

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Запорізький національний технічний університет**

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

# **ЕВОЛЮЦІЙНІ ОБЧИСЛЕННЯ ТА ПРОГРАМУВАННЯ**

*Навчальний посібник*

*Рекомендовано Міністерством освіти і науки України  
як навчальний посібник для студентів вищих навчальних закладів,  
які навчаються за спеціальністю  
«Програмне забезпечення автоматизованих систем»*

**Запоріжжя • ЗНТУ • 2010**

УДК 004.032.26:004.89  
ББК 32.973  
О53

*Рекомендовано до друку підкомісією з програмної інженерії  
Науково-методичної ради з галузі знань «Інформатика та обчислювальна  
техніка» Міністерства освіти і науки України  
(протокол № 5 від 29.04.2010 р.)*

*Рекомендовано до друку Вченою радою  
Запорізького національного технічного університету  
(протокол № 8 від 11.03.2010 р.)*

*Гриф надано Міністерством освіти і науки України  
(лист № 1/11-5846 від 01.07.2010 р.)*

Рецензенти: доктор технічних наук, професор, декан математичного факультету Запорізького національного університету *Гоменюк С. І.*;  
доктор технічних наук, професор, професор кафедри обчислювальної техніки та програмування Національного технічного університету «Харківський політехнічний інститут» *Дмитрієнко В. Д.*;  
доктор технічних наук, професор, завідувач кафедри електронних обчислювальних машин Харківського національного університету радіоелектроніки *Руденко О. Г.*

**Олійник А. О.**

О53 Еволюційні обчислення та програмування : навчальний посібник / А. О. Олійник, С. О. Субботін, О. О. Олійник. – Запоріжжя : ЗНТУ, 2010. – 324 с.

ISBN 978–611–529–023–9.

Книга містить систематизований виклад основних понять, моделей і методів еволюційного пошуку, які ефективно можуть використовуватися при вирішенні завдань дискретної та безперервної оптимізації. Розглянуто еволюційні методи, генетичне та еволюційне програмування, багатокритеріальну еволюційну оптимізацію, еволюційні методи синтезу нейромережевих моделей. Поряд із відомими методами та моделями запропоновано оригінальні авторські розробки, що дозволяють виконувати відбір інформативних ознак та побудову штучних нейронних мереж. Наведено опис програмних засобів, що реалізують розглянуті методи та моделі.

Видання призначено для студентів комп'ютерних спеціальностей вищих навчальних закладів, а також може використовуватися аспірантами, науковими та педагогічними працівниками, практичними фахівцями.

**УДК 004.032.26:004.89**  
**ББК 32.973**

ISBN 978–611–529–023–9

© ЗНТУ, 2010  
© Олійник А. О., Субботін С. О.,  
Олійник О. О., 2010

# ЗМІСТ

<b>Вступ</b> .....	<b>7</b>
<b>1 Еволюційні методи</b> .....	<b>10</b>
1.1 Еволюційний пошук як метод оптимізації.....	10
1.2 Аналогія генетичних методів з поняттями генетики .....	14
1.3 Узагальнена схема роботи генетичних методів.....	15
1.4 Класифікація методів еволюційного пошуку .....	17
1.5 Моделі генетичного пошуку .....	21
1.5.1 Канонічні моделі .....	21
1.5.2 Породжувальна модель.....	22
1.5.3 Гібридний генетичний метод .....	23
1.5.4 Генетичний метод зі змінним часом життя особин.....	24
1.5.5 Мобільний генетичний метод .....	25
1.5.6 Паралельні та багаторівневі генетичні методи .....	28
1.5.7 Генетичний пошук зі зменшенням розміру популяції .....	32
1.6 Ініціалізація та запуск генетичного пошуку .....	34
1.6.1 Кодування параметрів, що оптимізуються.....	34
1.6.2 Визначення цільової функції.....	37
1.6.3 Ініціалізація.....	39
1.7 Генетичні оператори .....	40
1.7.1 Відбір.....	42
1.7.2 Схрещування.....	45
1.7.3 Мутація.....	62
1.7.4 Формування нового покоління.....	70
1.7.5 Критерії зупинення.....	71
1.8 Теоретичний аналіз генетичного пошуку .....	72
1.8.1 Основні поняття.....	72
1.8.2 Аналіз впливу генетичних операторів.....	73
1.8.3 Теорема схем.....	75
1.8.4 Будівельні блоки.....	77
1.9 Керування параметрами еволюційного пошуку .....	77
1.9.1 Неадаптивні методи керування параметрами .....	78
1.9.2 Адаптивні методи.....	80
1.10 Програмні засоби генетичної оптимізації.....	83
1.11 Контрольні питання .....	86

<b>2 Генетичне та еволюційне програмування .....</b>	<b>90</b>
2.1 Еволюційні стратегії .....	90
2.1.1 ( $\mu + \lambda$ ) та ( $\mu, \lambda$ ) еволюційні стратегії .....	90
2.1.2 Оператор мутації .....	91
2.1.3 Метод імітації відпалу .....	93
2.2 Генетичне програмування .....	96
2.2.1 Основні поняття .....	96
2.2.2 Етапи генетичного програмування .....	96
2.2.3 Оператори генетичного програмування .....	97
2.3 Еволюційне програмування .....	98
2.3.1 Основні поняття .....	99
2.3.2 Етапи еволюційного програмування .....	99
2.3.3 Оператор мутації .....	100
2.3.4 Метод групового врахування аргументів .....	100
2.4 Полімодальна еволюційна оптимізація .....	103
2.4.1 Методи ухилення від передчасної збіжності .....	103
2.4.2 Методи відновлення .....	111
2.5 Контрольні питання .....	115
<b>3 Багатокритеріальний еволюційний пошук з обмеженнями .....</b>	<b>117</b>
3.1 Постановка задачі та класифікація методів багатокритеріальної оптимізації .....	117
3.2 Априорні методи багатокритеріальної оптимізації .....	118
3.2.1 Метод агрегувальних функцій .....	118
3.2.2 Лексикографічне впорядкування .....	119
3.2.3 Мінімаксний метод .....	119
3.2.4 Метод досягнення заданого значення .....	120
3.3 Популяційний підхід .....	120
3.4 Багатокритеріальний еволюційний пошук, заснований на підході Парето .....	121
3.4.1 Основні поняття .....	121
3.4.2 Оцінювання хромосом .....	123
3.4.3 Методи ущільнення множини Парето .....	124
3.4.4 Методи багатокритеріального еволюційного пошуку .....	126
3.5 Критерії оптимальності .....	133
3.6 Підтримка обмежень .....	135
3.6.1 Використання штрафних функцій .....	136
3.6.2 Перетворення простору пошуку .....	137
3.6.3 Використання спеціальних еволюційних операторів .....	138
3.6.4 Відновлення неприпустимих рішень .....	138
3.6.5 Застосування багатокритеріального підходу .....	138
3.7 Контрольні питання .....	140

<b>4 Еволюційний синтез нейромережових моделей.....</b>	<b>142</b>
4.1 Загальна постановка задачі синтезу нейромережових моделей ...	142
4.1.1 Вибір системи інформативних ознак.....	145
4.1.2 Структурний синтез .....	147
4.1.3 Параметричний синтез.....	149
4.1.4 Оптимізація побудованої нейромоделі.....	151
4.2 Еволюційний відбір інформативних ознак .....	152
4.2.1 Кодування хромосом для виділення найбільш значимого набору ознак .....	152
4.2.2 Фітнес-функція .....	153
4.2.3 Еволюційний метод пошуку інформативного набору ознак	155
4.2.4 Керування параметрами еволюційного пошуку .....	156
4.2.5 Еволюційний метод з фіксацією частини простору пошуку .....	159
4.2.6 Методи еволюційного пошуку з використанням апріорної інформації про значущість ознак .....	163
4.3 Параметричний синтез нейромереж .....	169
4.4 Структурний синтез нейронних мереж .....	176
4.4.1 Подання інформації про структуру нейромережі в хромосомі .	176
4.4.2 Визначення фітнес-функції .....	180
4.4.3 Послідовність виконання структурного синтезу на основі методів еволюційної оптимізації .....	180
4.4.4 Еволюційні оператори для структурного синтезу нейромоделей..	181
4.4.5 Метод полімодального еволюційного пошуку з кластеризацією хромосом .....	182
4.5 Структурно-параметричний синтез .....	185
4.6 Адаптація навчаючих правил .....	189
4.7 Оптимізація структури нейромоделей.....	190
4.8 Критерії й методика порівняння еволюційних методів синтезу нейронних мереж.....	194
4.9 Критерії якості навчальних вибірок і нейромоделей .....	197
4.9.1 Характеристики і критерії порівняння навчальних вибірок .....	199
4.9.2 Характеристики і критерії порівняння нейромоделей .....	213
4.9.3 Властивості нейромоделей, обумовлені навчальною вибіркою	214
4.9.4 Властивості структури і параметрів нейромоделей, що не залежать явно від навчальної вибірки .....	224
4.9.5 Інтегральні показники властивостей і критерії порівняння нейромоделей.....	232
4.10 Автоматизована система еволюційного синтезу й оптимізації нейромережових моделей .....	235
4.11 Контрольні питання .....	241
<b>5 Приклади та ілюстрації.....</b>	<b>243</b>

<b>6 Практичні завдання.....</b>	<b>267</b>
6.1 Моделі та методи еволюційного пошуку .....	267
6.2 Дослідження еволюційних операторів .....	284
6.3 Статистичний аналіз результатів еволюційної оптимізації.....	287
6.4 Комбінаторна оптимізація за допомогою еволюційних методів .	289
6.5 Генетичне програмування .....	290
6.6 Багатокритеріальна еволюційна оптимізація.....	290
6.7 Еволюційний синтез нейромережових моделей .....	290
<b>7 Тестові завдання.....</b>	<b>292</b>
7.1 Завдання першого рівня.....	292
7.2 Завдання другого рівня .....	296
7.3 Завдання третього рівня.....	302
7.4 Відповіді до тестових завдань.....	306
7.5 Приклад білету до контролю знань студентів .....	307
<b>Література.....</b>	<b>312</b>
Основна література .....	312
Додаткова література .....	313
<b>Алфавітно-предметний покажчик .....</b>	<b>320</b>

## ВСТУП

При розв'язанні задач математичного моделювання, планування й прогнозування в економіці, техніці та медицині використовуються математичні методи оптимізації. Відомі методи багатовимірної оптимізації, як правило, є методами локального пошуку, сильно залежать від вибору початкової точки пошуку й висувають ряд вимог до виду цільової функції, що ускладнює їхнє застосування на практиці.

Досить ефективними методами розв'язання оптимізаційних задач є еволюційні методи, які є методами глобальної оптимізації, засновані на імовірнісному підході, не вимагають обчислення похідних цільової функції й не залежать від вибору початкової точки пошуку.

*Актуальність* створення, вивчення та використання методів, моделей та комп'ютерних систем, що в своїй роботі застосовують методи штучного інтелекту, зокрема й методи еволюційного пошуку, підтверджується Державною програмою розвитку промисловості 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 прототипи 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 нейромережевою архітектурою, проблемно-орієнтовані робочі станції, нейрокомп'ютери та нейромережеві технології, програмне забезпечення інформаційно-аналітичної обробки текстових, фактографічних та статистичних даних, конкурентоспромо-

жні інформаційні технології формування аналітичних електронних оглядів і довідок, засоби інтелектуалізації широкого застосування».

*Метою* даної книги є систематизований виклад основних понять, моделей і методів еволюційного пошуку, які ефективно можуть використовуватися при вирішенні завдань дискретної та безперервної оптимізації, зокрема завдань відбору інформативних ознак і побудови нейромережових моделей для діагностики, розпізнавання, оцінювання й керування.

У першому розділі розглянуто генетичні методи як клас найбільш часто використовуваних методів еволюційного пошуку, запропоновано систему критеріїв для класифікації методів еволюційної оптимізації, проаналізовано різні моделі генетичного пошуку, розглянуто еволюційні оператори відбору, схрещування, мутації і їхній вплив на ефективність пошуку, наведено теорему схем, що дозволяє аналізувати й прогнозувати динаміку поводження генетичного пошуку. Проаналізовано методи керування параметрами еволюційного пошуку, що знижують залежність еволюційної оптимізації від початкових параметрів пошуку. Наведено класифікацію таких методів, проаналізовані неадаптивні й адаптивні методи керування параметрами еволюційної оптимізації.

Другий розділ містить опис методів еволюційної оптимізації, які в порівнянні з генетичними методами одержали менше поширення, але є досить перспективними для подальшого вивчення. Серед таких методів виділено: еволюційні стратегії, генетичне програмування й еволюційне програмування. У цьому ж розділі розглянуто, класифіковано й проаналізовано методи полімодальної еволюційної оптимізації, що знижують імовірність небажаної передчасної збіжності до локального оптимуму й дозволяють знайти декілька субоптимальних рішень, розташованих у різних точках простору пошуку.

У третьому розділі наведено опис методів багатокритеріального еволюційного пошуку, що дозволяє знайти оптимальне рішення для заданого набору цільових функцій. Проаналізовано апіорні, прогресивні й апостеріорні методи багатокритеріальної еволюційної оптимізації, описано методи, що дозволяють враховувати обмеження при вирішенні реальних завдань.

У четвертому розділі розглянуто застосування методів еволюційного пошуку для побудови нейромережових моделей, запропоновано нові еволюційні методи для відбору інформативних ознак, параметричного, структурного і структурно-параметричного синтезу нейромоделей, а також їхнього спрощення.

Поряд із описом відомих методів та моделей розглянуто оригінальні авторські розробки, що дозволяють виконувати побудову розпізнаю-

чих нейромережових моделей на основі еволюційного підходу. У книзі також наведено опис програмних засобів, що реалізують розглянуті методи та моделі.

Для спрощення самостійного опрацювання та кращого засвоєння матеріалу книги наприкінці кожного розділу наведено контрольні питання. Комплексні приклади, практичні та тестові завдання наведено після теоретичної частини посібника у вигляді окремих розділів.

Видання орієнтоване на студентів комп'ютерних спеціальностей вищих навчальних закладів, а також може використовуватися аспірантами, науковими та педагогічними працівниками, практичними фахівцями.

Матеріал, наведений у книзі, призначений для вивчення та апробований авторами при читанні курсів «Нейроінформатика та еволюційні алгоритми», «Сучасний штучний інтелект», «Еволюційне програмування» у Запорізькому національному технічному університеті. Книга також може використовуватися при вивченні окремих розділів дисциплін «Системи штучного інтелекту», «Дискретні структури», «Математичні основи представлення знань» та інших.

Більш детальна інформація про використання матеріалу книги у навчальному процесі, а також електронні варіанти праць авторів та інші корисні посилання на літературні джерела та рекомендовані програмні засоби доступні на веб-сайті авторів за адресою: <http://csit.org.ua>.

Терміни та назви методів, визначення яких наводиться у наступному тексті виділено *курсивом*. Тексти програм виділено моноширинним шрифтом.

Для спрощення пошуку навчальних завдань, прикладів і літературних джерел використовується така *система умовних позначень*:



– контрольні питання,



– приклади, що розв'язуються вручну,



– приклади та завдання, що мають бути виконані з використанням комп'ютера,



– індивідуальні завдання,



– завдання, що виконуються вручну.

# 1 ЕВОЛЮЦІЙНІ 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 [37].

Тому для вирішення оптимізаційних задач, що виникають при вирішенні практичних задач прогнозування, класифікації, діагностування, розпізнавання, доцільним є використання еволюційних методів [1–12, 14], які є методами стохастичної оптимізації й поєднують комп'ютерні методи моделювання еволюційних процесів у природних і штучних системах.

## 1.1 Еволюційний пошук як метод оптимізації

Оптимізаційні методи [37] 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) 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) 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) 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) 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.

*Градiєнтні багатовимірні методи* (метод Коші, метод Ньютона, метод Левенберга-Марквардта [62]) для визначення напряму пошуку використовують значення похідних цільової функції різних порядків.

*Еволюційний (генетичний) пошук* включає групу багатовимірних, стохастичних, евристичних оптимізаційних методів [1–86], 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, пристосовність 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:

$$GM = GM(P_0, N, L, f, \Omega, \Psi, \Theta, T),$$

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 (розмір популяції);  $L$  – довжина хромосом (кількість генів);  $f$  – цільова функція (фітнес-функція, функція пристосованості, функція здоров'я, функція придатності);  $\Omega$  – оператор відбо-

ру;  $\Psi$  – оператор схрещування;  $\mathcal{G}$  – оператор мутації;  $T$  – критерії 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 рівносьильне рідкісному, але безперервному пошуку оптимуму у решті областей пошукового простору.

Таким чином, на кожній ітерації генетичного пошуку метод працює не з одним 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.

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*:

– 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, імовірнісні 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:

– якщо простір пошуку є дискретним і невеликим за розміром, то можна скористатися методом повного перебору, 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) 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, 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, тому вони мають менше шансів зійтися до локального оптимуму і надійно функціонують на багатоекстремальних поверхнях.

## 1.2 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 послідов-

ності поколінь (генетичні оператори), ідею про переважне розмноження найбільш пристосованих особин.

### 1.3 Узагальнена схема роботи генетичних методів

Сутність генетичного пошуку полягає в циклічній заміні однієї популяції наступною, більш пристосованою. 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. 1.1.

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 задоволено, 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.

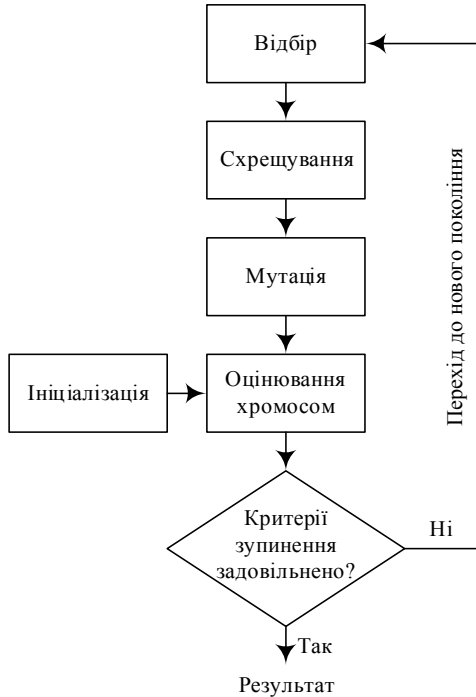


Рисунок 1.1 – Схема роботи узагальненого генетичного метода

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.

#### 1.4 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 до-

статньо широкий клас методів, часом мало схожих один на одного [2–12].

Методи еволюційного пошуку можуть бути класифіковані за різними критеріям.

Укрупнену класифікаційну схему методів еволюційного пошуку наведено на рис. 1.2. У кожному з наведених на рис. 1.2 генетичних методів є можливим застосування різних генетичних операторів, тому варіанти генетичного пошуку, що розрізняються лише генетичними операторами, в схему не включено.

Нижче наведено запропоновану класифікацію еволюційних методів.

1. *За класом методів* еволюційний пошук може бути розподілений на чотири групи: генетичні алгоритми, еволюційні стратегії, генетичне програмування та еволюційне програмування.

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.

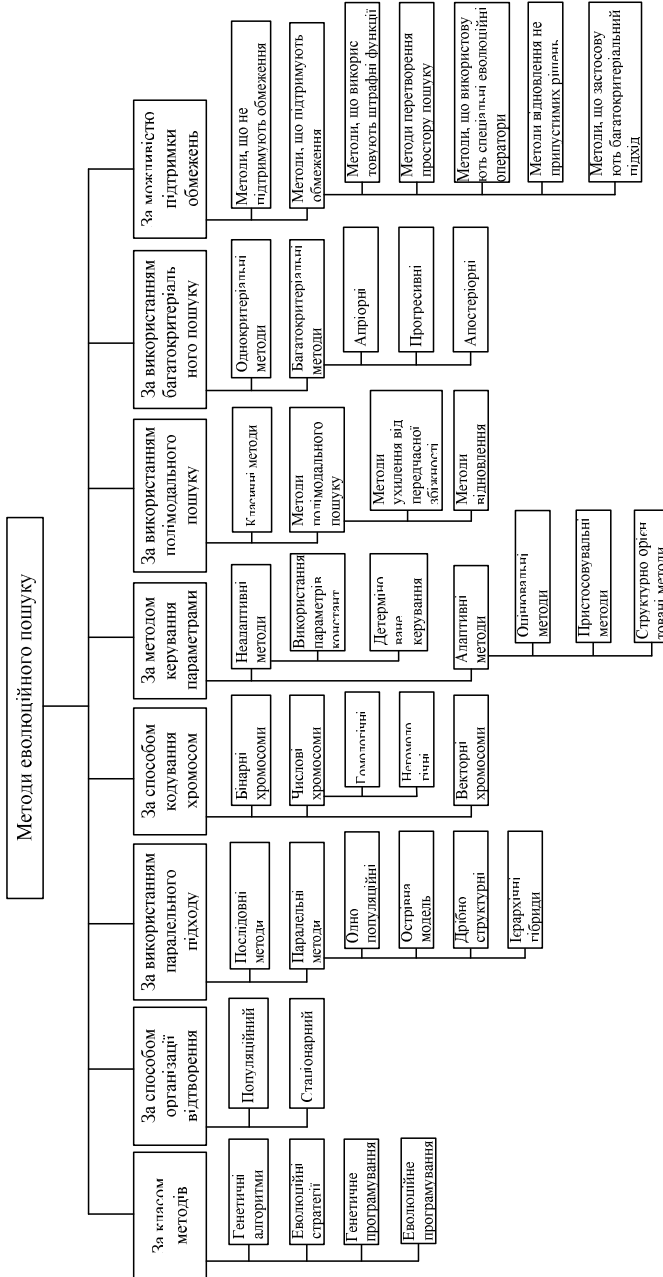


Рисунок 1.2 – Класифікація методів еволюційного пошуку

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:

- апріорні методи (метод агрегувальних функцій, лексикографічне впорядкування, мінімаксний 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.

## 1.5 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 популяцій, застосуванням специфічних генетичних операторів і т. ін. [2, 3, 4, 10–12, 15, 30]. Вибір моделі генетичного методу залежить від типу розв'язуваної задачі.

### 1.5.1 Канонічні моделі

Першою канонічною моделлю генетичного методу (canonical GA) є *репродуктивний план Холланда*. 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;
- особини для схрещування вибираються випадковим чином;
- односточкове схрещування і односточкова мутація;
- наступне покоління формується з нащадків поточного покоління без «елітизму». Насадки займають місця своїх батьків.

### 1.5.2 Породжувальна модель

У породжувальній моделі Genitor (Д. Уїтлі) використовується специфічна стратегія відбору.

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 породжувальної моделі:

- 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.

### 1.5.3 Гібридний генетичний метод

Використання гібридного генетичного методу дозволяє об'єднати переваги генетичного пошуку з перевагами класичних методів оптимізації. 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 пошукових 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.

### 1.5.4 Генетичний метод зі змінним часом життя особин

Генетичний метод зі змінним часом життя особин 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;  $f_{\text{cp}}$  – 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. 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. Сформувати 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.

### 1.5.5 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.

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 процеси-нгу.

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 «схдинках» 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 опера-

тори (розрізання, зчеплення, мутація) не використовуються. Необхідність оцінки кожного елемента 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.

### 1.5.6 Паралельні та багаторівневі генетичні методи

В генетичних методах використовуються два підходи до структуризації популяції.

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 «близкості» 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 багато-процесорних 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;  $\rho$  – параметр, що відображає вплив методу відбору або паралелізму;  $T_c$  – середній час для передачі даних між процесорами.

