкове обчислення значень вагових коефіцієнтів і зсувів.

Авторами в [74] як активні області визначення для таких функцій пропонується використовувати максимальні інтервали значень входів нейроелементів.

Однак при ініціалізації параметрів нейромережних моделей за допомогою методу Нгуена-Відроу вважається, що вхідні ознаки мають однакову значимість. Відомо [52, 67], що при рішенні реальних практичних завдань інформативність ознак навчальної вибірки не є однаковою, внаслідок

док чого використання методу Нгуена-Відроу для ініціалізації матриці вагових коефіцієнтів нейромережної моделі є недостатньо ефективним.

Тому в розробленому еволюційному методі параметричного синтезу нейромереж пропонується проводити ініціалізацію хромосом початкової популяції за допомогою модифікованого методу Нгуена-Відроу, що враховує апіорну інформацію про значущість ознак при ініціалізації нейронної мережі.

У еволюційному методі параметричного синтезу нейромереж [74] показники індивідуальної інформативності ознак пропонується також використовувати в операторі точкової мутації шляхом зниження ймовірності мутації генів, яким відповідають значення вагових коефіцієнтів зв'язків, що йдуть від вхідних ознак з оцінками індивідуальної значущості, вищої за середню.

Запропонований еволюційний метод параметричного синтезу нейромережних моделей з використанням апіорної інформації може бути поданий у вигляді такої послідовності кроків.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$.

Крок 2. Для кожної ознаки розрахувати значення оцінки її індивідуальної інформативності $I_i \in [0; 1]$. Зокрема, такими оцінками можуть виступати модуль коефіцієнту парної кореляції, коефіцієнт кореляції знаків, коефіцієнт кореляції Фехнера, дисперсійне відношення, коефіцієнт зв'язку, інформаційний критерій, ентропія ознаки [52, 59, 67].

Виконати ініціалізацію початкової популяції N хромосомами H_j (кроки 3–6), що містять інформацію про значення вагових коефіцієнтів мережі заданої структури.

Крок 3. Встановити лічильник згенерованих хромосом: $j = 1$.

Крок 4. Згенерувати j -ту хромосому, виконавши кроки 4.1–4.12.

Крок 4.1. Встановити лічильник шарів нейромережі, що відповідає j -ій хромосомі початкової популяції: $\mu = 1$.

Крок 4.2. Обчислити значення коефіцієнту α , що визначає ширину діапазону зміни значень вагових коефіцієнтів нейроелементів, за формулою:

$$\alpha = \begin{cases} \gamma \sqrt[\mu]{N_\mu}, & \text{якщо } \mu = 1; \\ \gamma^{N_{\mu-1}} \sqrt[N_\mu]{N_\mu}, & \text{якщо } \mu \neq 1, \end{cases}$$

де γ – коефіцієнт, що задається користувачем, $\gamma \in (0; 1)$, за замовчуванням пропонується встановлювати: $\gamma = 0,7$.

Крок 4.3. Обчислити кількість входів V_μ μ -го шару нейромережі:

$$V_\mu = \begin{cases} L, & \text{якщо } \mu = 1; \\ N_{\mu-1}, & \text{якщо } \mu \neq 1. \end{cases}$$

Крок 4.4. Визначити мінімальне $x_{v \min}^{(\mu)}$ та максимальне $x_{v \max}^{(\mu)}$ значення v -го входу нейронів μ -го шару мережі:

$$x_{v \min}^{(\mu)} = \begin{cases} x_{v \min}, & \text{якщо } \mu = 1; \\ \Psi_{v \min}^{(\mu-1)}, & \text{якщо } \mu \neq 1, \end{cases} \quad x_{v \max}^{(\mu)} = \begin{cases} x_{v \max}, & \text{якщо } \mu = 1; \\ \Psi_{v \max}^{(\mu-1)}, & \text{якщо } \mu \neq 1, \end{cases}$$

де $x_{v \min}$ та $x_{v \max}$ – мінімальне й максимальне значення v -ої ознаки навчальної вибірки; $\Psi_{v \min}^{(\mu-1)}$ та $\Psi_{v \max}^{(\mu-1)}$ – мінімально й максимально можливі значення функції активації v -го нейрона ($\mu - 1$ -го шару).

Крок 4.5. Для кожного ρ -го нейрона μ -го шару визначити мінімальне $x_{\rho \text{ акт. min}}^{(\mu)}$ та максимальне $x_{\rho \text{ акт. max}}^{(\mu)}$ значення активної області визначення функції активації.

Крок 4.6. Встановити лічильник нейронів μ -го шару: $\rho = 1$.

Крок 4.7. Згенерувати ваги зв'язків для ρ -го нейрона μ -го шару.

Крок 4.7.1. Встановити лічильник входів ρ -го нейрона μ -го шару: $v = 1$.

Крок 4.7.2. Згенерувати випадкове число r :

$$r = \begin{cases} \text{rand}[-I_v; I_v], & \text{якщо } \mu = 1; \\ \text{rand}[-1; 1], & \text{якщо } \mu \neq 1, \end{cases}$$

де I_v – значення оцінки індивідуальної інформативності v -ої ознаки навчальної вибірки; $\text{rand}[a; b]$ – випадково згенероване число в інтервалі $[a; b]$.

Крок 4.7.3. Обчислити значення v -го вагового коефіцієнту ρ -го нейрона μ -го шару:

$$w_{\rho v}^{(\mu)} = \alpha r \frac{x_{\rho \text{ акт. max}}^{(\mu)} - x_{\rho \text{ акт. min}}^{(\mu)}}{x_{v \max}^{(\mu)} - x_{v \min}^{(\mu)}}.$$

Крок 4.7.4. Встановити: $v = v + 1$.

Крок 4.7.5. Перевірити, чи розраховані значення всіх вагових коефіцієнтів ρ -го нейрона μ -го шару ($v > I_\mu$) нейромережі. У випадку, якщо умова $v > I_\mu$ виконується, тоді виконати перехід до кроку 4.8; у протилежному випадку – перейти до кроку 4.7.2.

Крок 4.8. Обчислити значення зсуву $w_{0\rho}^{(\mu)}$ для ρ -го нейрона μ -го шару за формулою:

$$w_{0\rho}^{(\mu)} = \sum_{v=1}^{I_\mu} \frac{x_{v \max}^{(\mu)} + x_{v \min}^{(\mu)}}{x_{\rho \text{ акт. max}}^{(\mu)} - x_{\rho \text{ акт. min}}^{(\mu)}} w_{\rho v}^{(\mu)} + b_\rho,$$

$$\text{де } b_\rho = \begin{cases} \frac{1}{2}(x_{\rho \text{ акт. max}}^{(\mu)} + x_{\rho \text{ акт. min}}^{(\mu)}) + \frac{1}{2}\alpha(x_{\rho \text{ акт. max}}^{(\mu)} - x_{\rho \text{ акт. min}}^{(\mu)}) \left(-1 + \frac{2(\rho-1)}{N_\mu - 1} \right), & \text{якщо } N_\mu \neq 1; \\ \frac{1}{2}(x_{\rho \text{ акт. max}}^{(\mu)} + x_{\rho \text{ акт. min}}^{(\mu)}) & \text{якщо } N_\mu = 1. \end{cases}$$

Крок 4.9. Встановити: $\rho = \rho + 1$.

Крок 4.10. Якщо розраховані значення вагових коефіцієнтів і зсувів всіх нейронів μ -го шару ($\rho > N_\mu$), тоді виконати перехід до кроку 4.11; у іншому випадку – перейти до кроку 4.7.

Крок 4.11. Збільшити лічильник шарів нейромережі: $\mu = \mu + 1$.

Крок 4.12. Якщо розраховані значення вагових коефіцієнтів і зсувів всіх нейронів всіх шарів нейромережі ($\mu > M$, де M – кількість нейронів мережі), тоді виконати перехід до кроку 5; у противному випадку – перейти до кроку 4.3.

Крок 5. Збільшити лічильник згенерованих хромосом: $j = j + 1$.

Крок 6. Перевірити, чи сформована повністю початкова популяція ($j > N$). Якщо згенеровані всі хромосоми початкової популяції, тоді виконати перехід до кроку 7, у противному випадку – перейти до кроку 4.

Крок 7. Оцінити хромосоми поточної популяції, декодувавши кожен хромосому в популяції в набір вагових коефіцієнтів нейронної мережі.

Крок 8. Перевірити критерії закінчення пошуку (досягнення прийнятної помилки синтезованої моделі, перевищення максимально припустимої кількості ітерацій, перевищення припустимого часу функціонування методу). У випадку, якщо критерії зупинення задовільнено, виконати перехід до виконання кроку 13.

Крок 9. Виходячи зі значення цільової функції, вибрати хромосоми для генерації нових рішень.

Крок 10. Застосувати оператор схрещування до хромосом, відібраних на попередньому кроці.

Крок 11. Виконати оператор точкової мутації над обраними хромосомами.

Імовірність мутації P_i i -го гену h_i мутуючої хромосоми пропонується розраховувати за формулою:

$$P_i = \begin{cases} \gamma \frac{\bar{I}}{I_i}, & \text{якщо } h_i = w_{\nu\rho}^{(1)}; \\ \gamma, & \text{якщо } h_i \neq w_{\nu\rho}^{(1)}. \end{cases}$$

де I_i – значення оцінки індивідуальної інформативності вхідної ознаки у зв'язку, обумовленому ваговим коефіцієнтом $w_{\nu\rho}^{(1)}$, якому відповідає ген

h_i ; $\bar{I} = \frac{1}{L} \sum_{i=1}^L I_i$ – середнє значення оцінок індивідуальної інформативності

ознак екземплярів навчальної вибірки; γ – імовірність мутації генів, яким відповідають ваги зв'язків нейронів, що перебувають на другому й наступних шарах нейромережі (пропонується встановити $\gamma = 0,01K$; K – кількість генів мутуючої хромосоми).

Таким чином, в результаті використання запропонованого оператора мутації знижується ймовірність мутації генів, яким відповідають значення вагових коефіцієнтів зв'язків, що йдуть від вхідних ознак з оцінками індивідуальної значимості, вищими за середню.

Крок 12. Збільшити лічильник ітерацій (часу): $t = t + 1$. Сформувати нове покоління з елітних хромосом і хромосом-нащадків, отриманих шляхом застосування еволюційних операторів схрещування й мутації. Перейти до виконання кроку 7.

Крок 13. Зупинення.

Запропонований метод еволюційної оптимізації для параметричного синтезу нейромереж дозволяє отримувати значення вагових коефіцієнтів і зсувів, не накладаючи обмеження на вид функцій активації нейронів. На відміну від класичних методів еволюційного пошуку, використовуваних для навчання нейромереж, у розробленому методі застосовуються спеціальні оператори ініціалізації й мутації, які враховують апріорну інформацію про значущість ознак, що підвищує ефективність пошуку й зменшує час еволюційної оптимізації.

9.4 Структурний синтез нейронних мереж

У випадку, якщо структуру нейромережної моделі неможливо вибрати заздалегідь, виходячи з апріорних знань про досліджуваний об'єкт, процес або систему, виконують структурний синтез.

При використанні еволюційного пошуку для синтезу структури нейромереж необхідно визначити спосіб подання структури мережі в хромосомі й вибрати еволюційні оператори, що використовуються в процесі пошуку.

9.4.1 Подання інформації про структуру нейромережі в хромосомі

Існують такі методи кодування інформації про структуру нейромоделі в хромосомах [49]:

- пряме кодування;
- параметричне подання;
- комбіноване подання;
- подання у вигляді породжуючих правил;
- фрактальне подання;
- популяційне подання;
- дерево подання.

шення кількості нейронів у базовій мережі приводить до значного збільшення часу, необхідного на еволюційний пошук.

Перетворення хромосоми в структуру нейромоделі (декодування) відбувається в такий спосіб.

Крок 1. Сформувати матрицю зв'язків нейромережі, що відповідає хромосомі (рис. 9.2, б).

Крок 2. Побудувати на основі матриці зв'язків граф (рис. 9.2, в).

Крок 3. Синтезувати нейромережу на основі побудованого на попередньому кроці графа, видаливши при цьому нейрони, що не мають вихідних зв'язків з нейронами наступних шарів (рис. 9.2, г).

Для зменшення розміру хромосоми можна використовувати апріорну інформацію про структуру мережі. Так, наприклад, якщо відомо, що між двома сусідніми шарами мережі використовуються повнозв'язні з'єднання, то в такому випадку в хромосомі може бути подана лише інформація про кількість схованих шарів і кількість нейронів на кожному шарі.

Якщо виникає потреба вибору функції активації нейронів при структурному синтезі, в хромосому можна ввести додаткові гени, що містять інформацію про вид функції активації для кожного нейрона.

9.4.1.2 Параметричне подання

З метою скорочення розміру хромосоми при використанні параметричного подання в хромосомі міститься набір параметрів мережі. До таких параметрів відносять: кількість схованих шарів, кількість нейронів на кожному шарі, кількість зв'язків між двома шарами і т. п. Декодування хромосоми в нейромережу здійснюється по заздалегідь встановлених правилах.

Такий підхід до подання нейромоделей ефективно може застосовуватися у випадках, коли апріорно відомий тип нейромережі, що найкращим чином відповідає досліджуваній системі.

9.4.1.3 Комбіноване подання

Комбіноване подання поєднує в собі ідеї прямого й параметричного кодування нейромережі в хромосомі. При такому способі кодування гени хромосоми містять інформацію про шари мережі (кількість нейронів у шарі). Зв'язки між шарами мережі визначаються за допомогою методу прямого кодування. Таким чином, вершини графа, що визначає зв'язки між нейронами при прямому способі кодування структури нейромережі, у комбінованому поданні визначають не нейрони, а шари нейронів. Можливі випадки використання одного або декількох шаблонів зв'язків (повнозв'язні, частковозв'язні за певними правилами).

9.4.1.4 Подання у вигляді породжуючих правил

При використанні подання структури мережі у вигляді породжуючих правил гени хромосоми є шаблонами, що визначають структуру зв'язків між нейронами мережі. При цьому шаблон являє собою фрагмент нейромережі, що складається з певної кількості нейронів і зв'язків між ними.

9.4.1.5 Фрактальне подання

Фрактальне подання засноване на використанні базового шаблона-генератора (фракталу), що визначає структуру всієї мережі. Фрактал є структурою, що складається із частин, які в деякому сенсі подібні цілому. Декодування хромосоми в такому випадку відбувається шляхом рекурентних перетворень базового фракталу аналогічно попередньому методу.

Хромосома при такому поданні відповідає характеристикам базового шаблона.

Важливо відзначити, що при необхідності вибору оптимальної функції активації для кожного нейрона в процесі структурного синтезу, гени хромосоми при використанні кожного з розглянутих методів також можуть містити інформацію про вид функції активації кожного нейрона.

9.4.1.6 Популяційне подання

При такому способі кодування хромосома в популяції відповідає не нейромережі, а окремому нейрону багаточислової нейромережної моделі. Нейромережа будується на основі декількох хромосом, обраних з поточної популяції.

Створюються дві популяції (популяція нейронів та популяція нейромереж), у кожній з яких виконується еволюційний пошук.

Кожна хромосома в популяції нейронів являє собою набір вагових коефіцієнтів одного нейрона. При цьому кількість міжнейронних зв'язків є фіксованою для всіх хромосом (нейронів). Ген такої хромосоми являє собою пари значень: номер (мітка) попереднього нейрона, значення відповідного вагового коефіцієнту. Хромосоми (нейрони) H_j у такій популяції оцінюються як середнє значення фітнес-функції хромосом з популяції нейромереж, які містять у собі нейрони, що відповідають оцінюваній хромосомі H_j .

Хромосома в популяції нейромереж визначає набір нейронів, що утворюють нейромережу.

Таке подання є мало ефективним, оскільки та сама хромосома з популяції нейронів, як правило, впливає по-різному при її використанні як ген хромосоми з популяції нейромереж, у результаті чого пошук може вестися в помилковому напрямку.

9.4.1.7 Дерево подання

При використанні генетичного програмування для структурного синтезу нейромереж використовується подання у вигляді дерева, що відповідає структурі мережі. Хромосома містить певну кількість генів (нефіксована кількість), кожний з яких описує певний нейрон мережі.

Вузли в дереві подання можуть бути двох видів: функціональні (нейрони) і термінальні (вхідні змінні).

Функціональні вузли (нейрони) містять: функцію активації, значення зсуву, значення вагових коефіцієнтів міжнейронних зв'язків. Значення виходів нейронів розраховуються послідовно в залежності від їхнього розташування в хромосомі.

9.4.2 Визначення фітнес-функції

У фітнес-функції при структурному синтезі, як правило, враховуються:

- складність синтезованої мережі, що відповідає оцінюваній хромосомі.
- У цьому критерії враховуються кількість шарів мережі, нейронів на кожному шарі, а також характер, топологія й кількість міжнейронних зв'язків;
- помилка прогнозування або класифікації за синтезованою нейромережною моделлю.

9.4.3 Послідовність виконання структурного синтезу на основі методів еволюційної оптимізації

Структурний синтез нейромоделі на основі еволюційного підходу може бути виконаний як така послідовність кроків [49, 51].

Крок 1. Сформувані початкове покоління хромосом, що містять інформацію про структуру мережі.

Крок 2. Виконати оцінювання хромосом поточної популяції.

Крок 2.1. Декодувати кожну хромосому популяції в архітектуру нейронної мережі.

Крок 2.2. Навчити кожну нейронну мережу за обраним правилом за допомогою даних з навчальної вибірки (виконати параметричний синтез нейромережі). Початкові значення вагових коефіцієнтів при навчанні задаються в залежності від обраного методу параметричного синтезу або випадковим чином.

Крок 2.3. Обчислити значення фітнес-функції хромосом, що враховує помилку нейромережі й складність її архітектури (кількість шарів, нейронів, міжнейронних зв'язків).

Крок 3. Перевірити критерії закінчення пошуку. У випадку задоволення таких критеріїв, виконати перехід до кроку 7.

Крок 4. Вибрати найбільш пристосовані хромосоми для виконання над ними еволюційних операторів схрещування й мутації.

Крок 5. Виконати оператори схрещування й мутації над відібраними раніше хромосомами.

Крок 6. Створити нове покоління з отриманих на попередньому кроці хромосом-нащадків і найбільш пристосованих хромосом поточного покоління. Виконати перехід до кроку 2.

Крок 7. Зупинення.

9.4.4 Еволюційні оператори для структурного синтезу нейромоделей

При виконанні структурного синтезу нейромоделей застосовуються, як правило, класичні оператори схрещування, що використовуються в генетичних методах і генетичному програмуванні.

Як оператори мутації, крім традиційно застосовуваних, можуть використовуватися:

- незміщена мутація – еволюційний оператор, що із заданою ймовірністю замінює значення кожного гена в хромосомі (ваги нейромережі) на випадково згенероване число з початкового інтервалу значень ваг нейронної мережі;

- зміщена мутація – із заданою ймовірністю збільшує (або зменшує) значення кожного гену в хромосомі на випадково згенероване число з початкового інтервалу значень ваг нейронної мережі;

- мутація нейронів – складається із двох кроків.

Крок 1. Вибрати K нейронів з нейромережі, що відповідає мутуючій хромосомі.

Крок 2. Змінити значення всіх вагових коефіцієнтів обраних на попередньому кроці нейронів;

- мутація найменш значущих нейронів – виконується в такій послідовності.

Крок 1. Вибрати найменш значущий нейрон з мережі, що відповідає мутуючій хромосомі. Найменш значущим вважається нейрон, при якому різниця між помилкою повної мережі й помилкою мережі, побудованої без цього нейрона, буде мінімальною.

Крок 2. Змінити значення всіх вагових коефіцієнтів найменш значущого нейрона.

Можливий вибір на першому кроці декількох найменш значущих нейронів.

При використанні генетичного програмування для синтезу структури нейромоделей застосовуються такі види мутації:

- зміна міжнейронних зв'язків;

- зміна нейронів на термінальні вузли або інші нейрони.

9.4.5 Метод полімодального еволюційного пошуку з кластеризацією хромосом

При вирішенні задачі структурного синтезу нейромережних моделей цільова функція, як правило, має кілька оптимумів [49, 105], розташованих у різних областях простору пошуку. При використанні класичних методів еволюційної оптимізації результатом пошуку є популяція рішень, що мало відрізняються одне від одного, в результаті чого знайдене рішення може виявитися локальним оптимумом багатоекстремальної функції. Таке рішення (структура нейромоделі), як правило, є неефективним при його використанні на практиці.

Тому для структурного синтезу нейромереж є доцільним використання еволюційних методів, здатних до пошуку декількох субоптимальних рішень. При використанні традиційних методів еволюційного пошуку для оптимізації полімодальних функцій існує проблема передчасної збіжності до локального оптимуму. Для подолання даної проблеми розроблені дві групи методів: методи відхилення від передчасної збіжності й методи відновлення [13, 27].

Методи першої групи зменшують генетичну збіжність популяції з метою забезпечення її розмаїтості й недопущення зациклення в локальних оптимумах. Однак при такому підході використовуються штрафні функції для зменшення ймовірності появи в популяції схожих рішень, що приводить до необхідності розрахунку значень таких функцій для кожної хромосоми й, отже, значно сповільнює процес еволюційного пошуку.

Методи відновлення спрямовані на збільшення розмаїтості в популяції шляхом використання стратегії перезапущу у випадках недостатності розмаїтості в популяції, що також вимагає значних часових витрат.

У розробленому методі полімодального еволюційного пошуку з кластеризацією хромосом [104] пропонується групувати рішення (хромосоми) у кластери по їхньому розташуванню в просторі пошуку.

В запропонованому методі у процесі еволюційної оптимізації визначаються групи близьких (подібних) хромосом і підвищується розмаїтість популяції шляхом погіршення значень фітнес-функції хромосом залежно від близькості до центра їхньої групи, не вимагаючи при цьому обчислення значень штрафних функцій і не використовуючи процедури перезапущу.

Розроблений полімодальний еволюційний пошук з кластеризацією хромосом передбачає виконання таких кроків.

Крок 1. Задати кількість оптимумів (оптимальних структур нейромережних моделей) $k \ll N$, які необхідно знайти в результаті еволюційної оптимізації.

Крок 2. Встановити лічильник ітерацій: $t = 1$.

Крок 3. Встановити кількість елітних особин (хромосом): $k_e = k$.

Крок 4. Ініціалізувати початкову популяцію у вигляді хромосом H_j , $j = 1, 2, \dots, N$.

Крок 5. Обчислити значення фітнес-функції $f(H_j)$ для кожної хромосоми H_j .

Крок 6. Згрупувати хромосоми в k кластерів за значенням їхніх фітнес-функцій і розташуванням в просторі пошуку.

Крок 6.1. Для кожної хромосоми H_j обчислити відстань Хеммінга (кількість незбіжних битів в однакових позиціях хромосом) від її до всіх інших хромосом у популяції. Відстань Хеммінга $d(H_j; H_l)$ між хромосомами H_j і H_l розраховується за формулою:

$$d(H_j; H_l) = \sum_{i=1}^L |h_{ij} - h_{il}|,$$

де L – розмір хромосом; h_{ij} і h_{il} – значення i -их генів хромосом H_j і H_l , відповідно.

Крок 6.2. Встановити лічильник сформованих кластерів: $m = 1$.

Крок 6.3. Вибрати хромосому з найкращим значенням фітнес-функції як центр m -го кластеру. При цьому розглядаються хромосоми, ще не згруповані по кластерах.

Крок 6.4. Ввести в кластер $(N/k - 1)$ хромосом, найближчих за відстанню Хеммінга до хромосоми, що є центром поточного m -го кластеру.

Крок 6.5. Якщо всі кластери сформовані ($m = k$), тоді виконати перехід до кроку 7.

Крок 6.6. Встановити: $m = m + 1$. Виконати перехід до кроку 6.3.

Крок 7. Збільшити значення фітнес-функцій хромосом, що не є кращими в кластері:

$$f_n(H_j) = \left(\frac{d(H_j; H_{\max, j})}{d(H_j; H_{c, j})} \right)^s f(H_j),$$

де $f_n(H_j)$ – нове значення фітнес-функції j -ої хромосоми; $f(H_j)$ – значення фітнес-функції до зміни j -ої хромосоми; $d(H_j; H_{c, j})$ – відстань Хеммінга від j -ої хромосоми до центра її групи; $d(H_j; H_{\max, j})$ – максимальна відстань Хеммінга в групі j -ої хромосоми; s – параметр, що визначає ступінь погіршення значень фітнес-функцій хромосом, що не є центрами кластерів, $s \geq 1$.

Крок 8. Відібрати хромосоми для схрещування й мутації.

Крок 9. Застосувати еволюційні оператори схрещування й мутації до відібраних на попередньому кроці хромосом. Як такі оператори пропонується використовувати класичні оператори (наприклад, точкове, однорідне або рівномірне схрещування й просту мутацію).

Крок 10. Сформуувати нове покоління. При цьому кращі (елітні) хромосоми в кожному кластері гарантовано переходять у нове покоління.

Крок 11. Якщо $t = T$, де T – максимально задана кількість ітерацій, тоді виконати перехід до кроку 14.

Крок 12. Встановити: $t = t + 1$.

Крок 13. Виконати перехід до кроку 5.

Крок 14. Оцінити кожну з k хромосом, що є центрами кластерів, за допомогою даних тестової вибірки. Вибрати найкращу хромосому. Нейромодель, що відповідає такій хромосомі, приймається як рішення.

Крок 15. Зупинення.

Розроблений метод полімодального еволюційного пошуку з кластеризацією хромосом підвищує розмаїтість популяції й дозволяє більш рівномірно покрити простір пошуку, підвищуючи в такий спосіб можливість відшукування глобального оптимуму й збільшуючи ймовірність успішного виконання процедури оцінювання знайдених рішень за допомогою зовнішніх критеріїв на тестовій вибірці, оскільки результатом пошуку є не єдина структура нейромережі, а множина різних рішень, що дозволяє вибрати таку нейромережну модель, що найкращим чином задовольняє зовнішнім критеріям оцінювання ефективності нейромоделей.

9.5 Структурно-параметричний синтез

Досить істотним недоліком існуючих методів структурного синтезу [49, 51, 105] є неоднозначність їхніх результатів, викликана двома факторами, пов'язаними з необхідністю навчання нейромережі для оцінювання її структури:

- випадковий вибір початкових значень вагових коефіцієнтів нейромоделі при параметричному синтезі (навчанні мережі), що приводить до ситуацій, коли мережам з однаковою архітектурою відповідають різні значення критерію оцінювання;

- одержання різних результатів навчання, отриманих при використанні різних методів настроювання вагових коефіцієнтів, внаслідок чого при виборі неоптимального для розв'язуваної задачі методу параметричного синтезу, оцінювання структур нейромоделей у процесі структурного синтезу буде необ'єктивним. Це, у свою чергу, приведе до неоптимальності знайденої структури нейронної мережі.

Для усунення такої неоднозначності при оцінюванні хромосом у процесі структурного синтезу необхідно виконувати багаторазовий (із застосуванням різних початкових значень вагових коефіцієнтів і з використанням різних методів навчання) параметричний синтез нейромережі, що відповідає оцінюваній структурі мережі. Однак такий підхід значно збільшує час, необхідний для виконання структурного синтезу.

Іншим, більш ефективним, методом подолання зазначеного недоліку є одночасний синтез структури нейромоделі й пошук оптимальних значень параметрів такої мережі – структурно-параметричний синтез.

При структурно-параметричному синтезі нейромоделей обираються оптимальні значення вагових коефіцієнтів нейромережі (вхідного шару й схованих шарів), зсувів нейронів, а також функцій активації нейронів. Тому пропонується рішення (хромосому) складати з декількох частин (рис. 9.3): у першій частині зберігати інформацію про значення ваг нейронів вхідного шару, у другій – значення вагових коефіцієнтів нейронів схованих шарів, у третій – значення зсувів нейронів, у четвертій – функції активації для кожного нейрона мережі.

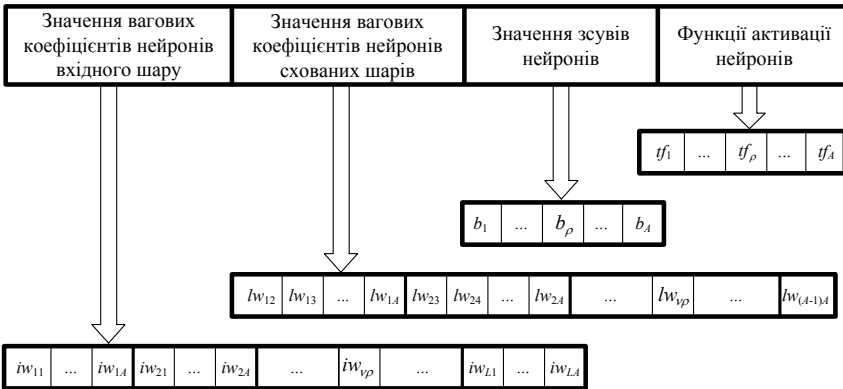


Рисунок 9.3 – Подання хромосоми при структурно-параметричному синтезі нейромереж

На рис. 9.3 використовуються такі позначення: iw_{vp} – значення вагового коефіцієнту зв'язку від v -ої ознаки до p -го нейрона; lw_{vp} – значення вагового коефіцієнта зв'язку від v -го нейрона до p -го нейрона; b_p – значення зсуву p -го нейрона; tf_p – функція активації p -го нейрона.

Як видно з рис. 9.3, друга частина хромосоми, що містить інформацію про значення вагових коефіцієнтів нейронів схованих шарів, кодується за допомогою способу, аналогічного параметричному поданню хромосоми при структурному синтезі.

Розмір хромосоми K визначається за формулою:

$$K = K_1 + K_2 + K_3 + K_4 = L \cdot A + \frac{A(A-1)}{2} + A + A = L \cdot A + \frac{A(A+3)}{2},$$

де K_1, K_2, K_3, K_4 – кількість генів першої, другої, третьої й четвертої частин хромосоми, відповідно.

З метою пошуку нейромоделей з мінімальною кількістю синаптичних з'єднань у запропонованому методі структурно-параметричного синтезу

розроблені спеціальні оператори схрещування й мутації, що зменшують кількість зв'язків у мережі.

Для підвищення ефективності еволюційної оптимізації при структурно-параметричному синтезі нейромоделей і скорочення часу пошуку пропонується на етапі ініціалізації хромосом, що відповідають нейромережним моделям, вибір початкових значень вагових коефіцієнтів і зсувів здійснювати таким чином, щоб рівномірно розподілити активну область визначення функції активації кожного нейрона по простору вхідних змінних, враховуючи при цьому індивідуальну інформативність ознак.

Розроблений еволюційний метод структурно-параметричного синтезу нейромережних моделей пропонується виконувати як таку послідовність кроків.

Крок 1. Сформулювати початкове покоління хромосом, що містять інформацію про структуру мережі і її параметри.

Крок 2. Виконати оцінювання хромосом поточної популяції, обчисливши значення фітнес-функції, що враховує помилку нейромережі й складність її архітектури (кількість міжнейронних зв'язків, нейронів, шарів). Для розрахунку помилки мережі, що відповідає оцінюваній хромосомі H_j , доцільно використовувати формулу:

$$E(H_j) = \sum_{p=1}^m (y_p - y(H_j, X_p))^2,$$

де $y(H_j, X_p) = y(\text{НМ}; X_p) = y_{Ap}$ – значення виходу нейромоделі НМ, побудованої на основі хромосоми H_j , обчислене для набору значень X_p за виразом:

$$y_{Ap} = f_A \left(b_A + \sum_{c=1}^L i w_{cA} x_{cp} + \sum_{c=1}^{A-1} l w_{cA} y_{cp} \right),$$

де y_{Ap} – значення виходу A -го нейрона мережі для p -го екземпляра; y_{cp} – значення виходу c -го нейрона мережі для p -го екземпляра, розраховується за формулою, аналогічною для обчислення значення y_{Ap} .

Крок 3. Перевірити критерії закінчення пошуку. У випадку задоволення таких критеріїв, виконати перехід до кроку 8.

Крок 4. Вибрати найбільш пристосовані хромосоми для виконання над ними еволюційних операторів схрещування й мутації.

Крок 5. Виконати оператор схрещування, що зменшує кількість синаптичних з'єднань у мережі.

Для частин хромосом, що містять інформацію про значення вагових коефіцієнтів нейронів вхідного й схованого шарів, значення i -го гену нащадків пропонується визначати за формулами:

$$h_{i\text{in}1} = \begin{cases} 0, & \text{якщо } h_{i1} \cdot h_{i2} < 0; \\ kh_{i1} + (1-k)h_{i2}, & \text{в іншому випадку,} \end{cases}$$

$$h_{in2} = \begin{cases} 0, & \text{якщо } h_{i1} \cdot h_{i2} > 0; \\ (1-k)h_{i1} + kh_{i2}, & \text{в іншому випадку,} \end{cases}$$

де h_{in1} і h_{in2} – значення i -их генів першого й другого нащадків, відповідно; h_{i1} і h_{i2} – значення i -их генів першого й другого батьків, відповідно; k – коефіцієнт, що задається користувачем, $k \in (0; 1)$.

Значення генів, що відповідають зсувам нейронів, визначити за формулами:

$$h_{in1} = kh_{i1} + (1-k)h_{i2} \quad \text{та} \quad h_{in2} = kh_{i2} + (1-k)h_{i1}.$$

Значення генів, що визначають функцію активації нейрона, пропонується обчислювати за правилами:

$$h_{in1} = \begin{cases} h_{i1}, & \text{якщо } h_{i1} = h_{i2} \text{ або } r > 0,5; \\ \text{rand}[\text{TF}], & \text{в іншому випадку,} \end{cases} \quad h_{in2} = \begin{cases} h_{i2}, & \text{якщо } h_{i1} = h_{i2} \text{ або } r \leq 0,5; \\ \text{rand}[\text{TF}], & \text{в іншому випадку,} \end{cases}$$

де r – випадково згенероване число в інтервалі $(0; 1)$; $\text{rand}[\text{TF}]$ – випадково обраний елемент множини TF, що містить функції активації, використовувани для побудови нейромережі.

Крок 6. Виконати оператор точкової мутації.

У випадку, якщо для мутації обраний ген хромосоми, що містить інформацію про значення вагових коефіцієнтів нейронів мережі, тоді нове значення i -го гену пропонується обчислювати за формулою:

$$h_{in} = \begin{cases} 0, & \text{якщо } |r| < 0,5h_i; \\ r, & \text{в іншому випадку,} \end{cases}$$

де h_i та h_{in} – значення i -го гену до та після мутації, відповідно; $r = \text{rand}[-h_i; h_i]$ – випадково згенероване число в інтервалі $[-h_i; h_i]$.

Якщо для мутації обраний ген хромосоми, що відповідає зсуву нейрона, тоді значення i -го гену після мутації h_{in} пропонується визначати за формулою:

$$h_{in} = \text{rand}[h_{i,\min}; h_{i,\max}],$$

де $h_{i,\min}$ та $h_{i,\max}$ – мінімальне та максимальне значення i -го гену в поточній популяції хромосом.

У випадку вибору для мутації гену хромосоми, що містить інформацію про функції активації нейрона, нове значення гену пропонується визначати за формулою:

$$h_{in} = \text{rand}[\text{TF}].$$

Крок 7. Створити нове покоління з отриманих на попередньому кроці хромосом-нащадків і найбільш пристосованих хромосом поточного покоління. Виконати перехід до кроку 2.

Крок 8. Зупинення.

Розроблений еволюційний метод структурно-параметричного синтезу нейромереж дозволяє будувати нейромережну модель, використовуючи

при цьому оператори схрещування й мутації, що зменшують кількість синаптичних з'єднань у мережі.

Для підвищення ефективності розробленого методу структурно-параметричного синтезу нейромоделей на основі еволюційного підходу доцільно суміщати наведену вище послідовність кроків із запропонованими методами параметричного синтезу, що прискорює пошук за рахунок використання апріорної інформації про значущість ознак, і структурного синтезу, що дозволяє одержувати різні структури нейромоделей, підвищуючи в такий спосіб імовірність синтезу нейромереж, що мають прийнятні апроксимаційні здатності як для даних навчальної, так і для даних тестової вибірок.

9.6 Адаптація навчальних правил

Методи еволюційної оптимізації також використовуються для пошуку оптимальних значень параметрів методів структурного й параметричного синтезу нейромоделей. Хромосома в такому випадку містить інформацію про параметри використовуваних методів синтезу нейромереж.

Крок 1. Створити початкову популяцію з хромосом, гени яких відповідають параметрам використовуваного методу структурно-параметричного синтезу.

Крок 2. Оцінити значення фітнес-функцій хромосом поточної популяції.

Крок 2.1. Згенерувати набір нейронних мереж довільної архітектури з випадковими значеннями вагових коефіцієнтів.

Крок 2.2. Навчити згенеровані на попередньому кроці нейромережі з використанням обраного заздалегідь методу структурно-параметричного синтезу. Параметри методу встановити в залежності від значень генів оцінюваної хромосоми.

Крок 2.3. Обчислити значення фітнес-функції оцінюваної хромосоми як середнє значення досягнутих результатів навчання різних нейромереж (помилка мережі, параметри структури).

Крок 3. Перевірити виконання критеріїв зупинення. Якщо критерії зупинення виконуються, тоді перейти до кроку 7.

Крок 4. Відібрати хромосоми для схрещування й мутації.

Крок 5. Виконати еволюційні оператори схрещування й мутації над хромосомами, обраними на попередньому кроці.

Крок 6. Сформувані нове покоління. Виконати перехід до виконання кроку 2.

Крок 7. Зупинення.

Еволюційний пошук оптимальних значень параметрів використовуваного методу структурно-параметричного синтезу нейромоделей відбувається, як правило, окремо від синтезу нейромережі досліджуваного об'єкту, процесу або системи.

9.7 Оптимізація структури нейромоделей

Відомо з [67, 91, 105], що нейромережні моделі мають різні характеристики: якість апроксимації, точність, інтерпретабельність, простота апаратної й програмної реалізації, обчислювальна й просторова складність і т. ін. Тому при оптимізації нейромоделі доцільним є використання набору різних критеріїв, що найбільш повним чином характеризує досліджувану залежність.

При оцінюванні структурної складності нейромоделі доцільно використовувати кількість синаптичних з'єднань (ваг) у мережі N_w , кількість нейронів N_n і кількість шарів N_c у нейромоделі. Пропонується використовувати такий критерій оцінювання складності структури нейромережі:

$$K_c = \left(1 + \frac{N_w}{N_n}\right) \left(1 + \frac{N_n}{N_c}\right) (1 + N_c), \quad N_n \geq 1, \quad N_c \geq 1.$$

Цей критерій враховує кількість шарів у нейромережі, а також середню кількість нейронів у шарі й середню кількість зв'язків, що доводяться на один нейрон.