### Однопопуляційні генетичні методи

В однопопуляційних генетичних методах (single-population master-slave GA, master-slave GA), використовують багато-процесорні системи (рис. 1.3 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*».

### 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. 1.3, б). Такий обмін особинами називається міграцією. Окрім частоти міграції, параметрами такого виду генетичного пошуку є також кількість мігруючих особин, напрям міграції, й метод відбору особин для міграції.

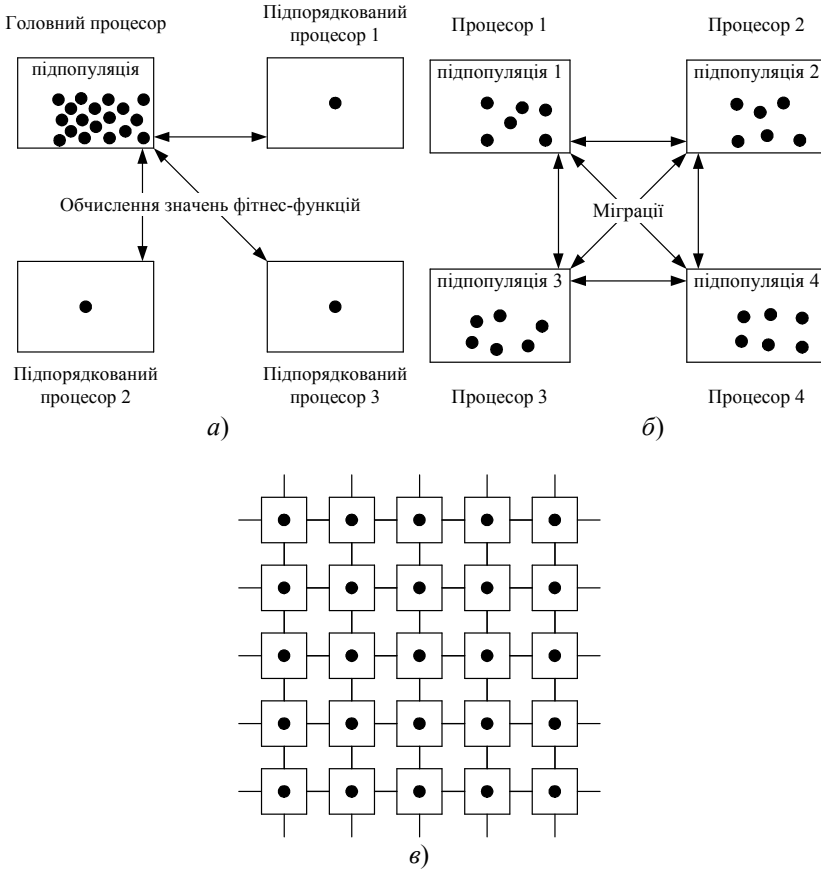


Рисунок 1.3 – Паралельні генетичні методи:

а) модель «ведучий-ведений»; б) острівна модель; в) дифузна модель

Ця модель генетичного методу має такі властивості:

- 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.

### **Дрібноструктурні генетичні методи**

Дрібноструктурні або *дифузійні генетичні методи* (fine-grained GA, cellular GA, diffusion-model GA) складаються з єдиної просторово структурованої популяції (рис. 1.3 в). Як правило, популяція подається у вигляді прямокутної сітки, у вузлах якої розташовані особини. В ідеалі, на кожен особину виділяється окремий процесор для обчислення її фітнес-функції. Батьківську 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.

### **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».

### **Дворівневий генетичний метод**

У дворівневому генетичному методі DAGA2 реалізовано ідею багаторівневої еволюції: генетичний метод використовується на двох рівнях. 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» найприспосованішого 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.

### 1.5.7 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 при наблизненні 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 особини;  $f_{best}^t$  – 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 особини, 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 особини 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 особини, 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.

## 1.6 Ініціалізація та запуск генетичного пошуку

### 1.6.1 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. При цьому кожен ген, тобто елемент інформації генотипу, має своє відображення у фенотипі. Таким чином, для розв'язання задач необхідно подати кожен 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 синоптичним 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 впорядковування 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);  $\varepsilon$  – 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  $\varepsilon$  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  $j$ -ої 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.

### 1.6.2 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 хромосом 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  $O$  відобража-

ється (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:

$$\begin{aligned} f(H_j) &= aO(H_j) + b \quad \text{або} \\ f(H_j) &= O(H_j) - (O_{\text{сєр.}} - c\sigma) \quad \text{або} \\ f(H_j) &= O^k(H_j), \end{aligned}$$

де  $a$ ,  $b$ ,  $c$  і  $k$  – константи, що залежать від вирішуваної задачі;  $O_{\text{сєр.}}$  – середнє значення цільової функції на поточній ітерації;  $\sigma$  – 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 «вузького 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.

### 1.6.3 Ініціалізація

Стандартні генетичні методи починають свою роботу з ініціалізації, 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. 1.4) використовуються такі:

- випадкове формування початкової популяції;
- рівномірне формування початкової популяції (сіткова ініціалізація). При такій стратегії генерується множина рішень, рівно віддалених одне від одного;
- 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.

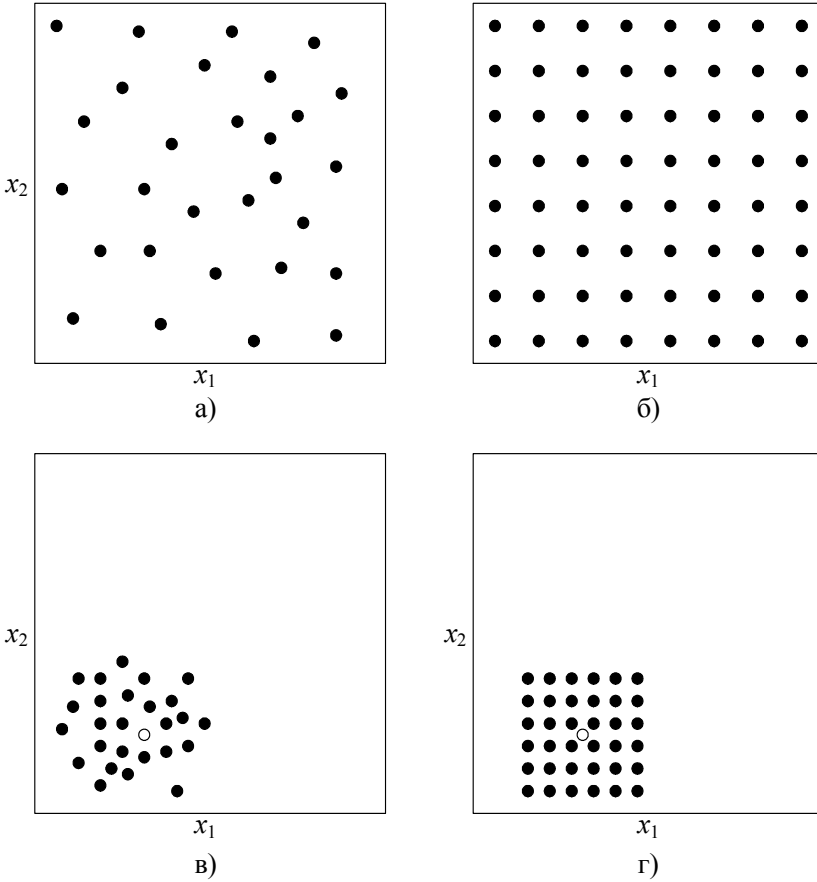


Рисунок 1.4 – Стратегії формування початкової популяції:  
 а) випадкова ініціалізація; б) сіткова ініціалізація;  
 в) випадкова ініціалізація, заснована на знаннях;  
 г) сіткова ініціалізація, заснована на знаннях

## 1.7 Генетичні оператори

Основними генетичними операторами (рис. 1.5) є:

- відбір;
- схрещування;
- мутація.

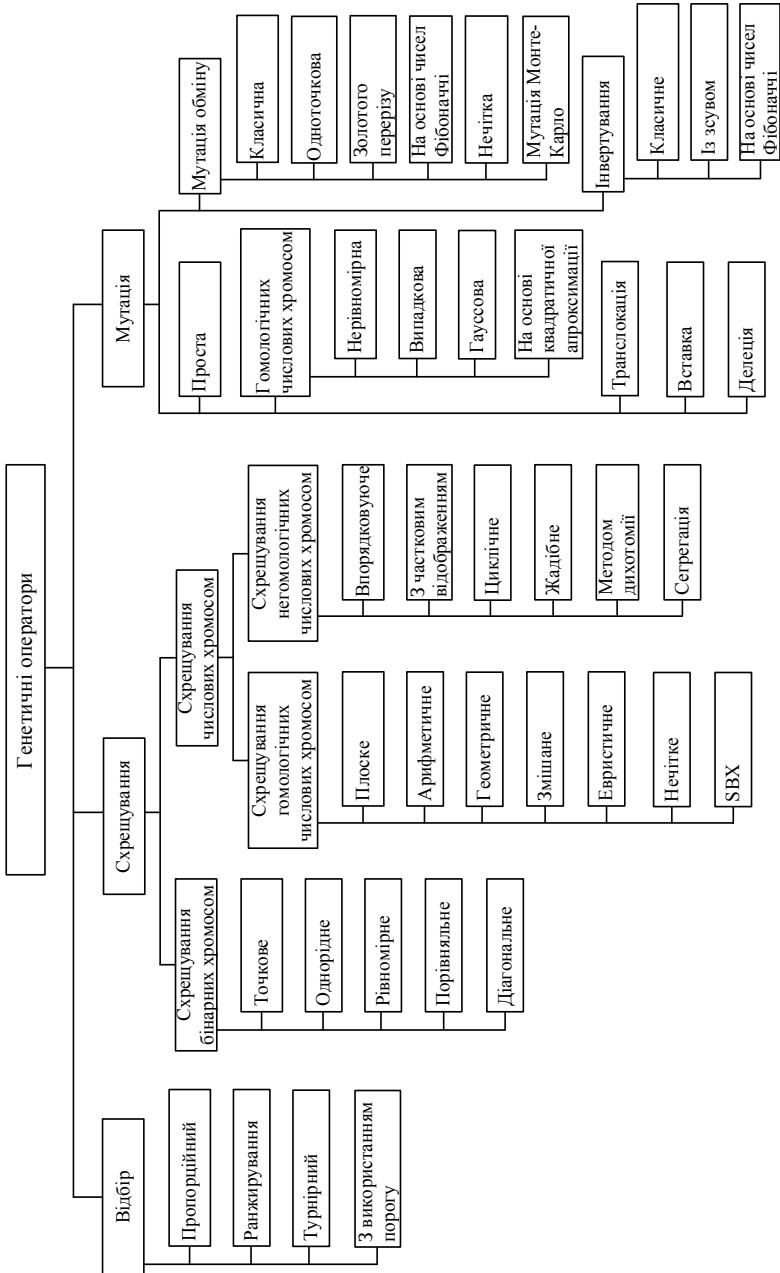


Рисунок 1.5 – Класифікація еволюційних операторів

### 1.7.1 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. 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-ймовірнісний);
- відбір ранжируванням;
- турнірний відбір;
- відбір з використанням порогу.

#### Пропорційний відбір

Даний вид відбору виконується у такій послідовності кроків.

Крок 1. Обчислити значення фітнес-функції кожної особини  $f_j$ .

Крок 2. Знайти середнє значення фітнес-функції  $f_{cp}$  популяції як середнє арифметичне значень фітнес-функцій всіх особин:

$$f_{cp} = \frac{1}{N} \sum_{j=1}^N f_j.$$

Крок 3. Для кожної особини обчислити відношення:

$$P_s(j) = \frac{f_j}{f_{cp}}.$$

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.

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  $2\pi$  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  $x_{\text{rnd}}$  генерується кожного разу випадковою функцією з рівномірним розподілом щільності ймовірності на відрізок  $[0; 1]$ .

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.

### Відбір ранжируванням

Відбір ранжируванням виконується за чотири кроки.

Крок 1. Обчислити пристосованість кожної особини  $f_j$ .

Крок 2. Відсортувати (ранжирувати) 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  $\mu$  – деяке фіксоване число перших членів популяції.

Крок 4. Залежно від величини  $P_s(j)$  відібрати певну частину особин для схрещування.

### Турнірний відбір

Турнірний відбір реалізує  $k$  турнірів, щоб вибрати  $k$  особин. 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 вибіркості у випадку, коли в ході еволюції всі елементи популяції стають приблизно рівними по значенню цільової функції.

### Відбір з використанням порогу

Відбір з використанням порогу (відбір усіканням) 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 популяції великого розміру і залежати також від методу сортування.

### 1.7.2 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.

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.

#### 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.

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 число 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. Сформувати 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 незакодзованому 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.

## Оператори схрещування для бінарних хромосом

До бінарних хромосом можуть бути застосовані такі оператори схрещування:

- точкове;
- однорідне;
- рівномірне;
- порівняльне;
- діагональне.

Оператори схрещування, розроблені для бінарних хромосом, також можуть застосовуватися для гомологічних числових і векторних хромосом.

### *Точкове схрещування*

Точкове схрещування 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 обмінюються 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.

### ***Однорідне схрещування***

Однорідне схрещування (uniform crossover) генерує нащадка шляхом випадкової передачі йому генетичної інформації від батьків. 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:  $h_{j0} = h_{jn}$ , де  $h_{j0}$  – 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.

### ***Рівномірне схрещування***

Рівномірне схрещування 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.

### ***Порівняльне схрещування***

При порівняльному схрещуванні 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.

### Діагональне схрещування

При діагональному схрещуванні для схрещування  $R$  батьків випадковим чином вибирається  $(R - 1)$  однакових точок схрещування в кожному 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 кожен 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).

### **Оператори схрещування для гомологічних числових хромосом**

До гомологічних числових хромосом крім вище розглянутих можуть бути застосовані такі оператори схрещування:

- 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.

#### ***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  $H_n$ . При цьому  $i$ -ий ген  $h_{ni}$  хромосоми-нащадка  $H_n$  визначається за формулою:

$$h_{ni} = \text{rand}[h_{1i}; h_{2i}],$$

де  $\text{rand}[h_{1i}; h_{2i}]$  – число, що випадково згенероване, в інтервалі  $[h_{1i}; h_{2i}]$ .

#### ***Арифметичне схрещування***

Арифметичне схрещування (arithmetical crossover) є найбільш вдалим для пошуку оптимуму функції багатьох дійсних змінних.

При використанні арифметичного схрещування (рис. 1.6) на основі значень генів хромосом-батьків  $H_1$  та  $H_2$  створюються два нащадки  $H_{1n}$  й  $H_{2n}$ , значення  $i$ -их генів  $h_{1ni}$  та  $h_{2ni}$  яких розраховуються за формулами:

$$h_{1ni} = k \cdot h_{1i} + (1 - k)h_{2i},$$

$$h_{2ni} = k \cdot h_{2i} + (1 - k)h_{1i},$$

де  $k \in [0; 1]$  – деякий дійсний коефіцієнт, який може залежати від  $d(H_1; H_2)$  – відстані між хромосомами.

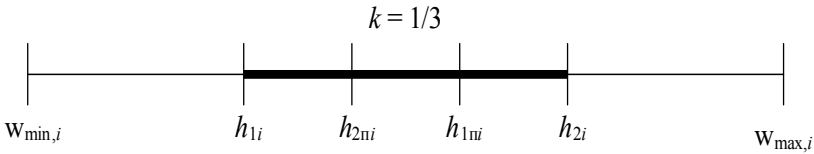


Рисунок 1.6 – Геометрична інтерпретація арифметичного схрещування

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  $H_{1pi} - H_{4pi}$ , значення  $i$ -их генів  $h_{1pi} - h_{4pi}$  яких розраховуються за формулами:

$$h_{1pi} = \frac{h_{1i} + h_{2i}}{2},$$

$$h_{2pi} = w_{\max,i}(1-k) + k \max(h_{1i}, h_{2i}),$$

$$h_{3pi} = w_{\min,i}(1-k) + k \min(h_{1i}, h_{2i}),$$

$$h_{4pi} = \frac{(w_{\max,i} + w_{\min,i})(1-k) + (h_{1i} + h_{2i})k}{2},$$

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  $H_{1pi} - H_{3pi}$ , значення  $i$ -их генів  $h_{1pi} - h_{3pi}$  яких розраховуються за формулами:

$$h_{1ni} = \frac{h_{1i} + h_{2i}}{2},$$

$$h_{2ni} = 1,5h_{1i} - 0,5h_{2i},$$

$$h_{3ni} = -0,5h_{1i} + 1,5h_{2i}.$$

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. 1.7).

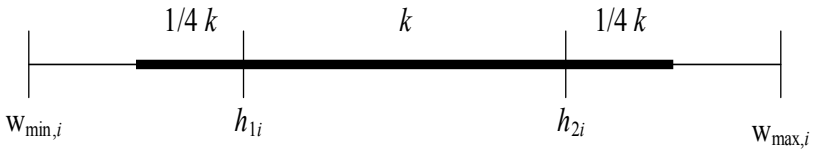


Рисунок 1.7 – Геометрична інтерпретація розширеного лінійчатого схрещування

### **Геометричне схрещування**

При геометричному схрещуванні (geometrical crossover) створюються два нащадки  $H_{1n}$  та  $H_{2n}$ . При цьому значення  $i$ -их генів  $h_{1ni}$  й  $h_{2ni}$  хромосом-нащадків  $H_{1n}$  та  $H_{2n}$ , відповідно, визначаються за формулами:

$$h_{1ni} = h_{1i}^k \cdot h_{2i}^{1-k},$$

$$h_{2ni} = h_{2i}^k \cdot h_{1i}^{1-k},$$

де  $k \in [0; 1]$  – дійсний коефіцієнт, що задається користувачем на етапі ініціалізації генетичного пошуку.

### **Змішане схрещування**

При змішаному схрещуванні (blend, BLX-alpha crossover) генерується один нащадок  $H_n$  (рис. 1.8). При цьому  $i$ -ий ген  $h_{ni}$  хромосоми-нащадка  $H_n$  визначається за формулою:

$$h_{ni} = \text{rand}[c_{\min} - I \cdot a; c_{\max} + I \cdot a],$$

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.

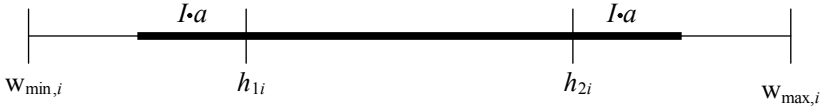


Рисунок 1.8 – Геометрична інтерпретація змішаного схрещування

Окремим випадком змішаного схрещування (при  $a = 0$ ) є плоске схрещування.

### ***Евристичне схрещування***

При евристичному схрещуванні (heuristic crossover) створюється нащадок  $H_n$ , 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. 1.9).

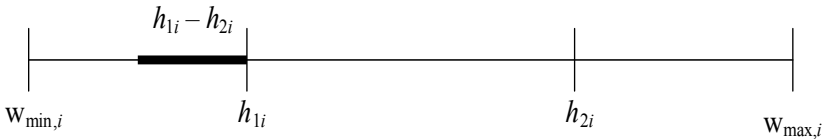


Рисунок 1.9 – Геометрична інтерпретація евристичного схрещування

При використанні евристичного схрещування значення  $i$ -го гена  $h_{ni}$  хромосоми-нащадка  $H_n$  визначається за формулою:

$$h_{ni} = k \cdot (h_{1i} - h_{2i}) + h_{1i}$$

де  $k \in [0; 1]$  – дійсний коефіцієнт, що задається користувачем на етапі ініціалізації генетичного пошуку.

### ***Нечітке схрещування***

При нечіткому схрещуванні (fuzzy recombination, FR-d crossover) створюються два нащадки  $H_{1n}$  и  $H_{2n}$  (рис. 1.10).

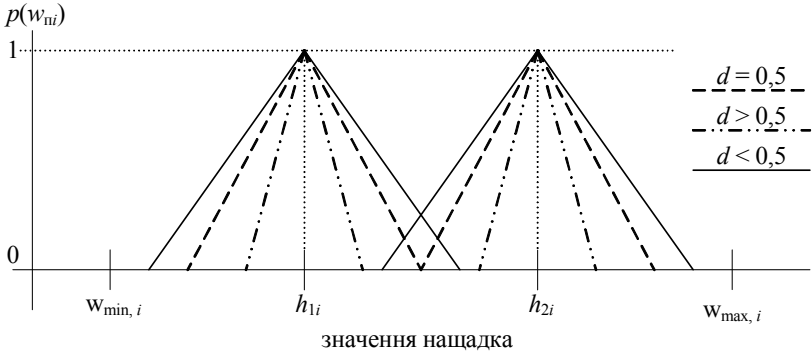


Рисунок 1.10 – Геометрична інтерпретація нечіткого схрещування

При цьому ймовірність  $p(w_{ni})$  того, що  $i$ -ий ген  $h_{ni}$  нащадка прийме значення  $w_{ni}$  визначається розподілом:

$$p(w_{ni}) \in \{\mu_1(w), \mu_2(w)\},$$

де  $\mu_1(w)$  і  $\mu_2(w)$  – розподіли ймовірностей трикутної форми (трикутні нечіткі функції приналежності):

$$\mu_1(w) = \begin{cases} 0, & \text{якщо } w \leq a_1; \\ \frac{w - a_1}{b_1 - a_1}, & \text{якщо } a_1 < w \leq b_1; \\ \frac{c_1 - w}{c_1 - b_1}, & \text{якщо } b_1 < w \leq c_1; \\ 0, & \text{якщо } w \geq c_1, \end{cases} \quad \mu_2(w) = \begin{cases} 0, & \text{якщо } w \leq a_2; \\ \frac{w - a_2}{b_2 - a_2}, & \text{якщо } a_2 < w \leq b_2; \\ \frac{c_2 - w}{c_2 - b_2}, & \text{якщо } b_2 < w \leq c_2; \\ 0, & \text{якщо } w \geq c_2, \end{cases}$$

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 приналежності. Як правило, задається  $d = 0,5$ .

### **SBX-схрещування**

SBX-схрещування (simulated binary crossover) імітує принципи роботи двійкового оператора схрещування.

SBX-схрещування виконується в такій послідовності кроків.

Крок 1. Згенерувати випадкове число  $u = \text{rand}[0; 1]$ .

Крок 2. Обчислити параметр  $\beta$ : 
$$\beta = \begin{cases} (2u)^{\frac{1}{n+1}}, & \text{якщо } u \leq 0,5; \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{n+1}}, & \text{якщо } u > 0,5, \end{cases}$$

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. 1.11 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.

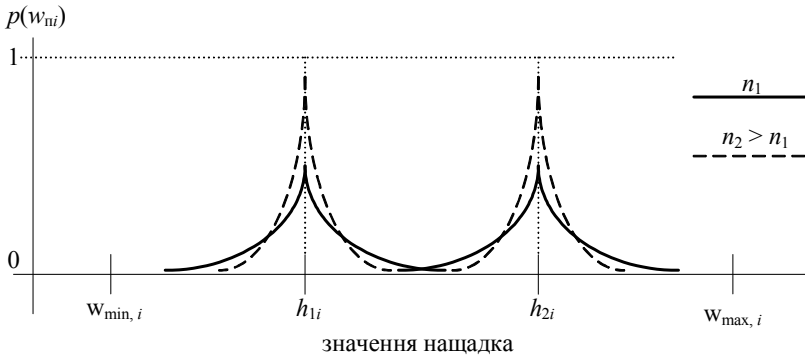


Рисунок 1.11 – Геометрична інтерпретація SBX-схрещування

### Оператори схрещування для негомологічних числових хромосом

Комбінаторні задачі оперують з дискретними структурами, незначні зміни яких часто викликають стрибкоподібну зміну фітнес-функції.

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 повторюють-

ся. У зв'язку з цим були запропоновані альтернативні види оператора схрещування.

### ***Впорядковуючий оператор схрещування***

Впорядковуючий оператор схрещування (order crossover, OX) був запропонований Д. Девісом в 1985 році для негомологічних числових хромосом. Схрещування може проводитися по одній або по двох точках. Точки схрещування вибираються випадково.

При *одноточковому* *впорядковуючому* *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* *впорядковуючому* *схрещуванні* змінюється частина хромосоми, яка знаходиться між точками схрещування.

Також застосовується *позиційно* *впорядковуюче* *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.

### ***Схрещування з частковим відображенням***

Схрещування з частковим відображенням (partially mapped crossover, РМХ-схрещування), запропоноване Д. Голдбергом і Р. Лінглом, полягає в прямому відображенні частини батьківської інформації в дитині. РМХ-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:

$$\begin{aligned} t_1 &= (m_1, P_1^{-1}(P_2(m_1))); \\ t_2 &= (m_1 + 1, (P_1 \circ t_1)^{-1}(P_2(m_1 + 1))); \\ &\dots\dots\dots \\ t_k &= (m_2, (P_1 \circ t_1 \circ t_2 \circ \dots \circ t_{k-1})^{-1}(P_2(m_2))), \end{aligned}$$

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.

### **Циклічне схрещування**

При циклічному схрещуванні (cycle crossover) 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  $C_1 = P_2 \circ (s_0, s_1, \dots, s_k) = P_2 \circ (s_0, s_1) \circ (s_1, s_2) \circ \dots \circ (s_{k-1}, s_k)$ , при цьому:

$$s_1 = P_2^{-1}(P_1(s_0));$$

$$s_2 = P_2^{-1}(P_1(s_1));$$

$$\dots\dots\dots$$

$$s_k = P_2^{-1}(P_1(s_{k-1})).$$

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.

### **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. 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  $\text{temp}$ .

Крок 5. У випадку, якщо хромосома-нащадок складена повністю ( $j = L$ ), перейти до кроку 8.

Крок 6. Виконати перевірку на передчасне замикання циклу:  
 $p_j = p_d$ ,  $d = 1, j - 1$ . 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 вирождження, а також зниження можливостей виходу з локального оптимуму.

### ***Схрещування методом дихотомії***

Схрещування методом дихотомії 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  $(1/2^i)$  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.

### ***Оператор сегрегації***

В результаті застосування оператора сегрегації  $R$  батьківських хромосом породжують одного нащадка.

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.

### ***Інші оператори схрещування***

Окрім жадібного оператора схрещування, гарні результати дають також інші оператори схрещування, орієнтовані на знання. 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  $D = \text{Ціле}(\tau \cdot L)$ .

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 визначитися 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, ...

### 1.7.3 Мутація

Оператор мутації полягає в зміні генів у випадково вибраних позиціях. 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.

### 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.

*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 на  $P_{\text{мг}}$ , 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 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.

## Мутація гомологічних числових хромосом

Такі види мутацій полягають в зміні вибраного для мутації гена  $h_{ij}$  (або всієї хромосоми  $H_j$ ) на деяку величину  $\Delta h_{ij}$ , розраховану за певними методами:

$$h_{ij}' = h_{ij} + \Delta h_{ij},$$

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:

$$h_{ij}' = \begin{cases} h_{ij} + \Delta(t, w_{\max,j} - h_{ij}), & \text{якщо } f(H_j + \Delta(t, w_{\max,j} - h_{ij})) \geq f(H_j - \Delta(t, h_{ij} - w_{\min,j})); \\ h_{ij} - \Delta(t, h_{ij} - w_{\min,j}), & \text{якщо } f(H_j + \Delta(t, w_{\max,j} - h_{ij})) < f(H_j - \Delta(t, h_{ij} - w_{\min,j})), \end{cases}$$

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:

$$w_i(t) = r \left(1 - \frac{t}{T}\right)^{k_i}, \text{ де } r = \text{rand}[0; 1] \text{ – число, що згенероване випадковим}$$

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  $\Delta h_{ij}$ , 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  $\varepsilon_i$  – випадкове число, одержане за нормальним законом розподілу:

$$e_i = N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(h_{ij}-\mu)^2}{2\sigma^2}},$$

де  $\mu$  – математичне сподівання, як правило  $\mu = 0$ ;  $\sigma$  – середньоквадратичне відхилення. Параметр  $\sigma$  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  $\varepsilon_i$  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 нормального розподілу для розрахунку значення параметра  $\varepsilon_i$  доцільно використовувати *розподіл Коші*:

$$e_i = \frac{1}{\pi} \frac{a}{a^2 + h_{ij}^2},$$

де  $a$  – параметр розподілу Коші.

При використанні закону Коші більш ймовірним є отримання великих значень параметру  $\varepsilon_i$ , 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  $\Delta h_{ij}$  можуть бути обчислені за формулами знаходження  $\Delta$  та  $\varepsilon$ , 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 параболу  $h_{ij, \min}$ .

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 необхід-

но знову перетворити до бінарного вигляду, застосувавши використований метод кодування.

### Мутація обміну

Мутація обміну 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  $H = \{h_1, h_2, \dots, h_D, h_{D+1}, \dots, h_L\}$  перетворюється 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 вибирають-

ся 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 методом 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 одноточкової 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 методом 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.

## Інвертування

Оператор інвертування (інверсії) змінює порядок проходження генів в ділянках хромосом. Мета перевпорядкування полягає в тому, щоб спробувати знайти порядок генів, який має кращий еволюційний потенціал. Перевпорядкування 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.

### **Транслокація, вставка і делеція**

Застосування операторів транслокації, вставки і делеції 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.

### 1.7.4 Формування нового покоління

Після схрещування і мутації необхідно створити нову популяцію. 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.

### 1.7.5 Критерії зупинення

Одним з важливих елементів генетичного пошуку є критерії зупинення. 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 точно, 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  $t_n$  та поріг коефіцієнта поліпшення значень фітнес-функції кращої хромосоми  $\rho_n$ .

Починаючи з циклу  $(t_n + 1)$ , 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  $f_{best_{t-1}}$ .

Після чого обчислюється коефіцієнт поліпшення  $\rho$  за формулою:

$$\rho = \frac{f_{best_t} - f_{best_{t-1}}}{f_{best_{t-1}}}.$$

Якщо значення  $\rho$  виявляється меншим, ніж  $\rho_n$ , то критерій зупинення вважається досягнутим, в іншому випадку виконується наступний цикл генетичного пошуку.

*Прийняття значення цільової функції  $f_n$*  також може використовуватися як критерій зупинення. Якщо в процесі функціонування генетичного методу значення цільової функції  $f$  деякої особи досягло значення  $f_n$  з певною наперед заданою точністю  $\epsilon$ , то метод зупиняється. При цьому рішенням задачі є набуте значення цільової функції  $f_n$ .

## 1.8 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, 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.

### 1.8.1 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*), відшукують-

є паралельним чином одночасно для всіх комбінацій. 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.

### 1.8.2 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. Для цього необхідно проаналізувати вплив генетичних операторів відбору, схрещування і мутації.

### 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 вимирає.

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.

### Вплив оператора схрещування

В результаті схрещування деякі схеми руйнуються, а інші виживають. Одержувана схема вважається *зруйнованою*, якщо вона не відповідає жодному з нащадків.

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 чи-

ном, 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.

### Вплив оператора мутації

Для того, щоб схема не була зруйнована внаслідок застосування оператора мутації, необхідно, щоб всі значущі позиції даної схеми залишилися незмінними.

Кожна аель 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:  $1 - O(H) \cdot P_m$ .

### 1.8.3 Теорема схем

Математично механізм роботи генетичного пошуку можна виразити за допомогою теореми схем. 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: однокрестове. 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 особини взяти участь в схрещуванні обчислюються відповідно до відношення  $f/f_{cp.}$ , де  $f$  – значення пристосованості даної особини, а  $f_{cp.}$  – 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:

$$m(H,t+1) \geq m(H,t) \frac{f(H,t)}{f_{cp.}(t)} \left[ 1 - P_c \frac{L(H)}{L-1} \left( 1 - \frac{m(H,t) f(H,t)}{N f_{cp.}(t)} \right) \right].$$

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:

$$m(H,t+1) \geq m(H,t) \frac{f(H,t)}{f_{cp.}(t)} \left[ 1 - P_c \frac{L(H)}{L-1} \left( 1 - \frac{m(H,t) f(H,t)}{N f_{cp.}(t)} \right) \right] (1 - P_m)^{O(H)}.$$

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 базову 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.

#### 1.8.4 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.

*Будівельні блоки* – це схеми, що мають високу пристосованість, низький порядок і коротку визначену довжину.

Після процедури відбору залишаються лише хромосоми з більшою високою пристосованістю. Отже, хромосоми, які є прикладами схем з високою пристосованістю, вибираються частіше. Холланд показав, що, тоді як генетичний метод явним чином обробляє  $N$  хромосом в кожному поколінні, в той же час неявно обробляються порядку  $N^3$  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.

#### 1.9 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 еволюційного пошуку (рис. 1.12).



Рисунок 1.12 – Методи керування параметрами еволюційного пошуку

### 1.9.1 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.

#### 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.

### **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.).

## 1.9.2 Адаптивні методи

Адаптивні методи управління змінюють параметри еволюційного пошуку в залежності від результатів, досягнутих на попередніх ітераціях.

### 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.

### 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 ( $w$ ,  $\sigma$ ). 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;  $\sigma_{t+1}$  й  $t$  – коефіцієнти, що визначають розмах зміни значень генів при використанні мутації на  $t$ -ій і  $(t + 1)$ -ій ітераціях, відповідно;  $\tau_0$  – 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:

$$\begin{aligned}\sigma_{j,t+1} &= \sigma_{j,t} \exp(\tau N(0,1) + \tau_j N_j(0,1)), \\ w_{j,t+1} &= w_{j,t} + \sigma_{t+1} N_j(0,1),\end{aligned}$$

де  $\tau_j$  – параметр, що керує рівнем адаптації для  $j$ -ої змінної залежно від  $N_j(0,1)$ ;  $\tau$  – параметр, що керує рівнем адаптації для всієї популяції;

– хромосома містить трійку значень  $(w, \alpha, \sigma)$ , де  $w$  та  $\sigma$  – вектори, аналогічні використовуваним в попередньому підході;  $\alpha$  – 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:

$$\begin{aligned}\sigma_{j,t+1} &= \sigma_{j,t} \exp(\tau N(0,1) + \tau_j N_j(0,1)), \\ \alpha_{jk,t+1} &= \beta N_{jk}(0,1), \\ w_{t+1} &= w_t + N(0, C(\sigma_{t+1}, \alpha_{t+1})),\end{aligned}$$

де  $\alpha_{jk}$  – кути обертання, що відповідають кореляції між  $j$ -ою та  $k$ -ою змінними  $w_j$  та  $w_k$ ;  $\beta$  – 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.

## Структурно-орієнтовані методи

Структурно-орієнтовані методи (population-structure-based methods) керують параметрами еволюційного пошуку в залежності від структури популяції.

### 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.

### ***Метод просторової популяційної структури***

Метод просторової популяційної структури (spatial control) відображає місце розташування кожної хромосоми на відповідну множину значень параметрів пошуку (рис. 1.13).

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.

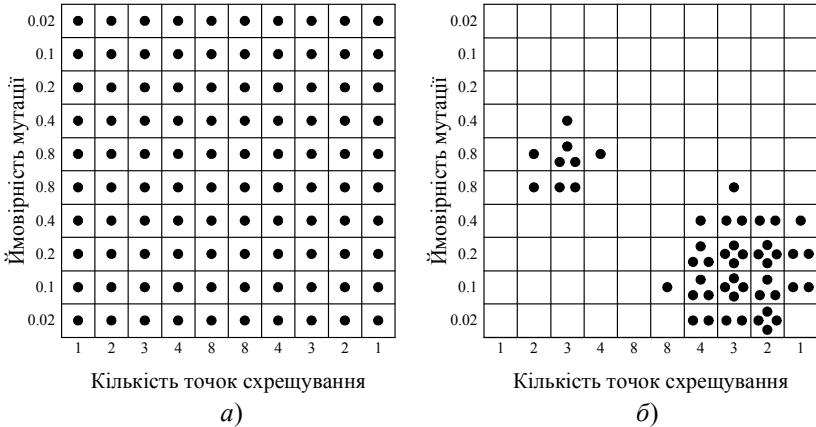


Рисунок 1.13 – Керування параметрами за допомогою методів просторової популяційної структури:  
 а) територіальний генетичний метод (the TBGA);  
 б) територіальна мозаїчна модель (the TBPM)

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 входи якої подається стан навколишнього середовища і внутрішній стан агента.

### 1.10 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  $F_{\min}$  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. 1.14. GAT містить два загальносистемні пункти меню: File і Help. Меню File дозволяє імпортувати й експортувати в робочу область 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;
- *expection* – 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;

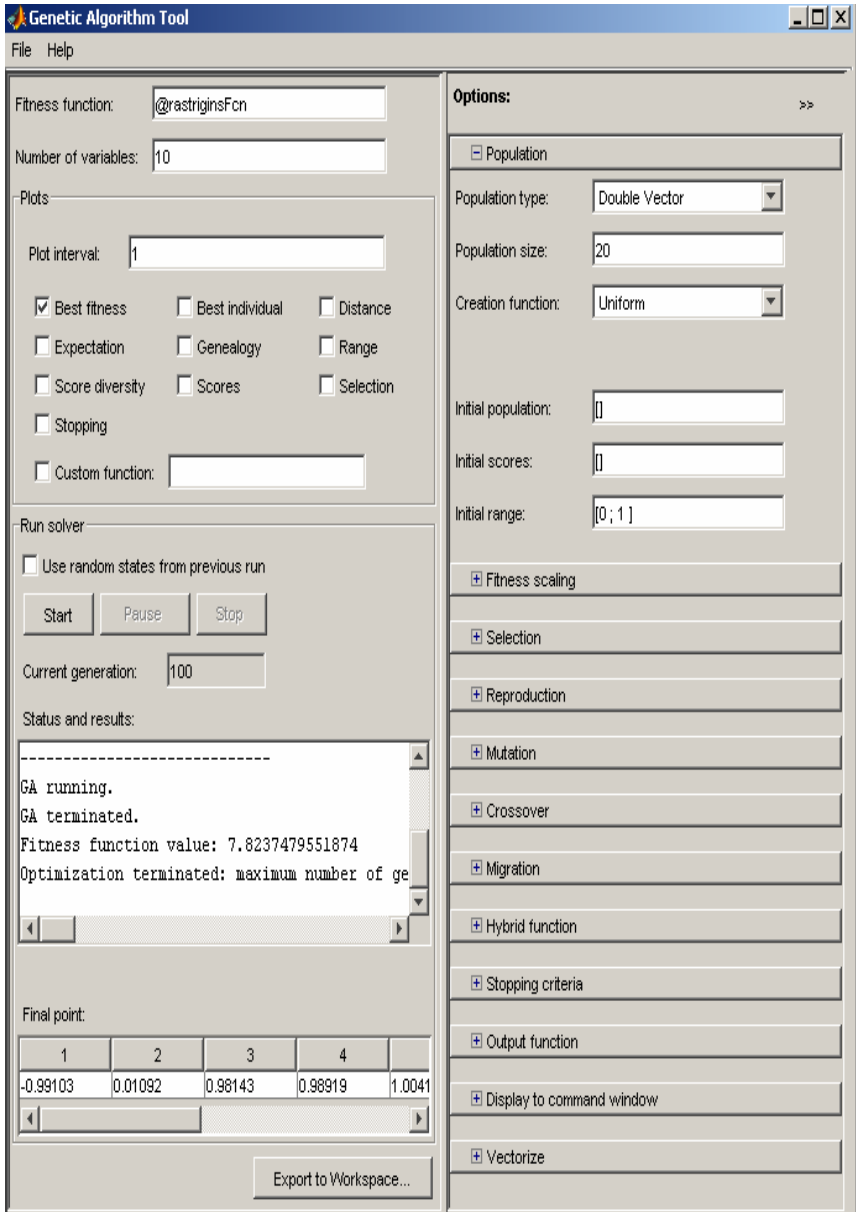


Рисунок 1.14 – Головна форма GAT

- 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.

## **? 1.11 Контрольні питання**

1. Порівняйте методи еволюційного пошуку з іншими методами оптимізації.
2. Які методи відносять до еволюційних?
3. В чому переваги еволюційних методів?
4. Проаналізуйте умови ефективного використання методів еволюційного пошуку.
5. Назвіть особливості еволюційних методів.
6. Які недоліки еволюційного пошуку та в чому вони полягають?
7. Дайте визначення основних термінів, що відносяться до теорії еволюційного пошуку: популяція, розмір популяції, число поколінь, хромосома, ген, локус, алель, фенотип, генотип.
8. Проаналізуйте узагальнену схему роботи еволюційних методів.
9. Наведіть послідовність виконання узагальненого еволюційного пошуку.
10. Які параметри необхідно визначати для роботи еволюційних методів?
11. Виконайте порівняльний аналіз канонічних моделей еволюційного пошуку.

12. В чому полягають особливості породжувальної моделі?
13. Що таке гібридний еволюційний метод? Які існують стратегії взаємодії класичних та еволюційних методів?
14. Назвіть відмінності моделі СНС від класичних еволюційних методів.
15. Які особливості еволюційного методу із змінним часом життя хромосом?
16. Порівняйте мобільний еволюційний метод з класичними методами еволюційного пошуку. Для чого призначені оператори CUT та SPLICE?
17. Проаналізуйте паралельні та багаторівневі еволюційні методи.
18. Наведіть послідовність виконання еволюційного пошуку із зменшенням розміру популяції.
19. Які існують способи кодування параметрів, що оптимізуються, при використанні еволюційних методів?
20. Що таке фітнес-функція?
21. Порівняйте стратегії створення початкової популяції.
22. Виконайте порівняльний аналіз операторів відбору (пропорційний відбір, відбір за допомогою ранжирування, турнірний відбір та відбір з використанням порогу).
23. Які способи формування батьківської пари використовуються в еволюційних методах?
24. Проаналізуйте оператори схрещування ( $n$ -точкове, рівномірне, порівняльне, арифметичне, діагональне).
25. Який вид кодування хромосом використовується в еволюційних методах при розв'язку комбінаторних задач?
26. В чому полягає суть оператора впорядковуючого схрещування? Які відмінності одноточечного від двохточечного впорядковуючого схрещування?
27. Проаналізуйте схрещування із частковим відображенням.
28. Яким чином відбувається схрещування із частковим відображенням?
29. Наведіть послідовність виконання жадібного схрещування. Які переваги та недоліки такого оператора?
30. За рахунок чого реалізується схрещування методом дихотомії?
31. Чим відрізняється оператор сегрегації від інших операторів схрещування? Яким чином реалізується такий оператор?
32. Дайте порівняльну характеристику операторів схрещування, яку використовуються методах при розв'язку задач комбінаторної оптимізації.

33. Для чого призначений оператор мутації? Які оператори мутації використовуються в еволюційних методах?

34. Порівняйте класичну із одноточною мутацією обміну.

35. Яка особливість мутації золотого перетину?

36. Наведіть спільні та відмінні риси мутації золотого перетину та мутації на основі чисел Фібоначчі. Які вони мають переваги та недоліки?

37. В чому полягає нечітка мутація на основі методу дихотомії?

38. В яких випадках застосовується оператор мутації Монте-Карло?

39. Яка мета оператору інвертування?

40. Наведіть послідовність виконання класичного інвертування.

41. Проаналізуйте інвертування із зсувом.

42. Порівняйте інвертування з використанням методу Фібоначчі з іншими видами інвертування.

43. Які особливості має інвертування на основі методу золотого перетину?

44. Порівняйте оператори мутації негомологічних числових хромосом.

45. Чому оператори транслокації, вставки та делеції не набули широкого поширення? Яким чином вони виконуються? Порівняйте їх з іншими еволюційними операторами.

46. Яким чином відбувається формування нового покоління?

47. Які критерії зупину використовуються при еволюційному пошуку?

48. Для чого призначена теорема схем? Дайте визначення основних понять, що використовуються в теоремі схем. В чому полягає теорема Холланда про схеми?

49. З якою метою виконують статистичний аналіз результатів еволюційної оптимізації?

50. Порівняйте поняття генеральної сукупності та вибірки. Що таке об'єм вибірки? Яка вибірка репрезентативною?

51. Чим відрізняються поняття “настройка параметрів” та “управління параметрами” еволюційного пошуку? Проаналізуйте методи управління параметрами еволюційного пошуку (адаптивні та неадаптивні).

52. Поясніть, яким чином впливають ймовірності виконання еволюційних операторів відбору, схрещування та мутації на його ефективність.

53. Обґрунтуйте вплив кількості елітних хромосом, розміру популяції та максимально допустимої кількості ітерацій на результати еволюційного пошуку.

54. Які параметри можна використовувати в функції `ga`? Яким чином вони задаються? Як отримати поточні параметри функції `ga`?

55. Проаналізуйте внутрішню структуру функції `ga` пакету `Matlab`: основні змінні, параметри, методи та допоміжні функції, їх призначення та використання.

56. Проаналізуйте візуальний модуль для роботи з методами еволюційного пошуку `gatoool`: призначення, використання, параметри, візуальні компоненти, методи представлення результатів еволюційного пошуку.

## 2 ГЕНЕТИЧНЕ ТА ЕВОЛЮЦІЙНЕ ПРОГРАМУВАННЯ

### 2.1 Еволюційні стратегії

Еволюційні стратегії орієнтовані на оптимізацію неперервних функцій з використанням операторів схрещування.

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.

#### 2.1.1 $(\mu + \lambda)$ та $(\mu, \lambda)$ 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  $(\mu, \lambda)$  еволюційні стратегії. Параметри  $\mu$  і  $\lambda$  означають число батьків і нащадків, відповідно.

У еволюційній  $(\mu + \lambda)$  стратегії  $\mu$  батьків породжують  $\lambda$  нащадків. При цьому батьки і нащадки беруть участь у загальному селективному відборі, внаслідок чого  $\mu$  кращих особин переходять в наступну популяцію.

Еволюційна  $(\mu, \lambda)$  стратегія передбачає, що  $\mu$  батьків, породивши  $\lambda$  нащадків, гинуть. У наступну популяцію переходять  $\mu$  особин з множини нащадків, причому в даному способі повинна виконуватися умова  $\lambda > \mu$ . Такий підхід може бути застосований 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  $(\mu, \lambda)$  еволюційними стратегіями стають меншими, якщо  $\eta$  є достатньо великим.

Канонічна еволюційна стратегія може бути подана такою послідовністю кроків.

Крок 1. Встановити:  $t = 0$ .

Крок 2. Ініціалізувати популяцію  $P_t$   $\mu$  випадковими індивідами.

Крок 3. Випадковим чином відібрати особини з  $P_t$  для генерації  $\lambda$  нащадків.

Крок 4. Виконати оператор мутації над нащадками.

Крок 5. Обчислити фітнес-функції нащадків.

Крок 6. Вибрати кращі  $\mu$  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.