На обчислювальну складність мережі істотно впливають функції активації нейронів. Тому як критерій оцінювання обчислювальної складності пропонується використовувати вираз, що враховує кількість обчислювально складних та обчислювально простих функцій активації:

$$K_e = \left(\frac{1 + N_{\text{скл.}}}{1 + N_{\text{пр.}}} \right),$$

де $N_{\text{скл.}}$ – кількість обчислювально складних функцій активації нейронів (логістична сигмоїдна, тангенціальна сигмоїдна, радіальнобазисна й ін.); $N_{\text{пр.}}$ – кількість обчислювально простих функцій активації (порогова, лінійна).

Для оцінювання апроксимаційних якостей мережі й складності її застосування доцільно використовувати середньоквадратичну помилку моделі E і час обчислення значення вихідного параметру за синтезованою нейромоделлю t , відповідно.

Запропонований набір критеріїв охоплює більшість характеристик нейромереж, що дозволяє використовувати його для багатокритеріальної оптимізації з метою одержання досить простих та інтерпретабельних нейромоделей, що забезпечують високу точність апроксимації.

Оскільки оптимізація по декількох критеріях є ефективною при невеликій кількості цільових функцій [2, 4], пропонується для спрощення структури нейромоделей використовувати два критерії, що інтегрують основні характеристики нейромереж:

$$f_1 = E \cdot K_c = E \left(1 + \frac{N_w}{N_n}\right) \left(1 + \frac{N_n}{N_c}\right) (1 + N_c),$$

$$f_2 = t \cdot K_e = t \left(\frac{1 + N_{\text{скл.}}}{1 + N_{\text{пр.}}} \right).$$

Перший критерій f_1 характеризує якість апроксимації нейромоделі і її структурну складність. Критерій f_2 поєднує час обчислення вихідного параметру за синтезованою мережею та її обчислювальну складність.

Для вирішення задачі оптимізації побудованих нейромережних моделей розроблений метод багатокритеріального еволюційного пошуку з використанням узагальненої цільової функції, що заснований на популяційному підході як одному з найменш витратних методів з погляду ресурсів. Для усунення недоліків існуючих методів, що використовують популяційний підхід, пропонується ввести узагальнюючу функцію, що дозволяє оцінювати хромосоми не за однією цільовою функцією, а за всіма функціями, що оптимізуються. В результаті цього пошук буде проводитися у всьому просторі цільових функцій.

Запропонований метод багатокритеріального еволюційного пошуку з використанням узагальненої цільової функції [37] складається з таких кроків.

Крок 1. Встановити лічильник ітерацій: $t = 0$. Сформувати початкову популяцію P_0 з N хромосом. Розмір популяції N вибирають таким чином, щоб забезпечити можливість одночасного дослідження досить великих областей простору пошуку.

Рішення (хромосоми) при оптимізації нейромоделей кодуються за допомогою подання, аналогічного тому, що використовується при структурно-параметричному синтезі нейромереж за допомогою еволюційного підходу. При цьому хромосома складається з декількох частин: у першій частині зберігається інформація про значення ваг нейронів вхідного шару, у другий – значення вагових коефіцієнтів нейронів схованих шарів, у третій – значення зсувів нейронів, у четвертій – функції активації для кожного нейрона мережі.

Крок 2. Розбити популяцію P_t на K підпопуляцій розміром N/K кожна ($N/K \geq 2$), де K – кількість цільових функцій розв'язуваної задачі.

Крок 2.1. Для кожної хромосоми H_j розрахувати Евклідову відстань від неї до всіх інших хромосом у популяції. Евклідова відстань d між хромосомами H_j і H_l обчислюється за формулою:

$$d(H_j; H_l) = \sqrt{\sum_{i=1}^L (h_{ij} - h_{il})^2},$$

де L – розмір хромосом; h_{ij} і h_{il} – значення i -их генів хромосом H_j і H_l , відповідно.

Крок 2.2. Встановити лічильник сформованих підпопуляцій: $c = 1$.
Сформувати множину хромосом, що не ввійшли в підпопуляцію, $A = P_t$.

Крок 2.3. Ініціалізувати c -у підпопуляцію: $V_c = \emptyset$.

Крок 2.4. Вибрати з множини A дві хромосоми H_j та H_l з максимальною відстанню між ними.

Крок 2.5. Включити обрані хромосоми H_j та H_l у c -ту підпопуляцію:
 $V_c = V_c \cup \{H_j, H_l\}$. Виключити хромосоми H_j і H_l з A : $A = A \setminus \{H_j, H_l\}$.

Крок 2.6. Якщо c -та підпопуляція повністю сформована ($|V_c| = N/K$), тоді перейти до кроку 2.9.

Крок 2.7. Вибрати з множини A хромосому H_k , сума відстаней від якої до хромосом з V_c є максимальною.

Крок 2.8. Включити хромосому H_k у c -ту підпопуляцію:
 $V_c = V_c \cup \{H_k\}$. Виключити хромосому H_k з A : $A = A \setminus \{H_k\}$. Виконати перехід до кроку 2.6.

Крок 2.9. Якщо сформовані всі підпопуляції ($c = K$), тоді перейти до виконання кроку 3.

Крок 2.10. Збільшити лічильник сформованих підпопуляцій:
 $c = c + 1$. Виконати перехід до кроку 2.3.

Крок 3. Виконати однокритеріальний еволюційний пошук у кожній з K підпопуляцій. При цьому оцінювання хромосом в c -ій підпопуляції виконувати за c -ої цільовою функцією.

Крок 4. Вибрати в кожній підпопуляції $N_b \leq N/K$ кращих особин.

Крок 5. Сформувати об'єднану популяцію з всіх особин, обраних на попередньому кроці.

Крок 6. Виконати еволюційний пошук над хромосомами з об'єднаної популяції. При цьому оцінювання хромосоми H_j проводити за допомогою обчислення узагальненого значення цільової функції $F(H_j) = F(f_1(H_j), f_2(H_j), \dots, f_k(H_j))$, що визначається за правилом:

$$F(H_j) = \sum_{k=1}^K \frac{B_k}{\Delta_k},$$

$$\text{де } B_k = \begin{cases} f_k(H_j) - \min(f_k), & \text{якщо } f_k \rightarrow \min; \\ \max(f_k) - f_k(H_j), & \text{якщо } f_k \rightarrow \max, \end{cases}$$

$\Delta_k = \max(f_k) - \min(f_k)$; $\max(f_k)$ і $\min(f_k)$ – відповідно, максимальне й мінімальне значення k -ої цільової функції на поточній ітерації. Таким чином, $F(H_j) \in [0; K]$.

З метою зменшення кількості синаптичних з'єднань у мережі як еволюційні оператори використовувати оператори схрещування й мутації, запропоновані в еволюційному методі структурно-параметричного синтезу нейромоделей.

Крок 7. Виконати перевірку критеріїв закінчення пошуку. Якщо критерії задовільнено, тоді виконати перехід до кроку 9.

Крок 8. Збільшити лічильник ітерацій: $t = t + 1$. Виконати перехід до кроку 2.

Крок 9. Зупинення.

Для оцінювання хромосом H_j з об'єднаної популяції замість використання узагальненої цільової функції можна застосовувати узагальнений ранг хромосом, що отримується в результаті додавання рангів по кожній функції й обчислюється за формулою:

$$R_j = R_{j1} + R_{j2} + \dots + R_{jk},$$

де R_j – ранг j -ої хромосоми; R_{jk} – ранг j -ої хромосоми для цільової функції f_k (кількість хромосом з більш оптимальним значенням цільової функції f_k).

Таке ранжирування відрізняється від ранжирування за принципом невідомінування тим, що хромосоми ранжируються за кожною цільовою функцією окремо, а потім знаходиться сумарний ранг кожної хромосоми. Крім того, запропоноване ранжирування вимагає менше обчислювальних ресурсів, ніж ранжирування за принципом невідомінування.

Таким чином, запропонований багатокритеріальний еволюційний пошук з використанням узагальненої цільової функції складається з двох етапів.

На першому етапі відбувається пошук оптимуму кожної цільової функції окремо, що дозволяє знайти більш оптимальні рішення в порівнянні з існуючими методами, що використовують популяційний підхід, при яких відбувається звичайний вибір декількох кращих хромосом по різних цільових функціях.

На другому етапі здійснюється багатокритеріальний еволюційний пошук, що заснований на використанні узагальненої функції, яка поєднує в собі значення всіх цільових критеріїв. Даний підхід дозволяє відбирати й одержувати ті хромосоми, які не є оптимальними для жодної з заданих цільових функцій, але в той же час є найбільш прийнятними для набору цих функцій.

9.8 Критерії й методика порівняння еволюційних методів синтезу нейронних мереж

Для забезпечення можливості комплексного аналізу еволюційних методів побудови нейромережних моделей необхідно розробити набір критеріїв, що дозволяє оцінювати ефективність створених методів еволюційної оптимізації для побудови нейромоделей.

Тому для дослідження ефективності еволюційних методів відбору ознак і синтезу нейронних мереж пропонується використовувати систему критеріїв [100], що включає в себе: класичні критерії [62], критерії оцінювання однорідності популяції, критерії оцінювання еволюційної адаптації, а також критерії оцінювання синтезованих нейромережних моделей.

Класичні критерії пропонується використовувати для аналізу еволюційного пошуку як методу оптимізації. Як такі критерії є доцільним обрати:

– точність рішення щодо оптимального або бажаного значення цільової функції $f_{\text{опт.}}$, яке визначається за формулою:

$$\varepsilon = |f_{\text{опт.}} - f_{\text{опт.}}^*|,$$

де $f_{\text{опт.}}$ – знайдене за допомогою аналізованого методу оптимальне значення цільової функції; $f_{\text{опт.}}^*$ – відоме оптимальне або необхідне значення цільової функції.

Оскільки в завданнях моделювання як цільова функція, як правило, використовується середньоквадратична помилка, необхідне оптимальне значення якої дорівнює нулю, то значення критерію може бути визначене в такий спосіб: $\varepsilon = f_{\text{опт.}}$;

– кількість обчислень значень цільової функції n , необхідне для одержання оптимального рішення тестової задачі із заданим ступенем точності. Кількість обчислень значень функцій показує, скільки разів виникає необхідність у визначенні значень цільової функції й (або) тієї або іншої функції із сукупності функцій, що задають обмеження задачі (а також числових значень похідних згаданих вище функцій), перш ніж буде знайдене рішення розглянутої оптимізаційної задачі;

– машинний час τ , необхідний для виконання послідовності процедур оптимізації, є найбільш часто використовуваним критерієм, що дозволяє порівнювати ефективність різних оптимізаційних методів;

– коефіцієнт стохастичності еволюційного пошуку характеризує ступінь випадковості при формуванні хромосом і може бути визначений як відношення кількості звертань до функції, що генерує псевдовипадкові числа, до загальної кількості генів всіх хромосом, згенерованих в процесі еволюційної оптимізації. Коефіцієнт стохастичності еволюційного пошуку пропонується обчислювати за формулою:

$$K_s = \frac{N_f}{N \cdot L \cdot T},$$

де N_f – кількість звертань в процесі еволюційної оптимізації до функції, що генерує псевдовипадкові числа;

– коефіцієнт детермінованості еволюційного пошуку характеризує ступінь використання визначених процедур при формуванні хромосом. Коефіцієнт детермінованості може бути визначений у такий спосіб: $K_d = 1 - K_s$.

Для визначення розмаїтості хромосом у популяції при еволюційному пошуку пропонується використовувати критерії оцінювання однорідності популяції:

- коефіцієнт однорідності генотипу λ_g ;
- коефіцієнт однорідності генотипу-фенотипу λ_{gf} ;
- коефіцієнт однорідності пристосованості λ_E ;

$$\lambda_E = \frac{1}{N} \sum_{j=1}^N \frac{1}{1 + D_E(f_j, \bar{f})},$$

де f_j – значення цільової функції j -ої хромосоми;

$$\bar{f} = \frac{1}{N} \sum_{j=1}^N f_j \text{ – середнє значення цільової функції в популяції;}$$

$$D_E(f_j, \bar{f}) = |f_j - \bar{f}| \text{ – відстань між цільовими функціями } f_j \text{ і } \bar{f}.$$

Поряд з коефіцієнтами однорідності можна використовувати коефіцієнти неоднорідності, що показують, як сильно відрізняються між собою по генотипу й фенотипу хромосоми популяції:

- коефіцієнт неоднорідності генотипу: $v_g = 1 - \lambda_g$;
- коефіцієнт неоднорідності генотипу-фенотипу: $v_{gf} = 1 - \lambda_{gf}$;
- коефіцієнт неоднорідності пристосованості: $v_E = 1 - \lambda_E$.

З метою визначення стійкості й здатності еволюційних методів до поліпшення середніх значень цільової функції й невлученню в локальні оптимуми розроблені критерії оцінювання еволюційної адаптації:

- критерій стабільності t -ої популяції C_t ;
- критерій оцінювання швидкості зміни цільової функції $K_{\Delta f}$;
- частина хромосом, що мають значення цільової функції, краще за середнє значення цільової функції в популяції, визначається як відношення:

$$\eta = \frac{G}{N},$$

де G – кількість хромосом, значення цільових функцій яких краще за середнє значення цільової функції в популяції;

– коефіцієнт ефективності ітерацій. Ефективною будемо вважати ітерацію, протягом якої одержали хромосому (рішення), значення цільової функції якої є більш оптимальним у порівнянні з досягнутим раніше значенням цільової функції. Коефіцієнт ефективності ітерацій пропонується обчислювати за формулою:

$$K_e = \frac{1}{T} \sum_{t=1}^T b_t,$$

$$\text{де } b_t = \begin{cases} 1, & \text{якщо } f_{\text{опт.,}t} \neq f_{\text{опт.,}t-1}; \\ 1, & \text{якщо } f_{\text{опт.,}t} = f_{\text{опт.,}t-1}, \end{cases}$$

$f_{\text{опт.,}t}$ та $f_{\text{опт.,}t-1}$ – кращі значення цільової функції на t -ій та $(t-1)$ -ій ітераціях, відповідно;

- коефіцієнт неефективності ітерацій: $K_n = 1 - K_e$.

При оцінюванні ефективності синтезованих за допомогою запропонованих методів нейромережних моделей пропонується використовувати такі критерії:

- середньоквадратичну помилку нейромережної моделі E ;
- час розрахунку значення вихідного параметра за синтезованою нейромережею $\tau_{\text{НМ}}$;
- критерій оцінювання складності структури нейромоделі K_c ;
- критерій оцінювання обчислювальної складності нейромережі K_g ;
- інтегральний критерій f_1 :

$$f_1 = E \cdot K_c = E \left(1 + \frac{N_w}{N_n} \right) \left(1 + \frac{N_n}{N_c} \right) (1 + N_c);$$

- інтегральний критерій f_2 :

$$f_2 = t \cdot K_g = t \left(\frac{1 + N_{\text{скл.}}}{1 + N_{\text{пр.}}} \right).$$

Методика проведення експериментів по вирішенню практичних задач за допомогою запропонованих методів може бути подана як така послідовність кроків.

Крок 1. Ввести вибірку даних, що характеризує досліджуваний об'єкт або процес.

Обрати еволюційні оператори відбору, схрещування й мутації для виконання еволюційного пошуку, встановити параметри еволюційної оптимізації (кількість особин у популяції, кількість елітних особин, імовірність схрещування), задати критерій закінчення пошуку.

Крок 2. Синтезувати нейромережну модель на основі навчальної вибірки, виконавши етапи відбору комбінації найбільш інформативних ознак, структурно-параметричного синтезу й оптимізації побудованої нейромоделі, вимірюючи при цьому для кожного методу кількість обчислень значень цільової функції n і машинний час τ , витрачений на еволюційну оптимізацію.

На кожній ітерації еволюційного пошуку для всіх методів виконувати розрахунок і фіксацію значень таких критеріїв і показників: значення цільової функції кожної хромосоми в популяції $f(H_i)$, кількість звертань N_i у процесі еволюційної оптимізації до функції, що генерує псевдовипадкові числа, коефіцієнт однорідності генотипу λ_g , коефіцієнт однорідності генотипу-фенотипу λ_{gf} , коефіцієнт однорідності пристосованості λ_E , критерій стабільності t -ої популяції C_t , критерій оцінювання швидкості зміни цільової функції $K_{\Delta f}$, частка хромосом, що має значення цільової функції, краще за середнє значення цільової функції в популяції η .

Після завершення роботи кожного з еволюційних методів на основі розрахованих на кожній ітерації показників визначити значення

коефіцієнту стохастичності еволюційного пошуку K_s і коефіцієнту ефективності ітерацій K_e .

Розрахувати значення критеріїв λ_g , λ_{gf} , λ_E , C_t , $K_{\Delta f}$, η як середні значення аналогічних показників, розрахованих на кожній ітерації.

Обчислити значення коефіцієнту детермінованості еволюційного пошуку K_d , коефіцієнту неоднорідності генотипу v_g , коефіцієнту неоднорідності генотипу-фенотипу v_{gf} , коефіцієнту неоднорідності пристосованості v_E , коефіцієнту неефективності ітерацій K_n , використовуючи значення критеріїв K_s , λ_g , λ_{gf} , λ_E , K_e , відповідно.

Для побудованих за допомогою запропонованих методів нейромережних моделей визначити значення таких критеріїв: середньоквадратична помилка E , час розрахунку значення вихідного параметра за нейромережею τ_{NM} , критерій оцінювання складності структури K_c , критерій оцінювання обчислювальної складності нейромережної моделі K_e , інтегральний критерій f_1 , інтегральний критерій f_2 .

Крок 3. Звести отримані результати в узагальнюючі таблиці, побудувати графіки, проаналізувати результати.

9.9 Автоматизована система еволюційного синтезу й оптимізації нейромережних моделей

Запропоновані методи синтезу нейромоделей та відбору інформативних ознак [36, 37, 74, 83, 95, 104] були програмно реалізовані мовою пакету Matlab. Розроблена автоматизована система еволюційного синтезу й оптимізації нейромережних моделей [77] дозволяє виконувати відбір інформативних ознак, параметричний синтез нейромереж, структурний синтез, структурно-параметричний синтез, спрощення побудованої нейромоделі.

Система складається з бібліотеки методів, що реалізує основні функції системи, і графічного інтерфейсу, що спрощує взаємодію користувача із системою.

Структурна схема програми подана на рис. 9.4.

Основними блоками системи є:

- блок введення-виведення – організує завантаження вхідних даних (моделі системи, вибірки даних) із вхідних файлів і запис результатів роботи у вихідні файли;

- блок еволюційної оптимізації – призначений для виконання еволюційного пошуку при відборі ознак і синтезі нейромоделей;

- блок відбору інформативних ознак – реалізує методи відбору ознак, а також містить функції для оцінювання інформативності ознак;

- блок синтезу нейромережних моделей – призначений для побудови нейромоделей за навчальними вибірками даних;

- блок інтерфейсу користувача й візуалізації (форми, графіки).

Система містить такі основні модулі.

Модуль реалізації еволюційного пошуку містить процедури, що реалізують узагальнений метод еволюційної оптимізації, і може застосовуватися в задачах багатокритеріальної оптимізації, відбору ознак і інших задачах, які можна звести до мінімізації значення цільової функції. У даному модулі програмно реалізовані класичні оператори еволюційного пошуку. Вибір найбільш прийнятних еволюційних операторів для кожної конкретної задачі здійснює користувач системи. Система дозволяє користувачу реалізовувати свої розширення за рахунок заміни одного або декількох операторів.

Модуль обчислення значень цільової функції складається з функції, що безпосередньо розраховує значення цільового критерію, а також з ґештаблиці, що зберігає вже обчислені значення. Ключем таблиці є комбінація вхідних параметрів, а значенням – обчислене на попередньому етапі значення цільової функції оціненої раніше хромосоми.

Модуль керування параметрами призначений для керування параметрами еволюційного пошуку в процесі оптимізації в залежності від критеріїв однорідності популяції. Для реалізації цього модулю в головний цикл (функція `main_ga`) введений виклик функції `gaUpdateParameters`, що запам'ятовує початкові значення параметрів еволюційного пошуку та при необхідності змінює поточний масив параметрів на скореговані. Після відновлення стабільності популяції функція відновлює вихідні значення параметрів.

Модуль критеріїв оцінювання інформативності ознак містить функції, що дозволяють обчислити інформативність наборів ознак, що відповідають оцінюваним хромосомам.

Модуль підтримки обмежень призначений для запобігання можливості появи наборів неприпустимих рішень.

Модуль методів виділення інформативної комбінації ознак містить програмну реалізацію розроблених методів відбору інформативних ознак.

Модуль параметричного синтезу нейромоделей використовується для навчання нейронної мережі заданої структури. Для навчання нейромережної моделі можуть застосовуватися такі методи: метод зворотного поширення помилки, метод класичного еволюційного пошуку, запропонований метод з використанням апріорної інформації.

Модуль структурного синтезу нейромережних моделей дозволяє синтезувати оптимальну структуру нейромережної моделі одним з методів: послідовного додавання синаптичних з'єднань, послідовного видалення синаптичних з'єднань, класичним еволюційним пошуком, полімодальним еволюційним пошуком. У даному модулі реалізований також запропонований еволюційний метод структурно-параметричного синтезу нейромережних моделей.

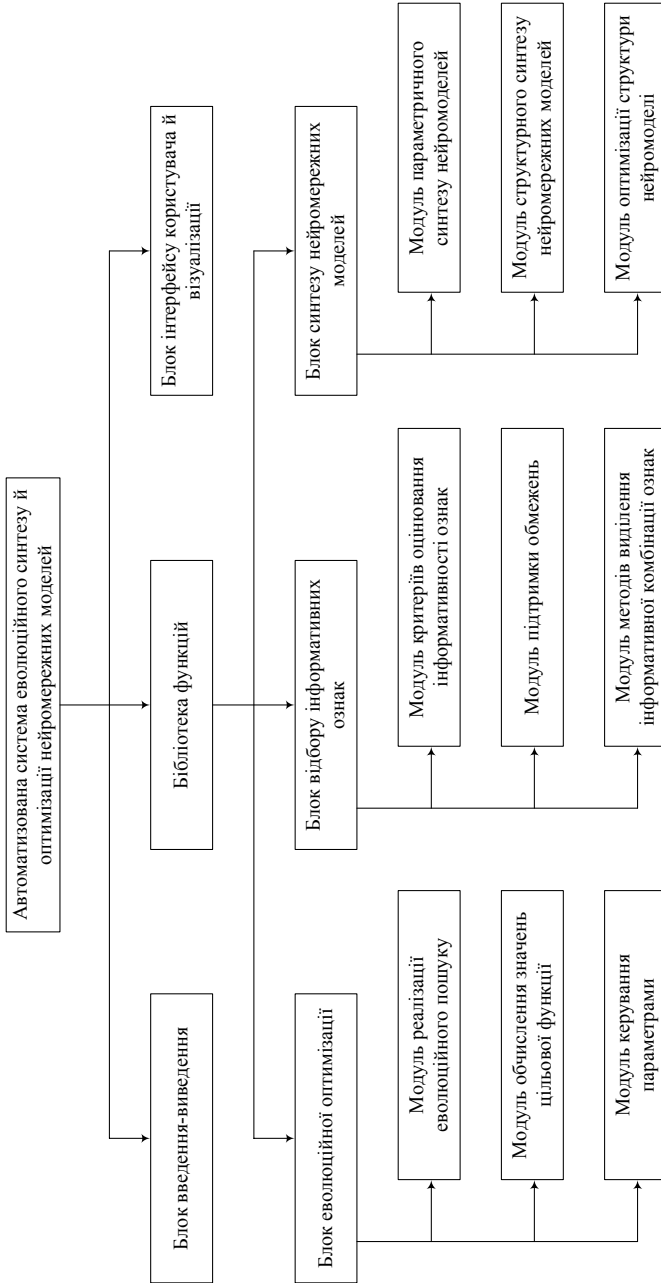


Рисунок 9.4 – Структурна схема автоматизованої системи еволюційного синтезу й оптимізації нейромережних моделей

Модуль оптимізації структури нейромоделі призначений для спрощення структури побудованої нейронної мережі. Для оптимізації нейромоделі реалізовано: метод видалення найменших ваг, метод видалення нейронів, метод багатокритеріального еволюційного пошуку.

Модуль інтерфейсу користувача й візуалізації являє собою засіб взаємодії системи з кінцевим користувачем і містить різні екранні форми для задавання параметрів пошуку, вхідної вибірки даних і інших параметрів. Цей модуль дозволяє переглядати результати роботи у вигляді таблиць і графіків, а також дозволяє підготувати для кінцевого користувача різні звіти для друку.

Інтерфейс системи служить для полегшення взаємодії з користувачем. Після запуску системи відображається головна форма, у якій необхідно вибрати режим роботи, натиснувши відповідну кнопку. Після цього з'явиться форма методу, що дозволяє вирішувати обрану задачу. Основні інтерфейсні форми автоматизованої системи еволюційного синтезу й оптимізації нейромережних моделей відображено на рис. 9.5.

Схему функціонування автоматизованої системи еволюційного синтезу й оптимізації нейромережних моделей зображено на рис. 9.6.

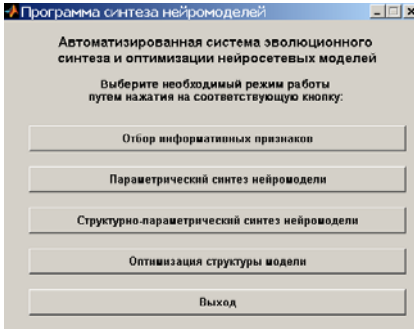
При роботі з програмою користувач, використовуючи інтерфейсні форми, задає вибірку даних, що характеризують досліджуваний об'єкт, процес або систему, а також встановлює основні параметри пошуку.

Після цього в залежності від розв'язуваної задачі відбувається звертання до функціональних блоків і модулів (блок відбору інформативних ознак, модуль параметричного синтезу нейромоделей, модуль структурного синтезу нейромоделей, модуль оптимізації структури побудованої нейромоделі), які входять до єдиної бібліотеки функцій.

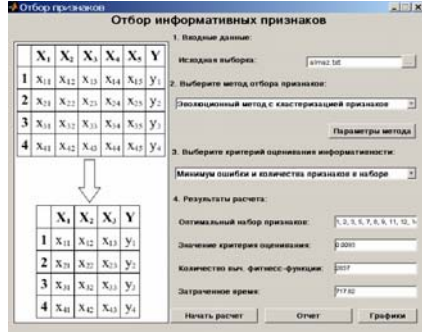
Бібліотека функцій являє собою комбінацію стандартних функцій пакету Matlab і функцій, реалізованих у даній системі.

Після закінчення роботи функціональних модулів результати виводяться на екран у вигляді звіту, що містить значення оптимізованих параметрів і значення критеріїв оптимальності. За бажанням користувача система може відобразити графіки зміни різних критеріїв. Результати роботи програми можуть бути записані у файл.

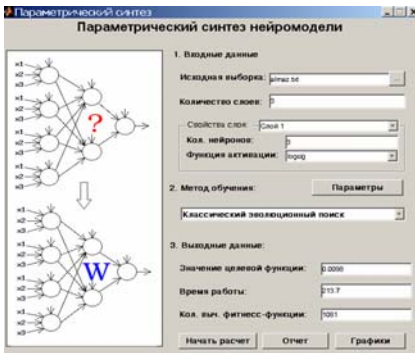
З метою перевірки ефективності еволюційних методів побудови нейромережних моделей вирішувався ряд завдань технічної й біомедичної діагностики, автоматичної класифікації образів [54, 76, 78, 81, 87, 96, 102, 103]. Процес вирішення наведених завдань полягав у послідовному виконанні таких етапів: відбір інформативних ознак, структурно-параметричний синтез, оптимізація побудованої нейромоделі. На кожному етапі використовувалися різні методи з метою порівняння розроблених методів з існуючими аналогами.



а)



б)



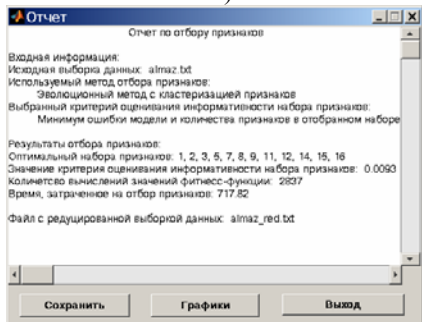
в)



г)



д)



е)

Рисунок 9.5 – Основні діалогові форми системи:
 а) головна форма; б) форма відбору інформативних ознак;
 в) форма параметричного синтезу моделей;
 г) форма структурно-параметричного синтезу моделей;
 д) форма оптимізації структури моделі; е) приклад звіту

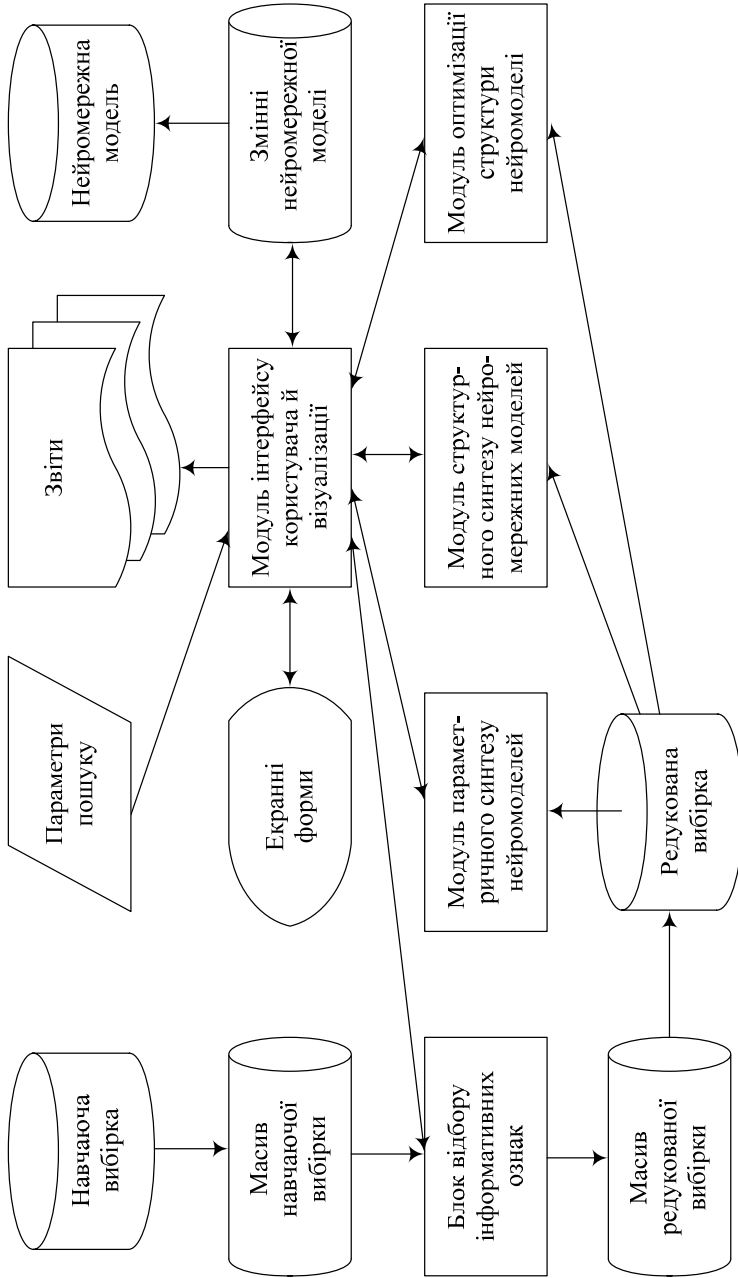


Рисунок 9.6 – Схема функціонування автоматизованої системи еволюційного синтезу й оптимізації нейромережних моделей

Результати експериментів показали, що запропоновані еволюційні методи відбору ознак і синтезу нейромоделей за рахунок використання додаткової інформації про досліджуваний об'єкт дозволяють скоротити час, необхідний на пошук, а нейромоделі, побудовані з їхньою допомогою, містять найбільш значущі міжнейронні зв'язки й зв'язки між вхідними ознаками й нейронами. Тому синтезовані моделі є більш простими, оскільки мають значно меншу кількість зв'язків при аналогічній кількості, нейронів і шарів.

ЛІТЕРАТУРА ДО ЧАСТИНИ II

1. Cantu-Paz E. Efficient and accurate parallel genetic algorithms. – Massachusetts: Kluwer Academic Publishers, 2001. – 162 p.
2. Coello C. A short tutorial on evolutionary multiobjective optimization // Evolutionary Multi-Criterion Optimization: Proceeding of the International Conference EMO2001 (7–9 March 2001). – Zurich: Springer-Verlag, 2001. – P. 21–40.
3. Cohen S., Ruppim E., Dror G. Feature selection based on the shapley value // Artificial Intelligence: Proceedings of the International Joint Conferences IJCAI'2003 (9-15 August 2003). – Acapulco: CAI, 2003. – P. 665–670.
4. Cvetkovic D., Coello C. Human preferences and their applications in evolutionary multiobjective optimization // IEEE Transactions on Evolutionary Computation. – 2002. – № 6. – P. 42–57.
5. Dash M., Liu H. Feature selection for classification // Intelligent Data Analysis. – 1997. – №1. – P. 131–156.
6. Dong M., Kothari R. Feature subset selection using a new definition of classifiability // Pattern Recognition Letters. – 2003. – №24. – P. 1215–1225.
7. Eiben A. E., Raue P. E., Ruttkay Z. Genetic algorithms with multi-parent recombination // Parallel Problem Solving from Nature: Proceedings of the 3rd International Conference. – Berlin: Springer-Verlag, 1994. – P. 78–87.
8. Eiben A. E., Hintering R., Michalewicz Z. Parameter control in evolutionary algorithms // IEEE Transactions on Evolutionary Computation. – 1999. – № 3. – P. 124–141.
9. Elalfi A. E., Haqueeb R., Elalami M. E. Extracting rules from trained neural network using genetic algorithm for managing E-business // Applied Soft Computing. – 2004. – №4. – P. 65–77.
10. Emmanouilidis C., Hunter A., MacIntyre J., Cox C. Multiple-criteria genetic algorithms for feature selection in neurofuzzy modeling // International Joint Conference on Neural Networks: Proceedings of the International Conference IJCNN'99 (10-16 July 1999). – Washington: IEEE Press, 1999. – P. 4387–4392.
11. Fogarty T. C. Varying the probability of mutation in the genetic algorithm // Genetic Algorithms: Proceedings of the Third International Conference. – San Mateo: Morgan Kaufmann, 1989. – P. 104–109.
12. Gen M., Cheng R. Genetic algorithms and engineering design. – New Jersey: John Wiley & Sons, 1997. – 352 p.
13. Goldberg D. E., Richardson J. Genetic algorithms with sharing for multimodal function optimization // Genetic Algorithms: Proceedings of the Second International Conference (3–5 June 1987). – Cambridge: Lawrence Erlbaum Associates, 1987. – P. 41–49.
14. Grefenstette J. J. Optimization of control parameters for genetic algorithms // IEEE Transactions on Systems, Man, and Cybernetics. – 1986. – № 16. – P. 122–128.

15. Gremling J. T., Passino K. M. Genetic adaptive parameter estimation // *International Journal of Intelligent Control and Systems*. – 1999. – № 3. – P. 465–503.
16. Guyon I., Elisseeff A. An introduction to variable and feature selection // *Journal of Machine Learning Research*. – 2003. – №3. – P. 1157–1182.
17. Handl J., Knowles J. Semi-supervised feature selection via multiobjective optimization // *Neural Networks: Proceedings of the International Joint Conference IJCNN'2006 (16–21 July 2006)*. – Vancouver: IEEE Press, 2006. – P. 6351–6358.
18. Haupt R., Haupt S. *Practical genetic algorithms*. – New Jersey: John Wiley & Sons, 2004. – 261 p.
19. Holland J. H. *Adaptation in natural and artificial systems*. – Ann Arbor: The University of Michigan Press, 1975. – 97 p.
20. Hyvarinen A., Karhunen J., Oja E. *Independent component analysis*. – New York: John Wiley & Sons, 2001. – 481 p.
21. Ishibuchi H., Nakashima T. Multi-objective pattern and feature selection by a genetic algorithm // *Genetic and Evolutionary Computation: Proceedings of the International Conference GECCO-2000 (8–12 July 2000)*. – Las Vegas: Morgan Kaufmann, 2000. – P. 1069–1076.
22. Jin Y., Okabe T., Sendhoff B. Adapting weighted aggregation for multiobjective evolution strategies // *Genetic and Evolutionary Computation: Proceedings of the International Conference GECCO-2001 (7–11 July 2001)*. – San Francisco: Morgan Kaufmann, 2001. – P. 1042–1049.
23. Jong K., Spears. W. On the state of evolutionary computation // *Genetic Algorithms: Proceedings of the Fifth International Conference*. – San Mateo: Morgan Kaufman, 1993. – P. 618–623.
24. Kuo R. J., Chen J. A. A decision support system for order selection in electronic commerce based on fuzzy neural network supported by real-coded genetic algorithm // *Expert Systems with Applications*. – 2004. – №26. – P. 141–154.
25. Lam H. K., Ling S. H., Leung F. H., Tam P. K. Function estimation using a neural-fuzzy network and an improved genetic algorithm // *International Journal of Approximate Reasoning*. – 2004. – №36. – P. 243–260.
26. Lee M. A., Esbensen H. Fuzzy multi-objective genetic systems for intelligent systems design tools and components // *Fuzzy Evolutionary Computation*. – 1997. – № 1. – P. 57–80.
27. Muhlenbein H., Schlierkamp-Voosen D. Predictive models for the breeder genetic algorithm // *Evolutionary Computation*. – 1993. – № 1. – P. 25–50.
28. Pendharkar P. C., Rodger J. A. An empirical study of impact of cross-over operators on the performance of non-binary genetic algorithm based neural approaches for classification // *Computers and Operations Research*. – 2004. – №31. – P. 481–498.

29. Poli R. Exact schema theory for genetic programming and variable-length genetic algorithms with one-point crossover // *Genetic Programming and Evolvable Machines*. – Las Vegas: Morgan Kaufmann. – 2001. – P. 469–476.

30. Radcliffe N.J. Equivalence class analysis of genetic algorithms // *Complex Systems*. – 1990. – № 2. – P. 183–205.

31. Rasheed K., Hirsh H. Informed operators: speeding up genetic-algorithm-based design optimization using reduced models // *Genetic and Evolutionary Computation: Proceedings of the International Conference GECCO-2000 (8-12 July 2000)*. – Las Vegas: Morgan Kaufmann, 2000. – P. 628–635.