### 2.1.2 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.

#### 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  $(\mu, \lambda)$  еволюційних стратегій, в яких  $\mu$  хромосом-батьків породжують  $\lambda$  нащадків, функцію успішності  $u_l$  виконання мутації на  $t$ -ій ітерації розраховують за формулою:

$$u_l = \begin{cases} 1, & \text{якщо } \frac{\sum_{j=1}^{\mu} f(H_j^l)}{\mu} < \frac{\sum_{j=1}^{\lambda} f(H_j^{l-1})}{\lambda}; \\ 0, & \text{якщо } \frac{\sum_{j=1}^{\mu} f(H_j^l)}{\mu} \geq \frac{\sum_{j=1}^{\lambda} f(H_j^{l-1})}{\lambda}. \end{cases}$$

Коефіцієнт успішності виконання мутації від  $(t-A-1)$ -ої до  $(t-1)$ -ої ітерації при використанні  $(\mu + \lambda)$  або  $(\mu, \lambda)$  еволюційних стратегій розраховується за формулою:

$$k = \sum_{l=t-A-1}^{t-1} u_l.$$

### Мутація в багатоелементних еволюційних стратегіях

Якщо використовують багатоелементну стратегію для еволюції популяції з  $\mu$  батьків та  $\lambda$  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 багатоеlementних 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;  $\gamma$  – коефіцієнт, що характеризує ступінь зміни значень вектора середньоквадратичних відхилень.

Збіжність еволюційних стратегій чутлива до вибору коефіцієнта  $\gamma$  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  $\mu$  та  $\lambda$ . 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.

### 2.1.3 Метод імітації відпалу

Метод імітації відпалу (simulated annealing) є імовірнісним варіантом локального спуску з можливістю переходу до гірших рішень. 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 контро-

льованому охолодженні субстанції. В результаті виходить міцна кристалічна структура, яка відрізняється від структури з дефектами, що утворюється при швидкому неконтрольованому охолодженні. *Структура* тут є кодованим 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  $\delta E = E_{\text{ws}} - E_{\text{cs}}$  – різниця енергії робочого рішення  $E_{\text{ws}}$  і енергії поточного рішення  $E_{\text{cs}}$ ;  $T$  – 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;  $\alpha$  – 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.

## 2.2 Генетичне програмування

Генетичне програмування застосовує еволюційний пошук для оптимізації комп'ютерних програм.

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.

### 2.2.1 Основні поняття

У генетичному програмуванні гіпотези про вигляд залежності цільової функції від незалежних змінних формуються 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.).

### 2.2.2 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  $P_{\text{temp}}$  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.

### 2.2.3 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.

#### 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.

#### 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.

### **2.3 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. 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.

### 2.3.1 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.

### 2.3.2 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 сформувані 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.

### 2.3.3 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.

### 2.3.4 Метод групового врахування аргументів

Різновидом еволюційного програмування є метод групового врахування аргументів, 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 виду:

$$y_k^{(1)} = g(x_{k_1}, x_{k_2}),$$

де  $y_k^{(1)}$  –  $k$ -ге 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 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 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 (навчаюча 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, вибирати 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 порада 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 нейромережових 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 апіорних припущень автора про результати моделювання;
- дає можливість відшукування незміщеної фізичної моделі об'єкту (закону або кластеризації) – однієї й тієї ж для всіх майбутніх вибірок.

## **2.4 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. 2.1).

Методи полімодального еволюційного пошуку 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.

### **2.4.1 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.

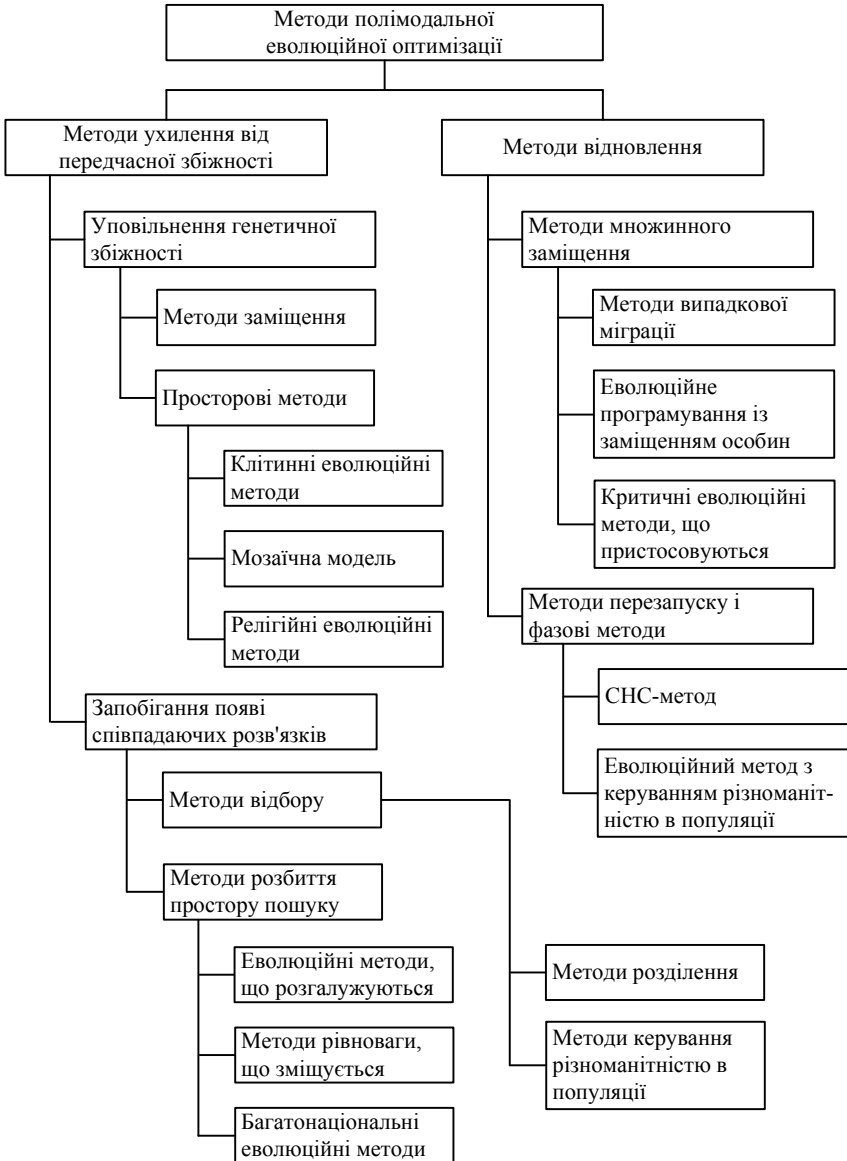


Рисунок 2.1 – Класифікація методів полімодального еволюційного пошуку

## Уповільнення генетичної збіжності

### 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.

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  $H_j^t$  нащадком  $H'_j{}^t$ , яка розраховується за формулою:

$$p(H_j^t, H_j^{t'}) = \frac{f(H_j^{t'})}{f(H_j^{t'}) + f(H_j^t)}.$$

### **Просторові методи**

До просторових методів (spatial population topology) відносять: клітинні еволюційні методи, мозаїчну модель та релігійні 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 зациклення в локальному оптимумі.

**Мозаїчна модель** (patchwork model) є гібридом еволюційних методів, що поєднують в собі ідеї дифузійної моделі, мультиагентних систем і острівної 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. 2.2). 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).

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,  $w_{ij} \in [0; 1]$ ;  $\text{map}_{ij}$  – функція перетворення значення  $s_j$ , одержаного з  $j$ -го сенсора для  $i$ -ої дії;  $S$  – кількість сенсорів.

**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, використовувани 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.

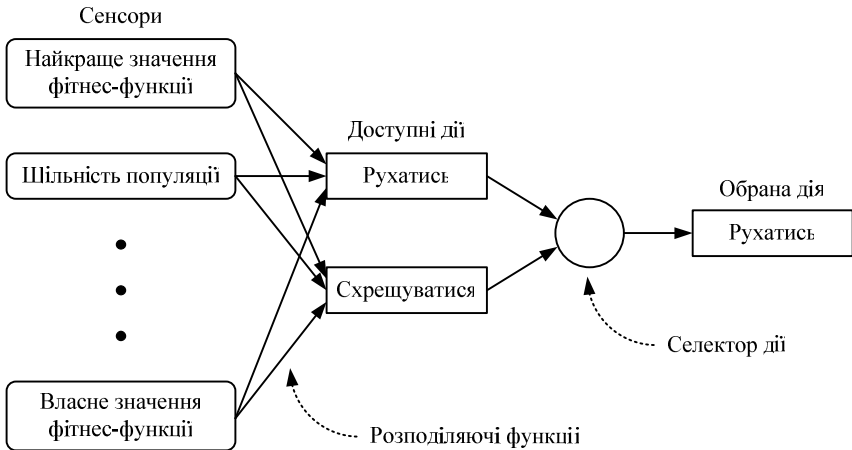


Рисунок 2.2 – Мотиваційна мережа

### Запобігання появі співпадаючих рішень

До методів, що запобігають появі співпадаючих рішень (prevent overlap of solutions), відносять:

- методи відбору;
- методи розбиття простору пошуку.

#### *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.

**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  $H_k$ ;  $\sigma_{sh}$  – порогова відстань між хромосомами (якщо  $d(H_j, H_k) < \sigma_{sh}$ , то вважається, що хромосоми розташовані близько одна до одної);  $\alpha_{sh}$  – коефіцієнт, що визначає ступінь погіршення значення фітнес-функції, як правило,  $\alpha_{sh} = 1$ .

**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 по-

пуляцію. 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  $\alpha$  – 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.

### **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.

**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.

**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 відбирається 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  $H_k$ .

**Багатонаціональні еволюційні методи** (multinational evolutionary algorithms) об'єднують в собі ідеї самоприспособування, адаптації до задачі, розбиття простору пошуку, використання підпопуляцій (рис. 2.3).

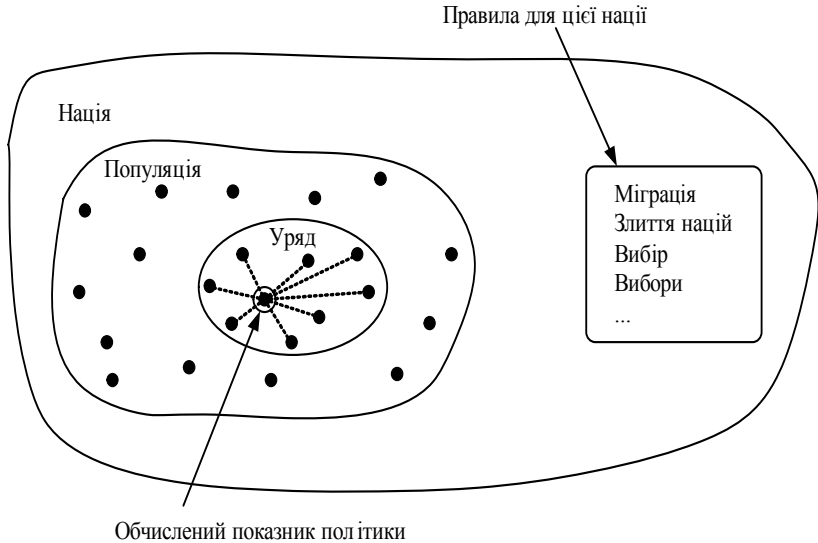


Рисунок 2.3 – Елементи нації  
в багатонаціональному еволюційному методи

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.

#### 2.4.2 Методи відновлення

Методи відновлення (repair strategy) засновані на відновленні різноманітності в популяції і включають такі групи методів:

- методи множинного заміщення;

– методи перезапуску і фазові методи.

### **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.

#### ***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.

#### ***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  $\alpha$  – 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.

#### ***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 використо-

ується один з адаптивних методів керування параметрами еволюційного пошуку.

Нові хромосоми генеруються за допомогою мутації хромосом, що вижили.

### **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.

### ***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 НУХ-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 значен-

ня 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.

### ***Еволюційний метод з керуванням різноманітністю в популяції***

Еволюційний метод з керуванням різноманітністю в популяції (diversity-guided evolutionary algorithm) використовує критерій різноманітності  $\text{diversity}(P)$  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. 2.4).

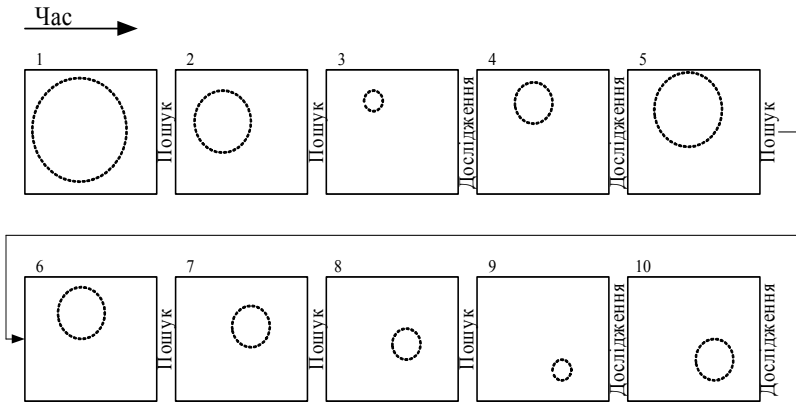


Рисунок 2.4 – Фази еволюційного методу з керуванням різноманітністю популяції

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 мутовують за допомогою застосування Гаусової мутації, для якої середнє значення встановлюється на певній відстані від центральної точки.

## ? 2.5 Контрольні питання

1. Порівняйте еволюційні стратегії з генетичними алгоритмами та методом імітації відпалу.
2. Чим відрізняються  $(\mu + \lambda)$  від  $(\mu, \lambda)$  еволюційних стратегій?
3. В чому полягає особливість оператора мутації, що застосовується в еволюційних стратегіях?
4. Наведіть особливості генетичного програмування.
5. Поясніть етапи генетичного програмування.
6. Проаналізуйте еволюційні оператори, що використовуються у генетичному програмуванні.
7. Виконайте порівняльний аналіз генетичного та еволюційного програмування.
8. Поясніть призначення еволюційного програмування.

9. Які етапи виконуються при еволюційному програмуванні?
10. За допомогою якого оператора формуються нові рішення в еволюційному програмуванні?
11. Поясніть метод групового врахування аргументів.
12. Що таке полімодальна еволюційна оптимізація?
13. Виконайте класифікацію методів полімодального еволюційного пошуку.
14. Проаналізуйте методи ухилення від передчасної збіжності.
15. Наведіть особливості методів відновлення як методів полімодальної еволюційної оптимізації.

### 3 БАГАТОКРИТЕРІАЛЬНИЙ ЕВОЛЮЦІЙНИЙ ПОШУК З ОБМЕЖЕННЯМИ

#### 3.1 Постановка задачі та класифікація методів багатокритеріальної оптимізації

Задача багатокритеріальної оптимізації 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  $m$  обмеженнях-нерівностей виду  $g_i(x) \leq 0$  та  $p$  обмеженнях-рівностей 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 можна класифікувати таким чином (рис. 3.1):

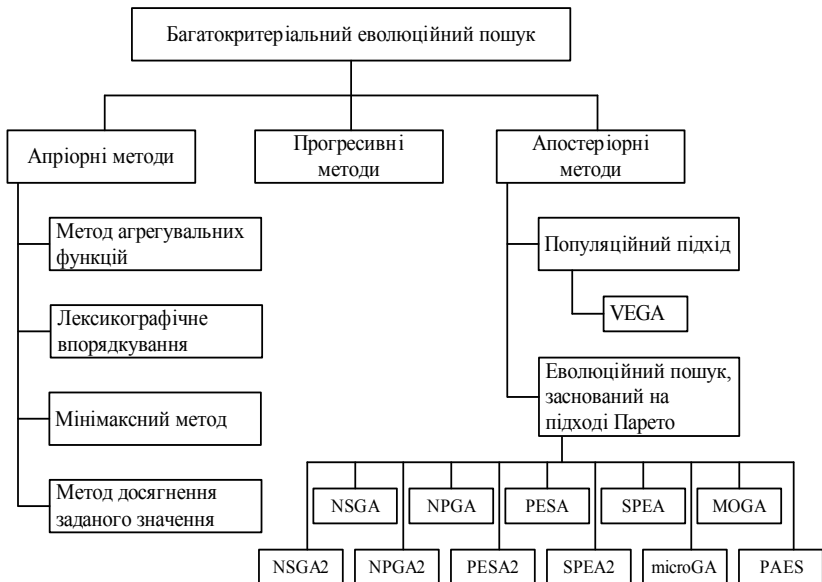


Рисунок 3.1 – Методи багатокритеріального еволюційного пошуку

– *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.

### 3.2 Апріорні методи багатокритеріальної оптимізації

При використанні апріорних методів формується одна цільова функція на основі всієї сукупності функцій  $F(x)$ . Потім проводиться еволюційний пошук оптимуму одержаної функції, який вважається рішенням багатокритеріальної задачі оптимізації.

Виділяють такі апріорні методи:

- метод агрегувальних функцій;
- лексикографічне впорядкування;
- мінімаксний метод;
- метод досягнення заданого значення.

#### 3.2.1 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. 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  $(\mu, \lambda)$  еволюційних стратегій, де  $\lambda$  – кількість нащадків, породжуваних  $\mu$  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.

### 3.2.2 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.

### 3.2.3 Мінімаксний метод

Мінімаксний метод передбачає перетворення багатокритеріальної задачі оптимізації в задачу оптимізації однієї функції за формулою:

$$f(x) = \max_{k \in 1, m} w_k \frac{|f_k^0 - f_k(x)|}{|f_k^0|},$$

де  $f_k^0$  – окремо отриманий мінімум  $k$ -ої цільової функції;  $w_k$  – ваговий коефіцієнт  $k$ -ої цільової функції;  $f(x)$  – узагальнена цільова функція для мінімізації.

### 3.2.4 Метод досягнення заданого значення

При використанні методу досягнення заданого значення 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.

### 3.3 Популяційний підхід

Популяційний підхід 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.

### 3.4 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. При такому підході 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  $f_i(x) \leq f_i(x^*)$  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*.

#### 3.4.1 Основні поняття

Методи багатокритеріального еволюційного пошуку, засновані на підході Парето, побудовані на виявленні *множини Парето*, 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:  $f(H_1) < f(H_2)$ .

На рис. 3.2 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.

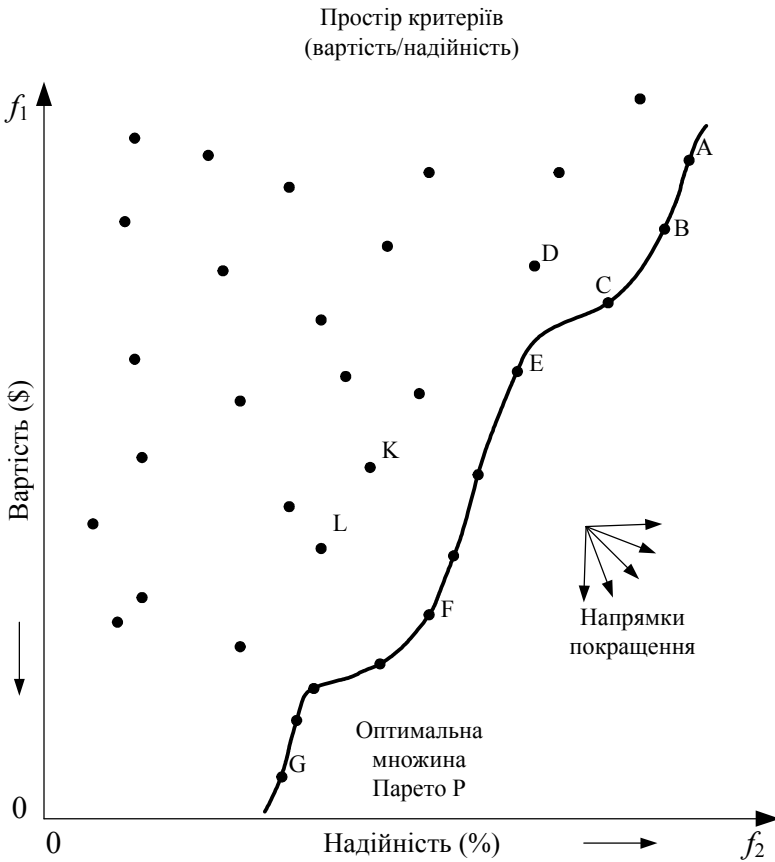


Рисунок 3.2 – Фронт Парето для двоцільової задачі

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.

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.

### 3.4.2 Оцінювання хромосом

При багатокритеріальному еволюційному пошуку, заснованому на підході Парето, використовується один з двох методів оцінювання хромосом (рис. 3.3):

- сортування за принципом недомінантності;
- ранжирування за принципом недомінантності.

Сортування за принципом недомінантності (*nondominated sorting*) передбачає групування хромосом за їх розташуванням у фронтах Парето. При цьому значення фітнес-функцій 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.

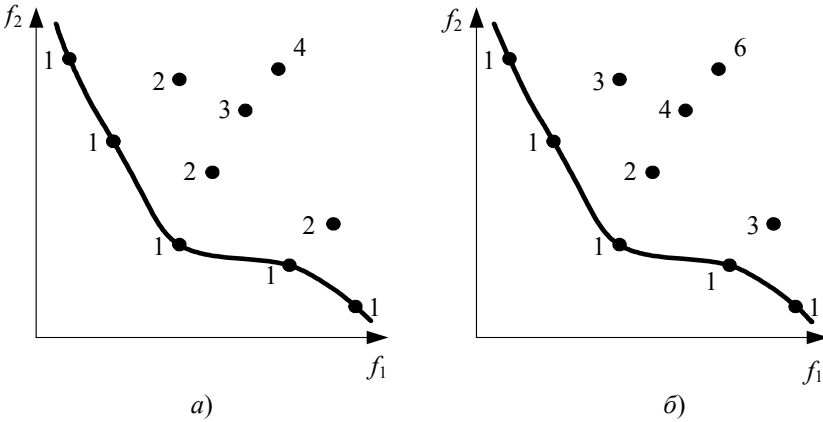


Рисунок 3.3 – Сортування (а) і ранжирування (б) за недомінантністю

### 3.4.3 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. 3.4).

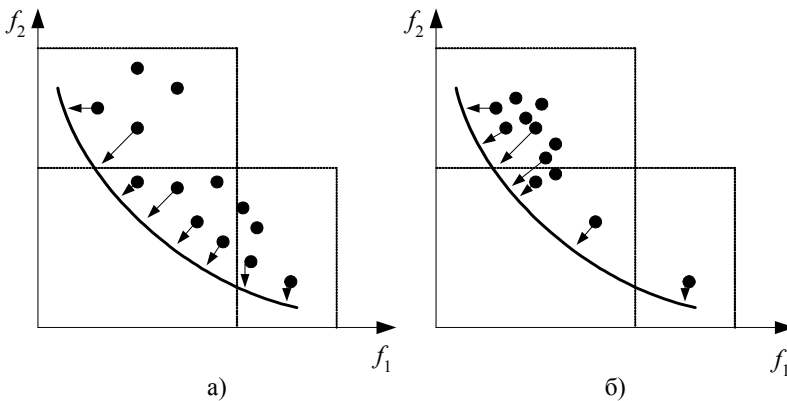


Рисунок 3.4 – Приклад наявності (а) та відсутності (б) різноманітності в популяції

Для запобігання перепоповнюванню множини Парето використовують різні методи її ущільнення (рис. 3.5).

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.