32. Roshdy S. Y., Carla N. P. Combining genetic algorithms and neural networks to build a signal pattern classifier // *Soft Computing Systems – Design, Management and Applications: Proceedings of the International Conference (1–4 December 2002)*. – Santiago: IOS Press, 2002. – P. 735–744.

33. Sarimveis H., Alexandridis A., Mazarakis S., Bafas G. A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms // *Computers and Chemical Engineering*. – 2004. – №28. – P. 209–217.

34. Siedlecki W., Sklansky J. A note on genetic algorithms for large-scale feature selection // *Pattern Recognition Letters*. – 1989. – №10. – P. 335–347.

35. Subbotin S., Oleynik An. Entropy based evolutionary search for feature selection // *The experience of designing and application of CAD systems in Microelectronics: Proceedings of the IX International Conference CADSM-2007 (20–24 February 2007)*. – Lviv: Publishing house of Lviv Polytechnic, 2007. – P. 442–443.

36. Subbotin S., Oleynik An. The feature selection method based on the evolutionary approach with a fixation of a search space // *Modern problems of radio engineering, telecommunications and computer science: Proceedings of the IX International Conference TCSET'2006 (21–25 February 2006)*. – Lviv: Publishing house of Lviv Polytechnic, 2006. – P. 574–575.

37. Subbotin S., Oleynik An. The multi objective evolutionary feature selection // *Modern problems of radio engineering, telecommunications and computer science: Proceedings of the IX International Conference TCSET'2008 (19–23 February 2008)*. – Lviv: Publishing house of Lviv Polytechnic, 2008. – P. 115–116.

38. Subbotin S.A., Oleynik An.A, Oleynik Al.A. Feature selection based on evolutionary approach // *Intelligent Systems Design: Neural Networks, Fuzzy Logic, Evolutionary Computation, Swarm Intelligence, and Complex Systems: Proceedings of the 16-th International Conference ANNIE 2006 (5–8 November 2006)*. – Missouri-Rolla: ASME Press, 2006. – P. 125–130.

39. *The practical handbook of genetic algorithms. Applications* / Ed. L.D. Chambers. – Florida: CRC Press, 2000. – Vol. I. – 520 p.

40. *The practical handbook of genetic algorithms. Complex coding systems* / Ed. L.D. Chambers. – Florida: CRC Press LLC, 2000. – Vol. III. – 659 p.

41. The practical handbook of genetic algorithms. New frontiers / Ed. L.D. Chambers. – Florida: CRC Press, 2000. – Vol. II. – 421 p.
42. Ursem R. K. Diversity-guided evolutionary algorithms // Parallel Problem Solving from Nature: Proceedings of the Seventh International Conference (7–11 September 2002). – Berlin: Springer-Verlag, 2002. – P. 462–471.
43. Vafaie H., Imam I. Feature selection methods: genetic algorithms vs. greedy-like search // Fuzzy and Intelligent Control Systems: Proceedings of the Third International Conference (22–26 March 1994). – Louisville: IST, 1994. – P. 381–390.
44. Vafaie H., Jong K. Genetic algorithms as a tool for feature selection in machine learning // Tools with Artificial Intelligence: Proceeding of the 4th International Conference TAI'92 (10–13 November 1992). – Arlington: IEEE Press, 1992. – P. 200–204.
45. Whitley D. A genetic algorithm tutorial // Statistics and Computing. – 1994. – № 4. – P. 65–85.
46. Yang J., Honavar V. Feature subset selection using a genetic algorithm // Genetic Programming: Proceedings of the II International Conference GP-97 (13–16 July 1997). – Stanford: Publishing house of Stanford University, 1997. – P. 58–63.
47. Yang J., Tiyyagura A., Chen F., Honavar V. Feature subset selection for rule induction using RIPPER // Genetic and Evolutionary Computation: Proceedings of the International Conference GECCO-1999 (13–17 July 1999). – Las Vegas: Morgan Kaufmann, 1999. – P. 981–988.
48. Yao X. An overview of evolutionary computation // Chinese Journal of Advanced Software Research. – 1996. – № 3. – P. 12–29.
49. Yao X. Evolving artificial neural network // Proceedings of the IEEE. – 1999. – № 9(87). – P. 1423–1447.
50. Zhang P., Verma B., Kumar K. Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection // Pattern Recognition Letters. – 2005. – №7. – P. 909–919.
51. Zhengjun L., Aixia L., Changyao W., Zheng N. Evolving neural network using real coded genetic algorithm for multispectral image classification // Future Generation Computer Systems. – 2004. – №20. – P. 1119–1129.
52. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: Исследование зависимостей. – М.: Финансы и статистика, 1985. – 487 с.
53. Байдык Т. Н., Куссуль Э. М. Применение генетических алгоритмов для оптимизации нейросетевых распознающих устройств // Кибернетика и системный анализ. – 1999. – №5 – С. 23–32.
54. Богуслаев А. В., Олейник Ан. А., Пухальская Г. В., Субботин С. А. Отбор геометрических параметров и синтез модели частотной характеристики лопаток компрессора на основе эволюционного поиска // Вісник двигунобудування. – 2006. – № 1. – С. 14–17.

55. Букатова И. Л., Михасев Ю. И., Шаров А. М. Эвоинформатика: Теория и практика эволюционного моделирования. – М.: Наука, 1991. – 206 с.
56. Васильев В. И., Ильясов Б. Г. Интеллектуальные системы управления с использованием генетических алгоритмов // Информационные технологии. – 2002. – №12 – С. 29–34.
57. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г. К. Вороновский, К. В. Махотило, С. Н. Петрашев, С. А. Сергеев. – Харьков: Основа, 1997. – 111 с.
58. Глибовец Н. Н., Медвидь С. А. Генетические алгоритмы и их использование для решения задачи составления расписания // Кибернетика и системный анализ. – 2003. – № 1. – С. 95–108.
59. Горелик А. Л. Скрипкин В. А. Методы распознавания. – М.: Высшая школа, 2004. – 261 с.
60. Дисперсионная идентификация / Под ред. Н. С. Райбмана. – М.: Наука, 1981. – 336 с.
61. Дли М. И. Нечеткая логика и искусственные нейронные сети. – М.: Физматлит, 2001. – 225 с.
62. Дубровін В. І, Субботін С. О. Методи оптимізації та їх застосування в задачах навчання нейронних мереж: Навчальний посібник. – Запоріжжя: ЗНТУ, 2003. – 136 с.
63. Дюк В., Самойленко А. Data mining: учебный курс. – СПб.: Питер, 2001. – 368 с.
64. Емельянов В. В. Курейчик В. В. Курейчик В. М. Теория и практика эволюционного моделирования. – М.: ФИЗМАТЛИТ, 2003. – 432 с.
65. Жамбю М. Иерархический кластер-анализ и соответствия. – М.: Финансы и статистика, 1988. – 342 с.
66. Ивахненко А. Г., Юрачковский Ю. П. Моделирование сложных систем по экспериментальным данным. – М.: Радио и связь, 1987. – 118 с.
67. Интеллектуальные средства диагностики и прогнозирования надежности авиадвигателей: Монография / В. И. Дубровин, С. А. Субботин, А. В. Богуслаев, В. К. Яценко. – Запорожье: ОАО «Мотор-Сич», 2003. – 279 с.
68. Кисляков А. В. Генетические алгоритмы: математический анализ некоторых схем репродукции // Информационные технологии. – 2000. – № 12. – С. 9–14.
69. Кисляков А. В. Генетические алгоритмы: операторы скрещивания и мутации // Информационные технологии. – 2001. – № 1. – С. 29–34.
70. Круглов В. В., Борисов В. В. Искусственные нейронные сети: Теория и практика. – М.: Горячая линия-Телеком, 2001. – 382 с.
71. Куприянов М. С., Матвиенко Н. И. Генетические алгоритмы и их реализации в системах реального времени // Информационные технологии. – 2001. – №1. – С. 17–21.
72. Курейчик В. М. Генетические алгоритмы: Монография. – Таганрог: ТРТУ, 1998. – 242 с.

73. Минаев Ю. Н., Филимонова О. Ю., Лиес Б. Методы и алгоритмы идентификации и прогнозирования в условиях неопределенности в нейросетевом логическом базисе. – М.: Горячая линия-Телеком, 2003. – 205 с.
74. Олейник Ан. А. Эволюционный метод обучения нейромоделей // Интеллектуальный анализ информации: Сборник трудов восьмой международной конференции ИАИ-2008 (14–17 мая 2008 р.). – К: Просвіта, 2008. – С. 351–361.
75. Олейник Ан. А., Субботин С. А. Метод эволюционного поиска с сокращением размера популяции // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: Матеріали Міжнародної науково-практичної конференції (13–15 квітня 2006 р.). – Запоріжжя: ЗНТУ, 2006. – С. 179–181.
76. Олейник Ал. А., Олейник Ан. А., Субботин С. А., Яценко В. К. Синтез моделей коэффициента упрочнения деталей авиадвигателей после алмазного выглаживания на основе оптимизационного подхода // Вісник двигунобудування. – 2005. – №3. – С. 25–30.
77. Олейник Ан. А. Автоматизированная система построения нейромоделей на основе эволюционного подхода // Радіоелектроніка і молодь в ХХІ сторіччі: Матеріали 12-го міжнародного молодіжного форуму (1–3 квітня 2008 р.). – Харків: ХНУРЕ, 2008. – Ч. 2. – С. 162.
78. Олейник Ан. А. Выбор системы информативных признаков для классификации транспортных средств на основе эволюционного поиска // Комп'ютерне моделювання та інтелектуальні системи: Збірник наукових праць / За ред. Д. М. Пізи, С. О. Субботіна. – Запоріжжя: ЗНТУ, 2007. – С. 134–146.
79. Олейник Ан. А. Синтез нейросетевых моделей на основе методов эволюционной оптимизации // Радіоелектроніка і молодь в ХХІ сторіччі: Матеріали 12-го міжнародного молодіжного форуму (1–3 квітня 2008 р.). – Харків: ХНУРЕ, 2008. – Ч. 2. – С. 316.
80. Олейник Ан. А. Эволюционный отбор заданного количества признаков // Радіоелектроніка і молодь в ХХІ сторіччі: Матеріали 11-го міжнародного молодіжного форуму (10–12 квітня 2007 р.). – Харків: ХНУРЕ, 2007. – Ч. 2. – С. 142.
81. Олейник Ан. А., Павленко Д. В., Субботин С. А. Определение влияния режимов высокоскоростного фрезерования на параметры нежестких деталей на основе эволюционного подхода // Вісник двигунобудування. – 2008. – № 1. – С. 84–90.
82. Олійник А. О. Виділення системи заданої кількості інформативних ознак на основі еволюційного пошуку // Тиждень науки: Збірник наукових праць. – Запоріжжя: ЗНТУ, 2007. – С. 59–61.
83. Пат. 18294 Україна, МПК G06F 19/00. Спосіб відбору інформативних ознак для діагностики виробів: Пат. 18294 Україна, МПК G06F 19/00 С. О. Субботін, А. О. Олійник (Україна); Запорізький національний техні-

чний університет. – № u200603087; Заявл. 22.03.06; Опубл. 15.11.06, Бюл. № 11. – 4 с.

84. Прикладная статистика: Классификация и снижение размерности: Справ. изд. / С. А. Айвазян, В. М. Бухштабер, И. С. Енюков, Л. Д. Мешалкин. – М.: Финансы и статистика, 1989. – 607 с.

85. Рассел С., Норвиг П. Искусственный интеллект: современный подход. – М.: Вильямс, 2006. – 1408 с.

86. Риоло Р. Л. Естественный отбор в мире битов // В мире науки. – 1992. – № 9. – С. 160–163.

87. Розробка математичного та інформаційного забезпечення інтелектуальної системи візуального контролю транспортних засобів: Звіт про НДР / ТОВ «МПА Групп». – № ДР 0106U012013. – Запоріжжя, 2007. – 259 с.

88. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. – Винница: УНІВЕРСУМ-Вінниця, 1999. – 320 с.

89. Ротштейн А. П., Митюшкин Ю. И. Извлечение нечетких баз знаний из экспериментальных данных с помощью генетических алгоритмов // Кибернетика и системный анализ. – 2001. – № 4. – С. 45–53.

90. Ротштейн А. П., Ракитянская А. Б. Решение задачи диагностики на основе нечетких отношений и генетического алгоритма // Кибернетика и системный анализ. – 2001. – № 6. – С. 162–170.

91. Руденко О. Г., Бодяньский Є. В. Штучні нейронні мережі. – Харків: Компанія СМІТ, 2006. – 404 с.

92. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. – М.: Горячая линия – Телеком, 2004. – 452 с.

93. Симанков В. С., Частикова В. А. Генетические алгоритмы и поиск оптимальных решений // Автоматизация и современные технологии. – 2003. – № 6. – С. 39–45.

94. Субботин С. А., Олейник Ан. А. Сравнение эволюционных методов при решении оптимизационных задач // Моделирование неравновесных систем: Материалы X Всероссийского семинара (12–14 октября 2007 г.). – Красноярск: ИВМ СО РАН, 2007. – С. 152–153.

95. Субботин С. А., Олейник Ан. А. Выбор набора информативных признаков для синтеза моделей объектов управления на основе эволюционного поиска с группировкой признаков // Искусственный интеллект. – 2006. – № 4. – С. 488–494.

96. Субботин С. А., Олейник Ан. А. Синтез нейросетевых моделей для выбора оптимальных режимов обработки деталей энергетических установок // Физика высоких энергий, ядерная физика и ускорители: Материалы 6-ой конференции (25–29 февраля 2008 р.). – Харьков: ННЦ ХФТИ, 2008. – С. 61–62.

97. Субботин С. А., Олейник Ан. А. Ускоренный метод эволюционного отбора признаков // Автоматика-2006: Тезисы доповідей тринадцятої міжнародної науково-технічної конференції (25–28 вересня 2006 р.). – Вінниця: Універсум-Вінниця, 2006. – С. 409.

98. Субботин С.А., Олейник Ан.А. Эволюционный синтез моделей сложных объектов и процессов // Радиоэлектроника и информатика. – 2007. – № 2. – С. 99–104.

99. Субботин С. А., Олейник Ан. А. Использование энтропии как меры неопределенности состояния объекта для эволюционного отбора признаков // Автоматика-2007: Тезисы доповідей чотирнадцятої міжнародної науково-технічної конференції (10–14 вересня 2007 р.). – Севастополь: СКУАЕтаП, 2007. – С. 100–102.

100. Субботин С. А., Олейник Ан. А. Критерии сравнения эволюционных методов синтеза нейромоделей // Научная сессия МИФИ–2007. IX Всероссийская научно-техническая конференция «Нейроинформатика-2007»: Сборник научных трудов (23–26 января 2007 г.). – М.: МИФИ, 2007. – Ч. 2. – С. 177–184.

101. Субботин С. А., Олейник Ан. А. Модифицированный оператор жадного кроссовера для эволюционного отбора информативных признаков // Нейроинформатика и ее приложения: Материалы XIV Всероссийского семинара (6–8 октября 2006 г.). – Красноярск: ИВМ СО РАН, 2006. – С. 116–118.

102. Субботин С. А., Олейник Ан. А. Синтез нейромоделей зависимости режимов высокоскоростного фрезерования деталей авиадвигателей от уровня вибрации // Научная сессия МИФИ–2008. X Всероссийская научно-техническая конференция «Нейроинформатика-2008»: Сборник научных трудов (22–25 января 2008 г.). – М.: МИФИ, 2008. – Ч. 2. – С. 160–170.

103. Субботин С. А., Яценко В. К., Олейник Ан. А., Олейник Ал. А. Эволюционный синтез нейромоделей коэффициента упрочнения лопаток авиадвигателей // Научная сессия МИФИ–2006. VIII Всероссийская научно-техническая конференция «Нейроинформатика-2006»: Сборник научных трудов (26–28 января 2006 г.). – М.: МИФИ, 2006. – Ч. 3. – С. 141–148.

104. Субботин С. О., Олейник А. О. Поліморфний еволюційний пошук з кластеризацією хромосом для відбору інформативних ознак // Автоматика-2006: Матеріали тринадцятої міжнародної науково-технічної конференції (25–28 вересня 2006 р.). – Вінниця: Універсум-Вінниця, 2006. – С. 448–451.

105. Хайкин С. Нейронные сети: полный курс. – СПб: Издательский дом «Вильямс», 2005. – 1104 с.

106. Холланд Х. Д. Генетические алгоритмы // В мире науки. – 1992. – № 9. – С. 32–40.

107. Эволюционные методы компьютерного моделирования: Монография / А. Ф. Верлань, В. Д. Дмитриенко, Н. И. Корсунов, В. А. Шорох. – К: Наукова думка, 1992. – 256 с.

ЧАСТИНА III

ІНТЕЛЕКТУАЛЬНІ МЕТОДИ МУЛЬТИАГЕНТНОЇ ОПТИМІЗАЦІЇ (SWARM INTELLIGENCE)

РОЗДІЛ 10

ОСНОВИ SWARM INTELLIGENCE

Новим напрямом розвитку методів штучного інтелекту є мультиагентні методи інтелектуальної оптимізації, що моделюють колективний інтелект (КІ) суспільних тварин, комах та інших живих істот, – методи Swarm Intelligence. Даний напрям штучного інтелекту є молодим і ще мало дослідженим, проте мультиагентні методи показують гарні результати при вирішенні різних задач оптимізації, що говорить про перспективність його розвитку.

Головною особливістю мультиагентних методів КІ є їхня біонічна природа. При цьому слід зазначити, що дані методи моделюють поведінку груп різних соціальних тварин, комах і інших живих істот, яка дозволяє цим групам вирішувати різні важкі практичні задачі в природі, що свідчить про ефективність їхньої поведінки, а, отже, і про ефективність роботи даних методів. Тому актуальним є вивчення методів, можливих областей їхнього застосування, а також розробка нових математичних моделей, заснованих на поведінці колективних тварин, в аспекті вирішення оптимізаційних задач, до яких дані методи ще не застосовувалися.

При вирішенні оптимізаційних задач досить перспективним є використання мультиагентних методів [1, 2]. При реалізації даних методів використовується парадигма агентно-орієнтованого програмування, що сприяє підвищенню продуктивності обчислень.

До інтелектуальних методів мультиагентної оптимізації, заснованих на моделюванні суспільного інтелекту, відносяться: метод мурашиних колоній (Ant Colony Optimization, ACO) [3–5], метод бджолоїної колонії (Bee Colony Optimization, BCO) [6, 7], оптимізація за допомогою рою часток (Particle Swarm Optimization, PSO) [8] та інші методи [9]. Дані методи вже ефективно застосовуються для вирішення різних задач оптимізації. Наприклад, ACO застосовується для вирішення задачі комівояжера [10, 11], задачі календарного планування [12], відбору інформативних ознак [13–15], кластеризації [16] і ін. [17–21], BCO – для вирішення задачі календарного планування [22], вирішення задачі комівояжера [23, 24], вирішення транспортного завдання [25] і ін. [26, 27].

10.1 Мультиагентні системи

Мультиагентну систему (МАС) в загальному випадку можна подати у вигляді множини з трьох елементів [29, 30]: Агенти, Середовище, Зв'язки між Середовищем та Агентами:

$$\text{МАС} = \langle \text{Агенти}, \text{Середовище}, \text{Зв'язки} \rangle.$$

Кожен i -ий Агент_i описується за допомогою множини чотирьох елементів: Стан_i , Вхід_i , Вихід_i , Процес_i :

$$\text{Агент}_i = \langle \text{Стан}_i, \text{Вхід}_i, \text{Вихід}_i, \text{Процес}_i \rangle,$$

де Стан_i – це множина змінних, що повністю визначають агента; Вхід_i та Вихід_i – підмножина Стану_i , елементи яких пов'язані із середовищем.

Процес_i – автономний метод, що виконує відповідні зміни над Станом . «Автономний метод» означає, що даний метод викликається без будь-якого зовнішнього впливу.

Середовищем являється множина з двох елементів:

$$\text{Середовище} = \langle \text{Стан}_c, \text{Процес}_c \rangle,$$

де індекс c інформує про те, що дані елементи відносяться до середовища, а не до будь-якого іншого агента.

Важливою особливістю такого подання Середовища є те, що Середовище є саме по собі активним, оскільки воно містить свій власний Процес_c , котрий може змінювати Стан_c незалежно від агентів, що входять в це Середовище .

Вхід_i та Вихід_i різноманітних агентів пов'язані з елементами Стану_c , але середовище не відрізняє, які з елементів Стану_c знаходяться в залежності з ними. Відсутність Входу та Виходу у Середовища означає, що воно, на відміну від агентів, є необмеженим. Якби в Середовище були додані Вхід та Вихід , то це означало б, що середовище обмежене та є фактично агентом високого рівня. Таким чином, може моделюватися взаємодія агентів високого та низького рівнів.

У відповідності до наведеної концепції подання мультиагентних систем можуть створюватися різноманітні типи мультиагентних систем, при цьому така концепція організації систем може бути використана не тільки в програмних (штучних) системах, а й в природних системах. Така класифікація мультиагентних систем [28] подана на рис. 10.1.

У даному розділі розглядається розподілений штучний інтелект, що базується на колективному інтелекті [31]. Робочі групи у людей описані в [32, 33], соціолобіологія у тварин – в [34, 35], мультироботні системи – в [36], штучне життя – в [37, 38].

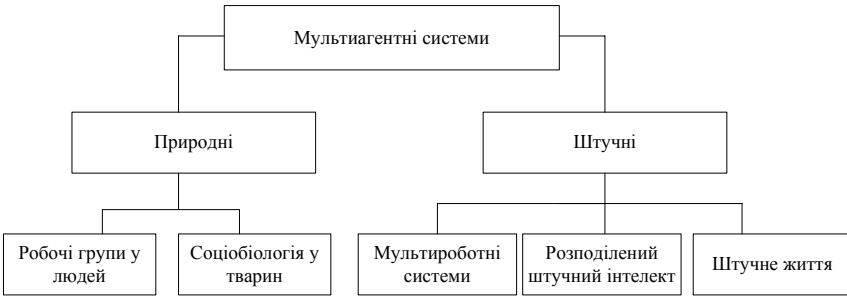


Рисунок 10.1 – Класифікація мультиагентних систем

10.2 Основні принципи колективного інтелекту

До мультиагентних розподілених систем штучного інтелекту відносяться мультиагентні методи інтелектуальної оптимізації (методи колективного інтелекту, Swarm Intelligence). Дані методи мають біонічну природу, тобто вони засновані на моделюванні поведінки комах, птахів, тварин і т.п., поведінка яких носить колективний характер, за рахунок чого досягається, так званий, колективний інтелект.

Тому для визначення основних принципів колективного інтелекту досить розглянути принципи поведінки різних колективних комах. До колективних комах відносяться комахи, які живуть колоніями, наприклад, мурахи, бджоли, терміти, деякі види ос і т.п. В основі поведінки колоній таких комах лежить самоорганізація [39, 40].

Самоорганізація – множина динамічних механізмів, відповідно до яких система регулюється на глобальному рівні за рахунок взаємодії її компонентів на нижньому рівні без прямої взаємодії між цими компонентами. Самоорганізація базується на чотирьох складових.

1. Позитивний зворотний зв'язок: досягається за рахунок виконання простих поведінкових емпіричних прийомів, які забезпечують створення рішень. Наприклад, при фуражировці мурах збільшення феромонів на шляху до джерела їжі – це різновид позитивного зворотного зв'язку, оскільки в такий спосіб створюється мережа з можливих шляхів до джерела їжі. У бджіл позитивний зворотний зв'язок полягає в тому, що бджоли, ґрунтуючись на отриманій інформації від інших бджіл, починають летіти до зазначеного джерела ресурсів.

2. Негативний зворотний зв'язок: урівноважує позитивний зворотний зв'язок, що веде до стабілізації колективної поведінки. При фуражировці мурах негативний зворотний зв'язок може бути викликаний такими фак-

торами: обмежена кількість фуражирів, виснаження джерела їжі, випаровування феромону. У бджіл негативний зворотний зв'язок полягає в тому, що бджола, грунтуючись на інформації, отриманій від інших бджіл, може вирішити, що знайдене нею джерело гірше.

3. Нестійкість позитивного зворотного зв'язку: поведінка соціальних комах засновано на стохастичних правилах. Випадкові відхилення в рішеннях – основа для нових рішень. Крім того, хаотичність може бути корисною, оскільки це дозволяє колонії виявити нові рішення. Наприклад, фуражир, що збився з курсу, може знайти нові, неопрацьовані джерела харчування, і потім привести за собою інших робочих особин для розробки цього джерела їжі.

4. Вимога множинної взаємодії між особинами, за рахунок чого досягається поява нових гарних і витривалих рішень. Множинність взаємодії у бджіл полягає в тому, що інформація про джерело ресурсів, знайденого однією бджолою, доступна для всіх інших у вулику за допомогою виконання, так званого, виляючого танцю.

Крім основних складових, самоорганізація характеризується певними властивостями.

1. Динамічність: формування рішень вимагає постійної взаємодії між особинами колонії й навколишнім середовищем. Ця взаємодія забезпечує позитивний зворотний зв'язок, за рахунок якого створюються колективні групи й можливості для їхнього проживання, у той час як негативний зворотний зв'язок знижує вплив прямого зворотного зв'язку.

2. Емерджентність: властивості систем, що самоорганізуються, є більш складними, ніж властивості окремих агентів, що входять у неї. Такі властивості системи є результатом нелінійної комбінації взаємодій між агентами.

3. Нелінійні взаємодії між агентами ведуть також до розгалуження в поведінці систем, що самоорганізуються. Розгалуження (біфуркація) – поява нових стійких рішень внаслідок зміни певних параметрів системи. За рахунок цього досягається якісне поліпшення в колективній поведінці.

4. Мультистійкість – для заданих встановлених параметрів система може досягти різних стійких станів в залежності від початкового стану й випадкових відхилень.

Для розуміння колективного інтелекту необхідно визначити функції, які виконують соціальні комахи в процесі вирішення різних задач. Можна виділити чотири функції такого виду: координація, кооперація, колективне прийняття рішень, спеціалізація. Вони не є взаємно виключними, а навпаки – спільно вносять вклад у досягнення поставлених перед колонією цілей.

Координація – відповідна організація задач окремих індивідів у часі й просторі, що дозволяє вирішити виниклу перед колонією проблему. Дана функція приводить до певного просторово-часового розподілу особин, їхніх дій та/або результатів їхніх дій для досягнення поставленої мети.

Наприклад, координація виконується при організації переміщення гнізда бджіл або рою сарани [41, 42]. У цьому випадку взаємодії між окремими особинами приводять до синхронного (часова організація) і спрямованого (просторова організація) руху особин до певної мети.

Координація також проявляється при фуражировці у мурах шляхом застосування феромону, що залишають мурахи при своєму переміщенні. За рахунок виділення феромону створюються мережі, які приводять до просторової організації у фуражировці [43, 44].

Як останній приклад координації можна навести будівельну діяльність у колоніях комах. У процесі будівництва гнізда в деяких різновидах соціальних ос [45–48] або термітів [49, 50] за рахунок процесів стигмерджі (stigmergy) особини поліпшують отримане гніздо (просторова організація), попередньо досягнуте (часова організація) іншими особинами.

Кооперація досягається за рахунок того, що особини виконують разом загальну задачу, що не могла б бути вирішена окремим індивідом. Індивіди повинні об'єднувати свої зусилля, щоб успішно вирішити загальну задачу, що перебуває поза межами можливостей окремих індивідів.

Кооперація спостерігається в процесі видобутку їжі, коли окремого індивіда недостатньо, щоб доставити їжу. Відомо багато прикладів спільного транспортування їжі різними видами мурах: *Oecophylla longinoda* [51], *Eciton burchelli* [52], *Formica* [53, 54]. Таке транспортування їжі може бути дуже ефективним при доставці їжі в гніздо. Наприклад, мурахи *Pheidologeton diversus* спільно можуть транспортувати в десять разів більше ваги, ніж якби вони робили це окремо одне від одного [55].

Колективне прийняття рішень відноситься до механізмів, які спрацьовують, коли колонія зіштовхується із проблемою вибору. Цей механізм завершується колективним вибором одного з можливих рішень.

Наприклад, медоносні бджоли обирають більш продуктивні ділянки із квітками шляхом вербування незайнятих бджіл за допомогою виляючого танцю, виконуваним фуражиром, що повернувся від джерела нектару [56].

Спеціалізація полягає в тому, що різні дії виконуються окремими спеціалізованими групами індивідів. Наприклад, фуражировка або догляд за потомством [57, 58, 35]. Така спеціалізація може здійснюватися як за рахунок поведінкової диференціації, так і залежно від віку індивіда.

У співтоваристві медоносних бджіл праця особин є чітко розподіленою. Одні з них добувають і приносять у вулик їжу, інші – забирають її та складають, треті – чистять комірочки, четверті – крильцями, як вентилятором, провітрюють вулик, п'яті – несуть вартову службу, чергуючи при вході, і не пропускають ворогів. У родині існують розвідники для пошуку їжі, няньки для догляду за малятами, бджоли-прибиральники, гробарі, які виносять із вулика мертвих. Бджоли-

солдати організовані в спеціальні військові підрозділи, які несуть службу по захисту вулика від викрадачів меду.

Таким чином, організація колективної поведінки у соціальних комах може бути розглянута як сукупність чотирьох складових: координація, кооперація, колективне прийняття рішень і спеціалізація. Кожна із цих складових проявляється на колективному рівні за рахунок непрямой взаємодії між особинами. Така взаємодія забезпечує можливість оцінювання зовнішньої інформації відповідно до двох основних напрямків.

1. Координація й спеціалізація забезпечують формування просторових, часових і соціальних зв'язків, що виникають у процесі роботи колонії. Координація регулює просторово-часове розміщення індивідів, у той час як за рахунок спеціалізації досягається розподіл їхніх дій.

2. Кооперація й колективне прийняття рішень є інструментами, за допомогою яких колонія може реагувати на зміни в навколишньому середовищі. Колективне прийняття рішень надає механізми, які забезпечують рішення колонії, у той час як кооперація надає механізми, які дозволяють вийти за рамки можливості окремих індивідів.

Разом чотири складові в поведінці соціальних комах формують уявлення, що колонія в цілому планує свою роботу для досягнення поставлених цілей.

РОЗДІЛ 11

МЕТОД МУРАШИНИХ КОЛОНІЙ

11.1 Біологічні основи

Відомо, що комахи, включаючи мурах різних видів, термітів і деякі різновиди бджіл та ос, живуть у колоніях, складених з великої кількості взаємодіючих індивідумів.

Для рою комах необхідна деяка форма зв'язку, щоб співпрацювати при вирішенні задачі. Цей зв'язок між індивідумами колонії може бути більш-менш прямим, залежно від певних різновидів. Коли бджола знаходить джерело нектару, вона повідомляє напрямок і відстань до місця розташування, де вона знайшла нектар іншим бджолам, виконуючи характерний танець. Це приклад прямого зв'язку, оскільки інші бджоли повинні відчувати танець, що виконує одна бджола, щоб визначити місцезнаходження джерела нектару. Інші форми прямого зв'язку: збудження фізичним контактом або обміном їжею або рідиною.

Непрямий зв'язок між індивідумами колонії є більш тонким й вимагає, щоб один індивідум змінив навколишнє середовище таким чином, щоб це змінило поведінку індивідумів, що проходять через це змінене навколишнє середовище в майбутньому.

Один варіант, де існує цей тип екологічно викликаної дії в природі – це, коли терміти будують гніздо, що має дуже складну структуру й показує властивості подібні до керування кліматом. Щоразу, коли стадія будівництва закінчується, середовище робітника змінюється, і допускається наступна стадія роботи, що у свою чергу закінчується новим середовищем, і т. д.

Інший приклад непрямого зв'язку – накладення слідів феромону, виконуване деякими різновидами мурах. Феромони – хімічні речовини, вироблювані екзокринними залозами (або спеціальними клітинами) живих істот; виділяючись у зовнішнє середовище одними особинами, феромони впливають на поведінку, а іноді на ріст і розвиток інших особин того ж виду. Особливо важливу роль феромони відіграють у житті комах. У соціальних комах вони регулюють склад колонії й специфічну діяльність її членів. Мураха при кормодобуванні відзначає доріжку, залишаючи певну кількість феромону за собою, за рахунок чого спонукає йти по його шляху іншу мурашу, що також займається кормодобуванням.

Принцип зміни навколишнього середовища як засіб зв'язку для зміни поведінки називають *stigmergy*. Він є основним в організації в мурашиних колоніях. Хоча мурахи мають королеву, вона – спеціалізована мураха, що

є відповідальною тільки за те, щоб класти яйця й не має ніякої керуючої функції. Замість цього, мурашині колонії самоорганізуються.

Термін самоорганізація використовується, щоб описати складну поведінку, що виникає при взаємодії порівняно простих агентів. За допомогою самоорганізації мурахи здатні вирішувати складні задачі, з якими вони зіштовхуються щодня. Вигоди самоорганізації при вирішенні задачі особливо очевидні в її розподіленому характері. Мурашина колонія може ефективно підтримувати осмислену поведінку, навіть якщо велика кількість мурах нездатна до сприяння.

Щоб краще зрозуміти механізм і здатність мурашиних колоній сходиться до гарних рішень при пошуку найкоротшого шляху від гнізда до джерела їжі були проведені деякі експерименти [17, 18]. При проведенні експериментів колонії мурах Аргентини *Linep-ithema humile* було надано два шляхи ідентичної довжини, і після того, як пройшов певний час, було відзначено, що мурахи сходилися до однієї з доріжок, після чого фактично була виключена альтернатива. Щоб перевірити, чи сходився б цей вид мурах до найбільш короткого з множини шляхів, був проведений подвійний експеримент мосту, де мурахи повинні були вибирати двічі між коротким і довгим шляхами (рис. 11.1).

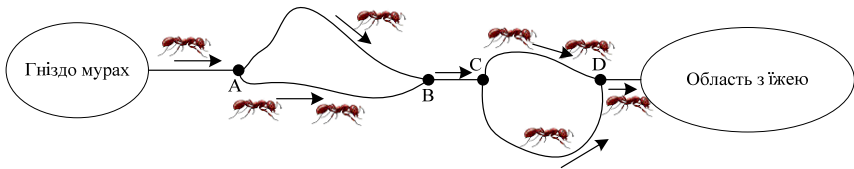


Рисунок 11.1 – Подвійний експеримент мосту

Мураха Аргентини є фактично сліпою, тому вона не має ніяких безпосередніх засобів ідентифікації короткого шляху. Однак, незважаючи на цей недолік, як показали результати експерименту, колонія мурах здатна до виявлення найкоротшого шляху, що з'єднує гніздо з областю, що містить їжу.

Спочатку, всі мурахи розташовуються на ділянці гнізда. Множина мурах відправляється від гнізда в пошуку їжі, кожна мураха залишає феромон на своєму шляху, і досягає першоївилки в точці А. Оскільки мурахи не мають ніякої інформації про те, який шлях обрати, тобто ніяка інша мураха ще не проходила тут раніше й не залишала за собою слід феромону, кожна мураха буде обирати йти вправо або вліво з рівною ймовірністю.

Згодом приблизно одна половина мурах вибере більш короткий шлях, інші – більш довгий маршрут до перетину з В. Мурахи, які оберуть

більш короткий шлях, досягнуть цього перетину швидше, і повинні будуть вирішити, який спосіб вибрати, щоб повернутися.

Знову немає ніякої інформації для мурах, щоб використати її як орієнтир, через це половина мурах, що досягла перетину В, повернеться до гнізда, у той час, як інші продовжуватимуть переміщення до області, що містить їжу.

Мурахи на більш довгій ділянці між перетинами А та В, незачепленій іншими мураками, яких вони зустріли раніше на прямій ділянці, досягають перетину В і також розділяться; однак, оскільки інтенсивність сліду феромону, розміщеного на шляху назад до гнізда є приблизно вдвічі більшою, ніж сліду феромону, що досягає області із їжею, більшість мурах повернеться до гнізда, прибуваючи туди в той же час як і інші мурахи, які вибрали довгий шлях. Цікаво, що, оскільки більше мурах тепер йшло по короткій ділянці між перетинами А і В у порівнянні з довгою, то наступні мурахи, що залишають гніздо, тепер уже будуть більш схильні вибрати короткий шлях, що є першим вдалим вибором при пошуку найкоротшого шляху.

Поведінка мурах на другому мосту в перетинах між С і D фактично ідентична поведінці, показаній раніше на першому мосту між перетинами А та В. В остаточному підсумку, більшість мурах досягне їжі й збере деяку кількість їжі, щоб принести назад до гнізда.

Досягаючи перетину D, мурахи віддадуть перевагу короткій ділянці через ті ж самі причини, що й мурахи, які починають переміщення із гнізда, і те ж саме відбувається знову в перетині В.

Оскільки кількість феромону в перетинах А та С на шляху назад до гнізда приблизно дорівнює сумі кількості феромону на цих двох ділянках від гнізда, то також найбільш ймовірним є найкоротший повний шлях від області із їжею назад до гнізда при виборі мурах при поверненні.

Оскільки мурахи постійно виділяють феромон під час свого переміщення короткий шлях постійно підсилюється все більшою кількістю мурах, у той час як кількість феромону, розміщеного надалі на інших альтернативних маршрутах настільки низька, що фактично всі мурахи будуть використовувати найкоротший шлях, тобто система зійдеться до самого короткого шляху через самопідсилення.

Слід відзначити, що феромон, використовуваний мураками, повільно випаровується через якийсь час, що не залишає сумнівів при поясненні подвійного експерименту мосту. Дійсно, довжини шляхів, які не були обрані протягом деякого часу, є незмінно більшими, і ці шляхи не будуть містити майже ніяких слідів феромону через їхній випар після певного проміжку часу, далі збільшується ймовірність вибору мураками коротких шляхів. Варто звернути увагу, що кількість феромону на самому короткому шляху має максимальну цінність, що досягнута більшою кількістю феромону, залишеною мураками.

11.2 Метод мурашиних колоній

Метод мурашиних колоній базується на моделюванні взаємодії декількох штучних аналогів мурах, що програмно подаються у вигляді інтелектуальних агентів, що є членами великої колонії. Модельовані агенти, переміщуючись по графу рішень, спільно вирішують проблему й допомагають іншим агентам у подальшій оптимізації рішення. Таким чином, оптимізаційна задача вирішується агентами, що знаходяться у непрямому зв'язку одне з одним. У методі мурашиних колоній такий зв'язок забезпечується завдяки моделюванню виділення феромонів агентам при переміщенні

Узагальнену схему роботи методу мурашиних колоній подано на рис. 11.2.