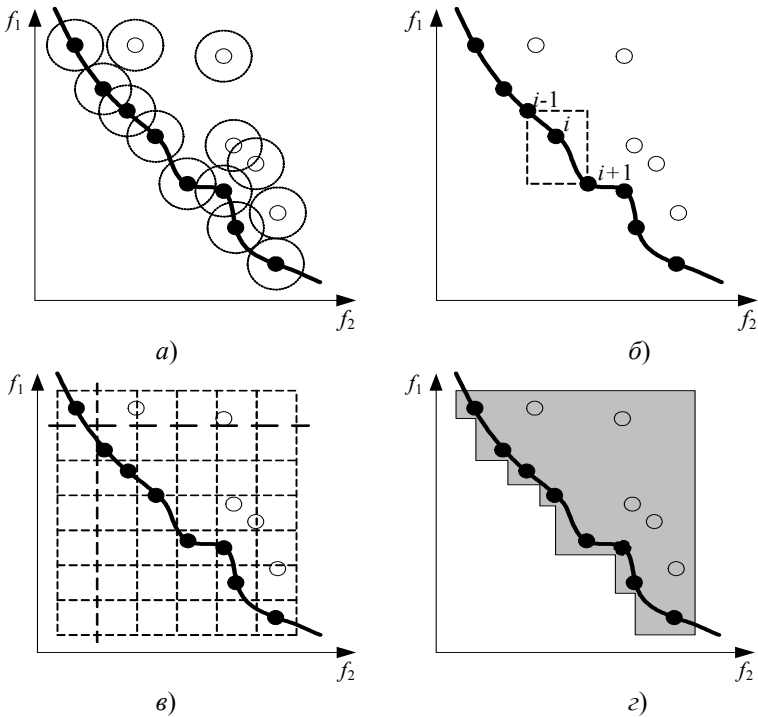


Рисунок 3.5 – Методи ущільнення множини Парето:

- а) розподілення фітнес-функції (*fitness sharing*);
- б) NSGA2-ущільнення (*crowding*);
- в) сіткове розбиття простору цільових функцій (*grid-based niching*);
- г) архівування недовідоманих хромосом ( $\epsilon$ -*dominance archiving*)

Під насиченістю (*niche counts  $m_j$* ) тут розуміється функція, що залежить від кількості розташованих в одному кластері з оцінюваною хромосомою 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 (гіпер-

сфера радіусу  $\sigma_{sh}$ ), – чим ближче розташовуються інші хромосоми до неї, тим більше величина насиченості.

Крок 1. Задати радіус кластерів  $\sigma_{sh}$ .

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.

Крок 3. Обчислити  $f_{j,sh} = \frac{1}{m_j} f_j$ .

*Метод ущільнення (crowding)*, використовується в NSGA2) ранжує хромосоми шляхом вимірювання відстані до їх найближчих недомініємих 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 опиняється декілька хромосом, то ймовірність відбору кожної з них знижується пропорційно кількості хромосом в чарунці.

*Метод архівування недомінантних хромосом ( $\epsilon$ -dominance archiving)* – недомінантні хромосоми в архіві утворюють область з межами, що визначаються параметром  $\epsilon$ . Таким чином, нові недомінантні рішення (хромосоми), що знаходяться близько (на відстані, менше  $\epsilon$ ) не увійдуть до архіву.

### 3.4.4 Методи багатокритеріального еволюційного пошуку

#### Генетичний метод з сортуванням за принципом недомінантності

При використанні генетичного методу з сортуванням за принципом недомінантності (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. 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. Сформувати початкову популяцію з  $N$  хромосом.

Крок 3. Обчислити значення цільових функцій нових хромосом.

Крок 4. Виконати сортування за принципом недомінантності (non-dominated 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 незадоволено, перейти до кроку 3.

Крок 10. Зупинення.

### **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.

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 домінувальною 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 домінувальні 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.

### **Багатокритеріальний еволюційний метод**

Багатокритеріальний еволюційний метод (multi-objective genetic algorithm, MOGA) використовує ранжирування за принципом недомінантності (nondominated ranking). 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 хромосомою  $H_j$ .

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).

### **Еволюційний метод з архівуванням недомінантних хромосом**

Еволюційний метод з архівуванням недомінантних хромосом (strength Pareto evolutionary algorithm, SPEA) використовує архів недо-

мінантних (кращих) хромосом, 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 хромосомою  $H_j$ .

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 зниженням 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  $H_j$  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.

### **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.

Для ущільнення архіву використовується метод сіткового розбиття простору цільових функцій (grid-based niching).

### **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.

Крок 1. Встановити лічильник ітерацій:  $t = 0$ . Випадковим чином сформувані 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. Сформувані 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.

### **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. 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.

### **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. Для відбору хромосом та керування розмаїтістю в популяції використовує метод сіткового розбиття простору цільових функцій.

У зовнішню популяцію (архів) заносяться хромосоми, що є недомінантними по відношенню до хромосом як внутрішньої, так і зовнішньої популяції. Після додавання нових хромосом у зовнішню популяцію хромосоми, що раніше містилися в ній, перевіряються на недомінантність, після чого хромосоми, що не є недомінантними, 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.

## Мікрогенетичний метод

Мікро генетичний метод (micro genetic algorithm, microGA) – метод еволюційного пошуку з малим розміром популяції і процедурою реініціалізації.

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 недомінантності хромосом.

Крок 3.2. Виконати двочкове схрещування.

Крок 3.3. Виконати просту мутацію.

Крок 4. Сформувати 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.

## 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. 3.1.

### 3.5 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).

Як критерії використовуються:

– *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;

– *узгаальнена 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 хромосоною в знайденому фронті Парето і найближчою до неї хромосоною в реальному фронті Парето;

Таблиця 3.1 – Порівняння методів багатокритеріального еволюційного пошуку, заснованого на підході Парето

№	Назва методу	Спосіб кодування хромосом	Метод оцінювання хромосом	Метод ущільнення множини Парето	Генетичні оператори			Стратегія елітизму
					оператор відбору	оператор схрещування	оператор мутації	
1	NSGA	бінарні	сортування за принципом недомінантності	розподілення фітнес-функції	будь-який	будь-який	будь-який	не використовується
2	NPGA	бінарні	перевірка на домінуючість перед обраною множиною хромосом	розподілення фітнес-функції	турнірний	будь-який	будь-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	будь-який	будь-який	зовнішній архів
5	SPEA2	бінарні	ранжирування за принципом недомінантності	розподілення фітнес-функції	турнірний	будь-який	будь-який	зовнішній архів
6	PAES	бінарні	порівняння із хромосомами в архіві	сіткове розбиття простору цільових функцій	не використовується	не використовується	будь-який	зовнішній архів
7	NSGA2	числові	сортування за принципом недомінантності	ущільнення (crowding)	турнірний	SBX-схрещування	будь-який	класичний елітизм
8	NPGA2	бінарні	ранжирування за принципом недомінантності	розподілення фітнес-функції	турнірний	будь-який	будь-який	класичний елітизм
9	PESA	бінарні	сортування за принципом недомінантності	сіткове розбиття простору цільових функцій	турнірний	будь-який	будь-який	зовнішній архів
10	PESA2	бінарні	сортування за принципом недомінантності	сіткове розбиття простору цільових функцій	регіональний турнірний	будь-який	будь-який	зовнішній архів
11	імітоGA	бінарні	сортування за принципом недомінантності	сіткове розбиття простору цільових функцій	парний турнірний	двоточковий	проста мутація	зовнішній архів

– 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:

$$C(A, B) = K_{AB} / |B|, C(B, A) = K_{BA} / |A|,$$

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 недомінувальною хромосомами фронту  $A$ . Якщо  $C(A, B) = 1$ , тоді над всіма хромосомами фронту Парето  $B$  домінує принаймні одна хромосома фронту Парето  $A$ .

### 3.6 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:

- методи, що використовують штрафні функції;
- перетворення простору пошуку;
- використання спеціальних еволюційних операторів;

- відновлення неприпустимих рішень;
- застосування багатокритеріального підходу.

### 3.6.1 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:

– *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  $k_i = 0$ ;  $\beta$  – коефіцієнт, що задається користувачем, як правило  $\beta$  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:

$$f^*(x) = f(x) + |opt_t - opt_{f,t}| \sum_{i=1}^m \left( \frac{g_i(x)}{q_{i,t}} \right)^\beta,$$

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.

### 3.6.2 Перетворення простору пошуку

Перетворення простору пошуку направлене на гарантування допустимих рішень. 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.

### **3.6.3 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. 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.

### **3.6.4 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.

### **3.6.5 Застосування багатокритеріального підходу**

Застосування багатокритеріального підходу передбачає перетворення задачі оптимізації  $K$  цільових функцій з  $m$ -обмеженнями до багатокритеріальної оптимізації  $(m + K)$  функцій.

### **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 кількості порушень обмежень. 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.

### 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. 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.

### 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:

$$f_2(H_j) = \sum_{k=1}^m (\max\{0; g_k(H_j)\})^\beta,$$

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  $d$  проєктується на вісь хромосоми  $H_j$ .

Недоліками даного методу є необхідність підтримки різноманітності рішень в популяціях і великий обсяг необхідних ресурсів для виконання пошуку.

## Мінімаксна стратегія

Мінімаксна стратегія 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.

### ? 3.7 Контрольні питання

1. Порівняйте задачі однокритеріальної та багатокритеріальної оптимізації.

2. В чому полягають особливості еволюційного пошуку при розв'язанні задач багатокритеріальної оптимізації?

3. Виконайте класифікацію методів багатокритеріальної еволюційної оптимізації.

4. Які методи відносять до апріорних методів багатокритеріальної еволюційної оптимізації?

5. Наведіть особливості методу агрегувальних функцій.

6. Порівняйте метод лексикографічного впорядкування та метод досягнення заданого значення.

7. В чому полягає популяційний підхід при розв'язанні задач багатокритеріальної оптимізації за допомогою еволюційного пошуку?

8. Поясніть поняття: «фронт Парето», «множина Парето».

9. Як виконується оцінювання хромосом при еволюційному пошуку, заснованому на підході Парето.

10. Порівняйте методи ущільнення множини Парето.

11. Проаналізуйте еволюційні методи багатокритеріальної оптимізації, що використовують підхід Парето.

12. Які критерії оптимальності використовують при багатокритеріальній еволюційній оптимізації?

13. Виконайте аналіз методів підтримки обмежень при еволюційній оптимізації.

14. У чому полягає використання штрафних функцій при еволюційному пошуку?

15. Проаналізуйте використання спеціальних еволюційних операторів для підтримки обмежень при розв'язанні задач багатокритеріальної оптимізації.

## 4 ЕВОЛЮЦІЙНИЙ СИНТЕЗ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ

Застосування градієнтних методів локального пошуку, традиційно використовуваних для побудови нейромоделей, у ряді випадків (багато-екстремальність цільової функції, недиференційованість функцій активації нейроелементів та ін.) є неприйнятним або неможливим [6, 9, 12, 13, 38].

Тому доцільним є використання методів еволюційного пошуку, 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 навчаючого правила, ініціалізація значень вагових коефіцієнтів, витяг правил з побудованої мережі, пошук оптимальних значень параметрів досліджуваної системи за наявною нейромоделлю і т. п.

### 4.1 Загальна постановка задачі синтезу нейромережеских 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 [7–10, 12, 13, 31, 32, 36].

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 Гауса, 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;  $\psi$  – функція активації;  $\phi$  – 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)  $w_0$ ;
- використовувана 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 нейромережевих 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 нейромережевих моделей прямого поширення, що мають досить просту структуру та є інтерпретабельними.

Багатoshарова нейронна мережа прямого поширення складається з формальних нейронів. При цьому зовнішній сигнал керованих змінних  $x$  надходить тільки на входи нейронів першого шару, виходи нейронів  $\mu$ -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 трьох шарів достатньо для одержання ефективного рішення за допомогою нейронних мереж прямого поширення. Тому при пошуку оптимальної структури неромережевої 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, нейромережева 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 нейромережевої моделі за навчаючою 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  $\text{HM}$  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 нейромережевої моделі  $\text{HM}$  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:

$$\text{fM} \sum_{p=1}^m (y_p - y(\text{HM}, Z_p))^2,$$

де  $Z_p$  – набір значень ознак для  $p$ -го екземпляру;  $y(\text{HM}, Z_p)$  – значення виходу нейромоделі  $\text{HM}$ , обчислене для набору значень  $Z_p$ .

Для побудови ефективної нейромережевої 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 нейромережевих моделей на основі відомої вибірки вихідних даних може бути поданий у вигляді послідовності етапів:

- 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.

#### 4.1.1 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.

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.

Тому побудова нейромережових 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.

1. Ідеалізована постановка: виділити комбінацію ознак  $X^*$  з вихідного масиву даних, при якій досягається мінімум заданого критерію оцінювання набору ознак:

$$J(X^*) = \min_{Xe \in XS} J(Xe),$$

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:

$$J(X^*) = \min_{Xe \in XS, |Xe| \leq L_0} J(Xe),$$

де  $|Xe|$  – кількість елементів у множині  $Xe$ .

3. Знайти набір ознак мінімального розміру, що забезпечує досягнення заданого значення критерію оцінювання значущості набору ознак:

$$|X^*| = \min_{X \in \mathcal{X}, J(X) < \varepsilon} |X|,$$

де  $\varepsilon$  – 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:

- 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 вимагає оцінки значної кількості комбінацій ознак, складених з вихідної сукупності ознак, що унеможливує використання такого підходу при великій кількості ознак у вихідному наборі, оскільки вимагає величезних обчислювальних витрат.

Методи евристичного пошуку 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, 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, 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.

#### **4.1.2 Структурний синтез**

Етап структурного синтезу нейромоделей є найважливішим завданням при побудові нейромереж, 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. Оскільки для вирішення більшості прак-

тичних 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.

Головною метою дослідника на етапі структурного синтезу є визначення загального виду структури шуканого зв'язку, поданої у вигляді нейромережевої 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 строго теоретичну базу для вирішення завдання структурного синтезу нейромоделей. 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 вибірових даних, на основі яких вона оцінюється.

Існуючі методи автоматичного пошуку оптимальної структури нейромережевих моделей, як правило, використовують жадібну стратегію пошуку. 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.

### 4.1.3 Параметричний синтез

На етапі параметричного синтезу відбувається навчання нейромережевої моделі, тобто підбираються такі значення вагових коефіцієнтів мережі, а іноді й параметрів функцій активації, при яких мережа найбільш ефективним чином дозволяє вирішувати поставлене завдання.

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 багат шарових 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 нейромережеві 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 нейромережевих 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:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \frac{\partial \xi}{\partial w_{ij}} \Big|_{w_{ij}=w_{ij}(t)},$$

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 нейромережевих моделей з диференційованими функціями активації.

У розглянутих вище детермінованих методах навчання використовується чітка послідовність дій, виконувана за певними правилами.

Поряд з детермінованими методами також можуть бути використані стохастичні методи настроювання вагових коефіцієнтів нейромережевої 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 використовуюють 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.

#### 4.1.4 Оптимізація побудованої нейромоделі

На можливість використання нейромережових моделей на практиці істотно впливають складність побудованої нейромережі й швидкість обчислення значення цільового параметра за набором даних, що не входить у навчаючу вибірку.

Тому актуальним є спрощення структури синтезованої нейронної мережі. Завдання оптимізації побудованої нейромережової 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 нейромережової моделі:

- 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 нейромережової 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, 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 нейромережових моделей, що здатні виконувати пошук у всьому просторі можливих структур нейронних мереж прямого поширення.

## **4.2 Еволюційний відбір інформативних ознак**

Як показано вище, для синтезу ефективних нейромережових 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.

### **4.2.1 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 набір. 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.

#### 4.2.2 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:

– 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:

– 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.

#### 4.2.3 Еволюційний метод 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.

#### **4.2.4 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, 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 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:

- 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, 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  $K_{\Delta f}$ :

$$K_{\Delta f} = \left| \bar{f}_t - \bar{f}_{t-1} \right|,$$

де  $\bar{f}_t = \frac{1}{N_t} \sum_{j=1}^{N_t} f_j$  та  $\bar{f}_{t-1} = \frac{1}{N_{t-1}} \sum_{j=1}^{N_{t-1}} f_j$  – 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 ймовірність 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  $\lambda_t$ .

Крок 2. Якщо виконується умова:  $\lambda_t > \lambda_{\Pi}$ , де  $\lambda_{\Pi}$  – 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  $p_{in}$  ( $p_{in} \in [0,01; 0,5]$ ) 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  $\lambda_{\Pi}$  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  $\rho$  – 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.

#### 4.2.5 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.

**Теорема.** Зменшення простору пошуку в  $\varphi$  разів при використанні еволюційних методів для відбору інформативних ознак скорочує час, що витрачається на пошук, не менш, ніж в  $\varphi^n$  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  $\mu$ :  $K1_c = \mu L/2$ .

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 використовованого 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 зневажити. 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  $\gamma$ , 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  $\phi = 1/\mu$  раз час, необхідний на еволюційний пошук найбільш інформативної комбінації ознак, зменшується не менш, ніж в  $\phi^n$  раз, де показник ступеню  $n > 1$  – залежить від методу оцінювання хромосом. Теорему доведено.  $\nu$

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.

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;
- для дискретних ознак – коефіцієнт кореляції Фехнера або коефіцієнт кореляції знаків.

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  $\mu$ , 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  $\alpha$  – 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  $(1 - \alpha)L$  залишених після виконання попереднього кроку ознак.

Крок 3.1. Встановити лічильник ітерацій (часу):  $t = 0$ . Ініціалізувати початкову популяцію  $N$  хромосомами розмірністю  $(1 - \alpha)L$ .

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.

#### **4.2.6 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.

## Еволюційний пошук з групуванням ознак на основі індивідуальних оцінок значущості ознак

У методі еволюційного пошуку з групуванням ознак пропонується аналізувати інформативність ознак на етапі ініціалізації параметрів шляхом застосування неітеративних методів оцінювання індивідуальної інформативності ознак. 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 навчаючої 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:

$$c = \arg \max_{q_i=0, i=1, L} I_i.$$

Установити:  $q_c = 1$ ,  $g_c = c$ ,  $N_g = N_g + 1$ , 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:  $d_{ic} \geq d_{\text{ср.}}$  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:

$$f'(H_j) = f(H_j) \frac{\sum_{i=1}^L h_{ij}}{1 + \sum_{i=1}^L I_i h_{ij}},$$

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.

Ймовірність 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;  $\alpha$  – 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.

### **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, 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 навчаючої вибірки, відповідно.

Крок 1.2. На основі розрахованих раніше відстаней між екземплярами, використовуючи методи кластера-аналізу, сформувати 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 масці 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 відшукання комбінації ознак, що володіє максимальною інформативністю.

### **Острівна модель еволюційного пошуку з урахуванням апріорної інформації про значущість ознак**

Пропонована острівна модель еволюційного пошуку для відбору інформативних ознак на основі апріорної 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 кластериза-

цією ознак. При цьому через задану як параметр методу  $t_m$  кількість ітерацій (інтервал міграцій) відбуваються міграції особин з одного острова на інший.

Крок 1. Встановити лічильник ітерацій (часу):  $t = 0$ . Ініціалізувати популяцію кожного острова з використанням апріорної 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, використовує апріорну інформацію про значущість ознак і дозволяє організувати паралельний пошук по двох різних стратегіях, внаслідок чого підвищується ймовірність знаходження глобального оптимуму.

### 4.3 Параметричний синтез нейромереж

У випадку, якщо структура нейромережевої 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 нейромережових 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.

Хромосома при параметричному синтезі (рис. 4.1) складається з  $K$  генів, що містять значення ваг і зсувів всіх нейронів мережі. При цьому для подання значень вагових коефіцієнтів у хромосомах застосовується дійсне кодування.

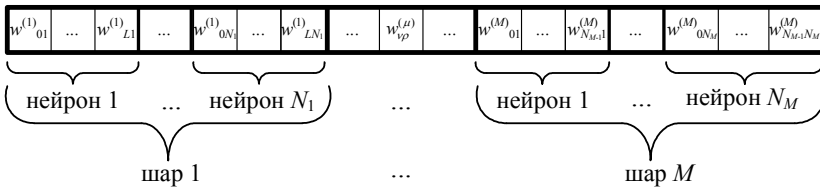


Рисунок 4.1 – Схематичне подання хромосоми при параметричному синтезі нейромоделей

Розмір хромосоми визначається за формулою:

$$K = N_1(L + 1) + \sum_{\mu=1}^M N_{\mu}(N_{\mu-1} + 1),$$

де  $N_{\mu}$  – кількість нейронів на  $\mu$ -ому шарі;  $L$  – кількість ознак у навчаючій 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 значення помилки синтезованої нейромережевої моделі, перевищення максимально допустимої кількості ітерацій, перевищення припустимого часу функціонування методу). У випадку, якщо критерії зупинення задовільнено, 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 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 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 Нгуєна-Відроу, 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 Нгуєна-Відроу для нейронів, що мають такі функції активації, припускає випадкове обчислення значень вагових коефіцієнтів і зсувів.

Авторами в [52] як активні області визначення для таких функцій пропонується використовувати максимальні інтервали значень входів нейроелементів.

Однак при ініціалізації параметрів нейромережових моделей за допомогою методу Нгуєна-Відроу вважається, що вхідні ознаки мають однакову значимість. Відомо, що при розв'язанні реальних практичних завдань інформативність ознак навчаючої вибірки не є однаковою, внаслідок чого використання методу Нгуєна-Відроу для ініціалізації матриці вагових коефіцієнтів нейромережової моделі є недостатньо ефективним.

Тому в розробленому еволюційному методі параметричного синтезу нейромереж пропонується проводити ініціалізацію хромосом початкової популяції за допомогою модифікованого методу Нгуєна-Відроу, що враховує апріорну інформацію про значущість ознак при ініціалізації нейронної мережі.

У еволюційному методі параметричного синтезу нейромереж показники індивідуальної інформативності ознак пропонується також вико-

ристовувати 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, вищою за середню.

Запропонований еволюційний метод параметричного синтезу нейромережних моделей з використанням апріорної 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.

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  $\alpha$ , що визначає ширину діапазону зміни значень вагових коефіцієнтів нейроелементів, за формулою:

$$\alpha = \begin{cases} \gamma^L \sqrt{N_\mu}, & \text{якщо } \mu = 1; \\ \gamma^{N_{\mu-1}} \sqrt{N_\mu}, & \text{якщо } \mu \neq 1, \end{cases}$$

де  $\gamma$  – коефіцієнт, що задається користувачем,  $\gamma \in (0; 1)$ , за замовчуванням пропонується встановлювати:  $\gamma = 0,7$ .

Крок 4.3. Обчислити кількість входів  $V_\mu$   $\mu$ -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  $\mu$ -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 навчаючої вибірки;  $\psi_{v \min}^{(\mu-1)}$  та  $\psi_{v \max}^{(\mu-1)}$  – мінімальне й максимальне можливі значення функції активації  $v$ -го нейрона ( $\mu - 1$ -го шару).

Крок 4.5. Для кожного  $\rho$ -го нейрона  $\mu$ -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  $\mu$ -го шару:  $\rho = 1$ .

Крок 4.7. Згенерувати ваги зв'язків для  $\rho$ -го нейрона  $\mu$ -го шару.

Крок 4.7.1. Встановити лічильник входів  $\rho$ -го нейрона  $\mu$ -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 навчаючої вибірки;  $\text{rand}[a; b]$  – випадково згенероване число в інтервалі  $[a; b]$ .

Крок 4.7.3. Обчислити значення  $v$ -го вагового коефіцієнту  $\rho$ -го нейрона  $\mu$ -го шару:

$$w_{v\rho}^{(\mu)} = \alpha r \frac{x_{\rho \text{ акт. max}}^{(\mu)} - x_{\rho \text{ акт. min}}^{(\mu)}}{x_{v \text{ max}}^{(\mu)} - x_{v \text{ min}}^{(\mu)}}.$$

Крок 4.7.4. Встановити:  $v = v + 1$ .

Крок 4.7.5. Перевірити, чи розраховані значення всіх вагових коефіцієнтів  $\rho$ -го нейрона  $\mu$ -го шару ( $v > V_\mu$ ) нейромережі. У випадку, якщо умова  $v > V_\mu$  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  $\rho$ -го нейрона  $\mu$ -го шару за формулою:

$$w_{0\rho}^{(\mu)} = \sum_{v=1}^{V_\mu} \frac{x_{v \text{ max}}^{(\mu)} + x_{v \text{ min}}^{(\mu)}}{x_{\rho \text{ акт. max}}^{(\mu)} - x_{\rho \text{ акт. min}}^{(\mu)}} w_{v\rho}^{(\mu)} + b_\rho,$$

$$\text{де } b_\rho = \begin{cases} \frac{1}{2}(x_{\rho \text{ акт. max}}^{(\mu)} + x_{\rho \text{ акт. min}}^{(\mu)}) + \frac{1}{2}\alpha(x_{\rho \text{ акт. max}}^{(\mu)} - x_{\rho \text{ акт. min}}^{(\mu)}) \left(-1 + \frac{2(\rho-1)}{N_\mu - 1}\right), & \text{якщо } N_\mu \neq 1; \\ \frac{1}{2}(x_{\rho \text{ акт. max}}^{(\mu)} + x_{\rho \text{ акт. min}}^{(\mu)}) & \text{якщо } N_\mu = 1. \end{cases}$$

Крок 4.9. Встановити:  $\rho = \rho + 1$ .

Крок 4.10. Якщо розраховані значення вагових коефіцієнтів і зсувів всіх нейронів  $\mu$ -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 значення 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 інформати-

вності ознак екземплярів навчаючої вибірки;  $\gamma$  – 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 зна-

чення 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.

#### **4.4 Структурний синтез нейронних мереж**

У випадку, якщо структуру нейромережевої 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.

##### **4.4.1 Подання інформації про структуру нейромережі в хромосомі**

Існують такі методи кодування інформації про структуру нейромоделі в хромосомах:

- пряме кодування;
- параметричне подання;
- комбіноване подання;
- подання у вигляді породжуючих правил;
- фрактальне подання;
- популяційне подання;
- дерево подання.

## Пряме кодування

При прямому кодуванні наявність кожного можливого міжнейронного зв'язку безпосередньо описується в бінарній матриці зв'язків  $C$  нейромережі, у якій одиничне значення в чарунці  $c_{ij}$  відповідає наявності зв'язку від  $i$ -го до  $j$ -го нейрона. Таким чином, нейронна мережа подається у вигляді матриці суміжності.

Хромосома при прямому кодуванні подається бітовим рядком, що містить інформацію про наявність міжнейронних зв'язків.

При цьому довжина хромосоми дорівнює  $B^2$ , де  $B = L + A$  – максимально припустима кількість вузлів (сума загальної кількості ознак  $L$  у навчаючій вибірці даних і максимально припустимої кількості нейронів  $A$ ) у нейромоделі.

У випадку структурного синтезу нейромереж прямого поширення всі значення елементів матриці зв'язків, що розташовані на головній діагоналі й нижче її, дорівнюють нулю, тому хромосому можна спростити, залишивши в ній тільки елементи матриці зв'язків, що перебувають вище головної діагоналі (рис. 4.2), у результаті чого кількість генів у хромосомі визначається за формулою:  $K = B \times (B - 1) / 2$ .

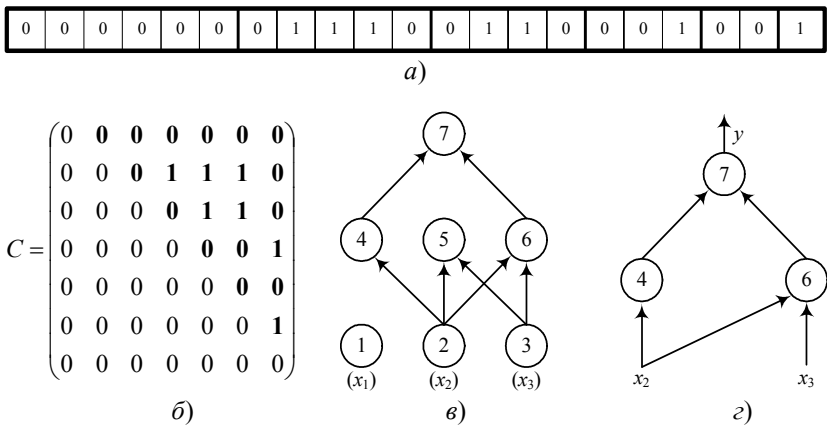


Рисунок 4.2 – Приклад хромосоми та відповідної їй нейромережі при структурному синтезі:

- a) хромосома;
- б) матриця зв'язків нейромережі, що відповідає хромосомі;
- в) побудований на основі матриці зв'язків граф;
- г) синтезована нейромережа (без надлишкових нейронів і зв'язків)

Як видно, при такому поданні структури мережі розмір хромосоми пропорційний квадрату кількості нейронів мережі, внаслідок чого збільшення 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. Сформувані матрицю зв'язків нейромережі, що відповідає хромосомі (рис. 4.2, б).

Крок 2. Побудувати на основі матриці зв'язків граф (рис. 4.2, в).

Крок 3. Синтезувати нейромережу на основі побудованого на попередньому кроці графа, видаливши при цьому нейрони, що не мають виділених зв'язків з нейронами наступних шарів (рис. 4.2, 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 функцій активації нейронів при структурному синтезі, в хромосомі можна ввести додаткові гени, що містять інформацію про вид функції активації для кожного нейрона.

### **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.

### **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).

### **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.

### **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.

### **Популяційне подання**

При такому способі кодування хромосома в популяції відповідає не нейромережі, а окремому нейрону багат шарової нейромережевої 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.

### **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.

#### **4.4.2 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 нейромережевою моделлю.

#### **4.4.3 Послідовність виконання структурного синтезу на основі методів еволюційної оптимізації**

Структурний синтез нейромоделі на основі еволюційного підходу може бути виконаний як така послідовність кроків.

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 навчаючої 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 поточно-го покоління. Виконати перехід до кроку 2.

Крок 7. Зупинення.

#### **4.4.4 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.

#### **4.4.5 Метод полімодального еволюційного пошуку з кластеризацією хромосом**

При вирішенні задачі структурного синтезу нейромережних моделей цільова функція, як правило, має кілька оптимумів, 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.

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 перезапуску у випадках недостатності розмаїтості в популяції, що також вимагає значних часових витрат.

У розробленому методі полімодального еволюційного пошуку з кластеризацією хромосом 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 перезапуску.

Розроблений полімодальний еволюційний пошук з кластеризацією хромосом передбачає виконання таких кроків.

Крок 1. Задати кількість оптимумів (оптимальних структур нейромережкових 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-функції 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. Сформувати 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 кожен 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 класифікацією 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 нейромережеву модель, що найкращим чином задовольняє зовнішнім критеріям оцінювання ефективності нейромоделей.

#### 4.5 Структурно-параметричний синтез

Досить істотним недоліком існуючих методів структурного синтезу 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. 4.3): 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.

На рис. 4.3 використовуються такі позначення:  $lw_{vp}$  – 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.

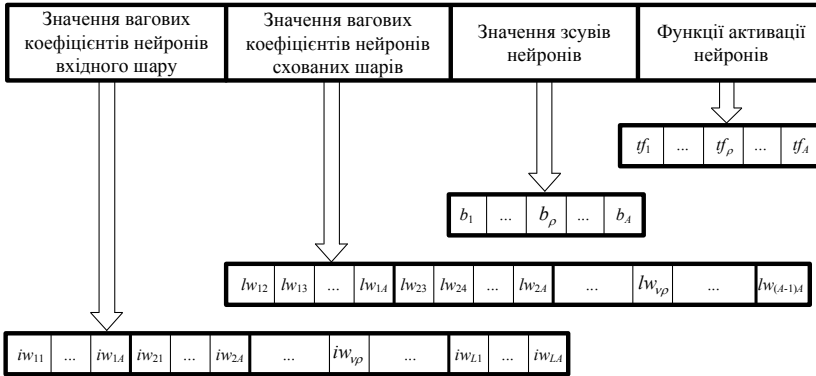


Рисунок 4.3 – Подання хромосоми при структурно-параметричному синтезі нейромереж

Як видно з рис. 4.3, 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 нейромережевим 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 нейромережевих моделей пропонується виконувати як таку послідовність кроків.

Крок 1. Сформувати 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:

$$y_{Ap} = tf_A \left( b_A + \sum_{c=1}^L iw_{cA} x_{cp} + \sum_{c=1}^{A-1} lw_{cA} y_{cp} \right),$$

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 синоптичних 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:

$$h_{i1} = \begin{cases} 0, & \text{якщо } h_{i1} \cdot h_{i2} < 0; \\ kh_{i1} + (1-k)h_{i2}, & \text{в іншому випадку,} \end{cases}$$

$$h_{i2} = \begin{cases} 0, & \text{якщо } h_{i1} \cdot h_{i2} > 0; \\ (1-k)h_{i1} + kh_{i2}, & \text{в іншому випадку,} \end{cases}$$

де  $h_{i1}$  і  $h_{i2}$  – 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:

$$h_{i1} = kh_{i1} + (1-k)h_{i2} \quad \text{та} \quad h_{i2} = kh_{i2} + (1-k)h_{i1}.$$

Значення генів, що визначають функцію активації нейрона, пропонується обчислювати за правилами:

$$h_{in1} = \begin{cases} h_{i1}, & \text{якщо } h_{i1} = h_{i2} \text{ або } r > 0,5; \\ \text{rand}[\text{TF}], & \text{в іншому випадку,} \end{cases}$$

$$h_{in2} = \begin{cases} h_{i2}, & \text{якщо } h_{i1} = h_{i2} \text{ або } r \leq 0,5; \\ \text{rand}[\text{TF}], & \text{в іншому випадку,} \end{cases}$$

де  $r$  – випадково згенероване число в інтервалі  $(0; 1)$ ;  $\text{rand}[\text{TF}]$  – випадково обраний елемент множини TF, що містить функції активації, використовувані 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 поточно-го покоління. Виконати перехід до кроку 2.

Крок 8. Зупинення.

Розроблений еволюційний метод структурно-параметричного синтезу нейромереж дозволяє будувати нейромережеву 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.

#### 4.6 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 нейромоделі відбувається, як правило, окремо від синтезу нейромережі досліджуваного об'єкту, процесу або системи.

#### 4.7 Оптимізація структури нейромоделей

Відомо з, що нейромережеві моделі мають різні характеристики: якість апроксимації, точність, інтерпретабельність, простота апаратної й програмної реалізації, обчислювальна й просторова складність і т. ін. Тому при оптимізації нейромоделей доцільним є використання набору різних критеріїв, що найбільш повним чином характеризує досліджувану залежність.

При оцінюванні структурної складності нейромоделей 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:

$$K_g = \left( \frac{1 + N_{\text{скл.}}}{1 + N_{\text{пр.}}} \right),$$

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, пропонується для спрощення стру-

ктури 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 нейромережових 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 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 другій – 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 недовідування тим, що хромосоми ранжуються за кожною цільовою функцією окремо, а потім знаходиться сумарний ранг кожної хромосоми. Крім того, запропоноване ранжирування вимагає менше обчислювальних ресурсів, ніж ранжирування за принципом недовідування.

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.

#### 4.8 Критерії й методика порівняння еволюційних методів синтезу нейронних мереж

Для забезпечення можливості комплексного аналізу еволюційних методів побудови нейромережових 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, що включає в себе: класичні критерії, критерії оцінювання однорідності популяції, критерії оцінювання еволюційної адаптації, а також критерії оцінювання синтезованих нейромережових 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 знайдено рішення розглянутої оптимізаційної задачі;

– машинний час  $\tau$ , 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:

$$K_s = \frac{N_r}{N \cdot L \cdot T},$$

де  $N_r$  – 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  $\lambda_g$ ;
- коефіцієнт однорідності генотипу-фенотипу  $\lambda_{gf}$ ;
- коефіцієнт однорідності пристосованості  $\lambda_E$ :

$$\lambda_E = \frac{1}{N} \sum_{j=1}^N \frac{1}{1 + D_E(f_j, \bar{f})},$$

де  $f_j$  – значення цільової функції  $j$ -ої хромосоми;

$\bar{f} = \frac{1}{N} \sum_{j=1}^N f_j$  – середнє значення цільової функції в популяції;

$D_E(f_j, \bar{f}) = |f_j - \bar{f}|$  – відстань між цільовими функціями  $f_j$  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 нейромережових моделей пропонується використовувати такі критерії:

– середньоквадратичну помилку нейромережової 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 нейромережову модель на основі навчаючої 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  $\tau$ , 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  $f(H_j)$ , кількість звертань  $N_r$  у процесі еволюційної оптимізації до функції, що генерує псевдовипадкові числа, коефіцієнт однорідності генотипу  $\lambda_g$ , коефіцієнт однорідності генотипу-фенотипу  $\lambda_{gf}$ , коефіцієнт однорідності пристосованості  $\lambda_E$ , 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  $\eta$ .

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  $\lambda_g$ ,  $\lambda_{gf}$ ,  $\lambda_E$ ,  $C_t$ ,  $K_{\Delta f}$ ,  $\eta$  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  $K_s$ ,  $\lambda_g$ ,  $\lambda_{gf}$ ,  $\lambda_E$ ,  $K_e$ , відповідно.

Для побудованих за допомогою запропонованих методів нейромережових моделей визначити значення таких критеріїв: середньоквадратична помилка  $E$ , час розрахунку значення вихідного параметра за нейромережею  $\tau_{\text{НМ}}$ , критерій оцінювання складності структури  $K_c$ , критерій оцінювання обчислювальної складності нейромережової моделі  $K_o$ , інтегральний критерій  $f_1$ , інтегральний критерій  $f_2$ .

Крок 3. Звести отримані результати в узагальнюючі таблиці, побудувати графіки, проаналізувати результати.

#### 4.9 Критерії якості навчальних вибірок і нейромоделей

Вирішення задачі автоматизації прийняття рішень в умовах відсутності або недостатності експертних знань, як правило, полягає в побудові моделі залежності прийнятого рішення від змінних, що спостерігаються за прецедентами [12].

Для побудови моделей прийняття рішень застосовують широкий арсенал методів математичної статистики й обчислювального інтелекту,

зокрема, штучні нейронні мережі [6, 7, 10, 12, 13, 32], нечіткі системи [8, 82], нейро-нечіткі мережі [82], дерева рішень [82], методи розпізнавання образів [29, 34, 42, 70], кластер-аналіз [1–3, 39, 70].

Однак, незважаючи на розходження в обробці даних і структурі моделей, властиві різним методам, загальним для них є використання навчальної вибірки спостережень для структурно-параметричної ідентифікації моделі прийняття рішень.

При цьому виникають дві задачі:

- вибір методу, здатного вирішити задачу щонайкраще при найменших витратах машинних і людських ресурсів;
- формування такої вибірки з наявного набору спостережень, що дозволила б синтезувати модель прийняття рішень щонайкраще при найменших витратах ресурсів.

Для вирішення даних задач пропонується використовувати комплекс критеріїв, що характеризують навчальну вибірку з різних сторін і відображують найбільш важливі для моделювання властивості вибірки. Це дозволить забезпечити рішення поставлених задач, а також істотно автоматизувати вибір методу побудови моделі з наявних у наборі, а також вибір екземплярів для включення в навчальну вибірку.

Серед інструментальних засобів при вирішенні задач технічної діагностики і розпізнавання образів широке застосування отримали штучні нейронні і нейро-нечіткі мережі, що мають здатності апроксимувати залежності за прецедентами. Відомо досить багато різних видів нейро-моделей і методів їхнього навчання [9, 10, 42]. Така велика розмаїтість відомих методів синтезу і моделей нейромереж висуває перед розробником розпізнавальної системи проблему вибору найкращої нейромоделі з множини можливих або синтезу оптимальної нейромоделі.

Традиційно використовувані критерії точності (помилки), а також часу побудови і роботи нейромоделі є необхідними, але недостатніми для обґрунтованого вибору нейромоделі, оскільки не відбивають множини інших властивостей нейромереж [1, 16, 31, 33, 47, 48, 49, 72].

Тому пропонується використовувати комплекс критеріїв, які дозволяють кількісно вимірити властивостей нейромоделей, для вирішення задач їхнього структурно-параметричного синтезу, а також порівняння отриманих моделей.

На основі розглянутих далі критеріїв можливо сконструювати користувальницькі цільові функції для використання в еволюційних методах з метою забезпечення урахування різних властивостей нейромоделей при їхньому синтезі на основі еволюційних методів.

#### 4.9.1 Характеристики і критерії порівняння навчальних вибірок

Нехай задано навчальну вибірку  $\langle x, y \rangle$ , де  $x = \{x^s\}$ ,  $x^s = \{x_j^s\}$ ,  $y = \{y^s_i\}$ ,  $s = 1, 2, \dots, S$ ,  $j = 1, 2, \dots, N$ ,  $i = 1, 2, \dots, N_M$ ,  $x^s$  –  $s$ -ий екземпляр вибірки,  $x_j^s$  – значення  $j$ -ої ознаки  $s$ -го екземпляра,  $S$  – кількість екземплярів вибірки,  $N$  – кількість вхідних ознак,  $N_M$  – кількість вихідних ознак,  $y^s_i$  – значення  $i$ -ої вихідної ознаки, співставлене  $s$ -му екземпляру вибірки, на основі якої синтезована нейронна або нейро-нечітка мережа (для мереж з одним виходом позначимо:  $y^s$  – значення вихідної ознаки, співставлене  $s$ -му екземпляру вибірки, на основі якої синтезована нейронна або нейро-нечітка мережа). Для задач класифікації позначимо:  $S^q$  – кількість екземплярів вибірки, що належать до  $q$ -го класу,  $K$  – кількість класів.

Навчальні вибірки в задачах розпізнавання образів характеризують таким набором найважливіших властивостей [27, 28, 44, 47]: репрезентативність, монотонність, повнота вибірки, розмірність, суперечливість, рівномірність, компактність розташування класів у просторі ознак, складність.

Для таких властивостей, як нерівномірність, повторюваність, складність, раніше були запропоновані чисельні критерії, що характеризують їх [41, 47]. Проте відомі критерії не відбивають усієї повноти властивостей навчальних вибірок, а також застосовні не для всіх задач (наприклад, застосовні тільки для задач з дійсною вихідною змінною [47]).

Тому є доцільним проаналізувати і доробити відомі критерії, а також розробити нові характеристики для формування комплексу показників, здатного охарактеризувати найважливіші властивості вибірки.

Деякі з розглянутих далі критеріїв вимагають задавання вихідної змінної як номера класу. У задачах оцінювання, де вихідна змінна є неперервною, застосування цих критеріїв припускає виділення псевдокласів, для чого можна використовувати розбиття діапазону значень вихідної змінної на рівномірні інтервали:

$$y^s = \text{round} \left( 1 + \frac{\left( y^s - \min_{p=1,2,\dots,S} \{y^p\} \right) (\text{round}(\ln S) - 1)}{\max_{p=1,2,\dots,S} \{y^p\} - \min_{p=1,2,\dots,S} \{y^p\}} \right),$$

де  $\text{round}(a)$  – функція округлення.

Якщо задана максимально допустима сумарна помилка  $\varepsilon$  для вибірки  $\langle x, y \rangle$  з неперервним виходом  $y$ , то номера класів для вибірки можна визначати за формулою:

$$y^s = 1 + \text{round}\left(\frac{S}{\varepsilon}\left(y^s - \min_{p=1,2,\dots,S}\{y^p\}\right)\right).$$

Будемо характеризувати властивості навчальної вибірки  $\langle x, y \rangle$  за допомогою таких характеристик.

*Розмірність вибірки* характеризує, з одного боку, просторову складність вибірки, а з іншого боку – мінімальну кількість операцій обробки вибірки.

Визначимо розмірність вибірки як:

$$Dm = NS.$$

Цей показник може змінюватися від 1 до деякої константи, оскільки число ознак і число екземплярів у навчальній вибірці повинні бути кінцевими. Проте, для формування узагальненого критерію даний критерій виявляється незручним через верхню межу, що плаває. Для усунення даного недоліку будемо використовувати відносну розмірність вибірки.

Відносну розмірність вибірки визначимо як:

$$Dr = 1 - \exp(-\ln(Dm)).$$

Величина  $Dr$  буде приймати значення в діапазоні  $[0, 1]$ . При цьому вона буде чутливою до малих розмірностей, на практиці є дуже корисним і зручним для порівняння різних вибірок, у тому числі, для автоматизації процесу формування вибірки на основі інтегрального критерію якості.

*Репрезентативність вибірки* – характеризує показність вибірки стосовно генеральної сукупності (на практиці дана властивість при невідомих характеристиках генеральної сукупності забезпечується достатністю обсягу і повнотою вибірки).

*Повнота вибірки* визначається забезпеченістю класів екземплярами.

*Рівномірність вибірки* показує, наскільки рівномірно розподілені екземпляри вибірки за класами.

Повноту і рівномірність вибірки пропонується характеризувати такими показниками, як:

– оцінка априорної імовірності (частоти)  $q$ -го класу у вибірці:

$$P(y = q) = \frac{S^q}{S};$$

– мінімальна частота класу у вибірці:

$$P_{\min} = S^{-1} \min_{q=1,2,\dots,K} \{S^q\};$$

– середнє відхилення частоти класу у вибірці:

$$\sigma = \sum_{q=1}^K \left( \frac{1}{K} - \frac{S^q}{S} \right)^2.$$

Дана величина буде змінюватися в діапазоні від нуля (якщо класи мають однакові частоти) до деякої позитивної константи (якщо класи мають неоднакові частоти). Причому вона буде тим більше, чим вище нерівномірність частот класів;

– інверсне нормоване середнє відхилення частоти класу у вибірці:

$$\sigma_{\text{норм.}} = \exp\left(-\sum_{q=1}^K \left(\frac{1}{K} - \frac{S^q}{S}\right)^2\right).$$

Дана величина буде змінюватися в діапазоні від нуля (якщо класи мають неоднакові частоти) до одиниці (якщо класи мають однакові частоти). Причому вона буде тим менше, чим вище нерівномірність частот класів.

*Нерівномірність навчальної вибірки* в [44] характеризується показником:

$$Rg = \sqrt{\sum_{q=1}^K \left(S^q - \frac{1}{K} \sum_{k=1}^K S^k\right)^2} = \sqrt{\sum_{q=1}^K \left(S^q - \frac{S}{K}\right)^2}.$$

Його недоліком є те, що він має рухливу верхню межу в області значень. Виконавши нормування, одержимо відносну характеристику нерівномірності навчальної вибірки:

$$Rg' = \frac{1}{S} \sqrt{\sum_{q=1}^K \left(S^q - \frac{1}{K} \sum_{k=1}^K S^k\right)^2} = \frac{1}{S} \sqrt{\sum_{q=1}^K \left(S^q - \frac{S}{K}\right)^2}.$$

Отриманий показник буде приймати значення в діапазоні від 0 до 1: чим менше буде його значення, тим більше рівномірним буде розподіл екземплярів вибірки по класах.

Відповідно, визначимо характеристику відносної рівномірності навчальної вибірки як:

$$Nr = 1 - Rg'.$$

Отриманий показник буде приймати значення в діапазоні від нуля до одиниці: чим більше буде його значення, тим більше рівномірним буде розподіл екземплярів вибірки по класах.