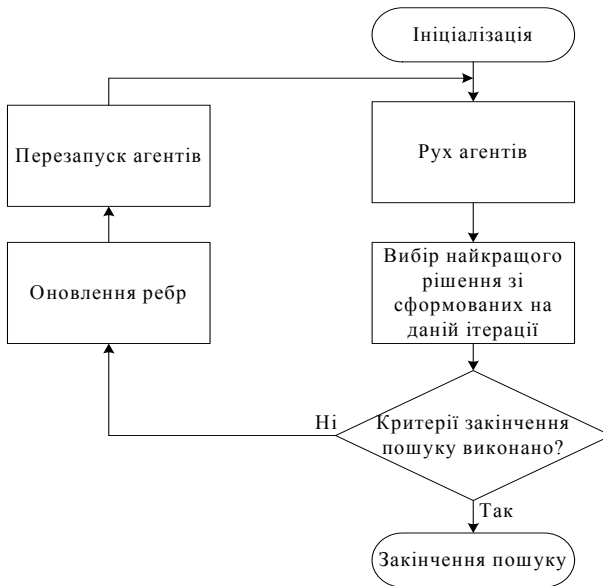


Рисунок 11.2 – Узагальнена схема роботи методу мурашиних колоній

Першою задачею, до якої був застосований метод мурашиних колоній, була задача комівояжера (Traveling Salesman Problem, TSP). Основною причиною, у зв'язку з якою була обрана ця задача, є те, що в ній необхідно знаходити найкоротший шлях, тому аналогія методу мурашиних колоній легко пристосовується для вирішення даної задачі.

Для вирішення даної задачі було розроблено кілька різних методів, заснованих на оптимізації за допомогою мурашиних колоній.

Першим методом був метод мурашиних систем (Ant System – AS) [59, 60]. Надалі цей метод послужив основою для багатьох інших методів, що працюють на принципі мурашиних колоній.

У методі мурашиних систем агент формує своє рішення в процесі переміщення від одного вузла до іншого на графі рішень. Метод працює до виконання t_{max} ітерацій. На кожній ітерації агенти формують свої рішення за n кроків, на кожному з яких застосовується правило вибору наступного вузла – правило вибору агентом, що перебуває у вузлі r , наступного вузла для переміщення в нього.

В [4, 59–62] було запропоновано три методи мурашиних систем, відмінних між собою способом оновлення шляхів – ребер. Ці методи називалися: щільнісний (ant-density), кількісний (ant-quantity) і циклічний (ant-cycle) методи мурашиних систем. У щільнісному й кількісному методах агенти залишали феромони в процесі формування рішення, у той час, як у циклічному методі агенти залишали феромони після закінчення переміщення, тобто після формування рішення.

Проведені експерименти по вирішенню тестових задач [4, 59, 60] показали, що циклічний метод мав значно кращі результати в порівнянні з іншими двома. У зв'язку із цим, два гірших методи було відкинуто. Тому, надалі, під методом мурашиних колоній розуміється саме циклічний метод мурашиних систем.

Кількість феромонів $\tau_{ru}(t)$, що залишає агент, відповідає дузі (r, u) – це кількість, що характеризує перевагу вибору даного ребра в порівнянні з іншими при переміщенні. Інформація про феромони на ребрах змінюється в процесі складання рішень. При цьому кількість феромонів, що залишається агентами, пропорційна якості рішення, сформованого відповідним агентом: чим менше шлях, тим більше буде залишено феромону, і навпаки, чим довше шлях, тим менше буде залишено феромону на відповідних ребрах. Такий підхід дозволяє забезпечити безпосередній пошук у напрямку знаходження кращого рішення.

Пам'ять про вузли, які були відвідані агентом, забезпечується шляхом введення, так званого списку табу $tList$ – у ньому зберігається бітовий масив, за допомогою якого визначаються відвідані й не відвідані вузли. Таким чином, агент повинен проходити через кожний вузол тільки один раз. Вузли в списку «поточної подорожі» $Path$ розташовуються в тому порядку, у якому агент відвідував їх. Пізніше список використовується для визначення довжини шляху між вузлами.

Метод мурашиних колоній включає такі основні кроки.

Крок 1. Задати параметри методу: α – коефіцієнт, що визначає відносну значущість шляху; β – параметр, що показує значимість відстані; ρ – коефіцієнт кількості феромону, що агент залишає на шляху, де $(1-\rho)$ по-

казує коефіцієнт випаровування феромону на шляху після його завершення; Q – константа, що відноситься до кількості феромону, що було залишено на шляху; $startPheromone$ – початкове значення феромону, що знаходиться на шляхах до початку моделювання.

Крок 2. Ініціалізація методу. Створення популяції агентів. Після створення популяції агенти рівномірно розподіляється у вузлах мережі. Необхідно розподіляти агенти рівномірно між вузлами, щоб всі вузли мали однакові шанси стати відправною точкою. Якщо всі агенти почнуть рух з однієї точки, це б означало, що дана точка вважається оптимальною для старту, а насправді вона такою може й не бути. Але при цьому, якщо кількість агентів не кратна кількості вузлів, то кількість агентів у вузлах буде різною, але ця різниця не повинна перевершувати 1.

Крок 3. Рух агентів. Якщо агент ще не закінчив шлях, тобто не відвідав всі вузли мережі, для визначення наступного ребра шляху розраховується ймовірність переходу в u -ий вузол, коли агент перебуває в r -му вузлі, за формулою (11.1):

$$P_{ru} = \frac{\tau_{ru}(t)^\alpha \cdot \eta_{ru}(t)^\beta}{\sum_{k \in J} \tau_{rk}(t)^\alpha \cdot \eta_{rk}(t)^\beta} > rand(1), \quad (11.1)$$

де P_{ru} – імовірність того, що агент переміститься в u -ий вузол з r -го вузла; $rand(1)$ – випадкове число в інтервалі $(0; 1)$; J – множина вузлів, ще не відвіданих агентом; $\tau_{ru}(t)$ – інтенсивність феромону на ребрі між вузлами r та u у момент часу t ; $\eta_{ru}(t)$ – функція, що репрезентує вимір зворотної відстані для ребра.

Агент переміщується тільки по тим вузлам, які ще не були відвідані (відзначені списком табу $tList$). Тому ймовірність розраховується тільки для ребер, які ведуть до ще не відвіданих вузлів.

Крок 3 повторюється доти, поки кожний агент не завершить шлях. Цикли заборонено, оскільки в метод включений список табу $tList$.

Крок 4. Після завершення переміщень агентів може бути підрахована довжина шляху. Вона дорівнює сумі всіх ребер, по яких подорожував агент. Кількість феромону, що була залишена на кожному ребрі шляху i -го агента, визначається за формулою (11.2):

$$\Delta\tau^i(t) = \frac{Q}{L^i(t)}, \quad (11.2)$$

де $\Delta\tau^i(t)$ – кількість феромону, що залишив i -ий агент; $L^i(t)$ – довжина шляху i -го агента.

Результат є засобом виміру шляху: короткий шлях характеризується високою концентрацією феромону, а більш довгий шлях – більш низькою. Потім отриманий результат використовується для збільшення кількості феромо-

ну уздовж кожного ребра пройденого i -им агентом шляху за формулою (11.3):

$$\tau_{ru}(t) = \tau_{ru}(t-1) + \rho \cdot \sum_{i=1}^{N_{ru}} \Delta \tau^i(t), \quad (11.3)$$

де r, u – вузли, що утворюють ребра, які відвідав i -ий агент; N_{ru} – загальна кількість агентів, що відвідали ребро ru .

Дана формула застосовується до всього шляху, при цьому кожне ребро позначається феромоном пропорційно довжині шляху. Тому варто дочекатися, поки агент закінчить переміщення і тільки потім оновити рівні феромону, у протилежному випадку справжня довжина шляху залишиться невідомою. Константа ρ приймає значення між 0 та 1.

На початку шляху в кожного ребра є шанс бути обраним. Щоб поступово видалити ребра, які входять у гірші шляхи в мережі, до всіх ребер застосовується процедура випару феромону. Використовуючи константу ρ з попереднього виразу, можна скласти формулу (11.4):

$$\tau_{ru}(t) = \tau_{ru}(t) \cdot (1 - \rho). \quad (11.4)$$

Крок 5. Перевірка на досягнення оптимального результату. Перевірка може виконуватися відповідно до обмеження на максимальну кількість ітерацій або перевірка може вважатися успішною, коли протягом декількох ітерацій не було відзначено змін у виборі найкращого шляху. Якщо перевірка дала позитивний результат, то відбувається закінчення роботи методу (перехід до кроку 7), у протилежному випадку – перехід до кроку 6.

Крок 6. Повторний запуск. Після того як шляхи агентів завершено, ребра оновлено відповідно до довжин шляхів й відбувся випар феромону на всіх ребрах, метод виконується повторно. Список табу очищається, довжини шляхів зануляються. Перехід до кроку 3.

Крок 7. Зупинення. Визначається кращий шлях, що i є рішенням.

11.3 Різновиди методу мурашиних колоній

У зв'язку з можливістю різного математичного опису поведінки мурах в [4, 10, 11, 59–66] були розроблені розширення методу мурашиних систем. До них відносяться: метод мурашиних систем, заснований на елітній стратегії [3, 59]; метод мурашиних систем, заснований на ранжируванні (AS_{rank}) [4]; метод системи мурашиних колоній [10, 11, 63]; максімічний метод мурашиних систем (MAX-MIN AS – MMAS) [64–66].

Першим розширенням методу мурашиних систем була елітна стратегія, запропонована в [3, 59]. Даний підхід ґрунтується на додатковому збільшенні кількості феромонів для кращого глобального шляху в момент часу t . Таким чином, процедура додавання феромону для ребер, які входять в кращий на

даний момент часу шлях, виконується повторно, при цьому кількість феромону, що додається, розраховується відповідно до довжини кращого шляху.

Далі був запропонований метод мурашиних систем, заснований на ранжируванні (AS_{rank}) [4]. Даний метод по своїй суті є розширенням елітної стратегії й полягає в наступному: агенти сортуються за довжиною складених ними шляхів, після чого на глобально кращому шляху феромони збільшуються з вагою w , а збільшення феромонів виконується тільки для ребер, що ввійшли в шлях $(w-1)$ кращих агентів; при цьому k -ий кращий агент буде додавати феромон з вагою $(w-k)$ відповідно до (11.5):

$$\tau_{ru}(t+1) = \rho \cdot \tau_{ru}(t) + w \cdot \Delta\tau_{ru}^{gb}(t) + \sum_{k=1}^{w-1} (w-k) \cdot \Delta\tau_{ru}^k(t), \quad (11.5)$$

де $\Delta\tau_{ru}^{gb}(t) = 1/L^{gb}(t)$, $L^{gb}(t)$ – довжина кращого глобального шляху.

Метод системи мурашиних колоній (Ant Colony System – ACS) [15–17] поліпшує метод мурашиних систем шляхом використання інформації, отриманої попередніми агентами, для вивчення простору пошуку. Це досягається за допомогою двох механізмів. По-перше, використовується сувора елітна стратегія при відновленні феромонів на ребрах. По-друге, агенти вибирають наступний вузол для переміщення, використовуючи, так зване, псевдовипадкове пропорційне правило [63]: з ймовірністю q_0 агент переміщається в пункт u , для якого добуток кількості феромонів і евристичної інформації є максимальним: $u = \arg \max_{u \in J^r} \{\tau_{ru} \cdot \eta_{ru}(t)^\beta\}$, у той час як з ймовірністю $1-q_0$ буде застосований базовий підхід при визначенні наступного пункту для переходу, описаний у методі мурашиних систем. Значення q_0 є константою. При цьому якщо q_0 наближається до 1, то використовується тільки псевдовипадкове пропорційне правило, коли ж $q_0 = 0$, тоді метод системи мурашиних колоній працює за принципом методу мурашиних систем.

При оновленні шляхів, як було сказано вище, застосовується сувора елітна стратегія, відповідно до якої тільки агент, що склав краще рішення, виділяє феромон на шляху свого переміщення. Тоді кількість феромонів на ребрах змінюється відповідно до формули (11.6):

$$\tau_{ru}(t+1) = \rho \cdot \tau_{ru}(t) + (1-\rho) \cdot \Delta\tau_{ru}^{best}(t). \quad (11.6)$$

За кращого агента може використовуватися агент, що одержав краще рішення на даній ітерації, або глобально кращий агент, що одержав краще рішення на всіх ітераціях від початку роботи методу.

Останньою відмінністю методу системи мурашиних колоній є те, що агенти оновлюють кількість феромонів у процесі складання рішення (подібно цільнісному й кількісному методам мурашиних систем). Такий підхід призводить до зменшення ймовірності вибору однакових шляхів всіма агентами. За рахунок цього знижується ймовірність зациклення в локальному оптимумі.

Максі-мінний метод мурашиних систем (MAX-MIN AS – MMAS) [64–66] вводить нижню й верхню межі для можливих значень феромонів на ребрі, а також цей метод відрізняється підходом до визначення їхнього значення при ініціалізації. Практично в MMAS використовується інтервал значень феромонів, обмежений $\forall \tau_{ri}: \tau_{min}$ та τ_{max} , тобто $\tau_{min} \leq \tau_{ri} \leq \tau_{max}$. Кількість феромонів ребер при ініціалізації задається рівним нижній границі інтервалу, що забезпечує краще дослідження простору рішень. В MMAS, також як і в ACS, тільки кращий агент (глобально кращий або локально) виконує додавання феромонів після кожної ітерації методу. Результати обчислень [17] показали, що кращі результати виходять, коли оновлення феромонів виконується з використанням глобально кращого рішення. В MMAS також часто застосовується локальний пошук для поліпшення його властивостей.

Надалі описані моделі методу мурашиних колоній застосовувалися для вирішення інших оптимізаційних задач. Квадратична задача про призначення (Quadratic Assignment Problem, QAP) вирішувалася за допомогою методу мурашиних систем [17, 67, 68], а також MMAS [69]. Застосування цих методів полягало у використанні відповідної евристичної інформації даної задачі. Також вирішувалися: задача календарного планування (Job-shop Scheduling Problem, JSP) [3, 12], транспортна задача (Vehicle Routing Problem, VRP) [18, 70–72], задача вибору найкоротшої загальної надпоследовності (Shortest Common Supersequence Problem, SCSP) [19, 73], задача розфарбування графа [20], задача послідовного упорядкування [74] та інші задачі [21, 75, 76]. При вирішенні тестових задач методи мурашиних колоній показали гарні результати в порівнянні з традиційними методами оптимізації, призначеними для вирішення даних задач [12, 18–21, 70–76].

У загальному виді відмінності між різновидами методу мурашиних колоній можна відобразити в табл. 11.1.

11.4 Використання методу мурашиних колоній до вирішення задачі відбору інформативних ознак

У загальному випадку метод мурашиних колоній може бути використаний до будь-якої комбінаторної задачі, що може бути погоджена з такими вимогами:

- відповідне подання задачі – задача повинна бути описана у вигляді графа з набором вузлів і ребер між вузлами;

- евристична придатність ребер – можливість застосування евристичної міри адекватності шляхів від одного вузла до кожного сусіднього вузла в графі;

- складання альтернативних рішень, за допомогою чого можна раціонально визначати припустимі рішення;

Таблиця 11.1 – Відмінності між різновидами методу мурашиних колоній

Критерій	Метод			
	AS	ASrank	ACS	MMAS
Додавання Феромону	Після одержання рішення	Після одержання рішення	У процесі одержання рішення	У процесі одержання рішення
Правило вибору наступного пункту	Традиційний підхід	Псевдовипадкове пропорційне правило	Традиційний підхід	Традиційний підхід
Застосування елітної стратегії	Всі агенти беруть участь у відновленні шляхів	Відновлення виконують ($w-1$) локально кращих агентів і глобально кращий агент	Відновлення виконує тільки кращий (глобально або локально) агент	
Використання обмежень для різних параметрів	Відсутні	Обмеження на кількість агентів, що беруть участь у відновленні шляхів	Відсутні	Використовується інтервал значень феромону
Застосування локальної оптимізації	Відсутнє	Відсутнє	Використовуються традиційні методи локальної оптимізації	Відсутнє
Розв'язувані задачі	TSP, QAP, JSP, VRP, SCSP	TSP	TSP, JSP	TSP, QAP
Вплив кількості агентів на знаходження результату	Сильний	Середній	Слабкий	Слабкий

– правило вибору наступного вузла – правило, що визначає ймовірність переміщення агента з одного вузла графа до іншого.

Задачею відбору ознак є знаходження з повного набору ознак скороченого набору найбільш інформативних ознак з кількістю елементів *outCF*, що задається. При цьому висновок про інформативність робиться на підставі побудованої певним чином моделі на підставі аналізованого

набору ознак. Таким чином, задачею модифікації методу мурашиних колоній є знаходження такого набору ознак H_0 , при якому буде досягтися задана помилка моделі, що будується на підставі отриманого набору.

Задача відбору інформативних ознак може бути подана у вигляді, що відповідає методу мурашиних колоній. Метод мурашиних колоній вимагає, щоб задача була подана у вигляді графа, вузли якого характеризують ознаки, а ребра між ними – вибір наступної ознаки. У цьому й полягають відповідні модифікації методу мурашиних колоній для вирішення задачі відбору інформативних ознак.

Для вирішення задачі відбору інформативних ознак пропонується використати такі модифікації методу мурашиних колоній: модифікація на основі подання пунктів призначення ознаками [77, 78, 80–87] і у вигляді інформативності ознак [80–84, 86, 87]; з використанням операцій над чіткими множинами [79, 80, 83, 85]; з використанням операцій над нечіткими множинами [79, 80, 83, 85].

11.4.1 Модифікація на основі подання пунктів призначення ознаками

Основна ідея даної модифікації методу мурашиних колоній в аспекті вирішення задачі відбору ознак полягає в наступному: припускається, що агент повинен виконати переміщення по заданій кількості пунктів призначення n , при цьому кожному пункту ставиться у відповідність ознака x_i , $i = \overline{1, n}$; тоді шлях, пройдений агентом, становить набір ознак H , що пропонується використати для побудови моделі. При цьому дані зберігаються не в бітовому виді, а у вигляді номерів ознак $j = \overline{1, n}$, які варто залишити. При одержанні підсумкового результату це дозволяє одержувати оцінку ступеня впливу кожної ознаки.

Крок 1. Задати параметри методу: $initPh$ – початкову кількість феромону; p – коефіцієнт кількості феромону, що агент залишає на шляху, де $(1-p)$ показує коефіцієнт випаровування феромону на шляху після його завершення; Q – константа, що відноситься до кількості феромону, що було залишено на шляху; $inCF$ – кількість ознак у вхідному наборі; $outCF$ – кількість ознак, яку варто залишити у скороченому наборі.

Крок 2. Ініціалізація. Створення популяції агентів. Після створення популяції агенти рівномірно розподіляються по пунктах призначення.

Крок 3. Рух агентів. Якщо агент ще не закінчив шлях, тобто не відвідав пункти в кількості рівній $outCf$, для визначення наступного пункту в шляху агента використовується формула (11.7):

$$P_k = \frac{\tau_k(t)}{\sum_{i=1}^{n^j} \tau_i^j(t) + \tau_k(t)} > rand(1), \quad (11.7)$$

де P_k – ймовірність того, що j -ий агент, що ще не відвідав n^j пунктів, для продовження шляху обере пункт k ; $\tau_i(t)$ – кількість феромону в i -му пункті в момент часу t ; $rand(1)$ – випадкове число в інтервалі $(0;1)$.

Агент подорожує тільки в пункти, які ще не були відвідані (які зазначені в списку табу $tList$). Тому ймовірність розраховується тільки для пунктів, які ще не були відвідані агентом.

Крок 3 триває доти, поки кожний агент не закінчить свій шлях. Шлях вважається пройденим до кінця, коли агент відвідає пункти в кількості, рівній $outCF$. Цикли виключені, оскільки в метод включений список табу $tList$.

Крок 4. На підставі пунктів, відвіданих агентом, будується модель і для неї визначається помилка

Крок 4.1. Послідовність номерів вузлів, відвіданих агентом, переводиться в бітовий рядок H^j за формулою (11.8):

$$a_i = \begin{cases} 0, & \text{якщо } i \notin L^j; \\ 1, & \text{якщо } i \in L^j, \end{cases} \quad (11.8)$$

де a_i – i -ий біт у бітовому рядку; i – номер вузла; L^j – шлях j -го агента.

Крок 4.2. На підставі отриманого бітового рядка H^j і експериментальних даних будується модель (наприклад, на основі регресії або нейромережі).

Крок 4.3. Розраховується помилка ε_j для отриманої моделі за формулою (11.9):

$$\varepsilon_j = \frac{1}{2} \sum_{i=1}^N (y_i - y_{i \text{ розр.}})^2, \quad (11.9)$$

де y_i – вхідні значення вихідної змінної; $y_{i \text{ розр.}}$ – розрахункове значення вихідної змінної по побудованій моделі; N – кількість експериментів.

Крок 5. Визначається кількість феромону, що було залишено в кожному пункті шляху для j -го агента за формулою (11.10):

$$\Delta\tau^j(t) = \frac{Q}{\varepsilon_j}, \quad (11.10)$$

де $\Delta\tau^j(t)$ – кількість феромонів, яку треба додати кожному пункту, що входить у шлях j -го агента в момент часу t ; Q – параметр, що знижує ступінь впливу помилки моделі ε_j , побудованої на підставі ознак, що входять у шлях j -го агента.

Результат формули (11.10) є засобом виміру якості отриманої моделі: чим менше помилка побудованої на підставі обраних ознак моделі, тим більша концентрація феромонів у пунктах, що ввійшли в шлях даного

агента. Потім $\Delta\tau^j(t)$ використовується у формулі (11.11), щоб збільшити кількість феромону кожного пункту, у якому побував агент:

$$\tau_i(t) = \tau_i(t-1) + \rho \cdot \sum_{j=1}^N \Delta\tau_i^j(t), \quad (11.11)$$

де N – кількість агентів, що обрали i -у ознаку.

Дана формула застосовується до всього шляху, при цьому кожний пункт позначається феромоном пропорційно помилці отриманої моделі. Тому варто дочекатися, поки агент закінчить переміщення і тільки потім оновити рівні феромону, у протилежному випадку реальна довжина шляхів залишиться невідомою. Константа ρ приймає значення між 0 та 1.

На початку шляху в кожного пункту є шанс бути обраним. Щоб поступово видалити пункти, які входять у гірші комбінації, до всіх шляхів застосовується процедура випаровування феромону. Використовуючи константу ρ із (11.11), можна скласти формулу (11.12):

$$\tau_i(t) = \tau_i(t) \cdot (1 - \rho). \quad (11.12)$$

Крок 6. Перевірка на досягнення оптимального результату. Перевірка може виконуватися відповідно до обмеження на максимальну кількість ітерацій або перевірка вважається успішною, коли протягом декількох ітерацій не було відзначено змін у виборі найкращого рішення. Якщо перевірка дала позитивний результат, тоді відбувається закінчення роботи методу (перехід до кроку 7), у протилежному випадку – перехід до кроку 6.

Крок 7. Повторний запуск. Після того як шлях агента завершений, ребра оновлені відповідно до довжини шляху й відбулося випаровування феромону на всіх ребрах, метод виконується повторно. Список табу очищується, і довжина шляху встановлюється у нуль. Агентам дозволяється переміщуватися по всій мережі, засновуючи вибір ребра на формулі (11.7). Перехід до етапу 3.

Крок 8. Зупинення. Визначається кращий шлях, що є рішенням. Кращий шлях вибирається на підставі помилок моделей, які розраховуються за принципом, описаним на кроці 4.

Крок 9. Після зупинення й вибору найкращого шляху будується фінальна модель, що дозволяє забезпечити необхідну точність при меншій кількості ознак, ніж у вхідному наборі.

11.4.2 Модифікація на основі подання пунктів призначення у вигляді інформативності ознак

Особливістю даної модифікації є те, що пунктам призначення $(\overline{1, n})$, де n – кількість вхідних ознак $inCF$, ставиться у відповідність двійковий набір $B = \{b_i | b_i = \langle 0; 1 \rangle, i = \overline{1, n}\}$, сформований випадково, і при цьому кількість одиничних елементів повинне бути рівною кількості ознак $outCF$, які необхідно залишити. Кожний агент повинен виконати переміщення по всіх пунктах призначення, після чого по отриманому шляху будується модель відповідно до двійкового набору.

Основні кроки даної модифікації наведено нижче.

Крок 1. Задати параметри методу: α – коефіцієнт, що визначає відносну значущість шляху (кількість феромону на шляху); $initPh$ – початкова кількість феромону; ρ – коефіцієнт кількості феромону, що агент залишає на шляху, де $(1-\rho)$ показує коефіцієнт випаровування феромону на шляху після його завершення; Q – константа, що відноситься до кількості феромону, що було залишено на шляху; $inCF$ – кількість ознак у вхідному наборі; $outCF$ – кількість ознак, яку варто залишити в скороченому наборі, $maxTime$ – максимальний час моделювання. Встановити часу моделювання: $curTime = 0$.

Крок 2. Ініціалізація. Створення двійкового набору B і популяції агентів. Двійковий набір B варто створювати випадковим чином. Пропонується одержати $outCF$ цілих випадкових чисел $rNum_j$ ($j = \overline{1, outCF}$) в інтервалі $[1; inCF]$ і кожному b_{rNum_j} прирівняти 1, інші елементи b_k прийняти

рівними 0. Кількість агентів у популяції пропонується вибирати не меншою за кількість ознак. Після створення популяції агенти рівномірно розподіляються по пунктах призначення.

Крок 3. Зміна кількості часу моделювання: $curTime = curTime + 1$.

Крок 4. Рух агентів. Кожний агент, поки не закінчить весь шлях, обирає наступний пункт призначення. При виборі наступного пункту враховується кількість феромону, вже залишеного в цьому пункті, і кількість феромону, залишеного в інших пунктах, ще не відвіданих агентом. Додавання пункту k у шлях j -го агента відбувається при виконанні умови (11.13):

$$\frac{\tau_k(t)}{\sum_{i=1}^{n^j} \tau_i^j(t) + \tau_k^\alpha(t)} > rand(1), \quad (11.13)$$

де $rand(1)$ – випадкове число в інтервалі $(0; 1)$; $\tau_i^j(t)$ – кількість феромону в i -ому пункті j -го агента, що ще не відвідав n^j пунктів.

Пересування відбувається доти, поки всі агенти не відвідають всі пункти.

Крок 5. Якщо кожний агент відвідав всі пункти, то відбувається відновлення шляхів (крок 6) і перезапуск агентів (крок 7). У протилежному випадку – перехід до кроку 3.

Крок 6. Відновлення шляхів. Даний етап необхідний для того, щоб понизити ймовірність подальшого вибору таких пунктів призначення, включення яких дає підвищення помилки побудованої моделі щодо інших моделей, побудованих без цього пункту.

Крок 6.1. Для кожного агента будується модель на підставі пунктів, відвіданих ним. При цьому відбувається декодування з послідовності номерів пунктів у двійковий код для визначення тих ознак, які варто залишити для побудови моделі. Декодування відбувається на підставі двійкового набору B . Кожному пункту x_i^j ($i=1, \overline{outCF}$) j -го агента ставиться у відповідність число a_i за принципом (11.14):

$$a_i = \begin{cases} 1, & \text{якщо } b_{x_i^j} = 1; \\ 0, & \text{якщо } b_{x_i^j} = 0. \end{cases} \quad (11.14)$$

Крок 6.2. На підставі отриманого бітового рядка $H^j = \{a_i, i=1, \overline{outCF}\}$ і експериментальних даних будується модель (наприклад, на основі регресії або нейромережі).

Крок 6.3. Розраховується помилка ε_j :

$$\varepsilon_j = \frac{1}{2} \sum_{i=1}^N (y_i - y_{i \text{ розр.}})^2, \quad (11.15)$$

де y_i – вхідні значення вихідної змінної; $y_{i \text{ розр.}}$ – розрахункове значення вихідної змінної по побудованій моделі; N – кількість експериментів.

Крок 6.4. Відповідно до розрахованої помилки кількість феромонів кожного пункту, що відвідав j -ий агент збільшується на $\Delta\tau(j)$:

$$\tau_i^j(t) = \Delta\tau^j(t) + (\tau_i^j(t) \cdot \rho), \quad (11.16)$$

де $\tau_i^j(t)$ – кількість феромону в i -ому пункті j -го агента;

$\Delta\tau^j(t)$ визначається виходячи з міркувань, що більше феромону повинно додаватися тим пунктам, використання яких дає меншу помилку, при цьому треба вплив значення самої помилки на додавану кількість феромону зробити не занадто істотним, оскільки може виникнути зациклення на одному шляху:

$$\Delta\tau^j(t) = \frac{Q}{\varepsilon_j}, \quad (11.17)$$

де Q – параметр, що знижує ступінь впливу помилки моделі ε_j , побудованої на підставі ознак, що входять у шлях j -го агента.

Крок 6.5. На початку шляху в кожного пункту є шанс бути обраним. Щоб поступово видалити пункти, які входять у гірші комбінації, до всіх шляхів застосовується процедура випару феромону:

$$\tau_i(t) = \tau_i(t) \cdot (1 - \rho). \quad (11.18)$$

Крок 7. Перезапуск агентів. На цьому кроці виконуються дії аналогічні кроку 2 (ініціалізація). Тільки в цьому випадку – уже наявна популяція розподіляється по початкових пунктах, обнуляється список табу $tList$. Також на цьому кроці вибирається кращий зі шляхів, які були отримані при даному запуску агентів і зберігається в $bestPath$.

Крок 8. Перевірка на зупинення. Здійснюється або на підставі часу моделювання $curTime$ (зупинення відбувається при досягненні межі $maxTime$), або на підставі помилки кращої з отриманих моделей ($\epsilon_{bestPath} < \epsilon$). Якщо перевірка на зупинення дала позитивний результат – перехід до кроку 9, у протилежному випадку – перехід до кроку 3.

Крок 9. Зупинення. Визначається кращий шлях, що є рішенням. Кращий шлях обирається на підставі помилок моделей, які розраховуються за принципом, описаним на кроці 4.

Крок 10. Після зупинення й вибору найкращого шляху будується фінальна модель, що дозволяє забезпечити необхідну точність при меншій кількості ознак, ніж у вхідному наборі.

11.4.3 Модифікація з використанням операцій над чіткими множинами

Головну ідею даної модифікація зображено на рис. 11.3.

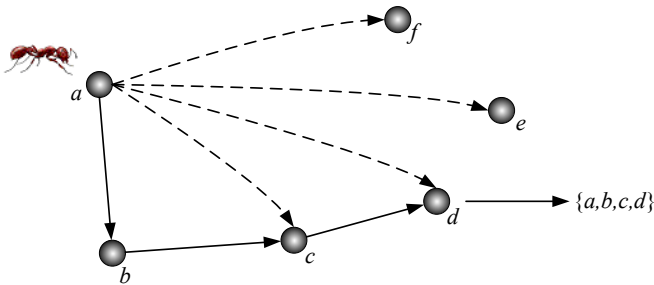


Рисунок 11.3 – Графічна інтерпретація методу мурашиних колоній для відбору ознак

Кожний агент переміщується по ознакам доти, поки не буде виконуватися певна умова закінчення переміщення. По закінченні переміщення агента отримується підмножина, що розглядається як кандидат для скорочення даних.

Основні кроки даної модифікації наведені нижче.

Крок 1. Робота методу починається з генерації k агентів, кожний з яких розміщується випадковим чином по графу (тобто кожний агент починає обхід з однієї випадкової ознаки). Кількість агентів, що розміщуються по графу, може дорівнювати кількості вхідних ознак; при цьому кожний агент починає формування шляху з різної ознаки.

Крок 2. З початкових позицій агенти обходять ребра на підставі ймовірного правила переходу, що позначає ймовірність того, що k -ий агент переміститься з ознаки i в ознаку j у момент часу t :

$$p_{ij}^k(t) = \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in J_i^k} \tau_{il}(t)^\alpha \cdot \eta_{il}^\beta}, \quad (11.19)$$

де J_i^k – множина ознак, не відвіданих k -им агентом; η_{ij} – евристична придатність вибору j -ої ознаки при знаходженні агента в i -ому вузлі; $\tau_{ij}(t)$ – кількість феромону на ребрі (i, j) .

Вибір параметрів α і β визначається експериментально. Кілька значень параметрів вибираються в інтервалі $[0, 1]$ і оцінюються експериментально.

Крок 3. Мінімальність підмножини й «якість» – два ключових фактори роботи даного методу, тому відновлення феромону повинне бути пропорційно «якості» і обернено пропорційно розміру підмножини.

Для застосування цього механізму при пошуку скороченого набору даних, необхідно використовувати функцію залежності γ' як критерій зупинення (11.20):

$$\gamma'_p(Q) = \frac{|POS_p(Q)|}{|U|}, \quad (11.20)$$

де U – універсальна кінцева множина рішень; $POS_p(Q)$ – позитивна область, що виражає відношення еквівалентності по U :

$$POS_p(Q) = \bigcup_{X \in U/Q} \underline{PX}, \quad (11.21)$$

де \underline{PX} – P -нижнє наближення множини P до множини X :
 $\underline{PX} = \{x \mid [x]_p \subseteq X\}$.

Це означає, що агент закінчить формування підмножини ознак, коли залежність підмножини досягне максимуму із вхідною множиною.

Крок 4. Збираються всі результуючі підмножини агентів і оцінюються певним методом, що може обиратися залежно від розв'язуваної задачі.

Шаг 5. Перевірка на зупинення. Якщо оптимальну підмножину знайдено або метод був виконаний певну кількість разів, то процес пошуку зупиняється й відбувається перехід до кроку 7.

Крок 6. Якщо перевірка на зупинення не була успішною, то феромони оновлюються. Феромон на кожному ребрі оновлюється відповідно до (11.22):

$$\tau' = (\rho \cdot \tau + \Delta\tau) \cdot (1 - \rho), \quad (11.22)$$

де τ' – нове значення феромону; τ – старе значення; $\Delta\tau$ – різниця, на яку повинен бути збільшений феромон; ρ – значення, що є постійною величиною, яка використовується для симуляції випаровування феромонів.

Якщо ребро (i, j) було відвідане, то $\Delta\tau$ розраховується за (11.23):

$$\Delta\tau_{ij}(t) = \sum_{k=1}^n (\gamma'(S^k) / |S^k|), \quad (11.23)$$

у протилежному випадку: $\Delta\tau_{ij}(t)=0$, де S^k – підмножина ознак, обраних k -им агентом; γ' – функція залежності.

Для однозначності всі агенти оновлюють феромони. Для цього можуть бути використані й альтернативні стратегії, наприклад, тільки агенти із кращими підмножинами, дозволяючи їм пропорційно збільшувати феромон.

Після оновлення феромонів відбувається перехід до кроку 1.

Крок 7. Зупинення. Обирається краща підмножина.

11.4.4 Модифікація з використання операцій над нечіткими множинами

Робота модифікації з використанням операцій над нечіткими множинами аналогічна роботі модифікації з використанням операцій над чіткими множинами, що описана вище, і складається з тих самих кроків.

Відмінностями є лише дані, з якими працює метод: у цій модифікації оброблюються нечіткі множини даних, а відповідно, формула для функції залежності γ' буде мати такий вигляд:

$$\gamma'_P(Q) = \frac{|\mu_{pos_P(Q)}(x)|}{|U|} = \frac{\sum_{x \in U} \mu_{pos_P(Q)}(x)}{|U|}, \quad (11.24)$$

де $\mu_A(x)$ – функція належності нечіткої підмножини A :

$$\begin{aligned} \mu_{pos_P(Q)}(x) &= \sup_{x \in U/Q} \mu_{PX}(x), \\ \mu_{PX}(x) &= \sup_{F \in U/P} \min(\mu_F(x), \inf_{y \in U} \max\{1 - \mu_F(y), \mu_x(y)\}). \end{aligned} \quad (11.25)$$

11.4.5 Порівняння модифікацій методу мурашиних колоній, що використовуються до вирішення задачі відбору інформативних ознак

У пп. 11.4.1–11.4.4 розглянуто чотири модифікації методу мурашиних колоній для відбору ознак:

- 1) з використанням операцій над чіткими множинами;
- 2) з використанням операцій над нечіткими множинами;

- 3) на основі подання пунктів призначення ознаками;
 - 4) на основі подання пунктів призначення як інформативності ознак.
- Основними відмінностями між модифікаціями є такі.

1. Відмінність видів даних, з якими працює метод. Модифікації 1, 3 та 4 працюють із чіткими множинами даних, а модифікація 2 – з нечіткими множинами даних. Внаслідок цього змінюються деякі етапи роботи модифікацій.

2. Відмінність аналогій вузлів. При приведенні задачі відбору ознак до виду задачі методу мурашиних колоній вузлам графа ставляться у відповідність різні аналогії: у модифікаціях 1, 2 та 3 їм відповідають безпосередньо ознаки, а в 4-ій модифікації вузли містять логічні числа (тобто 0 або 1), що вказують на інформативність, при цьому інформативність кожного вузла визначається не номером вузла, а порядком проходження агентом вузлів.

3. Різна приналежність феромону. Кількість феромону в модифікаціях 1 та 2 відноситься до ребра й характеризує його, тобто відображає можливість переходу від одного вузла до іншого, а в модифікаціях 3 та 4 характеризує вузол, тобто можливість включення даного вузла в результуючий набір.

4. Спосіб вибору наступного вузла при переміщенні агента. Вибір наступного вузла здійснюється за правилом переходу. Для модифікацій 1 та 2 правило переходу позначає ймовірність того, що k -ий агент переміститься з ознаки i в ознаку j у момент часу t . Правило переходу для модифікацій 3 та 4 позначає ймовірність того, що пункт i буде доданий у шлях агента.

5. Процедура зміни кількості феромонів. Сама процедура зміни кількості феромонів однакова для всіх модифікацій і є такою ж, як у базовому методі мурашиних колоній, і полягає в зміні феромонів для кожного ребра (у модифікаціях 1 та 2) або для кожного вузла (у модифікаціях 3 та 4). Розглянуті модифікації відрізняються розрахунком різниці, на яку повинен бути збільшений феромон.

6. Критерій зупинення роботи методу. Для модифікацій 1 та 2 критерієм зупину є γ' – метод зупиниться, коли залежність підмножини досягне максимуму із вхідною множиною ($\gamma'=1$ для несуперечливих множин). Для модифікацій 3 та 4 зупин здійснюється або на підставі часу моделювання (зупин відбувається при досягненні межі), або на підставі помилки кращої з отриманих моделей ($\epsilon_{\text{bestPath}} < \epsilon$).