Рівномірність розподілу екземплярів по осі значень  $i$ -ої ознаки визначимо як:

$$Ev_i = \frac{1}{S} \sum_{g=1}^S \omega_{ig},$$

$$\omega_{ig} = \begin{cases} \left(\sum_{s=1}^S \omega_i(x^s, g)\right)^{-1}, & \sum_{s=1}^S \omega_i(x^s, g) > 0; \\ 0, & \sum_{s=1}^S \omega_i(x^s, g) = 0, \end{cases}$$

$$\omega_i(x^s, g) = \begin{cases} 1, & (g-1) \leq \frac{(x_i^s - \min_{p=1,2,\dots,S}(x_i^p))S}{\max_{p=1,2,\dots,S}(x_i^p) - \min_{p=1,2,\dots,S}(x_i^p)} \leq g; \\ 0, & \text{в іншому випадку,} \end{cases}$$

або

$$\omega_i(x^s, g) = \begin{cases} \exp\left(-\left(x_i^s - \frac{1}{2S}(2g-1)\left(\max_{p=1,2,\dots,S}(x_i^p) - \min_{p=1,2,\dots,S}(x_i^p)\right)\right)^2\right), & (g-1) \leq \frac{(x_i^s - \min_{s=1,2,\dots,S}(x_i^s))S}{\max_{s=1,2,\dots,S}(x_i^s) - \min_{s=1,2,\dots,S}(x_i^s)} \leq g; \\ 0, & \text{в іншому випадку.} \end{cases}$$

Чим ближче значення  $Ev_i$  до одиниці, тим більш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки. У свою чергу, чим ближче значення  $Ev_i$  до нуля, тим менш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки.

Нерівномірність розподілу екземплярів вибірки по осі значень  $i$ -ої ознаки задамо як:

$$NEv_i = 1 - Ev_i.$$

Чим ближче значення  $NEv_i$  до одиниці, тим менш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки. У свою чергу, чим ближче значення  $NEv_i$  до нуля, тим більш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки.

Рівномірність покриття екземплярами вибірки простору ознак визначимо як:

$$Ev = \frac{1}{N} \sum_{i=1}^N Ev_i.$$

Чим ближче буде значення  $Ev$  до одиниці, тим рівномірніше будуть розподілені екземпляри в просторі ознак, що краще з погляду адекватності відображення властивостей генеральної сукупності вибіркою в розглянутій частині простору ознак, однак гірше з погляду можливої надмірності вибірки. У свою чергу, чим ближче значення  $Ev$  до нуля, тим менш рівномірно розподілені екземпляри в просторі ознак, що гірше з погляду адекватності відображення властивостей генеральної сукупності вибіркою в розглянутій частині простору ознак.

Нерівномірність покриття екземплярами простору ознак можна виразити як:

$$NEv = 1 - Ev.$$

Чим ближче значення  $NEv$  до одиниці, тим менш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки. У свою чергу, чим ближче значення  $NEv$  до нуля, тим більш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки.

Рівномірність розподілу екземплярів  $q$ -го класу по осі значень  $i$ -ої ознаки будемо визначати як:

$$Ev_i^q = \frac{1}{S^q} \sum_{g=1}^{S^q} \omega_{ig}^q,$$

$$\omega_{ig}^q = \begin{cases} \left( \sum_{s=1}^S \{ \omega_i^q(x^s, g) | y^s = q \} \right)^{-1} \cdot \sum_{s=1}^S \{ \omega_i^q(x^s, g) | y^s = q \} > 0; \\ 0, \sum_{s=1}^S \{ \omega_i^q(x^s, g) | y^s = q \} = 0, \end{cases}$$

$$\omega_i^q(x^s, g) = \begin{cases} 1, y^s = q, (g-1) \leq \frac{(x_i^s - \min_{p=1,2,\dots,S}(x_i^p))S}{\max_{p=1,2,\dots,S}(x_i^p) - \min_{p=1,2,\dots,S}(x_i^p)} \leq g; \\ 0, \text{в іншому випадку,} \end{cases}$$

або

$$\omega_i^q(x^s, g) = \begin{cases} \exp\left(-\left(x_i^s - \frac{1}{2S}(2g-1)\left(\max_{p=1,2,\dots,S}(x_i^p) - \min_{p=1,2,\dots,S}(x_i^p)\right)\right)^2\right), y^s = q, (g-1) \leq \frac{(x_i^s - \min_{s=1,2,\dots,S}(x_i^s))S}{\max_{s=1,2,\dots,S}(x_i^s) - \min_{s=1,2,\dots,S}(x_i^s)} \leq g; \\ 0, \text{в іншому випадку.} \end{cases}$$

Чим ближче значення  $Ev_i^q$  до одиниці, тим більш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки, і, отже, ситуація гірше з погляду гіпотези про компактність класів і розподіляючих властивостей  $i$ -ої ознаки. У свою чергу, чим ближче значення  $Ev_i^q$  до нуля, тим менш рівномірно розподілені екземпляри по осі значень  $i$ -ої ознаки і, отже, ситуація краще з погляду гіпотези про компактність класів і розподіляючих властивостей  $i$ -ої ознаки.

Нерівномірність розподілу екземплярів  $q$ -го класу по осі значень  $i$ -ої ознаки задамо як:

$$NEv_i^q = 1 - Ev_i^q.$$

Чим ближче значення  $NEv_i^q$  до одиниці, тим менш рівномірно розподілені екземпляри  $q$ -го класу по осі значень  $i$ -ої ознаки. У свою чергу, чим ближче значення  $NEv_i^q$  до нуля, тим більш рівномірно розподілені екземпляри  $q$ -го класу по осі значень  $i$ -ої ознаки.

Рівномірність покриття екземплярами  $q$ -го класу простору ознак подамо формулою:

$$Ev^q = \frac{1}{N} \sum_{i=1}^N Ev_i^q.$$

Чим ближче значення  $Ev^q$  до одиниці, тим рівномірніше розподілені екземпляри  $q$ -го класу в просторі ознак, що гірше з погляду гіпотези про компактність класів. У свою чергу, чим ближче значення  $Ev^q$  до нуля, тим менш рівномірно розподілені екземпляри в просторі ознак, що краще з погляду гіпотези про компактність класів.

Нерівномірність покриття екземплярами  $q$ -го класу простору ознак визначимо як:

$$NEv^q = 1 - Ev^q.$$

Чим ближче значення  $NEv^q$  до одиниці, тим менш рівномірно розподілені екземпляри  $q$ -го класу в просторі ознак.

Середня рівномірність покриття екземплярами класів простору ознак буде визначатися як:

$$\bar{E}v = \frac{1}{K} \sum_{q=1}^K Ev^q.$$

Мінімальний рівень рівномірності покриття екземплярами класів простору ознак буде визначатися як:

$$\bar{E}v = \min_{q=1,2,\dots,K} \{Ev^q\}.$$

*Повторюваність навчальної вибірки*, згідно з [44], може бути визначена як показник, що характеризує кількість однакових екземплярів, що належать до тому самому класу. Формально це, пронормувавши, можна подати як:

$$Rp(x, y) = \frac{2}{S(S-1)} \sum_{s=1}^S \sum_{g=s+1}^S \tau(x, y, s, g),$$

$$\tau(x, y, s, g) = \begin{cases} 1, y^s = y^g, \forall i = 1, 2, \dots, N : x_i^s = x_i^g; \\ 0, \text{в іншому випадку.} \end{cases}$$

Величина  $Rp$  буде мінімальною (рівною нулю) у випадку, якщо всі екземпляри навчальної вибірки є відмінними один від одного, і максимальною (рівною одиниці), якщо всі екземпляри є однаковими.

Проте, такий показник буде реагувати тільки на абсолютні збіги навчальних прикладів. На практиці ж часто приходиться мати справу з вибірками, у яких містяться не однакові, але близькі за властивостями ("майже однакові") екземпляри одного класу. Для урахування подібних випадків перевизначимо показник  $\tau$  як:

$$\tau(x, y, s, g) = \begin{cases} \exp\left(-\alpha \sum_{i=1}^N (x_i^s - x_i^g)^2\right), y^s = y^g; \\ 0, y^s \neq y^g. \end{cases}$$

Тут  $\alpha$  – коефіцієнт, що регулює положення межі локальної близькості екземплярів одного класу,  $\alpha > 0$ . У найпростішому випадку можна покласти  $\alpha = 1$ .

Отримана формула буде застосовна для задач розпізнавання образів, однак буде мало придатна для задач оцінювання. Для задач, де вихідна змінна приймає неперервні значення в певному діапазоні, перевизначимо показник  $\tau$  як:

$$\tau(x, y, s, g) = \begin{cases} \exp\left(-\alpha \sum_{i=1}^N (x_i^s - x_i^g)^2\right), & |y^s - y^g| \leq \delta; \\ 0, & |y^s - y^g| > \delta. \end{cases}$$

де  $\delta$  – константа, що регулює чутливість для визначення подоби значень вихідної змінної,  $\delta > 0$ . Значення константи  $\delta$  пропонується автоматично визначати попередньо на основі формули:

$$\delta = \frac{1}{2S-1} \left( \sqrt{\left( \max_{s=1,2,\dots,S} \{y^s\} - \min_{s=1,2,\dots,S} \{y^s\} \right)^2} + \sqrt{\sum_{s=1}^S \sum_{g=s+1}^S (y^s - y^g)^2} \right).$$

Як протилежну характеристику вибірки стосовно повторюваності визначимо унікальність екземплярів вибірки як:

$$Rn = 1 - Rp.$$

При побудові розпізнавальних моделей, часто висувається вимога незалежності вхідних змінних. Для оцінювання якості вибірки з погляду цієї вимоги будемо використовувати показники:

– усередненої незалежності вхідних змінних:

$$\bar{I}dp = 1 - \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N |r_{i,j}|,$$

де  $r_{i,j}$  – коефіцієнт парної кореляції  $i$ -ої та  $j$ -ої ознак [27];

– мінімальної незалежності вхідних змінних:

$$\check{I}dp = 1 - \max_{\substack{i=1,2,\dots,N; \\ j=i+1,\dots,N}} |r_{i,j}|;$$

– максимальної незалежності вхідних змінних:

$$\hat{I}dp = 1 - \min_{\substack{i=1,2,\dots,N; \\ j=i+1,\dots,N}} |r_{i,j}|.$$

Поряд із незалежністю вхідних змінних між собою при вирішенні задач побудови моделей висувається вимога наявності зв'язку між вихідною і вхідними змінними, причому переважніше лінійного зв'язку. Для характеристики відображення у вибірці зв'язку вхідних і вихідної змінних пропонується використовувати показники:

– максимального лінійного зв'язку вхідних і вихідної змінних:

$$\bar{Y}dp = \max_{i=1,2,\dots,N} |r_{i,y}|,$$

де  $r_{i,y}$  – коефіцієнт парної кореляції  $i$ -ої ознаки і вихідної ознаки [27];

– середнього лінійного зв'язку вхідних і вихідної змінних:

$$\bar{Y}dp = \frac{1}{N} \sum_{i=1}^N |r_{i,y}|;$$

– комбіновані показники незалежності вхідних змінних і лінійності зв'язку з вихідною змінною:

$$\begin{aligned}\widehat{I}_Y &= \max_{\substack{i=1,2,\dots,N; \\ j=i+1,\dots,N}} \left\{ r_{i,y} \left( 1 - |r_{i,j}| \right) \right\}, \\ \widetilde{I}_Y &= \min_{\substack{i=1,2,\dots,N; \\ j=i+1,\dots,N}} \left\{ r_{i,y} \left( 1 - |r_{i,j}| \right) \right\}, \\ \bar{I}_Y &= \frac{2}{N(N-1)} \sum_{i=1}^N \left( r_{i,y} \left| \sum_{j=i+1}^N \left( 1 - |r_{i,j}| \right) \right| \right).\end{aligned}$$

*Компактність* розташування класів у просторі ознак відбиває простоту рішення задачі розпізнавання (чим компактніше розташовані екземпляри кожного класу, тим простіше побудувати модель, що розпізнає).

Компактність розташування екземплярів  $q$ -го класу за  $i$ -ою ознакою визначимо як:

$$Co_i^q = 1 - \frac{2 \sum_{s=1}^S \sum_{g=s+1}^S \left\{ (x_i^s - x_i^g)^2 \mid y^s = y^g = q \right\}}{S^q (S^q - 1) \left( \max_{s=1,2,\dots,S} \{x_i^s \mid y^s = q\} - \min_{s=1,2,\dots,S} \{x_i^s \mid y^s = q\} \right)^2}.$$

Чим більше значення  $Co_i^q$ , тим, у середньому, компактніше розташовані екземпляри  $q$ -го класу за  $i$ -ою ознакою.

Компактність розташування екземплярів  $q$ -го класу подамо як:

$$Co^q = 1 - \frac{2 \sum_{s=1}^S \sum_{g=s+1}^S \sum_{i=1}^N \left\{ (x_i^s - x_i^g)^2 \mid y^s = y^g = q \right\}}{S^q (S^q - 1) \sum_{i=1}^N \left( \max_{s=1,2,\dots,S} \{x_i^s \mid y^s = q\} - \min_{s=1,2,\dots,S} \{x_i^s \mid y^s = q\} \right)^2}.$$

Чим більше значення  $Co^q$ , тим, у середньому, компактніше розташовані екземпляри  $q$ -го класу в просторі ознак.

Компактність розташування екземплярів  $q$ -го та  $p$ -го класів за  $i$ -ою ознаці визначимо з виразу:

$$Co_i^{q,p} = 1 - \frac{2 \sum_{s=1}^S \sum_{g=s+1}^S \left\{ (x_i^s - x_i^g)^2 \mid (y^s = q \vee y^s = p) \vee (y^g = q \vee y^g = p) \right\}}{(S^p + S^q)(S^p + S^q - 1) \left( \max_{s=1,2,\dots,S} \{x_i^s \mid (y^s = q \vee y^s = p)\} - \min_{s=1,2,\dots,S} \{x_i^s \mid (y^s = q \vee y^s = p)\} \right)^2}.$$

Чим більше значення  $Co_i^{q,p}$ , тим, у середньому, складніше відокремити  $q$ -й і  $p$ -й класи друг від друга за  $i$ -ою ознакою, але легше відокремити в сукупності  $q$ -й і  $p$ -й класи від інших класів за  $i$ -ою ознакою.

Компактність розташування екземплярів  $q$ -го і  $p$ -го класів у просторі ознак оцінимо за формулою:

$$Co(q, p) = 1 - \frac{2 \sum_{s=1}^S \sum_{g=s+1}^S \sum_{i=1}^N \left\{ (x_i^s - x_i^g)^2 \mid (y^s = q \vee y^s = p) \vee (y^g = q \vee y^g = p) \right\}}{(S^p + S^q)(S^p + S^q - 1) \sum_{i=1}^N \left( \max_{s=1,2,\dots,S} \{x_i^s \mid (y^s = q \vee y^s = p)\} - \min_{s=1,2,\dots,S} \{x_i^s \mid (y^s = q \vee y^s = p)\} \right)^2}.$$

Чим більше значення  $Co(q, p)$ , тим, у середньому, складніше відокремити  $q$ -й і  $p$ -й класи друг від друга, але легше відокремити в сукупності  $q$ -й і  $p$ -й класи від інших класів.

Усереднена компактність класів може бути розрахована за формулою:

$$\bar{Co} = \frac{1}{K} \sum_{q=1}^K Co^q.$$

Чим більше значення усередненої компактності класів, тим тісніше усередині кожного класу розташовані екземпляри, що свідчить на користь гіпотези про компактність класів.

Мінімальна компактність класів може бути оцінена за формулою:

$$Co^{\min} = \min_{q=1,2,\dots,K} (Co^q).$$

Чим більше значення мінімальної компактності класів, тим тісніше усередині кожного класу розташовані екземпляри, що свідчить на користь гіпотези про компактність класів.

*Умова компактності класу*, відповідно до [28], полягає в тому, що число екземплярів на межі класу є малим у порівнянні з загальним числом екземплярів даного класу.

Відповідно, для  $q$ -го класу визначимо показник компактності:

$$I_{comp}^q = 1 - \frac{1}{S^q} \sum_{s=1}^S \left\{ 1 - y^s = q, \min_{\substack{p=1,2,\dots,S; \\ s \neq p, y^s \neq y^p}} \left\{ \sum_{j=1}^N (x_j^s - x_j^p)^2 \right\} \leq \min_{\substack{g=1,2,\dots,S; \\ s \neq g, y^s \neq y^g}} \left\{ \sum_{j=1}^N (x_j^s - x_j^g)^2 \right\} \right\}.$$

Значення показника компактності класу буде знаходитися в діапазоні від 0 до 1. Чим більше буде значення показника компактності, тим легше буде відокремити даний клас.

Для всього набору класів визначимо усереднений показник компактності:

$$I_{comp}^{avg} = \frac{1}{K} \sum_{q=1}^K I_{comp}^q.$$

Даний показник є усередненою характеристикою роздільності класів. Чим більше буде його значення, тим компактніше розташовані класи, краще їхня роздільність і, отже, простіше задача.

Також визначимо мінімальний показник компактності класів:

$$I_{comp}^{\min} = \min_{q=1,2,\dots,K} \{I_{comp}^q\}.$$

Чим менше буде значення мінімального показника компактності класів, тим складніше буде задача поділу класів і, відповідно, тим гірше використовуваний набір ознак.

*Віддільність*  $q$ -го класу подамо як:

$$Se^q = \frac{1}{1 + \min_{p=1,2,\dots,K} Co(q,p)}.$$

Чим менше мінімальна спільна компактність  $q$ -го класу з всіма іншими класами, тим більше легко відокремити екземпляри  $q$ -го від інших класів, отже, буде більше значення віддільності  $q$ -го класу.

Віддільність класів оцінимо з виразу:

$$Se = \frac{1}{1 + \min_{\substack{q=1,2,\dots,K; \\ p=q+1,2,\dots,K}} Co(q,p)}.$$

Чим більше значення віддільності класів, тим більш компактно розташований кожний із класів і сильніше його віддільність від інших класів, що обумовлює застосування методів розпізнавання, заснованих на гіпотезі про компактність.

Спрощений показник компактності-віддільності класів визначимо за формулою:

$$SC = 1 - \exp \left( - \frac{\min_{\substack{s \neq p, \\ s=1,2,\dots,S; \\ p=s+1,\dots,S}} \left\{ \sum_{i=1}^N (x_i^s - x_i^p)^2 \mid y^s \neq y^p \right\}}{1 + \min_{\substack{s \neq p, \\ s=1,2,\dots,S; \\ p=s+1,\dots,S}} \left\{ \sum_{i=1}^N (x_i^s - x_i^p)^2 \right\}} \right).$$

Значення даного показника будуть розташовані в інтервалі від 0 до 1: чим менше значення критерію, тим тісніше розташовані (більш складно роздільні) різні класи і тем менш сконцентровані екземпляри того самого класу.

*Суперечливість* навчальної вибірки характеризує кількість однакових об'єктів вибірки, що належать до різних класів.

Суперечливість у [44] запропоновано характеризувати як (формула приведена в уточненому вигляді з підстановками):

$$Cnd = \frac{2}{S(S-1)} \sum_{s=1}^S \sum_{p=s+1}^S \frac{\sqrt{\sum_{i=1}^N (C_i^{y^s} - C_i^{y^p})^2}}{\sqrt{\sum_{i=1}^N (C_i^{y^s} - C_i^{y^p})^2} + \sqrt{\sum_{i=1}^N \frac{(x_i^s - x_i^p)^2}{S} \sum_{g=1}^S (x_i^g - \bar{x}_i)^2}},$$

$$\bar{x}_i = \frac{1}{S} \sum_{s=1}^S x_i^s, \quad C_i^q = \frac{1}{S^q} \sum_{s=1}^S \{x_i^s \mid y^s = q\}$$

Перевагою цього критерію є те, що його значення знаходяться в інтервалі від 0 до 1: чим більше значення критерію, тим більш суперечливою є вибірка.

Недоліком критерію є його залежність від гіпотези компактності образів: на практиці образи можуть бути подані множиною кластерів, а також містити взаємопроникнення. Цей критерій також є незастосовним для задач з дійсним виходом.

Відносну суперечливість навчальної вибірки будемо оцінювати за формулою:

$$Ic = \frac{1}{S(S-1)} \sum_{s=1}^S \sum_{g=s+1}^S \tau'(x, y, s, g),$$

де

$$\tau'(x, y, s, g) = \begin{cases} 1, y^s \neq y^g, \forall i = 1, 2, \dots, N : x_i^s = x_i^g; \\ 0, \text{в іншому випадку,} \end{cases}$$

або

$$\tau'(x, y, s, g) = \begin{cases} \exp\left(-\alpha \sum_{i=1}^N (x_i^s - x_i^g)^2\right), y^s \neq y^g; \\ 0, y^s = y^g, \end{cases}$$

або

$$\tau'(x, y, s, g) = \begin{cases} \exp\left(-\alpha \sum_{i=1}^N (x_i^s - x_i^g)^2\right), |y^s - y^g| > \delta; \\ 0, |y^s - y^g| \leq \delta. \end{cases}$$

Показник відносної суперечливості буде приймати значення в діапазоні від 0 до 1: чим менше буде його значення, тим менше частка однакових екземплярів, що належать до різних класів. У свою чергу, відносну несуперечність навчальної вибірки визначимо як:  $Cn = 1 - Ic$ . Показник відносної несуперечності буде приймати значення в діапазоні від 0 до 1: чим більше буде його значення, тим менше частка однакових екземплярів, що належать до різних класів.

*Розмаїтість* є однією з найважливіших характеристик систем і визначається числом різних станів системи. Відповідно до закону "необхідної розмаїтості" У. Р. Ешбі, створюючи систему, здатну справитися з вирішенням проблеми, яка має визначену, відому розмаїтість (складність), потрібно забезпечити, щоб система мала ще більшу розмаїтість (знання методів рішення), ніж розмаїтість розв'язуваної проблеми, або була здатна створити в собі цю розмаїтість (володіла би методологією, могла розробити методику, запропонувати нові методи вирішення проблеми) [86].

Для генеральної сукупності  $\langle X, Y \rangle$  при невідомих розподілах ознак умовно оцінимо граничну розмаїтість як:

$$I_{div}(X, Y) = \left( \prod_{i=1}^{N_M} \frac{(y_i^{\max*} - y_i^{\min*})}{\lambda_i^*} \right) \left( \prod_{j=1}^N \frac{(x_j^{\max*} - x_j^{\min*})}{\lambda_j^*} \right),$$

де  $\lambda_j$  – максимальна кількість десяткових розрядів, що може бути використана для фіксації значень  $j$ -ої ознаки (визначається можливостями вимірювальної апаратури й особливостями задачі);  $y_i^{\max*}, y_i^{\min*}$  – відповідно, максимальне і мінімальне можливі значення  $i$ -ої вихідної ознаки;  $x_j^{\max*}, x_j^{\min*}$  – відповідно, максимальне і мінімальне можливі значення  $j$ -ої вхідної ознаки.

Для навчальної вибірки будемо визначати абсолютну розмаїтість як:

$$I_{div}(x, y) = (N_M + N) \sum_{s=1}^S \left\{ 1 \left| \prod_{p=s+1}^S \left\{ 1 \left| \sum_{j=1}^N |x_j^s - x_j^p| = 0 \right. \right\} \right. \right\} = 0 \left. \right\},$$

де  $x \subset X, y \subset Y, \langle X, Y \rangle$  – генеральна сукупність, з якої витягається вибірка  $\langle x, y \rangle$ ,  $\lambda_j$  – кількість десяткових розрядів, використовувана для подання значень  $j$ -ої ознаки в ЕОМ (на практиці визначається типом даних і розрядною сіткою ЕОМ).