11.5 Кластер-аналіз на основі методу мурашиних колоній

Для вирішення задачі кластеризації за допомогою методу мурашиних колоній пропонується його модифікація, у якій виконується не моделювання феромонів, а моделювання трьох основних дій, виконуваних мурахами при виконанні сортування їжі: підйом їжі, перенос їжі й залишення їжі.

Модифікація методу мурашиних колоній для виконання кластеризації може бути подана у вигляді таких кроків.

Крок 1. Ініціалізація. Задануться початкові параметри методу: кількість агентів, величина кроку переміщення *step*, максимальна кількість ітерацій *maxIter*, кількість кластерів, на які треба розбити задану вибірку, розмірність сітки.

Крок 2. Встановлюється лічильник ітерацій: *curIter* = 1.

Крок 3. Всі елементи даних розміщуються випадковим чином по мережі. Всі агенти випадково моделюють підіймання по одному елементу даних, і також випадковим чином розміщуються в мережі.

Крок 4. Обирається наступний агент для переміщення. Агенти обираються один по одному. Обраний агент переміщується по мережі з кроком *step*. Далі агент приймає рішення, де залишити елемент даних. Ймовірність того, що агент залишить елемент даних у поточній позиції, куди він перемістився, розраховується за формулою (11.26):

$$P_i(i) = \begin{cases} 1, & \text{якщо } f(i) \geq 1; \\ f(i)^{n_i}, & \text{у протилежному випадку,} \end{cases} \quad (11.26)$$

де *i* – обраний елемент даних; *n_i* – коефіцієнт, що визначає ступінь впливу сусідніх вузлів при прийнятті рішення про залишення елемента даних. (значення коефіцієнту *n_i* обирається експериментально); *f(i)* – модифікована функція Люмера–Файета [16]:

$$f(i) = \begin{cases} B \sum_{j=1}^{\sigma} (1 - \frac{d(i,j)}{\alpha}), & \text{якщо } f(i) > 0 \text{ та } \forall j (1 - \frac{d(i,j)}{\alpha}) > 0; \\ 0, & \text{в протилежному випадку,} \end{cases} \quad (11.27)$$

де *B* – коефіцієнт, що визначає вплив порожніх вузлів мережі, $B = \sigma^{-2}$; σ – кількість сусідніх вузлів, у яких знаходяться дані; *d(i,j)* – відмінність між даними *i* та *j*; α – константа, що визначається експериментально.

Подана формула *f(i)* поєднує дві важливі властивості.

По-перше, використання коефіцієнта *B* забезпечує відсікання порожніх вузлів мережі, таким чином, забезпечуючи посилене об'єднання в кластери (а не тільки вільне сортування).

По-друге, додаткове обмеження

$$\forall j (1 - \frac{d(i,j)}{\alpha}) > 0$$

забезпечує сильне відсікання більших відмінностей, що значно поліпшує просторовий поділ між кластерами.

Якщо приймається рішення, що елемент даних не слід залишати в поточній позиції, то агент залишає його в сусідньому вузлі (обирається вільний вузол на основі випадкового пошуку).

Крок 5. Далі агент шукає наступний елемент даних, щоб взяти його для переміщення. Випадково обирається вільний елемент даних, для нього розраховується його функція $f(i)$ і відповідно до неї приймається рішення, брати для переміщення цей елемент даних чи ні, відповідно до формули (11.28):

$$P_p(i) = \begin{cases} 1, & \text{якщо } f(i) \leq 1; \\ \frac{1}{f(i)^{n_p}}, & \text{в протилежному випадку,} \end{cases} \quad (11.28)$$

де n_p – показує ступінь впливу сусідніх вузлів мережі, обирається експериментально, пропонується $n_p = 2$.

Крок 6. Після вибору елемента даних поточним агентом для переміщення виконуються кроки 4 і 5 для всіх інших агентів. Якщо всі агенти виконали кроки 4 і 5 – перехід до кроку 7.

Крок 7. Розраховується кількість кластерів, на яку було розбито вибірку. Якщо розрахована кількість кластерів збігається з кількістю, яку треба одержати, тоді відбувається перехід до кроку 9, у протилежному випадку – до кроку 8.

Крок 8. Збільшується лічильник ітерацій: $curIter = curIter + 1$. Якщо кількість $curIter$ перевищує максимальну кількість ітерацій: $curIter > maxIter$, тоді виконується перехід до кроку 9, у протилежному випадку – перехід до кроку 4.

Крок 9. Зупинення.

11.6 Переваги й недоліки методу мурашиних колоній

Грунтуючись на розглянутих основних принципах методу мурашиних колоній, його різновидах та областях застосування можна виділити переваги й недоліки методу мурашиних колоній.

До переваг методу мурашиних колоній можна віднести:

- можливість використання методу в динамічних додатках (адаптуються до змін навколишнього середовища);
 - використання пам'яті всієї колонії, що досягається за рахунок моделювання виділення феромонів;
 - гарантування збіжності до оптимального рішення;
 - стохастичність, тобто випадковість пошуку, за рахунок чого виключається можливість зациклення в локальному оптимумі;
 - мультиагентність;
 - вища швидкість знаходження оптимального рішення, ніж у традиційних методах;
 - можливість застосування до множини різних задач оптимізації.
- Можна виділити такі недоліки методу мурашиних колоній:

– складність теоретичного аналізу, оскільки підсумкове рішення формується в результаті послідовності випадкових рішень; розподіл ймовірностей змінюється по ітераціях; дослідження є більш експериментальними, ніж теоретичними;

– невизначеність часу збіжності при тому, що збіжність гарантується;

– висока ітеративність методу;

– сильна залежність результатів роботи методу від початкових параметрів пошуку, які підбираються експериментально.

Виходячи з наведених галузей застосування, переваг та недоліків методу мурашиних колоній можна зробити висновок, що метод рекомендовано використовувати при вирішенні дискретних оптимізаційних задач та в динамічних застосуваннях, оскільки він здатен пристосовуватися до змін навколишнього середовища. Варто також відзначити, що ефективність методу є більшою у випадках, коли задача характеризується великою розмірністю, оскільки традиційні методи зазвичай дають гарні результати, якщо розмірність є малою, а от у випадках з великою розмірністю вони можуть або зациклюватися в локальних оптимумах, або занадто довго працювати, що є небажаним при вирішенні практичних задач.

Не рекомендується застосовувати базовий метод мурашиних колоній до вирішення задач безперервної оптимізації, проте, оскільки метод характеризується гарною розширюваністю, то в нього можна легко вводити додаткові процедури, ідеї яких взяті з інших методів, заснованих на імовірнісному підході, внаслідок чого створюються гібридні системи на основі методу мурашиних колоній для вирішення задач й безперервної оптимізації.

РОЗДІЛ 12

МЕТОД БДЖОЛИНОЇ КОЛОНІЇ

12.1 Біологічні основи методу бджолиної колонії

Для опису поведінки бджіл у природі використовуються три основні поняття: джерело нектару (квітка), зайняті фуражири, незайняті фуражири.

Джерело нектару характеризується значущістю, що обумовлена різними факторами, такими як: дальність від вулика, концентрація нектару, зручність видобутку нектару.

Зайняті фуражири закріплені за окремим джерелом, на якому вони добувають нектар, тобто вони «зайняті» ним. Зайняті фуражири мають таку інформацію про дане джерело нектару, як: відстань і напрямок від вулика, корисність джерела.

Незайняті фуражири продовжують шукати джерела нектару для їхнього використання. Існує два типи незайнятих фуражирів: розвідники, які шукають нові джерела нектару, і спостерігачі, які чекають у вулику й можуть виконувати інші дії у вулику.

Середня кількість розвідників у рої становить близько 5–10%.

Незважаючи на колективний характер поведінки соціальних комах і на різні схеми цієї поведінки, окрема комаха також здатна виконувати різні складні дії [6], прикладом чого можуть служити збір і обробка нектару бджолами, виконання цих дій добре організовано.

Кожна незайнята бджола може полетіти до джерела нектару, слідуючи за бджолою-розвідником, що знайшла шлях до квітки. Це досягається за рахунок того, що кожний вулик має так звану закрити площадку для танцю, на якій бджоли, що виявили джерела нектару, виконують виляючий танець, тим самим, намагаючись залучити інших незайнятих бджіл летіти за ними. Якщо бджола вирішує залишити вулик, щоб одержати нектар, вона слідує за однією із бджіл-розвідників до області з нектаром. Таким чином, незайнята бджола стає зайнятою.

По досягненню області з нектаром зайнятий фуражир добуває нектар і повертається у вулик, залишаючи нектар там. Після того, як бджола залишає нектар, вона може виконати одну з таких трьох дій:

- залишити джерело нектару й знову стати незайнятим фуражиром;
- продовжити фуражування до того ж джерелу нектару, не вербуючи інших особин свого вулика;
- виконати танець і в такий спосіб виконати вербування.

Бджола обирає одну з вищезгаданих альтернатив з деякою ймовірністю. У межах області танцю, бджоли «рекламують» різні області нектару. Механізми, відповідно до яких бджола вирішує слідувати за іншою бджолою, до-

сліджені недостатньо добре, але припускається, що вербування серед бджіл з математичної точки зору завжди є функцією якості джерела нектару [6]. Також відзначено, що не всі бджоли починають фуражування одночасно.

Таким чином, виконується поділ функцій між зайнятими фуражирами й розвідниками на поліпшене вивчення знайдених областей з нектаром і на знаходження нових областей з нектаром, відповідно. За рахунок такого поділу праці досягається ефективна робота всього рою бджіл.

Самоорганізація бджолоїної колонії ґрунтується на чотирьох основних властивостях.

1. Позитивний зворотний зв'язок – полягає в тому, що бджоли, ґрунтуючись на отриманій інформації від іншої бджоли, починають літати до певного джерела нектару.

2. Негативний зворотний зв'язок – полягає в тому, що бджола, ґрунтуючись на інформації, отриманій від інших бджіл, може вирішити, що знайдене нею джерело значно гірше в порівнянні з іншими знайденими джерелами.

3. Нестійкість – бджоли-розвідники виконують випадковий пошук нових джерел ресурсів.

4. Множинність взаємодії – інформація про джерело ресурсів, знайдене однією бджолою, доступна для всіх інших у вулику за допомогою виконання, так званого виляючого танцю.

Основні особливості поведінки бджіл можна пояснити за допомогою рис. 12.1, де зображено випадок, коли вже знайдено два джерела нектару А та В, а також зображено бджолу – потенційного фуражиря. На самому початку ця бджола може прийняти одне із двох рішень:

– стати розвідником («С») і після цього направитися у пошуках нектару в будь-яку область;

– направитися у вулик і, надалі, брати участь у вербуванні («В»).

При цьому якщо бджола направляється у вулик на площадку для виконання танцю, то вона може стати або зайнятим фуражиром («ЗФ1»), або, у випадку, якщо бджола не була піддана вербуванню, теж стати розвідником.

Коли бджола стає зайнятим фуражиром, вона може бути спрямована як у джерело А, так і в джерело В. Це залежить від виконаного виляючого танцю. Після того, як бджола, будучи зайнятим фуражиром, набрала нектар у джерелі, вона повертається у вулик і залишає там зібраний нектар. При цьому, коли бджола-фуражир повертається, вона вже може залишатися як зайнятим фуражиром, якщо нектар ще залишився у джерелі, так і стати незайнятим фуражиром («НФ»). Після того, як бджола залишила нектар, вона може направитися на площадку для виконання танцю або відразу відправитися назад до джерела з нектаром («ЗФ2»). Після чого, даний цикл може повторюватися доти, поки не буде остаточно повністю використане досліджуване джерело нектару.

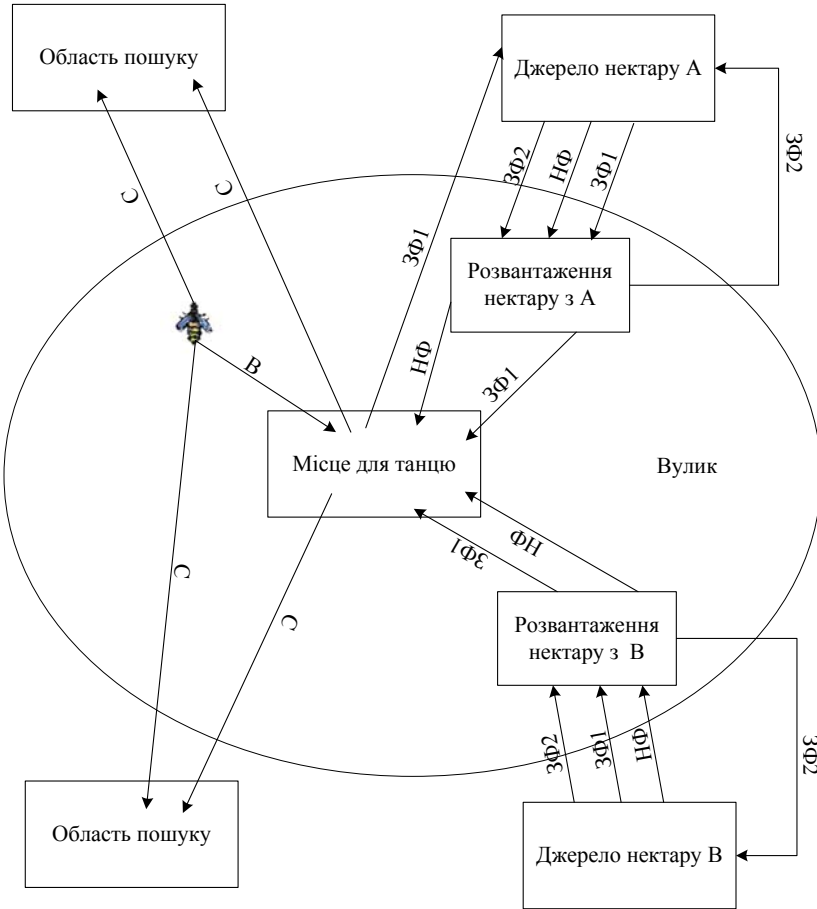


Рисунок 12.1 – Основні особливості поведінки бджіл

12.2 Формалізація поведінки бджіл у процесі фуражування

В [88] запропонована поведінкова модель самоорганізації колонії бджіл, у якій бджоли-фуражири, які переміщуються до квіток, повертаються у вулик, а також мають інформацію про корисність відповідного джерела. Зібраний нектар забезпечує зворотний зв'язок з поточним станом потоку нектару у вулику. Корисність – функція якості нектару, кількості нектару й відстані від вулика. Зворотний зв'язок встановлює відповідний поріг для сигналу вербування, також відомий як виляючий танець,

тривалість якого залежить як від порога, так і від корисності. Виляючий танець виконується на спеціальній площадці для танців, де окремі фуражири можуть спостерігати за їхнім виконанням. Фуражири можуть випадковим чином обирати танець для спостереження й відповідно до танцю вони можуть визначити місце розташування ділянки квітки та покинути вулик для фуражування. Модель самоорганізації забезпечує пропорційний зворотний зв'язок з якістю джерела нектару.

В [7] описано підхід, поданий в [88], у термінах числення зв'язних систем (Calculus of Communicating Systems, CCS).

CCS – математична модель (формальна мова), призначена для опису процесів, що звичайно використовується при вивченні паралелізму. CCS містить набір термів, операторів і аксіом, які використовуються для опису й керування складеними виразами. Вирази характеризують елементи паралельної системи, а керування цими виразами показує, як поводить система. Центральним елементом в CCS є унікально іменованій агент, що має специфічну поведінку. Поведінка агента визначається множиною подій і дій, які може виконувати агент. Множина подій, виконуваних агентом, описується за допомогою оператора «дія», що позначається як «.». Іншим важливим оператором в CCS є оператор «+», що є оператором вибору. Цей оператор використовується у випадку, якщо описується участь агента в одній з декількох альтернативних дій.

CCS в [89] має справу з агентами, які описують стан бджоли або стан групи бджіл, і з діями, які являють собою можливість переходу з одного стану в інший. Таким чином, колонія може бути подана у вигляді зв'язаного графа, вершинами якого є агенти, а ребра – діями. Наприклад, агент може бути поданий у такому вигляді:

Розвідник_b = гарний_b(s).Пошук_b(s) + поганий_b.Незайнятий_b.

Поданий агент b є розвідником, що шукає джерело нектару. У випадку, якщо він буде *погано* шукати нектар, то він стане *незайнятим* фуражиром, якщо ж він буде *добре* шукати, то він продовжить Пошук_b(s) деякого джерела s .

Тоді моделювання поведінки бджіл у термінах CCS можна зобразити на рис. 12.2.

Процес фуражування можна формалізувати за допомогою CCS в такому вигляді:

Вдало_b(s) = з_{b,s}.Вербування_b(s),

Невдало_b(s) = з_{b,s}.Незайнятий_b,

Вербування_b(s) = танець(θ, s).Вербування_b(s) + покинути_b.Використати_b(s),

Незайнятий_b = танець(b', s).Використати_b(s) + дослідити_b.Розвідник_b,

Розвідник_b = гарний_b(s).Пошук_b(s) + поганий_b.Незайнятий_b.

Використати_b(s) = до_{b,s}.Пошук_b(s),

Пошук_b(s) = нектар_{b,s}.Вдало_b(s) + нічого_s.Невдало_b(s),

У цьому описі: b – унікальний ідентифікатор одного агента, b' – ідентифікатор іншого агента, s – джерело нектару.

Таким чином, за допомогою запропонованої формалізації моделюється поведінка бджіл. Незайнятий агент перебуває у вулику доти, поки не одержить сигнал від зайнятого фуражира за допомогою виляючого танцю; якщо такий сигнал отримано, то агент переміщується до відповідного джерела нектару; у протилежному випадку – він може сам стати розвідником через якийсь час; якщо в передбачуваному джерелі нектару є нектар, то агент приносить його у вулик і моделює виконання виляючого танцю.

12.3 Методи на основі моделювання бджолої колонії для вирішення задач дискретної оптимізації

На основі запропонованого в [88] підходу в [22] було розроблено метод бджолої колонії для вирішення задачі календарного планування (Bee Colony Optimization for Job-Shop Scheduling Problem, BCO-JSSP).

Задача календарного планування може характеризуватися множиною робіт, кожна з яких складається з однієї або більше операцій. Операції виконуються на специфічній послідовності спеціальних машин. Метою планування є складання розкладу робіт, що мінімізує (максимізує) міру виконання.

Задача календарного планування відноситься до NP-важких. Міра виконання включає: завантаження устаткування (коефіцієнт використання устаткування), час виробничого циклу, продуктивність (витрату, пропускну здатність) і рівень запасів.

У загальному випадку задача календарного планування подається за допомогою диз'юнктивного графа. Граф складається з вузлів, що являють собою операції. Також є два додаткових вузли, які становлять ресурси й витрати. Множина орієнтованих дуг використовується для опису переваги кожної роботи.

Оскільки головними особливостями методу бджолої колонії є виляючий танець і процес фуражування, то запропонована модифікація [22] для рішення задачі календарного планування відрізняється саме цими етапами роботи методу бджолої колонії в порівнянні із запропонованим раніше методом [88]. Аналогією джерела нектару в даній модифікації є шлях, що може розглядатися як рішення задачі календарного планування.

Після повернення у вулик, агент виконує виляючий танець з ймовірністю p . Тривалість D_i виляючого танцю i -го агента розраховується за формулою:

$$D_i = d_i \cdot A,$$

де A – масштабуючий коефіцієнт, d_i – відносна корисність знайденого джерела нектару i -го агента.

Абсолютна корисність джерела нектару i -го агента Pf_i для задачі календарного планування розраховується за формулою:

$$Pf_i = \frac{1}{C_i},$$

де C_i – цільова функція для шляху i -го агента. У цьому випадку вона уявляє собою тривалість виконання всіх операцій всіх робіт для шляху.

Тоді, розрахувавши абсолютну корисність кожного агента, можна одержати середню корисність всієї колонії Pf_{colony} :

$$Pf_{colony} = \frac{1}{n} \sum_{j=1}^n Pf_j,$$

де n – кількість виляючих танців, що виконуються у момент часу t .

Таким чином, можна розрахувати відносну корисність d_i i -го фуражира:

$$d_i = \frac{Pf_i}{Pf_{colony}}.$$

Ймовірність p_i того, що за i -им агентом, після виконання ним танцю, підуть інші незайняті фуражири, визначається за формулою [26]:

$$p_i = \begin{cases} 0,60, & \text{якщо } Pf_i < 0,9 \cdot Pf_{colony}; \\ 0,20, & \text{якщо } 0,9 \cdot Pf_{colony} \leq Pf_i < 0,95 \cdot Pf_{colony}; \\ 0,02, & \text{якщо } 0,95 \cdot Pf_{colony} \leq Pf_i < 1,15 \cdot Pf_{colony}; \\ 0,00, & \text{якщо } 1,15 \leq Pf_i. \end{cases}$$

Оскільки в процесі кормодобування агенти формують рішення шляхом переміщення з вузла до вузла на графі, що описує можливі роботи, треба розрахувати ймовірність додавання в шлях агента заданого вузла.

Ймовірність P_{ij} того, що агент вибере наступний j -ий вузол, перебуваючи в i -му вузлі, розраховується за формулою:

$$P_{ij} = \frac{\rho_{ij}^\alpha \cdot d_{ij}^{-\beta}}{\sum_{j \in J^k} \rho_{ij}^\alpha \cdot d_{ij}^{-\beta}},$$

де ρ_{ij} – вартість дуги між j -им та i -им вузлами; d_{ij} – евристична відстань між j -им та i -им вузлами; $\alpha, \beta \in [0; 1]$ – коефіцієнти, обрані експериментально; J^k – множина вузлів, у які можна переміститися з i -го вузла.

Оцінка ρ_{ij} визначається за допомогою формули:

$$\rho_{ij} = \frac{1 - m\alpha}{k - m},$$

де k – кількість вузлів, у які можна переміститися з i -го вузла; m – число переваги шляху, що може бути дорівнює 1 або 0. Кращим вважається шлях, що на якій-небудь ітерації вважався придатним для виконання тан-

цю. При цьому кількість таких, так званих, елітних шляхів обмежена. Таким чином, на початковій ітерації всі ребра мають число $m=0$, що робить рівними шанси вибору будь-якого ребра.

Даний метод порівнювався з методом мурашиних колоній і з пошуком з табу. Експерименти [22] показали, що результати, отримані за допомогою методу бджолоїної колонії, майже не відрізняються від результатів, отриманих за допомогою методу мурашиних колоній, і незначно гірші результатів, отриманих за допомогою пошуку з табу.

Lucic і Teodorovic [23, 90] першими використали основні принципи колективного інтелекту бджіл для вирішення задач комбінаторної оптимізації. Вони розробили так званий метод системи бджіл (Bee System, BS) і перевірили його при вирішенні задачі комівояжера. На основі BS в [24] було запропоновано метаевристичний метод бджолоїної колонії (Bee Colony Optimization Metaheuristic, BCO) і метод нечіткої бджолоїної системи (Fuzzy Bee System, FBS).

У методі BCO на початку процесу пошуку всі агенти розташовані у вулику. Протягом процесу пошуку агенти зв'язуються один з одним опосередковано. Кожний агент робить ряд локальних переміщень, і в такий спосіб поступово формує рішення задачі. Процес пошуку складається з ітерацій. Перша ітерація вважається закінченою, коли агенти створять хоча б одне прийнятне рішення. Краще рішення зберігається, а потім відбувається перехід до наступної ітерації. Далі процес складання рішень повторюється. Загальна кількість ітерацій обмежується виходячи із задачі оптимізації.

При переміщенні в просторі пошуку агенти можуть переміщуватися в прямому або у зворотному напрямку. При переміщенні в прямому напрямку агенти формують різні часткові рішення, що досягається за рахунок індивідуального дослідження простору пошуку й за рахунок колективного досвіду, отриманого на попередніх ітераціях.

Після створення часткового рішення агенти переміщуються у зворотному напрямку, тобто вони повертаються у вулик, де можуть брати участь у процесі вербування шляхом виконання танцю, тим самим обмінюючись інформацією про різні створені часткові рішення. Після відвідування вулика агенти знову переміщуються у прямому напрямку й продовжують створювати часткові рішення. Ітерація закінчується тоді, коли створюється хоча б одне припустиме рішення. Таким чином, BCO, як і методи динамічного програмування, вирішує комбінаторні задачі оптимізації поетапно.

Виходячи з вищесказаного, можна відзначити, що в BCO, на відміну від описаного раніше методу бджолоїної колонії, немає поділу в ролях агентів (зайняті фуражири, незайняті фуражири й розвідники), і рішення

складається поетапно, у міру знаходження часткових рішень, поки не буде отримане повне прийнятне рішення.

Метод FBS призначений для вирішення задач, що характеризуються невизначеністю, агенти при вирішенні задачі використовують правила нечіткої логіки [91, 92] для організації зв'язку між агентами і їхніми діями.

Відповідно до FBS при додаванні компонента рішення до власного рішення агент може розглядати компонент рішення як: «менш корисний», «корисний» або «більш корисний». Також агенти здатні розрізнити додаткові властивості: «короткий», «середній» або «довгий», «нецінний», «середній» або «цінний».

При виборі наступного компонента рішення для визначення корисності компонента використовується таке правило:

ЯКЩО властивості компонента рішення – «дуже гарні», ТО розглянутий компонент рішення – «дуже корисний».

Ймовірність P_j того, що j -ий компонент буде доданий до власного рішення, розраховується за формулою:

$$P_j = \frac{f_j}{\sum_{k \in J^k} f_k},$$

де f_j – корисність j -го компонента рішення; J^k – множина компонентів рішення, які можуть бути додані.

В FBS для порівняння власних рішень агентів використовується концепція непридатності власних рішень [24]. Непридатність власного рішення визначається в як:

$$L_k = \frac{L^{(k)} - L_{\min}}{L_{\max} - L_{\min}},$$

де L_k – непридатність власного рішення, отриманого k -им агентом; $L^{(k)}$ – цільова функція, отримана за допомогою власного рішення k -го агента; L_{\min} – цільова функція, отримана на підставі кращого власного рішення, знайденого з початку пошуку; L_{\max} – цільова функція, отримана на підставі гіршого власного рішення, знайденого з початку пошуку.

Визначення придатності власного рішення ґрунтується на приблизних міркуваннях:

ЯКЩО отримане власне рішення – «погане», ТО його придатність – «низька».

Таким чином, агенти використовують приблизні міркування для порівняння отриманих власних рішень із кращим і з гіршим власними рішеннями, знайденими від початку процесу пошуку. За рахунок такого

підходу забезпечується вплив отриманих рішень на напрямки пошуку на майбутніх ітераціях.

Оскільки бджоли відносяться до соціальних комах, ймовірність P^* того, що агент продовжить переміщення по тому ж шляху без виконання виляючого танцю дуже низка ($P^* \ll 1$). Відповідно, ймовірність того, що агент буде виконувати процес вербування, дорівнює $(1 - P^*)$.

Для визначення кількості агентів, які покинуть знайдені ними шляхи й приєднаються до інших агентів, також існують певні правила. Кожне власне рішення, що розглядається в області для танцю, має дві характеристики: цільову функцію й кількість агентів, що пропонують це рішення.

Для визначення корисності пропонованого власного рішення використовується правило:

ЯКЩО пропонований шлях – «короткий» та кількість агентів, що його пропонують – «маленька», ТО корисність пропонованого шляху – «середня».

Таким чином, корисність пропонованого шляху може знаходитися в інтервалі $[0, 1]$.

Для визначення кількості агентів, що змінюють свій шлях, використовується правило:

ЯКЩО придатність власного рішення i -го агента – «низька» та корисність власного рішення j -го агента – «висока», ТО кількість агентів, що змінюють свій шлях на шлях j -го агента – «велика».

Таким чином, використовуючи колективні знання й обмін інформацією, агенти зосереджують пошук на найбільш перспективних напрямках пошуку.

Метод FBS був застосований до вирішення задачі про складання пар [24]. Проведені експерименти [24] показали, що застосування представленого методу характеризується достатньою точністю й швидкою збіжністю.

12.4 Багатовимірна оптимізація на основі методу бджолиної колонії

Однією з задач, що часто виникають у процесі моделювання складних об'єктів і систем, є знаходження глобального оптимуму багатовимірної функції. Для вирішення цієї задачі застосовується ряд методів (наприклад, методи Коші, Ньютона, Левенберга-Марквардта й т. п.), які, однак, вимагають безперервності, диференційованості та унімодальності цільових функцій. Тому пропонується застосовувати метод бджолиної колонії для вирішення задачі оптимізації багатовимірної функції, оскільки він не накладає ніяких вимог до цільової функції.

Запропонований метод можна подати як послідовність таких кроків.

Крок 1. Задаються початкові параметри роботи методу: початкова кількість агентів-розвідників B_s , максимальна енергія, коефіцієнт α , початкова температура T_{init} , кінцева температура T_{final} , максимальна кількість ітера-

цій $iter_{max}$. Також задаються функція f , що оптимізується, та параметри, пов'язані з нею: кількість змінних $argCnt$; мінімальна $Range^{min}$ і максимальна $Range^{max}$ межі для кожної зі змінних, тобто область, у якій можна шукати можливі рішення; напрямок оптимуму $optOrient$ (визначає максимізацію або мінімізацію функції); у випадку використання локальної оптимізації задається один із традиційних методів локальної оптимізації, що буде використовуватися, і задається клас агентів, до рішень яких буде застосовуватися локальна оптимізація – агент із кращим рішенням або агенти, що виконали моделювання виляючого танцю.

Крок 2. Створюються початкові агенти-розвідники.

Крок 2.1. Для кожного початкового агента-розвідника створюється випадкове рішення:

$$bee_i.x_j = rand \cdot (Range_j^{max} - Range_j^{min}) + Range_j^{min}, \\ \forall i = \overline{1, B_s}, j = \overline{1, argCnt},$$

де bee_i – i -ий агент-розвідник; $rand$ – випадкове число в інтервалі $[0; 1]$.

Крок 2.2. Для отриманих випадкових рішень розраховується користь даного джерела нектару як значення оптимізованої функції:

$$bee_i.profitability = optOrient \cdot f(bee_i.x_1, \dots, bee_i.x_{argCnt}), \forall i = \overline{1, B_s},$$

де $OptOrient = 1$, якщо виконується максимізація, у протилежному випадку – встановлюють: $OptOrient = -1$.

Встановлюється поточна кількість ітерацій: $iter=1$; встановлюється кількість агентів-розвідників: $B=B_s$; встановлюється поточна температура: $T=T_{init}$.

Крок 3. Обираються робочі агенти, тобто такі агенти, на базі яких будуть створюватися нові агенти за допомогою процедури схрещування.

Крок 3.1. Визначається агент $best$ з найбільшою користістю.

Крок 3.2. Процедура імітації відпалу [29]. Агент відноситься до робочих агентів $workBee$, якщо виконується умова:

$$\exp\left(-\frac{|bee_i.profitability - best.profitability|}{T}\right) > rand, \forall i = \overline{1, B}.$$

Крок 4. Схрещування. Оскільки реальні бджоли-розвідники при виборі джерела нектару користуються також генетичним матеріалом (у біології ще не вивчено, яким саме чином розвідники обирають одні квітки й пропускають інші, тобто вважається, що розвідники ґрунтуються на генетичному досвіді), то за допомогою процедури схрещування моделюється саме цей момент поведінки бджіл. Для схрещування використовуються раніше відібрані за допомогою процедури імітації відпалу робочі агенти $workBee$ і кращий агент за всі ітерації $best$. Нові агенти створюються у два етапи: на базі рішень робочих агентів і на базі рішення кращого агента.

Шаг 4.1. Створення нових агентів на базі робочих агентів $workBee$:

$$newWorkBee_i.x_j = workBee_i.x_j \pm rand \cdot (workBee_i.x_j - best.x_j), \forall i = \overline{1, B_w},$$

де B_w – кількість робочих агентів $workBee$; знак перетворення «+» або «-» обирається випадковим чином.

Крок 4.2. Створення нових агентів на базі кращого агента $best$:

$$newWorkBee_i.x_j = best.x_j \pm rand \cdot (workBee_i.x_j - best.x_j).$$

Крок 4.3. Для всіх нових агентів виконується корегування отриманих рішень, оскільки отримані в такий спосіб значення змінних можуть виходити за межі $[Range^{min}, Range^{max}]$.

Крок 4.4. Розраховується корисність отриманих рішень:

$$newWorkBee_i.profitability = f(newWorkBee_i.x_1, \dots, newWorkBee_i.x_{argCnt}), \forall i = \overline{1, B_n},$$

де B_n – кількість створених при схрещуванні агентів $newWorkBee$.

Крок 4.5 Обирається новий кращий агент $best$.

Крок 5. Моделювання виконання виляючого танцю. До можливого виконання танцю допускаються робочі агенти $workBee$, агенти, створені шляхом схрещування, $newWorkBee$, кращий агент за всі ітерації $best$. Моделювання виконання виляючого танцю відбувається в кілька етапів. У результаті цього моделювання обираються ті агенти, які за рахунок виконання танцю виконують вербування інших агентів для дослідження знайденого ними рішення.

Шаг 5.1. Виконується нормування корисностей агентів, допущених до можливості виконання танцю. При цьому нормування враховує напрямки оптимізації $optOrient$.

Крок 5.2. Додавання шумів до отриманих нормованих корисностей і їхнє корегування:

$$np_i = \begin{cases} 1, & \text{якщо } np_i + w_i > 1; \\ np_i + w_i, & \text{якщо } e_n < np_i + w_i < 1; \\ 0, & \text{якщо } 0 < np_i + w_i < e_n, \end{cases}$$

де np_i – нормована корисність i -го агента; w_i – шум у корисності. Шум рівномірно розподілений у межах $(-w; +w)$. Значення w обирається експериментально (пропонується $w = 0,1$), e_n – мінімальний поріг корисності. Мінімальний поріг обирається експериментально (пропонується $e_n = 0,1$).

Крок 5.3. Визначення переваги танцю кожного агента:

$$L_i = \max\{np_i - \eta \cdot \overline{np}, 0\},$$

де L_i – перевага танцю i -го агента; η – коефіцієнт, керуючий впливом величини \overline{np} на L_i ; \overline{np} – середнє значення нормованої корисності всіх агентів, які допущені до моделювання виконання танцю:

$$\overline{np} = \frac{1}{B_c} \sum_{i=1}^{B_c} np_i ,$$

де B_c – кількість агентів, допущених до виконання танцю.

Крок 5.4. Вибір тих агентів, які за рахунок виконання танцю виконують вербування інших агентів для дослідження знайденого ними рішення. Вважається, що агент виконав вербування, якщо

$$\frac{L_i}{\beta} > \gamma \cdot \overline{np}, \forall i = 1, B_c ,$$

де $\beta > 0$ – коефіцієнт, що знижує вплив переваги танцю на ймовірність виконання вербування; $\gamma \in [0; 1]$ – граничний коефіцієнт, що визначає, наскільки вище повинна бути перевага танцю цього i -го агента щодо середньої корисності рішень всіх B_c агентів.

Крок 6. Якщо потрібно проводити локальну оптимізацію для знайдених кращих рішень, тоді виконується локальна оптимізація за допомогою зазначеного методу локальної оптимізації (наприклад, метод Нелдера-Мида, Хука-Дживса, Пауелла та ін. [28]). Залежно від установлених параметрів локальна оптимізація виконується або для рішень тих агентів, які зробили вербування, або тільки для рішення кращого агента.

Крок 7. Обирається агент із кращим рішенням *best*.

Крок 8. Перезапуск агентів. Створюються агенти, які будуть розглядатися як агенти-розвідники для наступної ітерації.

До нових агентів-розвідників будуть відноситися:

– агенти, що виконали вербування, та кращий агент;

– агенти, які стали зайнятими фуражирами внаслідок вербування.

Оскільки такі агенти повинні виконувати поліпшене вивчення вже існуючого джерела з нектаром, то при створенні рішень для даних агентів повинні враховуватися рішення агентів, що їх завербували. У зв'язку із цим для завербованих агентів рішення створюється в такий спосіб:

$$x_j = \text{dancedBee} \cdot x_j + \text{range} \cdot \text{rand} - \frac{\text{range}}{2}, \forall j = 1, \text{argCnt} ,$$

де *range* – це межа, у якій величина змінної може відхилятися від значення даної j -ої змінної в рішенні агента, що моделював танець, *dancedBee*.

– агенти, рішення яких створюється випадковим чином:

$$x_j = \text{rand} \cdot (\text{Range}_j^{\max} - \text{Range}_j^{\min}) + \text{Range}_j^{\min}, j = 1, \text{argCnt} .$$

Також для всіх створених агентів розраховується корисність обраного рішення.

Крок 9. Відновлення динамічних параметрів методу:

– збільшується кількість ітерацій: $\text{iter} = \text{iter} + 1$;

– змінюється поточна температура: $T = \alpha \cdot T$;

– змінюється межа *range*:

$$range = range \cdot \frac{iter_{\max} - iter}{iter_{\max}}.$$

Крок 10. Перевірка на зупинення. Перевірка на зупинення вважається успішною, якщо виконується хоча б одна з двох умов:

- досягнуто максимальну кількість ітерацій, тобто $iter = iter_{\max}$;
- поточна температура дорівнює кінцевій температурі ($T = T_{final}$).

Якщо перевірка на зупин дала успішний результат, тоді виконується перехід до кроку 11, у протилежному випадку – до кроку 3.

Крок 11. Зупинення.

12.5 Відбір інформативних ознак на основі методу бджолоїної колонії

В [25] описано застосування методу бджолоїної колонії для вирішення задач, заснованих на розподілі ресурсів (наприклад, транспортної задачі). У відповідності, із запропонованими в [25] математичними моделями поведінки бджіл, розроблено модифікацію методу бджолоїної колонії для відбору інформативних ознак при побудові моделей, що класифікують, у задачах діагностики й розпізнавання образів за ознаками, яка складається з таких кроків.

Крок 1. Ініціалізація. Задаються основні параметри методу бджолоїної колонії: кількість агентів B , максимальна кількість ітерацій T_{\max} , початкова кількість агентів-розвідників Ex_{start} , обмеження максимальної кількості агентів-розвідників Ex_{\max} , граничне значення корисності s_{min} . Також задається загальна кількість ознак M і кількість ознак N , які варто залишити. Після чого створюється простір пошуку порядку $N \times M$.

Крок 2. Запуск розвідників. Розвідники випадковим чином розташовуються у просторі пошуку. При цьому обране розташування (точка в просторі пошуку) повинно бути таким, щоб не було однакових значень координат.

Крок 3. Відправлення зайнятих фуражирів. Зайняті фуражири прикріплені до певних джерел ресурсу. Початкова кількість зайнятих фуражирів $B_e = 0$, оскільки на початку роботи методу ще немає джерел ресурсів, за яким можуть бути закріплені зайняті фуражири.

Корисність перебування агента в джерелі h на ітерації t , за умови, що в цьому джерелі перебуває x_h агентів, розраховується за формулою:

$$s_h(t) = \frac{a_h}{x_h(t)}, \quad h = 1, N \times M, x_h > 0,$$

де a_h – кількість корисної речовини, вироблюваної джерелом в одиницю часу. Кількість корисної речовини a_h визначається після створення моделі на основі позиції відповідного джерела. В аспекті задачі відбору ознак

кількість корисної речовини a_h пропонується розраховувати як зворотне значення помилки моделі ε_h :

$$a_h = \frac{E}{\varepsilon_h}, \varepsilon_h > 0,$$

де E – коефіцієнт, що знижує ступінь впливу помилки ε_h . Якщо $\varepsilon_h = 0$, то точка h вважається оптимумом і робота методу закінчується, тобто виконується перехід до кроку 9.

Якщо корисність перебування $s_h(t)$ досягає граничного значення ($s_h < s_{min}$), тоді агент поміщається у прилеглу точку з точкою h простору пошуку. Нове положення визначається шляхом зміни значення однієї з координат поточного положення агента:

$$z_r = z_r + k \cdot \Delta z,$$

де z_r – координата, що змінюється; r – випадковим чином обраний номер координати для зміни; k – коефіцієнт, що визначає напрямок зміни значення координати, може дорівнювати $+1$ або -1 ; Δz – межа, у якій може змінюватися змінна.

За рахунок цього виконується дослідження області, у якій знаходиться раніше виявлене джерело ресурсів.

Крок 4. Розрахунок корисності отриманого ресурсу. Сумарна корисність фуражування зайнятого фуражира або розвідника i розраховується за формулою:

$$F^i(t) = \begin{cases} 1, & \text{якщо } J_f(h^i(t)) + w_f^i(t) > 1; \\ J_f(h^i(t)) + w_f^i(t), & \text{якщо } e_n < J_f(h^i(t)) + w_f^i(t) < 1; \\ 0, & \text{якщо } 0 < J_f(h^i(t)) + w_f^i(t) < e_n, \end{cases}$$

де $F^i(t)$ – корисність фуражування i -го агента, $w_f^i(t)$ – шум у сумарній корисності. Шум рівномірно розподілений у межах $(-w_f; +w_f)$. Значення w_f обирається експериментально (пропонується $w_f = 0,1$), e_n – мінімальний поріг корисності. Мінімальний поріг обирається експериментально (пропонується $e_n = 0,1$); $J_f(h^i(t))$ – корисність джерела h^i , у якому побував i -ий агент на ітерації t . Корисність джерела h пропонується розраховувати за формулою:

$$J_f(h) = \frac{\varepsilon^*}{\varepsilon_h}, \varepsilon_h > 0,$$

де ε^* – задана (необхідна) точність рішення.

Корисність незайнятих фуражирів і відпочиваючих встановлюється: $F^i(t) = 0$.

Крок 5. Вибір кращого результату й перевірка, чи досягається задана точність ε^* . Якщо точність досягається, тоді виконується перехід до кроку 9, у протилежному випадку – перехід до кроку 6.

Крок 6. Моделювання виконання танцю, за рахунок чого досягається обмін інформацією. Кожний агент приймає рішення: виконувати або не виконувати танець. При цьому ймовірність виконання виляючого танцю i -им агентом на ітерації t розраховується за формулою:

$$p(i, t) = \frac{1}{\beta} L_f^i(t),$$

де $\beta > 0$ – коефіцієнт, що знижує вплив переваги шляху на ймовірність виконання танцю; $L_f^i(t)$ – перевага танцю i -го агента на ітерації t . $L_f^i(t)$ розраховується за формулою:

$$L_f^i(t) = \max \{ (F^i(t) - \alpha \bar{F}(t)), 0 \},$$

де $\bar{F}(t)$ – середнє значення корисності всіх джерел; α – коефіцієнт, керуючий впливом величини $\bar{F}(t)$ на $L_f^i(t)$.

Крок 7. Виділення нових розвідників і вербування. Кожний незайнятий фуражир може стати розвідником або почати рух за іншим агентом.

Ймовірність того, що незайнятий фуражир стане розвідником, розраховується за формулою:

$$p_e(t) = \exp \left(- \frac{1}{2} \frac{L_t^2(t)}{\sigma^2} \right),$$

де σ – коефіцієнт, що необхідний для моделювання поведінки фуражування; $L_t(t)$ – сума переваги танців різних агентів:

$$L_t(t) = \sum_{i=1}^B L_f^i(t).$$

Крім того, незайнятий фуражир може бути підданий вербуванню, тобто піти за i -им агентом. Ймовірність того, що незайнятий фуражир піде за i -им агентом, пропонується розраховувати за формулою:

$$p_i(t) = \frac{L_f^i(t)}{\sum_{j=1, j \neq i}^B L_f^j(t)}.$$

Крок 8. Збільшується лічильник ітерацій: $t = t + 1$. Якщо $t < T_{max}$, тоді виконати перехід до кроку 2, у протилежному випадку – перехід до кроку 9.

Крок 9. Зупинення.

12.6 Особливості методу бджолоїної колонії

Особливості розглянутих модифікацій методу бджолоїної колонії та відмінності між ними наведено в таблиці 12.1.

Виходячи з різних застосувань методу бджолоїної колонії можна виділити такі переваги методу:

Таблиця 12.1 – Відмінності й особливості модифікацій методу бджолоїної колонії

Критерій порівняння	Метод		
	BCO-JSSP	BCO	FBS
Процедура виконання виляючого танцю	Моделюється тривалість виконання танцю	Виконується залежно від якості складеного рішення	Як такої процедури виляючого танцю немає. Моделюється шляхом виконання вербування
Вибір агентів-розвідників	Початкова кількість агентів-розвідників дорівнює загальній кількості агентів. Після вербування кількість розвідників зменшується	Розвідниками є всі незайняті фуражири	Всі незайняті фуражири є розвідниками
Особливості вибору рішень розвідником	Розвідники знаходять рішення випадковим чином	Розвідники створюють випадкові рішення	Використовується інформація, залишена попередніми агентами
Дослідження рішень за рахунок інших агентів	Здійснюється за рахунок того, що вербування здійснюється фактично в кожному вузлі	Завербовані агенти просто переміщуються по обраному рішенню	Не відбувається. Зайняті фуражири переміщуються по обраному рішенню
Клас розв'язуваних задач	Задача календарного планування	Задачі комбінаторної оптимізації	Задача про складання пар
Вербування	Здійснюється при виборі агентом наступного пункту в графі рішень	Здійснюється після створення власного рішення	Вербування відбувається виходячи з корисності рішення

– метод є несхильним до зациклення в локальних оптимумах, оскільки заснований на випадковому пошуку;

– мультиагентність реалізації;

– пошук кращого рішення ґрунтується на рішеннях агентів всієї колонії бджіл;

– може використовуватися в динамічних застосуваннях, оскільки здатен адаптуватися до змін навколишнього середовища;

– може використовуватися для вирішення як дискретних, так і безперервних задач оптимізації.

До недоліків методу бджолоїної колонії можна віднести:

- досить високу ітеративність;
- труднощі теоретичного аналізу процесу одержання рішень, обумовлені тим, що пошук рішення має стохастичну природу;
- апріорну невизначеність часу збіжності, хоча збіжність гарантується;
- залежність методу від настроювання параметрів, що підбирають експериментально.

Метод бджолоїної колонії має такі особливості:

1) Всі агенти поділяються на різні типи у відповідності з діями, які вони виконують у процесі вирішення задачі.

1.1) Зайняті фуражири забезпечують використання вже знайдених джерел нектару, тобто незначно змінюють уже знайдені раніше рішення.

1.2) Незайняті фуражири забезпечують продовження пошуку нових джерел нектару, тобто агенти такого типу виконують пошук нових припустимих рішень. Незайняті фуражири у свою чергу бувають двох типів.

1.2.1) Спостерігачі – очікують у вулику інших агентів. Вони не виконують ніяких дій, вони фактично очікують моменту, коли їм потрібно буде також почати пошук рішень;

1.2.2) Розвідники – забезпечують пошук нових джерел нектару. При цьому пошук здійснюється випадковим чином, тобто вони випадково обирають у просторі пошуку можливе рішення.

2) Зв'язок між рішеннями агентів здійснюється шляхом моделювання виконання бджолами виляючого танцю. При цьому виконання виляючого танцю забезпечує утворення двох типів зворотного зв'язку: позитивний зворотний зв'язок і негативний зворотний зв'язок.

Позитивний зворотний зв'язок полягає в тому, що агенти, ґрунтуючись на інформації про рішення інших агентів, можуть почати досліджувати рішення, отримане іншим агентом.

Негативний зворотний зв'язок полягає в тому, що агенти, одержавши інформацію про знайдені рішення іншими агентами, можуть прийняти рішення щодо припиненні дослідження свого рішення у зв'язку з гіршими характеристиками в порівнянні з іншими отриманими рішеннями.

3) Процес пошуку рішення забезпечується двома процедурами.

3.1) Пошук нових джерел нектару у всьому просторі пошуку, що досягається за допомогою агентів-розвідників. Таким чином, забезпечується дослідження всього простору пошуку.

3.2) Поглиблене використання областей, у яких розміщено уже знайдені джерела нектару (досягається за допомогою зайнятих фуражирів). Тобто знаходяться рішення, що перебувають у просторі пошуку поблизу від розглянутого рішення.

РОЗДІЛ 13 PSO–МЕТОД

Оптимізація з використанням рою часток (Particle Swarm Optimization, PSO) – це метод пошуку, що базується на понятті популяції, і моделює поведінку птахів у зграї й косяків риб. Первісною метою використання концепції рою часток була графічна імітація гарного й непередбачуваного руху птахів або риб у зграї, з метою виявлення базових принципів, завдяки яким птахи літають (риби плавають) синхронно та вміють змінювати напрямок руху з перегрупованням в оптимальні формації. З тих часів концепція розвинулася в простий і перспективний оптимізаційний метод [93].

13.1 Фундаментальні принципи PSO-методу

В PSO-методі, особини, що називаються частками, переміщуються в багатовимірному просторі рішень. Зміни координат часток усередині простору пошуку обумовлюються природною соціально-психологічною тенденцією часток конкурувати між собою. Отже, зміни в стані частки залежать від досвіду й знань її сусідів. У цьому випадку, слово «знання» є синонімом «інформації». Результат такого моделювання полягає в тому, що процес пошуку мотивує частки недетермінованим чином повертатися в оптимальні ділянки простору рішень.

Особини в рої часток мають дуже просту поведінку: вони прагнуть перевершити досягнення сусідніх часток і поліпшити власні. Таким чином, емергентна властивість даної системи заключається в дослідженні оптимальних ділянок багатовимірного простору пошуку.

PSO-метод керує роєм часток і кожна частка являє собою потенційне рішення [94, 95]. За аналогією з еволюційними стратегіями, рій можна трактувати як популяцію, а частку як індивіда (або хромосому). Таким чином, частка «літає» у багатовимірному просторі рішень, та її позиція визначається виходячи із власного досвіду, і досвіду своїх сусідів.

За допомогою $x_i(t)$ позначається позиція частки i у просторі пошуку у момент часу t (t позначає дискретні значення часу). Позиція частки змінюється шляхом додавання швидкості $v_i(t)$ до поточної позиції:

$$x_i(t + 1) = x_i(t) + v_i(t + 1). \quad (13.1)$$

Початковий стан визначається в такий спосіб: $x_i(0) = U(x_{\min}, x_{\max})$, де $U(a, b)$ є функцією генерації випадкових чисел з діапазону $[a, b]$. Формула (13.1) являє собою вектор швидкості й визначає сам оптимізаційний процес, а також відображає використання, як отриманих знань частки, так

й обмін інформацією із сусідніми частками. Власні знання самої частки, що також називаються *когнітивним компонентом* формули швидкості, прямо пропорційні поточній відстані частки від її найкращого положення, що було знайдено з моменту старту її життєвого циклу. А обмін інформацією даної особини з іншими є *соціальним компонентом* формули швидкості.

Історично було розроблено два підходи, які фактично є різновидами базового PSO-методу: *gbest* та *lbest*; вони відрізняються ступенем зв'язаності часток у просторі пошуку.

13.2 Підхід *gbest* PSO

У різновиді *gbest* PSO-методу кожна частка зв'язана з усім роєм [96, 97]. Частки утворюють так звану соціальну мережу, що в *gbest* відповідає топології типу «зірка» (рис. 13.1). Кожна частка може взаємодіяти з усіма іншими частками, і вона тяжіє до кращого рішення усього рою. Частка імітує загальне оптимальне рішення, тому її швидкість залежить від інформації, одержуваної від усіх інших. У цьому випадку, соціальним компонентом швидкості є найкраща досягнута позиція рою (у просторі рішень), і вона позначається як $y^*(t)$.

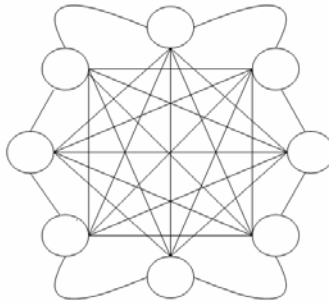


Рисунок 13.1 – Топологія типу «зірка»

В *gbest* PSO-методі швидкість частки розраховується за формулою:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[y^*_j(t) - x_{ij}(t)], \quad (13.2)$$

де $v_{ij}(t)$ – це швидкість частки i у вимірі j ($j = 1, \dots, n_x$) у момент часу t ; $x_{ij}(t)$ – це позиція частки i у вимірі j ; c_1 та c_2 – позитивні константи прискорення, які використовуються для варіювання ваг когнітивного й соціального компонентів швидкості.

льного компонентів швидкості частки відповідно; $r_{1j}(t), r_{2j}(t) = U(0,1)$ є випадковими значеннями з діапазону $[0,1]$. Ці випадкові величини приносять стохастичний елемент у роботу методу.

Величина y_i відображає найкращу позицію частки i , що вона відвідувала, починаючи з першої ітерації. Наступна оптимальна позиція частки i у момент часу $t + 1$ розраховується за формулою:

$$y_i(t+1) = \begin{cases} y_i(t), & \text{якщо } f(x_i(t+1)) \geq f(y_i(t)); \\ x_i(t+1), & \text{якщо } f(x_i(t+1)) < f(y_i(t)), \end{cases} \quad (13.3)$$

де $f: \mathfrak{R}^{n_x} \rightarrow \mathfrak{R}$ – фітнес-функція, де \mathfrak{R}^{n_x} – множина значень незалежних змінних, \mathfrak{R} – множина значень величини, що оптимізується. Також, як і в еволюційних підходах, вона є мірою близькості даного рішення до оптимального, також фітнес-функція визначає продуктивність, або якість частки.

Глобальний найкращий оптимум $y^*(t)$ у момент часу t , визначається як:

$$y^*(t) \in \{y_0(t), \dots, y_{n_s}(t)\} \mid f(y^*(t)) = \min\{f(y_0(t)), \dots, f(y_{n_s}(t))\}, \quad (13.4)$$

де n_s – загальна кількість часток у рої. Важливо відзначити, що відповідно до формули (13.4) y^* – це найкраща позиція, що була знайдена кожною із часток. Глобальний оптимум також може бути розрахований на основі інформації про частки з даного рою:

$$y^*(t) = \min\{f(x_0(t)), \dots, f(x_{n_s}(t))\}. \quad (13.5)$$

Метод *gbest* PSO може бути описаний у вигляді послідовності наступних кроків.

Крок 1. Якщо умови закінчення виконано, тоді вихід з роботи методу; у протилежному випадку – перехід до кроку 2.

Крок 2. Створити й ініціалізувати n_x -вимірний рій.

Крок 3. Встановити: $i = 1$.

Крок 4. Визначення локально кращої позиції. Якщо виконана умова $f(x_i) < f(y_i)$, тоді встановити $y_i = x_i$.

Крок 5. Визначення глобально найкращої позиції. Встановити: $y^* = y_i$.

Крок 6. Встановити: $i = i + 1$.

Крок 7. Якщо $i < n_s$, тоді виконати перехід до кроку 3, у протилежному випадку – до кроку 8.

Крок 8. Встановити: $i = 1$.

Крок 9. Оновити швидкість частки, використовуючи формулу (13.2).

Крок 10. Оновити позицію частки, використовуючи формулу (13.1).

Крок 11. Встановити: $i = i + 1$.

Крок 12. Якщо $i < n_s$, то виконати перехід до кроку 9, у протилежному випадку до кроку 1.

13.3 Підхід lbest PSO

У локально найкращому PSO-методі, або *lbest* PSO, частки утворюють соціально-мережну топологію типу «кільце» (рис. 13.2), і для кожної частки задається менша кількість сусідів, ніж при *gbest* підході [97].

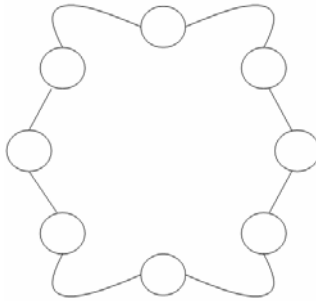


Рисунок 13.2 – Топологія типу «кільце»

Кожна частка намагається імітувати кращого зі своїх сусідів, тяжіючи до нього. Соціальний компонент швидкості відображає обмін інформацією між сусідами частки. Відповідно до формули (13.1) внесок у швидкість частки пропорційний відстані між нею й найкращим рішенням, знайденим її сусідами. Швидкість розраховується за формулою:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[y^*_{ij}(t) - x_{ij}(t)], \quad (13.6)$$

де $y^*_{ij}(t)$ – найкраща позиція, знайдена сусідами i -ої частки у вимірі j . Локально найкраща позиція частки y^*_i , тобто краща позиція, знайдена в сусідстві N_i та визначається як:

$$y^*_i(t+1) \in \{N_i | f(y^*_i(t+1))\} = \min\{f(x)\}, \forall x \in N_i, \quad (13.7)$$

де сусідство визначається за формулою:

$$N_i = \{y_{i-n_{N_i}}(t), y_{i-n_{N_i}+1}(t), \dots, y_{i-1}(t), y_i(t), y_{i+1}(t), \dots, y_{i+n_{N_i}}(t)\}, \quad (13.8)$$

для сусідства розмірністю n_{N_i} .

Метод *lbest* майже еквівалентний *gbest*, з тією лише різницею, що на кроці 9 швидкість частки оновлюється за формулою (13.6), а не (13.2), а зв'язані частки визначаються більш складним чином.

Крок 1. Якщо *умови закінчення* виконано, тоді вихід з роботи методу; у протилежному випадку – перехід до кроку 2.

Крок 2. Створити й ініціалізувати n_x -вимірний рій.

Крок 3. Встановити: $i = 1$.

Крок 4. Визначення локально найкращої позиції. Якщо виконано умову $f(x_i) < f(y_i)$, тоді встановити $y_i = x_i$.

Крок 5. Визначення глобально найкращої позиції. Встановити: $y^* = y_i$.

Крок 6. Встановити: $i = i + 1$.

Крок 7. Якщо $i < n_s$, тоді перехід до кроку 3, у протилежному випадку – перехід до кроку 8.

Крок 8. Встановити: $i = 1$.

Крок 9. Оновити швидкість частки, використовуючи формулу (13.6).

Крок 10. Оновити позицію частки, використовуючи формулу (13.1).

Крок 11. Встановити: $i = i + 1$.

Крок 12. Якщо $i < n_s$, тоді виконати перехід до кроку 9, у протилежному випадку – до кроку 1.

Важливо відзначити, що в базовому варіанті PSO-методу частки взагалі не зв'язані. Вибір зв'язаних часток здійснюється на основі їхніх індексів. Проте, були розроблені стратегії, де сусідні частки визначаються виходячи з їхньої просторової подоби.

Існує, принаймні, дві причини, чому треба віддати перевагу індексам часток у питанні визначення множини зв'язаних часток.

1. Це недорого в обчислювальному плані, оскільки не потрібно просторової нумерації часток. У підходах, де сусідство визначається на основі розрахунку відстаней, потрібно обчислювати Евклідову відстань між всіма парами часток, що має складність біля $O(n_s^2)$.

2. Даний підхід забезпечує інформацією про оптимальні рішення всі частки, незалежно від їхнього місця розташування в просторі.

13.4 Компоненти швидкості частки в PSO-методі

Розрахунок швидкості частки виконується за формулами (13.2) та (13.6). Розглянемо компоненти швидкості більш детально [92, 93].

1. *Попередня швидкість* $v_i(t)$ виступає в ролі пам'яті частки, тобто містить інформацію про її переміщення в минулому. Цю пам'ять можна ви-

значити як імпульс, що запобігає стрибкоподібним змінам напрямку руху частки. Дана швидкість також є інерційним компонентом.

2. *Когнітивний компонент* $c_1 r_1 (y_i - x_i)$ визначає продуктивність i -ої частки відносно минулих результатів і виступає в ролі індивідуальної пам'яті про найбільш оптимальні позиції даної частки. Використовуючи його, частка може повертатися в стани, які були найкращими для неї в минулому. Це є однією з емергентних властивостей системи в цілому.

3. *Соціальний компонент* $c_2 r_2 (y^* - x_i)$ в *gbest* PSO або $c_2 r_2 (y^*_i - x_i)$ в *lbest* PSO визначає продуктивність частки відносно сусідніх або зв'язаних з нею. Завдяки йому частка має можливість пересуватися в оптимальні позиції, які були знайдені сусідніми частками.

Ступінь внеску когнітивного й соціального компонентів визначається випадковими величинами $c_1 r_1$ та $c_2 r_2$, відповідно.

Ефект впливу швидкості на поведінку частки може бути продемонстрований графічно [98]. Розглянемо рух частки у двовимірному просторі рішень. Рисунок 13.3, *а*) відображає стан одиничного рою в момент часу t . У момент часу $t+1$, як показано на рис. 13.3, *б*), індивідуальна найкраща позиція частки не змінилася, але інші компоненти однаково пересувають її у напрямку кращої частки $y^*(t)$.

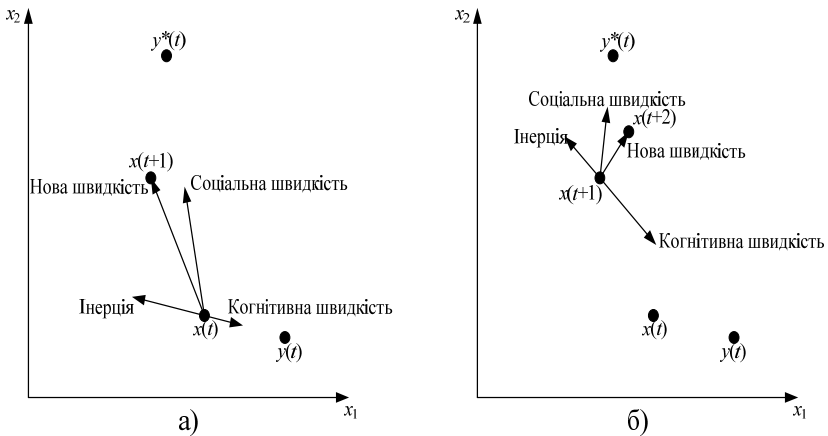


Рисунок 13.3 – Геометрична ілюстрація зміни швидкості й позиції частки:
а) момент часу t ; *б*) момент часу $t+1$

13.5 Ініціалізація та закінчення PSO-методу

Відповідно до методів *gbest* та *lbest* оптимізаційний процес є ітеративним і триває доти, поки не будуть виконані умови закінчення (хоча б одна з них) [93, 94]. Кожна ітерація полягає у виконанні всіх кроків, починаючи з першого, тобто це визначення індивідуально кращої позиції кожної частки рою, глобального оптимуму, і розрахунок швидкостей часток. У середині кожної ітерації певну кількість разів відбувається виклик фітнес-функції. У базовому варіанті PSO-методу ця функція викликається не більше n_s разів, де n_s – це кількість часток у рої.

На першому кроці базового PSO-методу ініціалізується рій і керовані параметри, тобто константи прискорення, c_1 та c_2 , початкові швидкості, позиції часток і їх індивідуальні кращі стани. Метод *lbest* також вимагає визначення кількості зв'язків кожної частки із сусідніми. Звичайно, позиції часток обираються таким чином, щоб рівномірно покривати весь простір пошуку. Також якість роботи PSO-методу визначається початковим ступенем різноманітності рою. Якщо деякі локальні ділянки не покриваються частками спочатку, то з великим ступенем ймовірності процес пошуку може взагалі не досягти цих підпросторів. Дані ділянки можуть бути досліджені, тільки якщо імпульси окремих часток перенесуть їх у ці ділянки.

Припустимо, що оптимум повинен бути знайдений у просторі, обмеженому двома векторами, x_{\min} та x_{\max} , які відповідно знизу й зверху обмежують діапазон значень кожного виміру. Тоді досить ефективний метод ініціалізації позицій часток можна записати в такий спосіб (у цьому випадку $r_j = U(0,1)$):

$$x(0) = x_{\min,j} + r_j(x_{\max,j} - x_{\min,j}), \forall j = 1, \dots, n_x, \forall i = 1, \dots, n_s. \quad (13.9)$$

Початкові швидкості часток можна прийняти:

$$v_i(0) = 0. \quad (13.10)$$

Також можливо ініціалізувати швидкості випадковими значеннями, але варто робити це з обережністю. Фактично, відповідно до фізичної аналогії, нульові швидкості часток говорять про те, що вони стаціонарні, у протилежному випадку – аналогія порушується. Випадкові значення швидкостей означають, що метод вже працює, і тоді частки мають також випадкові напрямки руху. Тому, навіть якщо застосовується подібний метод ініціалізації, не слід використовувати занадто великі випадкові числа, оскільки частки можуть покинути простір пошуку.

Індивідуальна найкраща позиція кожної частки при ініціалізації повинна дорівнювати позиції даної частки в момент часу $t = 0$, тобто:

$$y_i(0) = x_i(0). \quad (13.11)$$

Закінчення виконання PSO-метода відбувається, коли виконується хоча б один із критеріїв зупину. Існують два аспекти, які повинні бути враховані при виборі подібних мір.

1. Умова зупинення не повинна спричиняти передчасну збіжність PSO-методу. У протилежному випадку будуть отримані субоптимальні рішення.

2. При перевірці критеріїв зупинення фітнес-функція має викликатися мінімальну кількість разів, оскільки від цього сильно залежить швидкість роботи PSO-методу.

Звичайно використовуються такі критерії зупинення:

- досягнення максимальної кількості ітерацій;
- знаходження задовільного рішення;
- вихід при відсутності поліпшень протягом багатьох ітерацій;
- вихід за умови, що перша похідна фітнес-функції в околиці частки або часток дорівнює нулю.

13.6 Вибір параметрів PSO-методу

Базовий PSO-метод залежить від вибору множини керованих параметрів таких як: кількість часток, коефіцієнти прискорення, інерційні ваги, кількість зв'язків між частками, ваги соціальних і когнітивних компонентів [92, 93].

Розглянемо вплив кожного із цих параметрів по черзі.

Розмір рою n_s . Очевидно, що чим більше часток, тим більшу різноманітність вони мають. Також великий рій здатний більш рівномірно покривати простір пошуку. Однак це також збільшує обчислювальну складність на кожній ітерації, знижуючи при цьому їхню загальну кількість. Емпіричні дослідження показали, що PSO-метод може ефективно працювати навіть із малою кількістю часток: від 10 до 30 [99]. Але варто пам'ятати, що гладкий, на відміну від нерівного, простір пошуку вимагає меншої щільності покриття. Тому бажано величину n_s підбирати окремо для кожної задачі.

Кількість зв'язків між частками. Ця величина визначає ступінь соціальної взаємодії часток усередині рою. Чим вона нижче, тим менше кооперація. При малій кількості зв'язків PSO-метод сходиться до глобального рішення більш повільно, що запобігає потраплянню в локальні оптимуми. Для того, щоб досягти вигоди від цих двох різних підходів, варто починати пошук з малою кількістю зв'язків між частками, а потім їхню кількість рівномірно збільшувати з ходом ітерацій.

Кількість ітерацій. Це величина, що в остаточному підсумку визначає результат роботи PSO-методу, і вона також є проблемно-залежною. Очевидно, що при малій кількості ітерацій робота методу може закінчитися передчасно, а при великій – безпідставно збільшиться обчислювальна складність методу.

Коефіцієнти прискорення, c_1 та c_2 . Разом з випадковими векторами r_1 та r_2 , вони контролюють стохастичний ступінь впливу когнітивного й соціального компонентів на загальну швидкість частки. Коефіцієнти c_1 та c_2 також називаються довірчими вагами. При $c_1 = c_2 = 0$ частка пересувається доти, поки не покине простір пошуку. Якщо $c_1 > 0$, а $c_2 = 0$, то всі частки намагаються досягти оптимуму поза залежністю одна від одної. Але успіх PSO-методу забезпечується тільки взаємною кооперацією часток, тому важливо, щоб коефіцієнти були збалансовані, наприклад, $c_1 \approx c_2$. При малих значеннях коефіцієнтів частки переміщуються по гладких траєкторіях, при більших – спостерігається стрибкоподібний рух.

13.7 Багатокритеріальний PSO-метод

Відомо декілька модифікацій базового PSO-методу, які дозволяють виконувати багатокритеріальний пошук.

У методі *динамічного сусідства* MOPSO (Multiple Objective Particle Swarm), що був запропонований Хью та Ебернатом [100], динамічно визначаються групи зв'язаних часток на кожній ітерації, базуючись на їхніх відстанях у просторі. Нехай $f_1(x)$ буде першою цільовою функцією, а $f_2(x)$ – другою. Тоді, відповідно до методу, сусіди даної частки визначаються як найближчі в сенсі значень цільової функції $f_1(x)$. А сусідня найкраща частка визначається як найближча в сенсі значень функції $f_2(x)$. Індивідуально найкраща позиція частки змінюється тільки в тому випадку, якщо частка перейшла в кращий стан.

Проте, даний метод має кілька недоліків:

- метод може працювати тільки із двома цільовими функціями;
- метод припускає використання апріорних знань (яка із двох функцій є більш простою);
- метод залежить від порядку розташування цільових функцій і віддає перевагу другій при оптимізації.

Парсонулус і Врахатіс розробили векторно-оціночний PSO-метод VEPSO (Vector Evaluation Particle Swarm Optimization) [101], що базується на векторному генетичному алгоритмі, розробленому Шафером. VEPSO використовує дві підпопуляції, де кожний рій оптимізує одну цільову функцію. VEPSO являє собою один з різновидів еволюційного підходу. Глобально найкраще рішення першого рою використовується для розрахунку швидкостей другого рою, і навпаки. Таким чином:

$$\begin{aligned}
 S_1.v_{ij}(t+1) &= wS_1.v_{ij}(t) + c_1r_{1j}(t)(S_1.y_{ij}(t) - S_1.x_{ij}(t)) + \\
 &+ c_2r_{2j}(t)(S_2.y^*_i(t) - S_1.x_{ij}(t)), \\
 S_2.v_{ij}(t+1) &= wS_2.v_{ij}(t) + c_1r_{1j}(t)(S_2.y_{ij}(t) - S_2.x_{ij}(t)) + \\
 &+ c_2r_{2j}(t)(S_1.y^*_i(t) - S_2.x_{ij}(t)),
 \end{aligned}
 \tag{13.12}$$

де підпопуляція S_1 робить обчислення на базі цільової функції $f_1(x)$, а S_2 – на базі $f_2(x)$.

13.8 PSO-метод з керуванням зміною швидкості часток

Гарний оптимізаційний метод має володіти двома важливими властивостями: повністю досліджувати простір рішень задачі, і фіксувати пошук біля потенційних оптимумів. Інакше кажучи, метод повинен балансувати між цими двома станами. В PSO-методі баланс визначається формулою швидкості часток, а саме рівняннями (13.2) та (13.6) для $gbest$ та $lbest$ методів відповідно.

В ранніх реалізаціях PSO-методу було виявлено, що швидкості можуть занадто швидко зростати, особливо швидкості тих часток, які знаходяться далеко від свого власного оптимуму або загального оптимуму своїх сусідів. У результаті, такі частки можуть залишати простір пошуку, що винятково негативно впливає на роботу методу в цілому. Для запобігання подібних ситуацій варто контролювати діапазон зміни швидкостей часток. Якщо швидкість частки перевищує максимально допустиму, то її варто знизити до дозволеного рівня. Нехай $V_{\max,j}$ – максимально припустима швидкість частки у вимірі j . Тоді швидкість частки пропонується змінювати в такий спосіб:

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1), & \text{якщо } v_{ij}(t+1) = V_{\max,j}; \\ V_{\max,j}, & \text{якщо } v'_{ij}(t+1) \geq V_{\max,j}, \end{cases} \quad (13.13)$$

де значення v'_{ij} розраховується за формулою (13.2) або (13.6).

Величина $V_{\max,j}$ є досить важливою, оскільки вона контролює загальні масштаби пошуку. Більші значення змушують PSO-метод досліджувати простір задачі більш ретельно. Якщо значення $V_{\max,j}$ є досить невеликим, то PSO-метод може взагалі не покинути локальних ділянок, а також збільшується кількість ітерацій, потрібних для досягнення оптимуму.

Виходячи з вищесказаного очевидно, що $V_{\max,j}$ варто підбирати таким чином, щоб балансувати між швидким і повільним пересуваннями часток, і між фіксацією пошуку й загальним дослідженням простору. Пропонується розраховувати $V_{\max,j}$ у такий спосіб:

$$V_{\max,j} = \delta(x_{\max,j} - x_{\min,j}), \quad (13.14)$$

де $x_{\max,j}$ та $x_{\min,j}$ – величини, які, відповідно, зверху та знизу обмежують діапазон значень виміру j , а $\delta \in (0,1]$. Очевидно, що величина δ є проблемно-залежною.

Відзначимо дві важливі властивості запропонованої модифікації PSO-методу:

1. Величина $V_{\max,j}$ обмежує не простір пошуку, всередині якого пересуваються частки, а тільки швидкості часток (точніше діапазон їхньої зміни за одну ітерацію).

2. Максимальна швидкість часток визначається для кожного виміру окремо й обумовлюється тільки його розмірністю (формула (13.14)).

Незважаючи на те, що в запропонованій модифікації швидкості часток контролюються за допомогою величини $V_{\max,j}$, у даного методу є й недоліки, які слід розглянути. У першу чергу слід зазначити, що обмеження швидкості частки також може викликати зміну напрямку її пересування. Цей ефект зображено на рис. 13.4. У цьому випадку, $x_i(t+1)$ – це позиція i -ої частки без обмежень по швидкості. А величина $x'_i(t+1)$ вже відображає позицію частки після того, як швидкість стала рівною $V_{\max,j}$. З іншого боку, таке «втручання» може направити частки саме в напрямку до оптимуму.

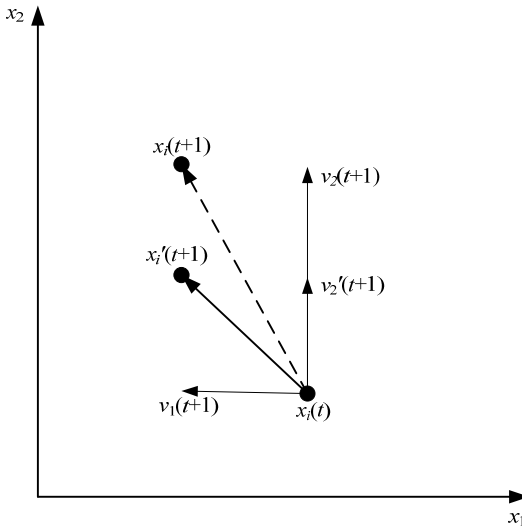


Рисунок 13.4 – Ефект використання обмеження швидкості частки

Другий небажаний ефект може виникнути у випадку, якщо швидкості всіх часток стануть рівними $V_{\max,j}$. Для його запобігання можна запропонувати з ходом ітерацій зменшувати значення $V_{\max,j}$. Тоді модифікований PSO-метод буде послідовно звужувати масштаби дослідження простору пошуку, що робить оптимізаційний процес більше ефективним.

13.9 Порівняння різновидів PSO-методів

Виконаємо порівняння різновидів базового PSO-методу. Включимо в аналіз також модифікований метод, що гнучко регулює швидкості часток у просторі пошуку. Порівняльний аналіз проведемо на основі найбільш важливих і істотних розходжень, виявлених при їхньому дослідженні, а також на основі запропонованих критеріїв. Аналіз наведено у табл. 13.1.

Таблиця 13.1 – Порівняльний аналіз різновидів PSO-методу

Критерії	Методи			
	Метод gbest PSO	Метод lbest PSO	PSO-метод з керуванням зміною швидкості часток	Багатокритеріальний PSO-метод
Адаптивність	Висока	Висока	Висока	Низька
Масштабованість	Середня	Середня	Середня	Середня
Гнучкість моделі	Середня	Висока	Висока	Низька
Швидкість збіжності	Висока	Низька	Середня	Сильно залежить від вирішуваної задачі
Якість оптимізації	Середня	Висока	Висока	Низька
Помилкостійкість	Висока	Висока	Висока	Низька
Кількість обчислень фітнес-функції	Середня	Низька	Середня	Висока

Різновиди lbest і gbest базового PSO-методу подібні в тому плані, що в обох випадках вони рухаються до глобального оптимуму при оновленні компонентів швидкості.

Але між цими підходами існує два важливих розходження, які впливають на збіжність:

– через більший ступінь зв'язності часток між собою (топология «зірка») gbest PSO-метод сходиться швидше; однак швидка збіжність веде до менш ретельного дослідження простору рішень;

– різновид lbest PSO-методу має менший шанс потрапити в локальний оптимум і знайти, таким чином, тільки субоптимальне рішення; з іншої сторони, даний метод працює більш повільно, ніж gbest.

13.10 Напрями використання PSO-методу

Перші напрями використання PSO-методу варто віднести до навчання нейронних мереж (НМ) прямого поширення [102]. З тих пір, у цілому ряді робіт було показано, що PSO-метод також можна застосовувати як метод навчання НМ будь-якої архітектури. У подібних розробках варто визначити потрібний вигляд фітнес-функції, що може бути подана як функція помилки, яку потрібно мінімізувати. Кожна частка в PSO-методі являє собою окреме рішення оптимізаційної проблеми, у цьому випадку – це ціла нейронна мережа. А кожна координата частки в просторі пошуку – це значення відповідного вагового коефіцієнту.

Еберат та Кеннеді надали перші результати застосування різних модифікацій PSO-методу в задачах навчання НМ з учителем і без нього. Мендес досліджував якість роботи цих методів з використанням різноманітних топологій (кільце, квадрат, зірка, піраміда). Хірата досліджував можливість *gbest* і *lbest* підходів у задачах навчання.

Жанг та Шао запропонували використання PSO-методу до безперервної оптимізації ваг НМ і її архітектури. Для цього варто використати два рої часток: перший шукає оптимальну архітектуру, а другий – оптимізує ваги. Частки в першому рої пересуваються тільки у двох вимірах: перша координата – кількість схованих шарів, а друга – щільність зв'язків. На першій ітерації роботи алгоритму частки ініціалізуються випадковим чином у заданих інтервалах. Частки другого рою шукають оптимальний набір ваг НМ. Даний PSO-метод для кожної частки з першого рою створює групу часток у другому. Після того як метод закінчить свою роботу, вибирається потрібна архітектура й відповідна їй оптимальна множина ваг НМ.

РОЗДІЛ 14

МЕТОД ОПТИМІЗАЦІЇ НА ОСНОВІ МОДЕЛЮВАННЯ ПЕРЕМІЩЕННЯ БАКТЕРІЙ

14.1 Біологічні основи

Однією з найбільш вивчених бактерій є бактерія *E. Coli* [108–112]. Ця бактерія існує в кишечнику більшості ссавців (у тому числі й у кишечнику людини). Бактерія *E. Coli* при відповідних умовах може самовідтворюватися (розщеплюватися) за 20 хв. Здатність рухатися бактерії *E. Coli* забезпечується за рахунок шести або більше жгутиків, які обертаються з частотою 100–200 обертів за секунду, кожний з яких управляється за допомогою свого власного біологічного мотору. Бактерія *E. Coli* чергує свої дії: то переміщується прямолінійно (зі швидкістю 10–20 мкм/сек.), то обертається (зміна напрямку). Коли жгутики обертаються за годинниковою стрілкою (або проти годинникової стрілки), вони працюють як пропелери й, отже, *E. Coli* може плисти або обертатися.

Загальна схема руху бактерії виглядає таким чином: коли мотор працює в одну сторону, всі жгутики в бактерії складаються й обертаються разом, і бактерія рухається прямолінійно. Наприкінці пробігу бактерія зупиняється, мотор перемикається й починає працювати в іншу сторону. Жгутики розчіплюються й «бвтаються» незалежно один від одного. Бактерія при цьому переорієнтується в просторі випадковим чином. Після цього, мотор знову перемикається й починає працювати в ту сторону, у яку жгутики працюють разом, і виникає наступний відрізок прямолінійного руху.

Хемотаксис – це рухова реакція бактерії у відповідь на появу в середовищі атрактанта (атрактант – речовина, що приваблює бактерії) або репеленту (репелент – речовина, що відштовхує бактерії) [113, 114]. Зрозуміло, що в природних умовах атрактантами є речовини, корисні для бактерій, а репелентами – ті, які бактеріям шкідливі.

При наявності просторових змін концентрації атрактантів або репелентів частота перекидань, а, отже, і довжина вільного пробігу бактерії змінюються. Довжина вільного пробігу бактерії, що пливе у бік зростаючої концентрації атрактанта, збільшується, а при русі у бік зростаючої концентрації репеленту зменшується.

Таким чином, можна виділити такі хемотаксичні дії бактерії *E. Coli* [115]:

– якщо бактерія знаходиться в нейтральному середовищі, то чергуються обертання й пересування, за рахунок чого виконується пошук.

– якщо бактерія переміщається по градієнту атрактанта, то переміщення продовжується в тому ж напрямку. Таким чином, забезпечується пошук більш сприятливого навколишнього середовища.

– якщо відбувається переміщення в напрямку протилежному градієнту репеленту, то забезпечується запобігання несприятливого навколишнього середовища.

Таким чином, бактерія може переміщатися по корисних речовинах і в той же час уникати небезпечних речовин. Датчики, використовувані *E. Coli* – білкові рецептори, які є дуже чутливими (незначна зміна в концентрації корисних речовин може викликати істотні зміни в поведінці бактерії). Від рецепторів сигнали надходять на метилакцептуючі білки, які збирають всі сигнали від рецепторів, і результируючий сигнал виходить на мотор жгутика, що управляє рухом бактерії залежно від співвідношення корисних і небезпечних речовин у навколишнім середовищі.

Бактерії часто вмирають або розчиняються, і це повинно враховуватися при моделюванні їхньої діяльності. Мутації в *E. Coli* відбуваються в нормальних умовах приблизно 10^{-7} у ген і в покоління.

E. Coli можуть формувати складні стійкі просторово-часові структури в деяких напівтвердих корисних середовищах. Бактерії можуть поглинати корисні речовини по їхньому радіусу, починаючи від зовнішньої границі, закінчуючи серединою, навіть у випадку якщо бактерії були спочатку розміщені в центрі корисних речовин. Крім того, за певних умов, вони можуть приховувати сигнали, що притягають, від клітини до клітини, за рахунок чого вони групуються й захищають одна одну. Таким чином, ці бактерії можуть групуватися в колонії.

14.2 Метод оптимізації на основі моделювання переміщення бактерій (Bacterial Foraging Optimization, BFO)

Даний метод призначений для знаходження мінімуму функції $J(X)$, $X \in R^p$ при невідомому градієнті $\nabla J(X)$, де X – позиція бактерії в просторі пошуку R^p , а за допомогою $J(\theta)$ моделюються корисні й шкідливі властивості середовища, тобто $J(X)$ характеризує, де знаходяться атрактанти й репеленти. Таким чином, $J < 0$, $J = 0$, $J > 0$ означає, що бактерія перебуває в корисному, нейтральному та шкідливому середовищі, відповідно.

Нехай $P(j, k, l) = \{X^i(j, k, l), i=1, 2, \dots, S\}$ описує позицію кожного члена популяції S бактерій на j -му хемотаксичному кроці, k -ому кроці відтворення та на l -ій події виключення-розсіювання.

Тоді роботу методу оптимізації на основі моделювання переміщення бактерій можна подати у вигляді послідовності виконання наступних кроків [116].

Крок 1. Ініціалізація. Задаються параметри, що впливають на роботу методу: S – кількість бактерій, N_{re} – кількість кроків відтворення, N_s – кількість кроків-повторень на одному хемотаксичному кроці, N_c – кількість

хемотаксичних кроків, N_{ed} – кількість подій виключення-розсіювання; P_{ed} – ймовірність розсіювання. Випадковим чином розподілити початкові значення X^i , $i = 1, 2, \dots, S$, по простору пошуку. Розраховуються початкові значення цільової функції для кожної бактерії J^i .

Крок 2. Встановити: $l = l + 1$.

Крок 3. Встановити: $k = k + 1$.

Крок 4. Встановити: $j = j + 1$.

Крок 5. Для кожної бактерії моделюється хемотаксис: обертання, переміщення й ковзання.

Крок 5.1. Встановити: $i = i + 1$.

Крок 5.2. Обертання. Моделювання обертання досягається за рахунок генерації вектора випадкових чисел $\varphi(j) \in R^p$:

$$\varphi = \frac{\Delta}{\sqrt{\Delta^T \Delta}},$$

де Δ – вектор випадкових чисел в інтервалі $[-1; 1]$.

Вектор φ являє собою множину довжин для відповідних вимірів.

Крок 5.3. Переміщення. Розраховується нове положення i -ої бактерії за формулою:

$$X^i(j+1, k, l) = X^i(j, k, l) + C(i)\varphi(j),$$

де $C(i) > 0$ – розмір кроку в певному напрямку, що дозволяє моделювати процес обертання.

Для нової позиції $X^i(j+1, k, l)$ розраховується відповідне значення цільової функції $J(i, j+1, k, l)$.

Крок 5.4. Ковзання. Якщо в позиції $X^i(j+1, k, l)$ значення $J(i, j+1, k, l)$ краще, ніж у позиції $X^i(j, k, l)$, тобто виконується умова: $J(i, j+1, k, l) < J(i, j, k, l)$, тоді виконується наступний хемотаксичний крок з тим же вектором φ і в тому ж напрямку (перехід до кроку 5.3), і таке повторення може відбуватися N_s раз. Якщо умова не виконується, тоді перехід до кроку 5.5.

Шаг 5.5 Якщо $i < S$, тоді перехід до кроку 5.1, у протилежному випадку – перехід до кроку 6.

Крок 6. Якщо $j < N_c$, тоді перехід до кроку 4, у протилежному випадку – перехід до кроку 7.

Крок 7. Відтворення. Менш здорові бактерії вмирають, а інші, більш здорові, розділяються на дві бактерії, при цьому нові бактерії розміщуються в ту ж точку простору пошуку. За рахунок такого підходу забезпечується незмінність загальної кількості бактерій. Для цього всі агенти сортуються відповідно до отриманих значень цільової функції, після чого гірша половина (менш здорові бактерії) відкидається, а краща (більше здорові бактерії) – дублюється.

Крок 8. Якщо $k < N_{re}$, тоді виконується перехід до кроку 3, у протилежному випадку – перехід до кроку 9.

Крок 9. Виключення та розсіювання. Життя популяції бактерій у навколишньому середовищі може змінюватися або поступово (наприклад, шляхом споживання корисних речовин), або раптово у зв'язку з деяким іншим впливом. Може відбутися так, що всі бактерії в області загинуть, або колонія бактерій буде розсіяна в іншу частину навколишнього середовища. Даний ефект може перешкодити можливому хемотаксичному прогресу, але в той же час цей ефект і допомагає, оскільки, у випадку розсіювання, бактерії можуть розміститися біля гарних джерел з корисними речовинами. Виключення й розсіювання допомагають понизити ймовірність стагнації, тобто зациклення в локальному оптимумі, що часто спостерігається в традиційних градієнтних методах оптимізації.

Відповідно до даного підходу кожна бактерія з ймовірністю P_{ed} розміщується у випадково обраній точці простору пошуку.

Таким чином, перевіряється умова: $U^i < P_{ed}$, де U^i – випадкове число в інтервалі $[0;1]$ для i -ої бактерії.

Якщо дана умова виконується, то бактерія переміщується в позицію $X^i(j, k, l)$, отриману випадковим образом.

Крок 10. Якщо $l < N_{ed}$, тоді виконується перехід до кроку 2, у протилежному випадку – до кроку 11.

Крок 11. Обирається й зберігається краще рішення J_{best} і відповідна позиція X_{best} , у якій досягається краще рішення J_{best} .

Крок 12. Перевірка на закінчення пошуку. Якщо були виконані всі цикли для всіх агентів, тоді виконується перехід до кроку 14, у протилежному випадку виконується перезапуск – перехід до кроку 13.

Крок 13. Перезапуск агентів: вибираються нові випадкові позиції для кожного агента X^i , $i = 1, 2, \dots, S$, і розраховуються відповідні значення цільової функції J^i , $i = 1, 2, \dots, S$. Встановлюються лічильники циклів: $j = 0$; $k = 0$; $l = 0$.

Крок 14. Зупинення.

Загальна схема роботи методу оптимізації на основі моделювання переміщення бактерій наведена на рис. 14.1.

Розглянутий метод оптимізації на основі моделювання переміщення бактерій не враховував одну з важливих особливостей поведінки бактерій: бактерії *E. Coli*, як було сказано раніше, здатні, групуючись, створювати просторово-часові структури. У зв'язку з цим надалі було запропоновано кілька підходів, що дозволяють враховувати в процесі моделювання переміщення бактерій і дану особливість їхньої поведінки, основними з яких є: моделювання сигналів між клітинами [116] і застосування PSO-оператора (Particle Swarm Optimization, PSO) [117].

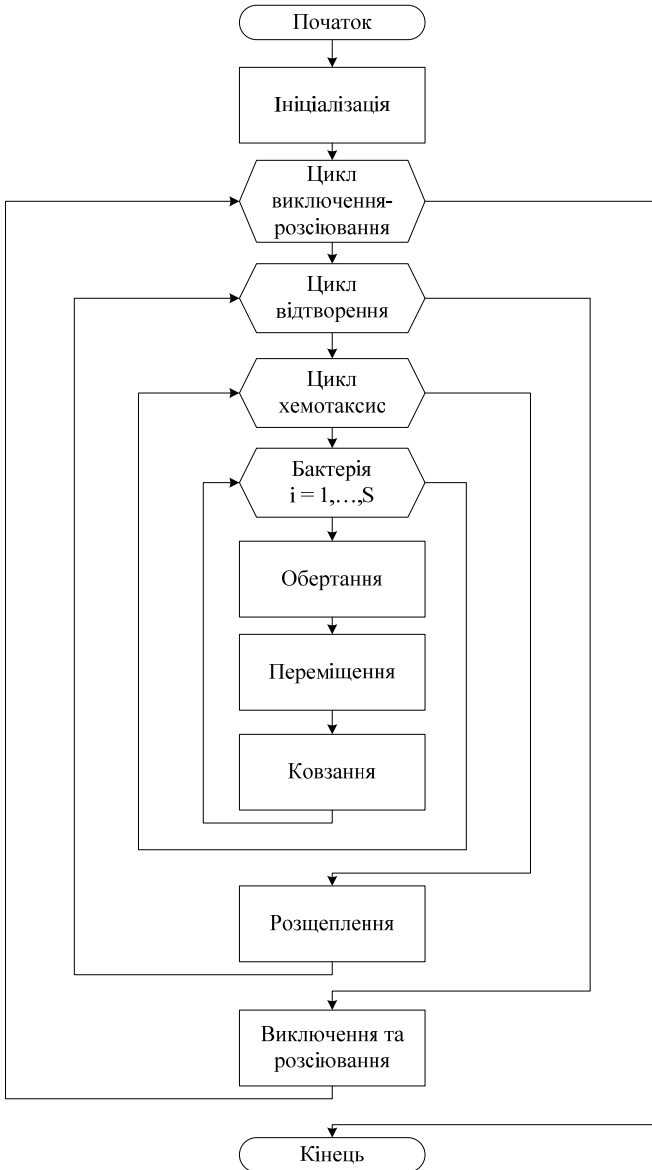


Рисунок 14.1 – Загальна схема роботи методу на основі моделювання переміщення бактерій

14.3 Метод оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок зв'язку між клітинами

Нехай $d_{attract}$ – глибина атрактанта (корисних речовин) клітини, а $w_{attract}$ – міра ширини атрактанта. Клітини виключають одна одну за допомогою локального споживання, а також за рахунок того, що клітини не є їжею одна для одної. Нехай $h_{repellent}=d_{attract}$ – висота шкідливої речовини (репелент), і $w_{repellent}$ – міра ширини репеленту. Таким чином, можна використати функцію $J_{cc}^i(X)$, $i=1, 2, \dots, S$ для моделювання сигналів між клітинами за допомогою виділення бактеріями атрактанта й репеленту:

$$J_{cc}(X) = \sum_{i=1}^S J_{cc}^i = \sum_{i=1}^S \left[-d_{attract} \exp\left(-w_{attract} \sum_{j=1}^p (X_j - X_j^i)^2\right) \right] + \sum_{i=1}^S \left[h_{repellent} \exp\left(-w_{repellent} \sum_{j=1}^p (X_j - X_j^i)^2\right) \right],$$

де $X=[x_1, \dots, x_p]^T$ – точка в просторі оптимізації.

Важливими особливостями такого підходу до розрахунку $J_{cc}(X)$ є:

- значення $J_{cc}(X)$ не залежить від значення цільової функції в точці θ ;
- за рахунок варіювання коефіцієнтів $w_{attract}$ та $w_{repellent}$ можна впливати на поведінку популяції бактерій у процесі пошуку: при малих значеннях $w_{attract}$ і більших значеннях $w_{repellent}$ популяція схильна до великого розсіювання в просторі пошуку, у протилежному випадку, тобто при малих значеннях $w_{attract}$ і більших значеннях $w_{repellent}$ популяція схильна до групування в невеликих областях пошуку.

Очевидно, що сила виділення хімічних речовин бактеріями залежить від навколишнього середовища, тобто бактерія, що перебуває в середовищі з високою концентрацією корисних речовин буде виділяти більш сильний атрактант, ніж та ж бактерія в області з низькою концентрацією корисних речовин. Тому в даному методі використовується функція $J_{ar}(\theta)$ для моделювання взаємодії між областями з урахуванням особливостей навколишнього середовища: $J_{ar}(X)=\exp(M-J(X))J_{cc}(X)$, де M – параметр, значення якого обирається експериментально. Отже, для пошуку оптимального необхідно мінімізувати вираз $(J(i, j, k, l) + J_{ar}(X^i(j, k, l)))$, що дозволяє бактеріям шукати корисні речовини, уникаючи областей зі шкідливими речовинами, і в той же час наближаючись до інших областей, але не надто близько до них.

Таким чином, метод оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок зв'язку між областями відрізняється від базового методу тим, що мінімізується не просто функція $J(i, j, k, l)$, а мінімізується сума: $J(i, j, k, l) + J_{ar}(X^i(j, k, l))$, що дозволяє враховувати положення всіх бактерій у популяції.

Функція $J_{ar}(X^i(j, k, l))$ забезпечує таку залежність: чим менше $J(X)$, тим більше $J_{ar}(X)$, і тоді більшою є привабливість, що залежить від $J_{cc}(X)$. Керуючий параметр M звичайно обирається з міркувань, що якщо M досить велике, то $J_{ar}(X)$ значно більше, ніж $J(X)$ і, таким чином, домінуючим напрямком у просторі пошуку є хімічний аттрактант, що виділяється бактерією. З іншого боку, якщо M є дуже малим, то $J_{ar}(X)$ значно менше, ніж $J(X)$ і, таким чином, домінують корисні речовини навколишнього середовища. Для $J_{ar}(X)$ обрана експоненціальна залежність, але вона не є єдино можливим вибором. Будь-які інші функції, які монотонно убувають й асимптотично наближаються до нуля – можливі кандидати, хоча можуть бути необхідні й деякі додаткові обмеження.

Метод оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок зв'язку між областями може бути описаний у вигляді послідовності таких кроків.

Крок 1. Ініціалізація. Задаються параметри, що впливають на роботу методу: $S, N_{re}, N_s, N_c, N_{ed}, P_{ed}$. Випадковим чином розподіляються початкові значення $X^i, i = 1, 2, \dots, S$, у просторі пошуку. Розраховуються початкові значення цільової функції для кожної бактерії J^i . Для кожної бактерії виконується перерахунок: $J^i = J^i + J_{ar}(X^i)$.

Крок 2. Встановити: $l = l + 1$.

Крок 3. Встановити: $k = k + 1$.

Крок 4. Встановити: $j = j + 1$.

Крок 5. Моделювання хемотаксису.

Крок 5.1. Встановити: $i = i + 1$.

Крок 5.2. Обертання й переміщення. Розраховується нове положення i -ої бактерії за формулою:

$$X^i(j+1, k, l) = X^i(j, k, l) + C(i) \frac{\Delta}{\sqrt{\Delta^T \Delta}}.$$

Розрахувати значення цільової функції $J(i, j+1, k, l)$.

Встановити: $J(i, j+1, k, l) = J(i, j+1, k, l) + J_{ar}(X^i(j+1, k, l))$.

Якщо $J(i, j+1, k, l) < J_{best}$, тоді встановити: $J_{best} = J(i, j+1, k, l)$, $X_{best} = X^i(j+1, k, l)$.

Крок 5.3. Ковзання. Поки виконується умова: $J(i, j+1, k, l) < J(i, j, k, l)$, повторюється крок 5.2. Таке повторення може відбуватися N_s разів. Якщо умова не виконується, тоді перехід до кроку 5.4.

Крок 5.4 Якщо $i < S$, тоді перехід до кроку 5.1, у протилежному випадку – перехід до кроку 6.

Крок 6. Якщо $j < N_c$, тоді перехід до кроку 4, у протилежному випадку – перехід до кроку 7.

Крок 7. Відтворення. Всі агенти сортуються відповідно до отриманих значень цільової функції, після чого гірша половина відкидається, а краща – дублюється.

Крок 8. Якщо $k < N_{res}$, тоді виконується перехід до кроку 3, у протилежному випадку – перехід до кроку 9.

Крок 9. Виключення й розсіювання. Перевіряється умова: $U^i < P_{ed}$.

Якщо дана умова виконується, тоді бактерія розміщується в позицію $X^i(j, k, l)$, отриману випадковим образом.

Крок 10. Якщо $l < N_{ed}$, тоді виконується перехід до кроку 2, у протилежному випадку – до кроку 11.

Крок 11. Вибирається й зберігається краще рішення J_{best} і відповідна позиція X_{best} .

Крок 12. Перевірка на закінчення пошуку. Якщо були виконані всі цикли для всіх агентів, тоді виконується перехід до кроку 14, у протилежному випадку виконується перезапуск – перехід до кроку 13.

Крок 13. Перезапуск агентів: вибираються нові випадкові позиції для кожного агента X^i , $i = 1, 2, \dots, S$, розраховуються відповідні значення цільової функції J^i , $i = 1, 2, \dots, S$. Встановлюються лічильники циклів: $j = 0$; $k = 0$; $l = 0$.

Крок 14. Зупинення.

14.4 Метод оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок використання PSO-оператору

PSO-метод базується на групуванні часток, тому логічним є застосування підходів, використовуваних в PSO, до методу оптимізації на основі моделювання переміщення бактерій, що й було запропоновано в [117].

Як відомо, в PSO-методі зв'язок між частками досягається за рахунок того, що швидкість переміщення частки залежить від рішень, отриманих іншими частками, що входять у популяцію. Тому в [117] запропоновано змінювати положення бактерії в просторі пошуку відповідно до PSO-оператору. Така зміна виконується в рамках циклу хемотаксису, після моделювання ковзання бактеріями.

Нехай $V^i = (v_1^i, v_2^i, \dots, v_p^i)$ – вектор швидкостей i -ої бактерії у вимірах $1, 2, \dots, p$, а $X^i = (x_1^i, x_2^i, \dots, x_p^i)$ – позиція i -ої бактерії в просторі пошуку. Тоді відповідно до PSO-методу швидкість і позиція i -ої бактерії на j -му хемотаксичному кроці, k -му кроці відтворення й на l -ій події виключення-розсіювання змінюється в такий спосіб:

$$\begin{aligned} V^i(j+1, k, l) &= \omega \cdot V^i(j, k, l) + C_1 \cdot \varphi_1 \cdot (X_{l_best}^i - X^i(j, k, l)) + \\ &\quad + C_2 \cdot \varphi_2 \cdot (X_{g_best} - X^i(j, k, l)), \\ X^i(j+1, k, l) &= X^i(j, k, l) + V^i(j+1, k, l), \end{aligned}$$

де ω – коефіцієнт інерції, C_1, C_2 – позитивні константи прискорення, які використовуються для варіювання ваг когнітивного й соціального компонентів швидкості бактерії відповідно; $\varphi_1, \varphi_2 = rand(0,1)$ є випадковими значеннями з діапазону $[0,1]$; X_{g_best} – глобально краща позиція, що була обрана із всіх рішень агентів, починаючи з першої ітерації пошуку; $X_{l_best}^i$ – найкраща позиція i -ої бактерії від початку пошуку.

Метод оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок використання PSO-оператора може бути описаний у вигляді послідовності наступних кроків.

Крок 1. Ініціалізація. Задаються параметри, що впливають на роботу методу: $S, N_{re}, N_s, N_{c_2}, N_{ed}, P_{ed}, \omega, C_1, C_2$. Випадковим чином розподілити початкові значення $X^i, i = 1, 2, \dots, S$, у просторі пошуку, задати нульові початкові швидкості бактерій: $v^j=0, i=1, 2, \dots, S, j=1, 2, \dots, p$. Розраховуються початкові значення цільової функції для кожної бактерії J^i . Для кожної бактерії встановлюється $X_{l_best}^i = X^i, J_{l_best}^i = J^i$, а також вибираються J_{g_best} та X_{g_best} .

Крок 2. Встановити: $l = l + 1$.

Крок 3. Встановити: $k = k + 1$.

Крок 4. Встановити: $j = j + 1$.

Крок 5. Моделювання хемотаксису.

Крок 5.1. Встановити: $i = i + 1$.

Крок 5.2. Обертання й переміщення. Розраховується нове положення i -ої бактерії за формулою:

$$X^{i,j+1, k, l} = X^{i,j, k, l} + C(i) \frac{\Delta}{\sqrt{\Delta^T \Delta}}.$$

Розрахувати значення цільової функції $J(i, j+1, k, l)$. Якщо $J(i, j+1, k, l) < J_{l_best}^i$, тоді встановити: $J_{l_best}^i = J(i, j+1, k, l), X_{l_best}^i = X^{i,j+1, k, l}$.

Крок 5.3. Ковзання. Поки виконується умова: $J(i, j+1, k, l) < J(i, j, k, l)$, повторюється крок 5.2. Таке повторення може відбуватися N_s разів. Якщо умова не виконується, тоді перехід до кроку 5.4.

Крок 5.4 Якщо $i < S$, тоді перехід до кроку 5.1, у протилежному випадку – перехід до кроку 6.

Крок 6. Застосування PSO-оператору. Для кожної бактерії розраховується швидкість і нова позиція:

$$\begin{aligned} \forall i = \overline{1, S} : V^i(j+1, k, l) &= \omega \cdot V^i(j, k, l) + C_1 \cdot \varphi_1 \cdot (X_{l_best}^i - X^i(j, k, l)) + \\ &+ C_2 \cdot \varphi_2 \cdot (X_{g_best} - X^i(j, k, l)), \\ X^i(j+1, k, l) &= X^i(j, k, l) + V^i(j+1, k, l). \end{aligned}$$

Крок 7. Якщо $j < N_c$, тоді перехід до кроку 4, у протилежному випадку – перехід до кроку 8.

Крок 8. Відтворення. Всі агенти сортуються відповідно до отриманих значень цільової функції, після чого гірша половина відкидається, а краща – дублюється.

Крок 9. Якщо $k < N_{res}$, тоді виконується перехід до кроку 3, у протилежному випадку – перехід до кроку 10.

Крок 10. Виключення й розсіювання. Перевіряється умова: $U^i < P_{ed}$, де U^i – випадкове число в інтервалі $[0; 1]$ для i -ої бактерії.

Якщо дана умова виконується, тоді бактерія розміщується в позицію $X^i(j, k, l)$, отриману випадковим чином.

Крок 11. Якщо $l < N_{ed}$, тоді виконується перехід до кроку 2, у протилежному випадку – до кроку 12.

Крок 12. Обирається й зберігається краще рішення J_{g_best} і відповідна позиція X_{g_best} .

Крок 13. Перевірка на закінчення пошуку. Якщо були виконані всі цикли для всіх агентів, тоді виконується перехід до кроку 15, у протилежному випадку виконується перезапуск – перехід до кроку 14.

Крок 14. Перезапуск агентів: вибираються нові випадкові позиції для кожного агента X^i , $i = 1, 2, \dots, S$, обнуляється вектор швидкостей V^i , розраховуються відповідні значення цільової функції J^i , $i = 1, 2, \dots, S$. Скидаються лічильники циклів: $j = 0$; $k = 0$; $l = 0$.

Крок 15. Зупинення.

Розглянутий метод оптимізації на основі моделювання переміщення бактерій і його різновиди з групуванням, засновані на різних підходах, застосовувалися для знаходження глобального оптимуму багатовимірних функцій.

Експерименти, проведені в [116, 117], показали, що результати методу без групування і його різновидів з групуванням є ідентичними по величині знайдених оптимальних значень цільової функції, однак швидкість збіжності до оптимального рішення краща в різновидах, які застосовують групування.

14.5 Області застосування й відповідні модифікації методу оптимізації на основі моделювання переміщення бактерій

Оскільки метод оптимізації на основі моделювання переміщення бактерій виявив себе з гарного боку при вирішенні задач оптимізації багатовимірних функцій, то його почали застосовувати до вирішення різних оптимізаційних задач, розроблюючи при цьому відповідні модифікації, що враховують специфіку тієї або іншої задачі.

Метод оптимізації на основі моделювання переміщення бактерій застосовувався до наступних задач.

1. Знаходження оптимального потоку енергії [118].

Для знаходження оптимального потоку енергії запропонована модифікація методу оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок зв'язку між областями, що дозволяє прискорити збіжність.

Основні особливості запропонованої модифікації наступні:

– сортування бактерій робиться на основі мінімальних значень цільової функції, які були отримані на даному кроці хемотаксису. За рахунок такого підходу бактерії із кращими рішеннями переходять у нове покоління, що сприяє більш швидкій збіжності до глобально кращого рішення;

– для групування розраховується відстань не між всіма бактеріями, а розраховується відстань від кожної бактерії до поточного кращого рішення, що було знайдено від початку роботи методу.

Проведені експерименти [118] показали, що запропонована модифікація забезпечує більш високу точність при вирішенні задачі знаходження оптимального потоку енергії в порівнянні із традиційним методом IPSLP (Interior Point Successive Linear Programming).

2. Аналіз незалежних компонентів [119].

Аналіз незалежних компонентів (Independent Component Analysis, ICA) – статистичний метод, мета якого полягає в лінійному поданні даних, розподілених за негаусівським законом, і таких, що компоненти незалежні або незалежні настільки, наскільки це можливо. Існуючі методи ICA зациклюються в локальних оптимумах, тому не знаходять глобальний оптимум. У зв'язку з цим для аналізу незалежних компонентів застосовуються методи, засновані на випадковому підході, наприклад, досить успішним було застосування для даної задачі генетичних алгоритмів [120]. Оскільки метод оптимізації на основі моделювання переміщення бактерій також є методом випадкового пошуку, то в роботі [119] він був застосований до аналізу незалежних компонентів.

В [119] пропонується використовувати ексцес як функцію розходження, яку варто максимізувати й, отже, відповідно до методу BFO виконується мінімізація функції концентрації корисних речовин:

$$J = \frac{1}{\text{ContrastFunction}},$$

де *ContrastFunction* – функція розходження.

Для знаходження мінімуму даної функції використався базовий метод оптимізації на основі моделювання переміщення бактерій.

Результати проведених експериментів [119] показали, що результати, отримані за допомогою BFO мали більш високу точність у порівнянні з розробленими раніше підходами fast ICA та CGAICA (constrained genetic algorithm based ICA).

3. Настроювання пропорційно-інтегрально-диференційного (ПІД) контролера [121].

Незважаючи на нові досягнення в теорії керування, пропорційно-інтегрально-диференційні (ПІД) контролери продовжують широко застосовуватися в багатьох промислових процесах. Більшість підходів до настроювання ПІД, розроблені за минулі роки, використовують традиційні методи, наприклад, метод частотних характеристик [122]. Такі методи вимагають великої кількості технічних експериментів, що є трудомісткою й дорогою задачею. Тому в [121] для настроювання параметрів ПІД контролера пропонується використовувати метод оптимізації на основі моделювання переміщення бактерій.

Як зворотну міру ККД пропонується використовувати інтеграл від зваженого за часом квадрата помилки:

$$PI = \int_0^{\infty} t(E(t))^2 dt ,$$

де $E(t)$ – помилка роботи контролера. $E(t)$ залежить від параметрів роботи контролера (p). Тоді для найкращого настроювання ПІД контролера треба знайти такі значення параметрів роботи контролера p , при яких буде досягатися мінімум PI : $\min_p PI(p)$.

Пошук мінімуму PI здійснювався методом оптимізації на основі моделювання переміщення бактерій з групуванням за рахунок зв'язку між областями.

Отримані результати порівнювалися з результатами, отриманими шляхом застосування інших методів, заснованих на випадковому підході: генетичні алгоритми й метод моделювання імунних систем. Результати всіх методів мали достатню точність і незначно відрізнялися між собою й, таким чином, підтвердили ефективність застосування методу оптимізації за допомогою моделювання переміщення бактерій для настроювання ПІД контролера.

4. Задача динамічного розподілу ресурсів [123].

Базовий метод BFO застосовувався в [123] для вирішення задачі динамічного розподілу ресурсів і тестувався за допомогою експериментальної платформи, що характеризувалася різною температурою в різних областях платформи. Експериментальна платформа складалася із двох частин: плата (що складається з різних конструкторських елементів) і рівня одержання даних, що знімає дані про температуру в різних областях плати.

Таким чином, моделювалося переміщення бактерій по експериментальній платформі, і виконувався пошук областей з найменшою температурою.

Робота методу BFO порівнювалася із традиційними методами, а також з методом мурашиних колоній. Отримані результати [123] показали, що метод BFO знаходить оптимальні рішення швидше в порівнянні з іншими методами.

Область застосування методу ВФО не обмежується наведеними задачами. Описані тільки основні й уже добре відомі напрямки, у яких знайшов застосування даний метод. Однак він використовується й при вирішенні інших оптимізаційних задач [124–126].

14.6 Метод оптимізації на основі моделювання переміщення бактерій для відбору інформативних ознак

Задача відбору інформативних ознак полягає у виборі набору ознак, які найбільш повно характеризують досліджуваний складний об'єкт або процес.

Для вирішення задачі відбору інформативних ознак на основі методу моделювання переміщення бактерій агенти, що моделюють поведінку бактерій, повинні здійснювати оптимізацію в просторі пошуку $X = (x_1, x_2, \dots, x_N)$, $x_i = \{0, 1\}$, де N – загальна кількість ознак, що характеризують вибірку. Таким чином, кожна точка простору пошуку являє собою бітовий рядок, координати якого дозволяють визначити, чи вважається ознака інформативною: інформативною вважається ознака, якщо відповідна координата дорівнює 1, у протилежному випадку – ознака вважається неінформативною. В якості цільової функції $J(X)$ пропонується використовувати помилку моделі, побудованої на основі ознак, які є інформативними для даної точки простору пошуку.

Для виконання переміщення по такому простору пошуку величини, що визначають процес руху в просторі пошуку повинні приймати такі значення:

– змінні, що визначають розмір кроку в певному напрямку, для i -ої бактерії: $C(i) \in (0; 1)$;

– вектор випадкових чисел на j -ому кроці хемотаксису: $\varphi(j) = \{-1, 0, +1\}$.

Приклад переміщення по простору пошуку з трьох ознак, якщо агент на початку пошуку перебував у точці $(0; 0; 0)$, вектор розміру кроку для даної бактерії $C = (1; 1; 1)$, вектор випадкових чисел на 1-ому кроці хемотаксису $\varphi(1) = (1; 0; +1)$, а на 2-ому кроці $\varphi(2) = (-1; +1; 0)$, наведений на рис. 14.2.

Тоді метод оптимізації на основі моделювання переміщення бактерій для відбору інформативних ознак може бути поданий у вигляді послідовності таких кроків.

Крок 1. Ініціалізація. Задаються параметри, що впливають на роботу методу: $S, N_{res}, N_s, N_c, N_{ed}, P_{ed}$. Випадковим чином розподіляються початкові значення $X^i, i = 1, 2, \dots, S$, у просторі пошуку. Розраховуються початкові значення цільової функції для кожної бактерії J^i . Для кожної бактерії виконується перерахунок: $J^i = J^i + J_{ar}(X^i)$.

Крок 2. Встановити: $l = l + 1$.

Крок 3. Встановити: $k = k + 1$.

Крок 4. Встановити: $j = j + 1$.

Крок 5. Моделювання хемотаксису.

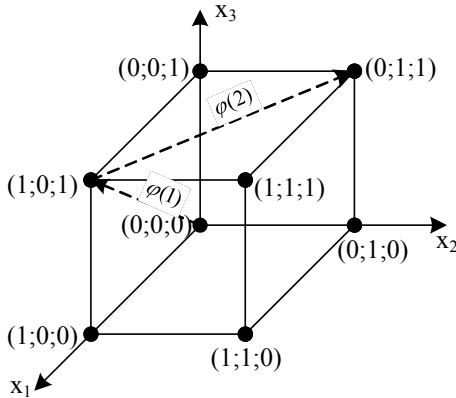


Рисунок 14.2 – Приклад переміщення в просторі пошуку з трьох ознак

Крок 5.1. Встановити: $i = i + 1$.

Крок 5.2. Обертання й переміщення.

Створюється вектор $C(i) = \text{rand}(0;1)$, де $\text{rand}(0;1)$ – випадкове дійсне число в діапазоні $[0; 1]$.

Генерація вектора випадкових чисел $\varphi(j) = \text{rand_int}(-1; +1)$, де $\text{rand_int}(-1; +1)$ – випадкове ціле число в діапазоні $[-1; +1]$.

Розраховується нове положення i -ої бактерії за формулою:

$$X^i(j+1, k, l) = \begin{cases} 1, & \text{якщо } X^i(j, k, l) + C(i) \cdot \varphi(j) \geq 1; \\ \text{round}(X^i(j, k, l) + C(i) \cdot \varphi(j)), & \text{якщо } 0 < X^i(j, k, l) + C(i) \cdot \varphi(j) < 1; \\ 0, & \text{якщо } X^i(j, k, l) + C(i) \cdot \varphi(j) \leq 0. \end{cases}$$

Розрахувати значення цільової функції $J(i, j+1, k, l)$.

Встановити: $J(i, j+1, k, l) = J(i, j+1, k, l) + J_{ar}(X^i(j+1, k, l))$.

Якщо $J(i, j+1, k, l) < J_{best}$, тоді встановити: $J_{best} = J(i, j+1, k, l)$, $X_{best}^i = X^i(j+1, k, l)$.

Крок 5.3. Ковзання. Поки виконується умова: $J(i, j+1, k, l) < J(i, j, k, l)$, повторюється крок 5.2. Таке повторення може відбуватися N_s разів. Якщо умова не виконується, тоді перехід до кроку 5.4.

Крок 5.4 Якщо $i < S$, тоді перехід до кроку 5.1, у протилежному випадку – перехід до кроку 6.

Крок 6. Якщо $j < N_c$, тоді перехід до кроку 4, у протилежному випадку – перехід до кроку 7.

Крок 7. Відтворення. Всі агенти сортуються відповідно до отриманих значень цільової функції, після чого гірша половина відкидається, а краща – дублюється.

Крок 8. Якщо $k < N_{re}$, тоді виконується перехід до кроку 3, у протилежному випадку – перехід до кроку 9.

Крок 9. Виключення й розсіювання. Перевіряється умова: $U^i < P_{ed}$.

Якщо дана умова виконується, тоді бактерія розміщується в позицію $X^i(j, k, l)$, отриману випадковим чином.

Крок 10. Якщо $l < N_{ed}$, тоді виконується перехід до кроку 2, у протилежному випадку – до кроку 11.

Крок 11. Обирається й зберігається краще рішення J_{best} і відповідна позиція X_{best} .

Крок 12. Перевірка на закінчення пошуку. Якщо були виконані всі цикли для всіх агентів, тоді виконується перехід до кроку 14, у протилежному випадку виконується перезапуск – перехід до кроку 13.

Крок 13. Перезапуск агентів: вибираються нові випадкові позиції для кожного агента X^i , $i = 1, 2, \dots, S$, розраховуються відповідні значення цільової функції J^i , $i = 1, 2, \dots, S$. Скидаються лічильники циклів: $j = 0$; $k = 0$; $l = 0$.

Крок 14. Зупинення.

14.7 Переваги та недоліки BFO

На основі розглянутих застосувань методу BFO можна зробити висновок, що метод BFO добре застосовується для задач безперервної оптимізації, у яких варто знайти мінімум певної цільової функції. Проведені експерименти по вирішенню практичних задач [118–126] засвідчили ефективність використання методу BFO як базового, так і з групуванням бактерій.

Можна виділити такі переваги методу BFO у порівнянні із традиційними методами багатовимірної оптимізації:

- метод не висуває абсолютно ніяких вимог до цільової функції, тобто функція може бути мультимодальною, негладкою, мати розриви;

- для знаходження оптимуму цільової функції не треба знати градієнт цільової функції;

- цей метод є методом глобального пошуку, у той час як традиційні методи часто є методами локальної оптимізації;

- метод має велику кількість керуючих параметрів, що сприяє його кращій адаптації до вирішення тієї чи іншої практичної задачі оптимізації;

- метод є мультиагентним, за рахунок чого досягається зниження часу виконання пошуку, оскільки пошук виконується деякою кількістю агентів паралельно.

– метод є добре розширюваним, тобто в його роботу можна легко додавати нові оператори й тим самим створювати гібридні системи. Одним з перспективних є додавання генетичних операторів, які дозволяли б моделювати більш правильні еволюційні процеси, які спостерігаються в життєдіяльності бактерій.

До недоліків методу можна віднести те, що:

– швидкість збіжності носить випадковий характер, хоч знаходження оптимуму гарантується;

– при використанні методу BFO з групуванням бактерій можливе зациклення в локальних оптимумах у випадку неправильного настроювання параметрів, що впливають на групування бактерій;

– робота методу сильно залежить від керуючих параметрів;

– неповнота моделі поведінки бактерій, що може бути дороблена, оскільки процедури відтворення й розсіювання ще не дороблені.

ЛІТЕРАТУРА ДО ЧАСТИНИ ІІІ

1. Beni G., Wang J. Swarm intelligence // Annual Meeting of the Robotics Society: Proceedings of Seventh International Conference. – Tokyo: RSJ Press, 1989. – P. 425–428.
2. Bonabeau E., Dorigo M., Theraulaz G. Swarm intelligence: from natural to artificial systems. – New York: Oxford University Press, 1999. – 320 p.
3. Dorigo M., Maniezzo V., Colomi A. The ant system: optimization by a colony of cooperating agents // IEEE Transactions on Systems, Man, and Cybernetics. – 1996. – Part B, №26 (1). – P. 29–41.
4. Bullnheimer B., Hartl R. F., Strauss C. A new rank-based version of the ant system: A computational study // Central European Journal for Operations Research and Economics. – 1999. – №7 (1). – P. 25–38.
5. Олейник Ал. А. Сравнительный анализ методов оптимизации на основе метода муравьиных колоний // Комп'ютерне моделювання та інтелектуальні системи: Збірник наукових праць / За ред. Д. М. Пізи, С. О. Субботіна. – Запоріжжя: ЗНТУ, 2007. – С. 147–159.
6. Camazine S., Sneyd J. A model of collective nectar source by honey bees: self-organization through simple rules // Journal of Theoretical Biology, 1991. – №149. – P. 547–571.
7. Sumpter D. J. T., Broomhead D. S. Formalising the link between worker and society in honey bee colonies // Lecture Notes In Computer Science: Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation (MABS '98). – Berlin: Springer, 1998. – P.95–110.
8. Kennedy J., Eberhart R. C. Particle swarm optimization // International Conference on Neural Networks: Proceedings of the 1995 IEEE. – NJ: IEEE Press, 1995. – P. 1942–1948.
9. De Castro L. N., Von Zuben F. J. Artificial immune systems. Part I. Basic theory and applications. – Technical Report No. Rt Dca 01/99. – Brazil: Feec/Unicamp, 1999. – 95 p.
10. Gambardella L. M., Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies // Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96). – NJ: IEEE Press, 1996. – P. 622–627.
11. Dorigo M., Gambardella L. M. Ant colonies for the traveling salesman problem // BioSystems. – 1997. – №43. – P. 73–81.
12. Colomi A., Dorigo M., Maniezzo V., Trubian M. Ant system for job-shop scheduling // Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL). – 1994. – №34. – P. 39–53.
13. Субботин С. А., Олейник А. А., Яценко В. К. Отбор информативных признаков на основе модифицированного метода муравьиных колоний // Радиоелектроніка та інформатика. – 2006. – № 1. – С. 65–69.

14. Субботин С. А., Олейник Ал. А. Сравнение модификаций метода муравьиных колоний при отборе информативных признаков // Моделирование неравновесных систем: Материалы IX Всероссийского семинара (13–15 октября 2006) / Под ред. В. В. Слабко. Отв. за выпуск М. Ю. Сенашова. – Красноярск: ИВМ СО РАН, 2006. – С. 175–177.

15. Subbotin S., Oleynik A. Modifications of ant colony optimization method for feature selection // The experience of designing and application of CAD systems in Microelectronics: Proceedings of the IX International Conference CADSM–2007 (20–24 February 2007). – Lviv: Publishing house of Lviv Polytechnic, 2007. – P. 493–494.

16. Субботин С. А., Олейник А. А. Кластер-анализ на основе мультиагентного подхода // Автоматика–2007: Тези доповідей чотирнадцятої міжнародної науково-технічної конференції (10–14 вересня 2007 р.). – Севастополь: СНУЯЕтаП, 2007. – С. 50–53.

17. Maniezzo V., Colomi A., Dorigo M. The ant system applied to the quadratic assignment problem. – Bruxelles: Universite Libre de Bruxelles, 1994. – 24 p.

18. Bullnheimer B., Hartl R. F., Strauss C. An improved ant system algorithm for the vehicle routing problem. – Vienna: Institute of Management Science, University of Vienna, 1997. – 28 p.

19. Michel R., Middendorf M. An island model based ant system with lookahead for the shortest supersequence problem // Proceedings of PPSN–V, Fifth International Conference on Parallel Problem Solving from Nature. – Berlin: Springer, 1998. – P. 692–701.

20. Costa D., Hertz A. Ants can colour graphs // Journal of the Operational Research Society. – 1997. – №48. – P. 295–305.

21. Fan X. P., Luo X., Yi S., Yang S. Y., Zhang H. Optimal path planning for mobile robots based on intensified ant colony optimization algorithm // Proceedings of the International Conference on Robotics, Intelligent Systems and Signal Processing. – Taipei, Taiwan: IEEE Press, 2003. – P. 131–136.

22. Chong S. C., Low M. Y. H. A bee colony optimization algorithm to job shop scheduling // Winter Simulation Conference: Proceedings of the 38th conference on Winter simulation. – Monterey: Monterey Press, 2006. – P. 1954–1961.

23. Lučić P., Teodorović D. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence // Transportation Analysis: Proceedings of the Triennial Symposium TRISTAN IV. – Sao Miguel: Azores Press, 2001. – P. 441–445.

24. Teodorović D., Dell’Orco M. Bee colony optimization – a cooperative learning approach to complex transportation problems // Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini–EURO Conference and 10th Meeting of EWGT (13–16 September 2005). – Poznan: Publishing House of the Polish Operational and System Research, 2005. – P. 51–60.

25. Quijano N., Passino K. M. Honey bee social foraging algorithms for resource allocation: theory and application. – Columbus: Publishing house of the Ohio State University, 2007. – 39 p.
26. Nakrani S., Tovey C. On honey bees and dynamic allocation in an internet server colony // *Adaptive Behavior*. – 2004. – №12. – P. 223–240.
27. Passino K. M., Seeley T. D. Modeling and analysis of nest-site selection by honey bee swarms: The speed and accuracy trade-off // *Behavioral Ecology and Sociobiology*. – 2006. – №59. – P. 427–442.
28. Parunak H. V. D. «Go to the ant»: Engineering principles from natural multi-agent systems // *Annals of Operation Research* – 1997. – №75. – P. 69–101.
29. Ferber J., Müller J. P. Influences and reactions: a model of situated multiagent systems // *Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS-96)*. – Menlo Park: IEEE Computer Society Press, 1996. – P. 72–79.
30. Müller J. P. The design of intelligent agents: a layered approach (lecture notes in computer science). – Berlin: Springer, 1996. – 227 p.
31. Gasser L., Hill R. W. Coordinated problem solvers // *Annual Review of Computer Science*, – 1990. – №4. – P. 203–253.
32. Hackman J. R. Groups that work (and those that don't). – San Francisco, California: Jossey-Bass, 1990. – 544 p.
33. Majchrzak A., Gasser L. Toward a conceptual framework for specifying manufacturing workgroups congruent with technological change // *International Journal of Computer Integrated Manufacturing*. – 1992. – №5. – P. 118–131.
34. Griffin D. R. Animal minds. – Chicago: The University of Chicago Press, 2001. – 376 p.
35. Wilson E. O. The insect societies. – Cambridge, Massachusetts: The Belknap Press, 1974. – 562 p.
36. Agah A., Bekey G. A. Phylogenetic and ontogenetic learning in a colony of interacting robots // *Autonomous Robots*. – 1997. – №4. – P. 85–100.
37. Джонс М. Т. Программирование искусственного интеллекта в приложениях. – М.: ДМК Пресс, 2004. – 312 с.
38. Dornhaus A., Klügl F., Puppe F., Tautz J. Task selection in honeybees – experiments using multi-agent simulation // *Proceedings of GWAL'98*. – Bochum: Verlag Harri Deutsch, 1998. – P. 171–183.
39. Bonabeau E., Theraulaz G., Deneubourg J. L., Aron S., Camazine S. Self-organization in social insects // *Trends in Ecology and Evolution*. – №12(50). – 1997. – P. 188–193.
40. Camazine S., Deneubourg J. L., Franks N. R., Sneyd J., Theraulaz G., Bonabeau E. Self-organization in biological systems. – NJ, Princeton: Princeton University Press, 2001. – 560 p.

41. Buhl J., Sumpter D. J. T., Couzin I. D., Hale J. J., Despland E., Miller E. R., Simpson S. J. From disorder to order in marching locusts // *Science*. – №312 (5778). – 2006. – P. 1402–1406.
42. Janson S., Middendorf M., Beekman M. Honeybee swarms: how do scouts guide a swarm of uninformed bees? // *Animal Behaviour*. – №70(2). – 2005. – P. 349–358.
43. Hölldobler B., Wilson E. O. *The ants*. – Cambridge: Harvard University Press, 1990. – 756 p.
44. Wilson E. O. Chemical communication among workers of the fire ants *Solenopsis saevisima* (Fr. Smith): I. The organization of mass foraging // *Animal Behaviour*. – №10. – 1962. – P. 134–147.
45. Downing H. A., Jeanne R. L. Nest construction by the paperwasp *Polistes*: a test of stigmergy theory // *Animal Behaviour*. – №36. – 1988. – P. 1729–1739.
46. Downing H. A., Jeanne R. L. The regulation of complex building behavior in the paperwasp *Polistes fuscatus* // *Animal Behaviour*. – №39. – 1990. – P. 105–124.
47. Karsai I., Theraulaz G. Nest building in a social wasp: postures and constraints // *Sociobiology*. – №26. – 1995. – P. 83–114.
48. Wenzel J. W. Learning, behaviour programs, and higher-level rules in the nest construction of *Polistes* // *Natural history of paper-wasps*. – №1. – P. 58–74.
49. Bruinsma O. H. An analysis of building behaviour of the termite *Macrotermes subhyalinus*: Ph. D. Thesis. – Lanbouwhogeschool te Wageningen, 1979. – 85 p.
50. Grasse P. P. La reconstruction du nid et les coordinations inter-individuelles chez *bellicositermes natalensis* et *cubitermes* sp. la theorie de la stigmergie : essai d'interpretation du comportement des termites constructeurs // *Insectes Sociaux*. – №6. – 1959. – P. 41–81.
51. Wojtusiak J., Godzinska E. J., Dejean A. Capture and retrieval of very large prey by workers of the African weaver ant *Oecophylla longinoda* (Latreille) // 12th congress of the international union for the study of social insects, Paris, Sorbonne, 21–27 August 1994. – Paris: Universite Paris Nord, 1994. – P. 548.
52. Franks N. R. Teams in social insects: group retrieval of prey by army ants (*Eciton burchelli*, Hymenoptera: Formicidae) // *Behavioral Ecology and Sociobiology*. – №18. – 1986. – P. 425–429.
53. Chauvin R. Sur le transport collectif des proies par *Formica polyctena* // *Insectes Sociaux*. – №15. – 1968. – P. 193–200.
54. Sudd J. H. The transport of prey by ants // *Behaviour*. – №25. – 1965. – P. 234–271.

55. Moffett M. W. Cooperative food transport by an Asiatic ant // *National Geographic Research*. – №4. – 1988. – P. 386–394.
56. Seeley T. D., Camazine S., Sneyd J. Collective decision-making in honey bees: how colonies choose among nectar sources // *Behavioural Ecology and Sociobiology*. – №28. – 1991. – P. 277–290.
57. Gordon D. M. Dynamics of task switching in harvester ants // *Animal Behaviour*. – №38. – 1989. – P. 194–204.
58. Gordon D. M. The organization of work in social insect colonies // *Nature*. – №380 (6570). – 1996. – P. 121–124.
59. Dorigo M. Optimization, learning and natural algorithms. – Milano: Politecnico di Milano, 1992. – 140 p.
60. Dorigo M., Maniezzo V., Colomi A. Positive feedback as a search strategy. – Milano: Politecnico di Milano, 1991. – 22 p.
61. Di Caro G., Dorigo M. Two ant colony algorithms for best routing in datagram networks // *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)* / Eds: Y. Pan, S. G. Akl, K. Li. – Anaheim: IASTED/ACTA Press, 1998. – P. 541–546.
62. Leguizamón G., Michalewicz Z. A new version of ant system for subset problems // *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*. – New Jersey: IEEE Press, 1999. – P. 1459–1464.
63. Dorigo M., Gambardella L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem // *IEEE Transactions on Evolutionary Computation*. – 1997. – №1(1). – P. 53–66.
64. Stützle T., Hoos H. H. The MAX–MIN ant system and local search for the traveling salesman problem // *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)* / Eds: T. Back, Z. Michalewicz, X. Yao. – New Jersey: IEEE Press, 1997. – P. 309–314.
65. Stützle T. Local search algorithms for combinatorial problems: analysis, improvements, and new applications. – Sankt Augustin: Infix, 1999. – 18 p.
66. Stützle T., Hoos H.H. MAX–MIN ant system // *Future Generation Computer Systems*. – 2000. – №16(8). – P. 889–914.
67. Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. – Bologna: Università di Bologna, 1998. – 102 p.
68. Maniezzo V., Colomi A. The ant system applied to the quadratic assignment problem // *Knowledge and Data Engineering*. – 1999. – №5 (11). – P. 769–778.
69. Stützle T., Hoos H. MAX–MIN ant system and local search for combinatorial optimization problems // *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* / Eds: S. Vo., S. Martello, I. H. Osman, C. Roucairol. – Boston: Kluwer, 1998. – P. 137–154.

70. Bullnheimer B., Strauss C. Tourenplanung mit dem ant system // Technical reports of Universit at Wien. – Wien: Instituts für Betriebswirtschaftslehre, 1996. – 26 p.

71. Bullnheimer B., Hartl R. F., Strauss C. Applying the ant system to the vehicle routing problem // Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization / Eds: I. H. Osman S. Vo., S. Martello, C. Roucairol. – Boston: Kluwer, 1998. – P. 109–120.

72. Gambardella L. M., Taillard E., Agazzi G. Macs-vrptw: a multiple ant colony system for vehicle routing problems with time windows // New Methods in Optimisation / Eds: D. Corne, M. Dorigo, F. Glover. – McGraw–Hill, 1999. – P. 63–79.

73. Michel R., Middendorf M. An ACO algorithm for the shortest common supersequence problem // New Methods in Optimisation / Eds: D. Corne, M. Dorigo, F. Glover. – McGraw–Hill, 1999. – P. 525–537.

74. Gambardella L. M., Dorigo M. HAS-SOP: An hybrid ant system for the sequential ordering problem. – Lugano: CH, 1997. – P. 237–255.

75. Branke J., Guntsch M. Solving the probabilistic TSP with ant colony optimization // Journal of Mathematical Modeling and Algorithms. – 2004. – №3. – P. 403–425.

76. Zhao D. B., Yi J. Q. A framework of mobile robot path planning with ant colony optimization // Proceedings of the International Conference on Sensing, Computing and Automation. – ChongQing: Watam Press, 2006. – P. 2009–2013.

77. Олейник Ал. А., Субботин С. А. Мультиагентный метод выделения информативных признаков для синтеза моделей объектов управления // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: Тези доповідей Міжнародної науково-практичної конференції, м. Запоріжжя, 13–15 квітня 2006 р. / Під заг. ред. Д. М. Пізи. – Запоріжжя: ЗНТУ, 2006. – С. 181–183.

78. Субботин С. А., Олейник А. А., Яценко В. К. Отбор информативных признаков на основе модифицированного метода муравьиных колоний // Радиоелектроніка та інформатика. – 2006. – № 1. – С. 65–69.

79. Субботин С. А., Олейник Ал. А. Сравнение модификаций метода муравьиных колоний при отборе информативных признаков // Моделирование неравновесных систем: Материалы IX Всероссийского семинара (13–15 октября 2006) / Под ред. В.В. Слабко. Отв. за выпуск М. Ю. Сенашова. – Красноярск: ИВМ СО РАН, 2006. – С. 175–177.

80. Олейник Ал. А., Олейник Ан. А., Субботин С. А., Яценко В. К. Синтез моделей коэффициента упрочнения деталей авиадвигателей после алмазного выглаживания на основе оптимизационного подхода // Вісник двигунобудування. – 2005. – № 3. – С. 25–30.

81. Олейник Ан. А., Олейник Ал. А. Эволюционный синтез моделей сложных объектов и процессов // 10-й міжнародний молодіжний форум «Радіоелектроніка і молодь в ХХІ сторіччі»: Зб. матеріалів форуму. – Харків.: ХНУРЕ, 2006. – С. 417.
82. Олейник Ал. А. Сравнительный анализ модификаций метода муравьиных колоний для отбора признаков // Радіоелектроніка і молодь в ХХІ сторіччі: Матеріали 11-го міжнародного молодіжного форуму (10–12 квітня 2007 р.). – Харків: ХНУРЕ, 2007. – Ч.1. – С. 353.
83. Subbotin S., Oleynik A. Modifications of ant colony optimization method for feature selection // The experience of designing and application of CAD systems in Microelectronics: Proceedings of the IX International Conference CADSM–2007 (20–24 February 2007). – Lviv: Publishing house of Lviv Polytechnic, 2007. – P. 493–494.
84. Розробка математичного та інформаційного забезпечення інтелектуальної системи візуального контролю транспортних засобів: Звіт про НДР / ТОВ «МПА Групп». – № ДР 0106U012013. – Запоріжжя, 2007. – 259 с.
85. Subbotin S., Oleynik A. Ant colony optimization for feature selection based on operations with crisp sets // Modern problems of radio engineering, telecommunications and computer science: Proceedings of the XI International Conference TCSET'2008 (19–23 February 2008). – Lviv: Publishing house of Lviv Polytechnic, 2008. – P. 117–118.
86. Субботин С. А., Яценко В. К., Олейник Ан. А., Олейник Ал. А. Эволюционный синтез нейромоделей коэффициента упрочнения лопаток авиадвигателей // Нейроинформатика-2006: Труды VIII Всероссийской научно-технической конференции (26–28 января 2005 г.). – М.: МИФИ, 2006. – Ч. 2. – С.89–96.
87. Субботин С. А., Олейник Ал. А. Методы отбора информативных признаков для синтеза нейромоделей на основе метода муравьиных колоний // Нейроинформатика–2007: Сборник научных трудов IX Всероссийской научно-технической конференции (23–26 января 2007 г.). – М.: МИФИ, 2007. – Ч. 2. – С.185–193.
88. Seeley T. D. The wisdom of the hive. – Cambridge: Harvard University Press, 1995. – 265 p.
89. Milner R. Communication and concurrency. – Edinburgh: Prentice Hall, 1989. – 272 p.
90. Lučić P., Teodorović D. Computing with bees: attacking complex transportation engineering problems // International Journal on Artificial Intelligence Tools. – 2003. – №12. – P. 375–394.
91. Zadeh L. Fuzzy sets // Information and Control. – 1965. – №8. – P. 338–353.

92. Zadeh L. From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions // *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*. – 1999. – №45. – P. 105–119.

93. Eberhart R., Kennedy J. A new optimizer using particle swarm theory // *Symposium on Micro Machine and Human Science: Proceedings of the Sixth International Symposium on Micro machine and Human Science*. – Nagoya: IEEE Service Center, 1995. – P. 39–43.

94. Engelbrecht A. *Computational intelligence: an introduction*. – Sidney: John Wiley and Sons Ltd., 2007. – 597 p.

95. Abraham A., Grosan G. *Swarm intelligence in data mining*. – Berlin: Springer, 2006. – 267 p.

96. Peer E., Engelbrecht A. Using neighborhoods with the guaranteed convergence PSO // *Swarm Intelligence Symposium: Proceedings of the IEEE SIS–2003*. – Indianapolis: IEEE Press, 2003. – P. 235–242.

97. Kennedy J. The particle swarm: social adaptation of knowledge // *Proceedings of the IEEE International Conference on Evolutionary Computation*. – New York: IEEE, 1997. – P. 303–308.

98. Engelbrecht A. A study of particle swarm optimization particle trajectories // *Information Sciences*, 2006. – № 176(8). – P. 937–971

99. Clerc M., Kennedy J. The particle swarm-explosion, stability and convergence in a multidimensional complex space // *IEEE Transactions on Evolutionary Computation*, 2002. – №6. – P. 58–73.

100. Coello C., Lechuga M. MOPSO: A proposal for multiple objective particle swarm optimization // *Proceedings of the IEEE Congress on Evolutionary Computation*. – Honolulu: IEEE Press, 2002. – P. 105–110.

101. Parsopoulos K., Tasoulis D., Vrahatis M. Multiobjective optimization using parallel vector evaluated particle swarm optimization // *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*. – Innsbruck: ACTA Press, 2004. – P. 823–828.

102. Braendler D., Henttlass T. The suitability of particle swarm optimization for training neural hardware // *Proceedings of the Fifteenth International Conference on Industrial and Engineering, Applications of Artificial Intelligence And Expert Systems*. – Cairns: Springer, 2002. – P. 190–199.

103. Li A., Chen X. A new stochastic PSO technique for neural network training // *Proceedings of Third International Symposium on Neural Networks*. – Chengdu: Springer, 2006. – P. 1442–1444.

104. Elgallad A., El-Hawary M. Phillips W.; Sallam A. PSO-based neural network for dynamic bandwidth reallocation // *Proceedings of Power Engineering 2002 Large Engineering Systems Conference*. – Halifax: IEEE Press, 2002. – P. 98–102.

105. Kennedy J., Eberhart R. C. Particle swarm optimization // Proceedings of the 4th IEEE International Conference on Neural Networks. –NJ: IEEE Service Center, 1995. – P. 1942–1948.

106. Mendes R., Kennedy J., Neves J. The fully informed particle swarm: simpler, maybe better // IEEE Transactions of Evolutionary Computation, 2004. – №1. – P. 204–210.

107. Zhang C., Shao H., Li Y. Particle swarm optimization for evolving artificial network // Proceedings of IEEE International Conference on System, Man and Cybernetics. – Las Vegas: IEEE Press, 2000. – P. 2487–2490.

108. Madigam M. T., Martinko J. M., Parker J. Biology of microorganisms. –New Jersey: Prentice Hall, 1997. – 1036 p.

109. Neidhardt F. C., Ingraham J. L., Schaechter M. Physiology of the bacterial cell: a molecular approach. – Sunderland: Sinauer Associates, 1990. – 520 p.

110. Alberts B., Bray D., Lewis J., Raff M., Roberts K., Watson J. D. Molecular biology of the cell. – New York: Garland Publishing, 1994. – 1408 p.

111. Berg H. C., Brown D. A. Chemotaxis in escherichia coli analyzed by three-dimensional tracking // Nature, 1972. – №239. – P. 500–504.

112. Berg H. C. Motile behavior of bacteria // Physics Today, 2000. – №53. – P. 24–29.

113. Segall J. E., Block S. M., Berg H. C. Temporal comparisons in bacterial chemotaxis // Proceedings of the National Academy of Sciences, 1986. – №83 (23). – P. 8987–8991.

114. Berg H. C. Random walks in biology. – Princeton: Princeton University Press, 1993. – 164 p.

115. Woodward D. E., Tyson R., Myerscough M. R., Murray J. D., Budrene E. O., Berg H. C. Spatio-temporal patterns generated by salmonella typhimurium // Biophysical Journal, 1995. – №68. – P. 2181–2189.

116. Passino K. M. Biomimicry of bacterial foraging for distributed optimization and control // IEEE Control System Magazine, 2002. – №3 (22). – P. 52–67.

117. Biswas A., Dasgupta S., Das S., Abraham A. Synergy of PSO and bacterial foraging optimization: a comparative study on numerical benchmarks // Proceedings of the Second International Workshop on Hybrid Artificial Intelligent Systems (HAIS 2007). – Salamanca: Springer, 2007. – P. 255–263.

118. Tripathy M., Mishra S., Lai L. L., Zhang Q. P. Transmission loss reduction based on FACTS and bacteria foraging algorithm // Proceedings of 9th International Conference of Parallel Problem Solving from Nature. – Heidelberg: Springer, 2006. – P. 222 – 231.

119. Acharya D. P., Panda G., Mishra S., Lakshmi Y. V. S. Bacteria foraging based independent component analysis // Proceedings of International Con-

ference on Computational Intelligence and Multimedia Applications (ICCIMA 2007). – Sivakasi: IEEE Computer Society, 2007. – P. 527–531.

120. Mishra S., Bhende C. N. Bacterial foraging technique-based optimized active power filter for load compensation // IEEE Transactions on Power Delivery, 2007. – №1 (22). – P. 457–465.

121. Kim D. H. Robust tuning of embedded intelligent PID controller for induction motor using bacterial foraging based optimization // Proceedings of International Conference on Embedded Software and Systems. – Hangzhou: Springer, 2005. – P. 137–142.

122. Xu J. X., Liu C., Hang C. C. Tuning of fuzzy PI controllers based on gain/phase margin specifications and itae index // ISA Transactions, 1996. – №35. – P.79–91.

123. Analysis and design of intelligent systems using soft computing techniques / Ed. Melin P., Castillo O. R., Ramirez E. G., Kacprzyk J. – Heidelberg: Springer, 2007. – 855 p.

124. Mishra S. A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation // IEEE Transactions on Evolutionary Computation, 2005. – №1(9). – P. 61–73.

125. Kim D. H., Cho C. H. Bacterial foraging based neural network fuzzy learning // Proceedings of the 2nd Indian International Conference on Artificial Intelligence (IICAI–2005). – Pune: IICAI, 2005. – P. 2030–2036.

126. Kim D. H., Abraham A., Cho J. H. A hybrid genetic algorithm and bacterial foraging approach for global optimization // Information Sciences, 2007. – №18 (177). – P. 3918–3937.

ВИСНОВКИ

З метою вирішення актуальних завдань розробки методів синтезу нейро-нечітких та нейромережних моделей залежностей запропоновано неітеративні, еволюційні та мультиагентні методи, здатні суттєво підвищити якість та швидкість побудови моделей на основі навчальних вибірок даних.

У **першій частині** монографії розроблено комплекс неітеративних методів синтезу нейро-нечітких та нейромережних моделей.

Для цього у першому розділі пропонується враховувати апріорну інформацію про екземпляри навчальної вибірки, а скорочення надмірності та підвищення якості апроксимації і рівня узагальнення моделей пропонується досягати шляхом редукції блоків мережі, що дублюються.

Другий та третій розділи присвячено побудові ієрархічних моделей.

У розділах першої частини отримано такі результати.

1. Визначено метод отримання апріорної інформації про навчальну вибірку.
2. Запропоновано метод редукції кількості нечітких термів.
3. Розроблено метод об'єднання суміжних термів по ознаках.
4. Створено метод виділення нечітких термів на основі інтервалів значень ознак, що перетинаються для різних класів.
5. Запропоновано метод синтезу тришарових розпізнавальних нейро-нечітких моделей.
6. Розроблено метод синтезу класифікувальних нейро-нечітких мереж з урахуванням значимості термів ознак.
7. Створено метод синтезу чотиришарових розпізнавальних нейро-нечітких моделей з урахуванням інформативності ознак.
8. Запропоновано метод ідентифікації нечітких, нейро-нечітких і нейромережних моделей за прецедентами для вирішення задач розпізнавання образів.
9. Розроблено метод синтезу нейро-нечітких мереж з групуванням ознак.
10. Створено методи синтезу нейромережних і нейро-нечітких розпізнавальних моделей з лінеаризацією, факторним групуванням та згорткою ознак.
11. Запропоновано метод синтезу ієрархічних логічно прозорих нейро-нечітких мереж, який узагальнює класичні методи нечіткого виведення і дозволяє будувати логічно прозорі нейро-нечіткі мережі, що є зручними не тільки для аналізу і сприйняття, але також мають добрі узагальнюючі властивості та легко реалізуються апаратно за рахунок спрощення структури оброблюючих елементів і скорочення кількості зв'язків між нейроелементами.
12. Розроблено метод об'єднання нечітких термів у кластери.
13. Створено метод нейро-нечіткої класифікації за узагальненою вісю, який дозволяє автоматично формувати ієрархічне перетворення координат екземплярів вибірки в багатовимірному просторі ознак на одномірну узагальнену вісь, виділяти нечіткі терми класів за узагальненою озна-

кою і налагоджувати їхні параметри, а також синтезувати нейро-нечіткі моделі якісних залежностей за прецедентами.

14. Запропоновано метод побудови нейро-нечіткої кластер-регресійної апроксимації за узагальненою віссю, що дозволяє автоматично формувати ієрархічне перетворення координат екземплярів вибірки в багатовимірному просторі ознак на одновимірну узагальнену вісь, виділяти нечіткі терми класів за узагальненою ознакою і налагоджувати їхні параметри, а також синтезувати часткові регресійні моделі кількісних залежностей для кластерів.

У **другій частині** книги на основі еволюційного підходу вирішено актуальне завдання розробки методів синтезу нейромережних моделей. В результаті проведених досліджень отримано таке.

1. Виконано аналітичний огляд етапів побудови нейромоделей, який показав, що всі етапи синтезу нейронних мереж передбачають оптимізацію деякого задалегідь заданого критерію оптимальності. Показано, що для синтезу нейромережних моделей ефективно можуть використовуватися методи еволюційного пошуку, які дозволяють у рамках єдиного підходу реалізовувати всі етапи побудови нейромоделей і не вимагають обчислення похідних цільової функції, працюють із деякою множиною рішень, що дозволяє в багатьох випадках аналізувати простір пошуку швидше в порівнянні з традиційними оптимізаційними методами, не висуваючи при цьому додаткових обмежень до виду цільової функції.

2. Дістав подальшого розвитку канонічний метод еволюційного пошуку, на основі якого розроблено модифікації для відбору ознак за навчальною вибіркою, які використовують апріорну інформацію про значущість ознак в еволюційних операторах, що дозволяє збільшити ймовірність одержання оптимальної системи інформативних ознак і прискорити еволюційний пошук.

3. Створено новий метод еволюційного пошуку для параметричного синтезу нейромереж, що дозволяє отримувати значення вагових коефіцієнтів і зсувів, не накладаючи обмеження на вид функцій активації нейронів, і, на відміну від класичних методів еволюційного пошуку, застосовує спеціальні оператори ініціалізації й мутації, що враховують апріорну інформацію про значущість ознак, що підвищує точність і скорочує час пошуку.

4. Вперше запропоновано метод полімодального еволюційного пошуку з кластеризацією хромосом для структурного синтезу нейромоделей, що рівномірно покриває простір пошуку, підвищуючи можливість відшукування глобального оптимуму й збільшуючи ймовірність успішного виконання процедури оцінки знайдених рішень за допомогою зовнішніх критеріїв на тестовій вибірці.

5. Розроблено новий метод еволюційної оптимізації для структурно-параметричного синтезу нейромоделей, що дозволяє вибирати оптимальну

структуру нейромоделей, налагоджуючи при цьому значення вагових коефіцієнтів і зсувів і не накладаючи обмежень на вид функцій активації нейронів.

6. Запропоновано новий метод багатокритеріального еволюційного пошуку з використанням узагальненої цільової функції, що дозволяє оптимізувати побудовану нейромоделю по декількох заданих цільових критеріях.

7. Вперше створено систему критеріїв та розроблено методичку, що дозволяють досліджувати й оцінювати ефективність запропонованих еволюційних методів побудови нейромоделей.

8. Розроблено автоматизовану систему еволюційного синтезу й оптимізації нейромережних моделей, що реалізує запропоновані методи та дозволяє виконувати відбір інформативних ознак, параметричний, структурний і структурно-параметричний синтез нейромоделей, а також їхнє спрощення.

9. Проведено експерименти по дослідженню властивостей і характеристик розроблених методів і програмних засобів шляхом синтезу нейромережних моделей для вирішення практичних задач технічної й біомедичної діагностики. Результати експериментів показали, що запропоновані методи за рахунок використання додаткової інформації про досліджуваний об'єкт дозволяють скоротити час, необхідний на пошук, а нейромоделі, побудовані за допомогою запропонованих методів, містять найбільш значущі міжнейронні зв'язки та зв'язки між вхідними ознаками та нейронами. Тому синтезовані моделі є більш простими, оскільки мають значно меншу кількість зв'язків при аналогічній кількості, нейронів і шарів.

Аналіз результатів теоретичних і експериментальних досліджень показав, що розроблені еволюційні методи й програмне забезпечення для відбору ознак і синтезу нейромережних моделей доцільно застосовувати при вирішенні практичних задач, що характеризуються великою розмірністю навчальної вибірки, де побудова моделей, що включають всі ознаки й містять всі можливі зв'язки між нейроелементами, є непринятною, оскільки такі моделі є неінтерпретабельними й вимагають значних витрат часових і апаратних ресурсів при їхньому використанні.

У **третьій частині** монографії розглянуто та запропоновано інтелектуальні методи мультиагентної оптимізації, а також їхні можливі області застосування. У результаті проведених досліджень отримані такі результати.

1. Надано огляд основних принципів і біологічних основ методів і моделей колективного інтелекту.

2. Проведено аналітичний огляд методу мурашиних колоній, його різновидів та їхніх особливостей. Надано порівняльну характеристику варіацій методу мурашиних колоній.

3. Вперше запропоновано методи відбору інформативних ознак, що засновані на мультиагентному підході з опосередкованим зв'язком між агентами, які моделюють поведінку мурах: на основі подання пунктів

призначення ознаками, на основі подання пунктів призначення у вигляді інформативності, з використанням операцій над чіткими множинами та з використанням операцій над нечіткими множинами. Перший метод передбачає пошук у просторі ознак, що дозволяє швидше виконувати пошук. Другий метод передбачає виконання відбору ознак у бітовому просторі, що дозволяє покращити ефективність оптимізаційного процесу. Останні два методи зорієнтовані на роботу з чіткими та нечіткими вибірками даних, відповідно, через що характеризуються значною ефективністю при використанні до таких вибірок даних.

4. Вперше розроблено метод кластеризації на основі мультиагентного підходу, що заснований на моделюванні колективного інтелекту агентів з опосередкованим зв'язком між ними, який передбачає виконання агентами дій підйому, переміщення та залишення елементів даних по простору пошуку, а також використання функцій подібності та відмінності сусідніх елементів, що дозволяє підвищити ефективність дослідження простору пошуку.

5. Розглянуто метод бджолоїної колонії й проведено порівняльний аналіз різних його модифікацій, використовуваних при вирішенні різноманітних задач оптимізації.

6. Виявлено особливості, переваги й недоліки методу бджолоїної колонії.

7. Дістав подальшого розвитку мультиагентний метод оптимізації з прямим зв'язком між агентами, що моделює поведінку бджолоїної колонії, на основі якого розроблено методи відбору інформативних ознак та знаходження оптимуму багатовимірної функції, що передбачають відсікання з часом гірших областей простору пошуку та використання еволюційних операторів, що дозволяє зменшити час оптимізації, а також підвищити точність.

8. Розглянуто оптимізацію з використанням рою часток (PSO-метод) і основні підходи (gbest та lbest), застосовувані в PSO.

9. Розроблено PSO-метод з керуванням зміною швидкості частки й наведено порівняльну характеристику його з іншими PSO-методами.

10. Наведено огляд методу оптимізації на основі моделювання переміщення бактерій і його різновидів, виявлено основні напрями застосування, недоліки й переваги методу.

11. Запропоновано модифікацію методу оптимізації на основі моделювання переміщення бактерій для вирішення задачі відбору інформативних ознак.

Таким чином, розглянуто інтелектуальні методи мультиагентної оптимізації, їхні основи та принципи роботи, основні різновиди цих методів та області застосування. На основі проведених досліджень можна відзначити, що мультиагентні методи інтелектуальної оптимізації можуть застосовуватися до вирішення задач як безперервної, так і дискретної оптимізації.

Наукове видання

**СУББОТІН Сергій Олександрович
ОЛІЙНИК Андрій Олександрович
ОЛІЙНИК Олексій Олександрович**

НЕІТЕРАТИВНІ, ЕВОЛЮЦІЙНІ ТА
МУЛЬТИАГЕНТНІ МЕТОДИ СИНТЕЗУ
НЕЧІТКОЛОГІЧНИХ І НЕЙРОМЕРЖНИХ МОДЕЛЕЙ

Монографія

Під загальною редакцією С. О. Субботіна

Оригінал-макет підготовлено на кафедрі програмних засобів
та у редакційно-видавничому відділі ЗНТУ

Верстання

Рибалка Ірина Сергіївна

Підписано до друку 04.12.2008. Формат 60x84/16. Бум. офс.
Різогр. друк. Ум. друк. арк. 21,8. Обл.-вид. арк. 18,6.
Тираж 300 прим. Зам. № 1811.

Запорізький національний технічний університет
Україна, 69063, м. Запоріжжя, вул. Жуковського, 64
Тел.: 8 (061) 769–82–96, 220–12–14

Свідоцтво про внесення суб'єкта видавничої справи
до державного реєстру видавців, виготівників
і розповсюджувачів видавничої продукції
від 27.12.2005 р., серія ДК № 2394