Для навчальної вибірки визначимо показник розмаїтості стосовно генеральної сукупності:

$$I_{div}(x, X, y, Y) = \frac{I_{div}(x, y)}{I_{div}(X, Y)}.$$

Чим ближче буде значення  $I_{div}(x, X, y, Y)$  до одиниці, тим більше відповідає вибірка генеральній сукупності за своїми характеристиками розмаїтості.

*Монотонність вибірки* визначається як властивість вибірки задовольняти умові:  $\forall s, p = 1, 2, \dots, S : x^s \leq x^p \rightarrow y^s < y^p$ .

Визначимо показник монотонності  $i$ -го виходу за  $j$ -им входом для навчальної вибірки  $\langle x, y \rangle$  як:

$$I_{mon}^{i,j} = \frac{1}{S(S-1)} \sum_{s=1}^S \sum_{p=1}^S \{1 | x_j^s \leq x_j^p, y_i^s \leq y_i^p, s \neq p\}.$$

Показник монотонності вибірки визначимо як:

$$I_{mon} = \frac{1}{N_M N S(S-1)} \sum_{i=1}^{N_M} \sum_{j=1}^N \sum_{s=1}^S \sum_{p=1}^S \{1 | x_j^s \leq x_j^p, y_i^s < y_i^p, s \neq p\}.$$

Показник монотонності вибірки буде дорівнювати одиниці для монотонної вибірки, і нулю – для немонотонної вибірки. Очевидно, що чим вище рівень монотонності вибірки, тим простіше здійснити поділ вибірки на класи і тим більш просту нейромережу можна обрати для вирішення задачі.

*Еластичність* функції  $y(x)$  за змінною  $x_j$ , згідно з [40], визначається як:

$$El_{x_j}(y) = \lim_{\Delta x_j \rightarrow 0} \frac{\frac{\Delta y}{y}}{\frac{\Delta x_j}{x_j}} = \left( \lim_{\Delta x_j \rightarrow 0} \frac{\Delta y}{\Delta x_j} \right) \frac{x_j}{y},$$

де  $\Delta y = \frac{y(x_j + \Delta x_j) - y(x_j)}{y(x_j)}$ ,  $x_j > 0$ ,  $y > 0$ .

Визначимо відносну оцінку еластичності навчальної вибірки  $\langle x, y \rangle$  з нормуванням для вихідної змінної  $y_i$  за вхідною змінною  $x_j$  як:

$$El_{x_j, y_i}(\langle x, y \rangle) = \frac{\sum_{s=1}^S \tilde{x}_j^s}{2S \sum_{s=1}^S \tilde{y}_i^s} \max_{\substack{s=1,2,\dots,S; \\ p=s+1,\dots,S}} \left\{ \frac{\max(\tilde{y}_i^s, \tilde{y}_i^p) - \min(\tilde{y}_i^s, \tilde{y}_i^p)}{\min(\tilde{y}_i^s, \tilde{y}_i^p) |\tilde{x}_j^s - \tilde{x}_j^p|} \left| \tilde{x}_j^s - \tilde{x}_j^p \right| > 0 \right\},$$

де

$$\tilde{x}_j^s = 1 + \frac{x_j^s - \min_{p=1,2,\dots,S}(x_j^p)}{\max_{p=1,2,\dots,S}(x_j^p) - \min_{p=1,2,\dots,S}(x_j^p)}, \quad \tilde{y}_i^s = 1 + \frac{y_i^s - \min_{p=1,2,\dots,S}(y_i^p)}{\max_{p=1,2,\dots,S}(y_i^p) - \min_{p=1,2,\dots,S}(y_i^p)}.$$

Чим більше буде показник еластичності, тим сильніше вихідна змінна реагує на зміну вхідних змінних.

*Складність навчальної вибірки* характеризує витрати ресурсів пам'яті (просторова складність) і обчислювальних ресурсів (обчислювальна складність) для обробки вибірки.

Складність навчальної вибірки  $\langle x, y \rangle$  для апроксимації функції  $y = f(x)$  у випадку, коли вихідна змінна є дійсною, може бути оцінена за допомогою константи Ліпшица [47]:

$$L(x, y) = \max_{\substack{s=1,2,\dots,S; \\ g=s+1,\dots,S}} \left\{ \frac{\sqrt{(y^s - y^g)^2}}{\sqrt{\sum_{i=1}^N (x_i^s - x_i^g)^2}} \right\}.$$

Для задач розпізнавання, коли вихідна змінна приймає дискретні значення, константа Ліпшица буде залежати в основному від знаменника. При цьому варто врахувати той факт, що номери класів у чисельнику можуть не виражати ступінь їхнього розходження. Тому визначимо складність апроксимації, модифікувавши константу Ліпшица в такий спосіб:

$$L'(x, y) = \max_{\substack{s=1,2,\dots,S; \\ g=s+1,\dots,S}} \left\{ \left( \sqrt{\sum_{i=1}^N (x_i^s - x_i^g)^2} \right)^{-0,5} \right\} = \left( \min_{\substack{s=1,2,\dots,S; \\ g=s+1,\dots,S}} \left\{ \sum_{i=1}^N (x_i^s - x_i^g)^2 \right\} \right)^{-0,5}.$$

Розглянуті показники складності вибірки сильно залежать від розмірностей вхідних і вихідної змінних та незручні у використанні при

порівнянні різних задач. Для усунення цього недоліку, а також оптимізації обчислень пропонується використовувати модифіковані показники складності навчальної вибірки (тут також забезпечується нерівність знаменника нулю):

– для задач з дійсною вихідною змінною:

$$L''(x, y) = \sqrt{\max_{\substack{s=1,2,\dots,S; \\ g=s+1,\dots,S}} \left\{ \frac{v_y (y^s - y^g)^2}{1 + \sum_{i=1}^N v_i (x_i^s - x_i^g)^2} \right\}},$$

$$v_x = \left( \frac{1}{\max_{s=1,2,\dots,S} \{x_i^s\} - \min_{s=1,2,\dots,S} \{x_i^s\}} \right)^2, \quad v_y = \left( \frac{1}{\max_{s=1,2,\dots,S} \{y^s\} - \min_{s=1,2,\dots,S} \{y^s\}} \right)^2;$$

– для задач з дискретною вихідною змінною:

$$L''(x, y) = \left( 1 + \min_{\substack{s=1,2,\dots,S; \\ g=s+1,\dots,S}} \left\{ \sum_{i=1}^N v_i (x_i^s - x_i^g)^2 \right\} \right)^{-0,5}.$$

Модифікований показник буде характеризувати відносну складність апроксимації залежності за навчальною вибіркою. При цьому його значення будуть знаходитися в діапазоні від нуля до одиниці: чим менше буде значення показника відносної складності, тим краще буде вибірка підходити для вирішення задачі.

Для показника відносної складності визначимо альтернативний йому показник відносної простоти апроксимації залежності за навчальною вибіркою як:

$$S_i = 1 - L''.$$

Значення показника відносної простоти апроксимації залежності за навчальною вибіркою будуть знаходитися в діапазоні від нуля до одиниці: чим більше буде значення показника відносної простоти, тим краще буде вибірка підходити для вирішення задачі.

*Інтегральні показники якості навчальної вибірки* можливо визначити на основі комплексу розглянутих характеристик у такий спосіб:

– критерій відбору екземплярів:

$$I_Q^{\text{екз.}} = \frac{S_{\max} \sigma_{\text{норм.}} (Nr + Ev)}{2S(1 + Rp)} \rightarrow \max,$$

де  $S_{\max}$  – максимально можливе число екземплярів у вибірці;

– критерій відбору ознак:

$$I_Q^{\text{призн.}} = \frac{N_{\max} (\bar{I}dp + \bar{I}dp + \bar{Y}dp + \bar{Y}y + \bar{I}y)(Se + Sc + \bar{C}o)I_{\text{мон.}}I_{\text{див.}}(x, X, y, Y)(I_{\text{comp}}^{\text{div.}} + I_{\text{comp}}^{\text{min}})}{36N \left( 1 + \frac{1}{6}(Ev + \bar{E}v + \bar{E}y)(Cnd + Ic)L'' \right)} \rightarrow \max,$$

де  $N_{\max}$  – максимально можливе число ознак у вибірці;

– узагальнений показник якості навчальної вибірки:

$$I_Q = \frac{\sigma_{\text{норм.}} N r R \bar{I} dp + \bar{I} dp + \bar{Y} dp + \bar{Y} dp + \bar{I} y + \bar{I} y)(Se + Sc + \bar{C} o) I_{\text{мон}} I_{\text{div}}(x, X, y, Y)}{18 + 3Dr(Ev + \bar{E}v + \bar{E}v)(Cnd + Ic)L''} \rightarrow \max.$$

Розглянуті показники, що характеризують окремі властивості навчальних вибірок, а також введені інтегральні показники можуть бути використані для аналізу властивостей навчальних вибірок, обґрунтування вибору методу і моделі нейромереж для рішення відповідної задачі, а також можуть використовуватися як цільові функції при формуванні вибірок і відборі ознак.

#### 4.9.2 Характеристики і критерії порівняння нейромоделей

Найбільше широко використовуваними критеріями порівняння нейромоделей є точність, яку характеризують помилкою навчання і помилкою розпізнавання тестових даних, час і кількість ітерацій, витрачених на процес навчання, час розпізнавання тестових даних, число шарів і нейронів мережі [42, 45].

У [47] запропонований ряд способів для оцінювання складності нейромереж. У [74, 75] запропоновано комплекс критеріїв, що характеризували складність, надмірність пам'яті, логічну прозорість і якість апроксимації нейромоделей.

Однак цей перелік критеріїв відбиває лише малу частину властивостей, які мають нейромоделі. Отже є доцільним розширити перелік розглянутих властивостей нейронних мереж.

Штучні нейронні та нейро-нечіткі мережі (розглянуті в більш загальному випадку як кібернетичні системи) характеризують такими властивостями, як: автономність, адаптивність, варіативність моделей, ієрархічність, інтерпретабельність, надійність, нелінійність, узагальнення, здатність до навчання, однорідність, паралелізм обчислень, перенавчаченість, пластичність, попередня організація в навчанні, розмаїтість, розподіленість пам'яті, робастність, симетрія, впевненість у прийнятті рішення, стійкість до шуму, функціональна блочність, чутливість, еквівалентність, еквіфінальність, еластичність, емерджентність, ефективність [1, 9, 13, 16, 31–33, 36, 38, 45, 47–49, 72, 73].

Для перелічених вище властивостей нейромереж пропонується використовувати критерії, що дозволяють кількісно оцінити для конкретної нейромоделі наявність у неї відповідного властивості. Це дозволить на практиці не тільки вирішувати задачу вибору нейромоделі, але також формувати комплексні критерії для побудови нейромоделей, що враховують множини властивостей.

Оскільки найбільш широко використовуваним типом мереж є шаруваті мережі прямого поширення сигналу [9], будемо характеризувати мережі кортежем  $\langle M, \{N_\eta\}, \{\langle \{w_j^{(\eta,i)}\}, \varphi^{(\eta,i)}, \psi^{(\eta,i)} \rangle\} \rangle$ , де  $M$  – число шарів мережі,  $N_\eta$  – кількість нейронів у  $\eta$ -му шарі мережі,  $w_j^{(\eta,i)}$  – ваговий коефіцієнт  $j$ -го входу  $i$ -го нейрона  $\eta$ -го шару,  $\varphi^{(\eta,i)}$  – дискримінантна функція  $i$ -го нейрона  $\eta$ -го шару,  $\psi^{(\eta,i)}$  – функція активації  $i$ -го нейрона  $\eta$ -го шару.

Також позначимо:  $N_n$  – число нейронів у мережі,  $\varphi(i)$ ,  $\psi(i)$  – відповідно, дискримінантна й активаційна функції  $i$ -го нейрона (при суцільній нумерації нейронів мережі),  $w_{ij}$  – вага зв'язку між  $i$ -м та  $j$ -м нейронами мережі (якщо зв'язок відсутній, то прийемо  $w_{ij} = 0$ ), де  $i, j = 1, 2, \dots, N_n$ ,  $N_w$  – число ваг у мережі,  $\psi^{(M,i)}(x^s)$  – розрахункове значення на  $i$ -му виході мережі при подачі на її входи екземпляра  $x^s$ .

#### 4.9.3 Властивості нейромоделей, обумовлені навчальною вибіркою

У групу властивостей нейромоделей, обумовлених навчальною вибіркою, входять ті властивості, значення яких можливо визначити тільки для навчених нейромереж.

*Розмаїтість* синтезованої нейромоделі *net* (для загального випадку повнозв'язної мережі з зовнішнім входом і власними зворотними зв'язками нейронів) за аналогією з раніше введеними показниками розмаїтості вибірки і генеральної сукупності, будемо оцінювати за допомогою показника граничної розмаїтості:

$$I_{div}(net) = \frac{NN_n^{\max} \sum_{i=1}^{N_n^{\max}} \vartheta_{np}(\varphi(i)) \vartheta_{np}(\psi(i))}{\text{round}\left(\frac{w_{\max} - w_{\min}}{\Delta w}\right)},$$

де  $N_n^{\max}$  – максимально можлива кількість нейронів у мережі *net*,  $\vartheta_{np}(element)$  – характеристика пластичності функціонального елемента *element* нейрона (будемо вважати  $\vartheta_{np}(element)$  рівним числу можливих станів елемента *element*: для дискримінантної функції  $i$ -го нейрона  $\vartheta_{np}(\varphi(i))$  буде дорівнювати числу можливих видів дискримінантних функцій, що можуть бути задані для даного нейрона; для функції активації  $i$ -го нейрона  $\vartheta_{np}(\psi(i))$  буде дорівнювати числу можливих видів функ-

цій активації, що можуть бути задані для даного нейрона),  $w_{\max}, w_{\min}$  – відповідно, максимальне і мінімальне можливі значення ваг мережі,  $\Delta w$  – мінімально можлива зміна ваги з урахуванням розрядності обчислювальної сітки ЕОМ,  $round$  – функція округлення до найближчого цілого числа.

На основі введених абсолютних показників граничної розмаїтості визначимо відносні показники.

Для нейронної мережі  $net$  визначимо показник розмаїтості стосовно навчальної вибірки:

$$I_{div}(net, \langle x, y \rangle) = \frac{I_{div}(net)}{I_{div}(x, y)}.$$

Чим більше буде значення  $I_{div}(net, \langle x, y \rangle)$ , тим більше у мережі буде можливостей для апроксимації вибірки. Але якщо значення  $I_{div}(net, \langle x, y \rangle)$  буде більше одиниці, то мережа є надлишковою, а якщо близьким до нуля, то або добре апроксимуючою вибірку (при прийнятному рівні помилки  $E$ ), або такою, що має дефіцит структурних елементів і пам'яті.

Для нейронної мережі  $net$  визначимо показник розмаїтості стосовно генеральної сукупності:

$$I_{div}(net, X, Y) = \frac{I_{div}(net)}{I_{div}(X, Y)}.$$

Якщо значення  $I_{div}(net, X, Y)$  буде більше одиниці чи близьким до неї, то мережа є надмірно надлишковою для вирішення задачі.

*Перенавченість* нейромоделі  $net$ , що розпізнає, на навчальній вибірці  $x$  щодо тестової вибірки  $x_{\text{тест.}}$ ,  $x_{\text{тест.}} \neq x$ , визначають як:

$$\delta_{net}(x, x_{\text{тест.}}) = v(net, x_{\text{тест.}}) - v(net, x),$$

$$n(net, x) = \frac{1}{S} \sum_{x=1}^S E(net, x^s),$$

де  $n(net, x)$  – частота помилок моделі  $net$  на вибірці  $x$  (у задачах класифікації:  $E(net, x^s) = \{1 \mid \psi^{(M,1)}(x^s) \neq y^s\}$ , у задачах оцінювання:

$$E(net, x^s) = \{1 \mid \delta \leq |\psi^{(M,1)}(x^s) - y^s|\}, \delta - \text{поріг помилки}.$$

Оскільки нейромоделі можуть на виході визначати значення не однієї, а декількох змінних, і поріг помилки для екземпляра на практиці не завжди може бути заданий, а також для більшої універсальності й однаковості при вирішенні різних задач визначимо:

$$E(net, x^s) = 1 - \exp \left( - \frac{1}{N_M} \sum_{i=1}^{N_M} \left( \frac{\Psi^{(M,i)}(x^s) - y_i^s}{\max_{s=1,2,\dots,S} (y_i^s) - \min_{s=1,2,\dots,S} (y_i^s)} \right)^2 \right).$$

Еластичність нейромоделі, подібно раніше введеному показнику еластичності для навчальної вибірки, будемо оцінювати за допомогою показника відносної еластичності за змінною  $x_j$  апроксимуючої функції  $y_i = net(x)$ , реалізованою нейромережею  $net$  на  $i$ -му виході  $y_i$ , навченою на основі навчальної вибірки  $\langle x, y \rangle$ , як:

$$El_{x_j}(net, y_i) = \frac{1}{2S} \sum_{s=1}^S \left( \frac{\tilde{x}_j^s \left( \tilde{\Psi}_{(\tilde{x}_j^s + \Delta x_j)}^{(M,i)} - \tilde{\Psi}_{(\tilde{x}_j^s)}^{(M,i)} \right)}{\Delta x_j \left( \tilde{\Psi}_{(\tilde{x}_j^s)}^{(M,i)} \right)^2} \right),$$

де  $\tilde{\Psi}_{(\tilde{x}_j^s)}^{(M,i)}$  – розрахункове значення на  $i$ -му виході нейромережі  $net$  при подачі на її входи нормованих значень ознак  $s$ -го екземпляра;  $\tilde{\Psi}_{(\tilde{x}_j^s + \Delta x_j)}^{(M,i)}$  – розрахункове значення на  $i$ -м виході нейромережі  $net$  при подачі на її входи нормованих значень ознак  $s$ -го екземпляра, а на  $j$ -й вхід – скорегованого на  $\Delta x_j$  нормованого значення  $j$ -ої ознаки  $s$ -го екземпляра.

Чутливість нейронної мережі до вхідних сигналів характеризують за допомогою розрахунків часткових похідних функції помилки мережі. Однак такий підхід виявляється обчислювально складним.

З метою спрощення обчислень визначимо усереднений нормований показник чутливості  $i$ -го виходу нейромережі до зміни вхідного сигналу як:

$$I_{tot}^i = \frac{1}{SN(y_i^{\max} - y_i^{\min})} \sum_{s=1}^S \sum_{j=1}^N \max(\tau_1, \tau_2),$$

$$\tau_1 = \left( \Psi^{(M,i)} \left( \begin{array}{l} x^{s*} \\ x_b^{s*} = x_b^s, b \neq j, b = 1, 2, \dots, N, \\ x_j^{s*} = x_j^s + \Delta_{\min} \end{array} \right) - y_i^s \right)^2,$$

$$\tau_2 = \left( \Psi^{(M,i)} \left( \begin{array}{l} x^{s*} \\ x_b^{s*} = x_b^s, b \neq j, b = 1, 2, \dots, N, \\ x_j^{s*} = x_j^s - \Delta_{\min} \end{array} \right) - y_i^s \right)^2,$$

де  $y_i^{\max}, y_i^{\min}$  – відповідно, максимальне і мінімальне межові значення  $i$ -ої вихідної ознаки.

Усереднений показник чутливості нейромоделі до зміни вхідного сигналу визначимо як:

$$I_{tol} = \frac{1}{N_M} \sum_{i=1}^{N_M} I_{tol}^i.$$

Значення запропонованих показників чутливості будуть знаходитися в інтервалі  $[0, 1]$ . Чим вище буде значення показника чутливості, тим сильніше мережа реагує на зміни у вхідному сигналі, тим більше категоризаційні можливості мережі. Однак занадто висока чутливість може свідчити про слабку стійкість мережі до шумів і завад у вхідному сигналі.

*Стійкість до шуму* – властивість нейромереж забезпечувати правильну реакцію на вхідний сигнал, що містить шум.

Визначимо показник стійкості навченої нейромережі до адитивного шуму у вхідному сигналі за  $j$ -м входом як:

$$I_{tolj}^{\ell} = \exp \left( -\frac{1}{2} \sum_{s=1}^S \left( (\psi^{(M,1)}(x^s) - \psi^{(M,1)}(x_{(j^+)}^s))^2 + (\psi^{(M,1)}(x_j^s) - \psi^{(M,1)}(x_{(j^-)}^s))^2 \right) \right),$$

$${}_{(j^+)}x_g^s = \begin{cases} x_g^s, & g \neq j, g = 1, 2, \dots, N; \\ x_g^s + \ell \left( \max_{p=1, 2, \dots, S} (x_g^p) - \min_{p=1, 2, \dots, S} (x_g^p) \right), & g = j, \end{cases}$$

$${}_{(j^-)}x_g^s = \begin{cases} x_g^s, & g \neq j, g = 1, 2, \dots, N; \\ x_g^s - \ell \left( \max_{p=1, 2, \dots, S} (x_g^p) - \min_{p=1, 2, \dots, S} (x_g^p) \right), & g = j, \end{cases}$$

$$x_{j-}^s = x_j^s - \ell \left( \max_{p=1, 2, \dots, S} (x_j^p) - \min_{p=1, 2, \dots, S} (x_j^p) \right),$$

де  $\ell$  – заданий рівень шуму,  $0 < \ell < 1$ .

Показник стійкості навченої нейромережі до випадкового адитивного шуму у вхідному сигналі визначимо як:

$$I_{tol}^{\ell} = \exp \left( -\frac{1}{2} \sum_{s=1}^S \left( (\psi^{(M,1)}(x^s) - \psi^{(M,1)}(x^s))_+^2 + (\psi^{(M,1)}(x_j^s) - \psi^{(M,1)}(x_j^s))_-^2 \right) \right),$$

$${}_+x_j^s = x_j^s + \ell \left( \max_{p=1, 2, \dots, S} (x_j^p) - \min_{p=1, 2, \dots, S} (x_j^p) \right), \quad {}_-x_j^s = x_j^s - \ell \left( \max_{p=1, 2, \dots, S} (x_j^p) - \min_{p=1, 2, \dots, S} (x_j^p) \right),$$

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

де  $\ell$  – заданий рівень шуму,  $0 < \ell < 1$ . Для автоматизації процесу задавання  $\ell$  пропонується використовувати вираз:

$$\ell = \min_{j=1, 2, \dots, N} \left\{ \begin{array}{l} \min_{\substack{s=1, 2, \dots, S; \\ p=s+1, \dots, S}} \left\{ x_j^s - x_j^p \right\} \Big| y^s \neq y^p \\ \max_{p=1, 2, \dots, S} (x_j^p) - \min_{p=1, 2, \dots, S} (x_j^p) \end{array} \right\},$$

*Нелінійність класифікатора* щодо вибірки  $x$  визначається як:

$$I_{nl}(net) = \frac{2}{S(S-1)} \sum_{s=1}^S \sum_{p=s+1}^S \left( \frac{\sum_{\ell=0}^S \left\{ \left| \Psi^{(M,1)} \left( \frac{\ell x^p}{S} + \left( 1 - \frac{\ell}{S} \right) x^s \right) \right| \neq \Psi^{(M,1)}(x^p) \right\}}{\sqrt{\sum_{j=1}^N (x_j^s - x_j^p)^2}} \right)$$

Показник нелінійності класифікатора буде приймати значення в діапазоні [0, 1]: чим більше буде його значення, тим більш нелінійною є модель. Недоліком даного показника є його застосовність тільки для нейромереж з одним виходом, причому винятково для задач класифікації.

Для нейромереж прямого поширення з декількома виходами для задач оцінювання визначимо показник нелінійності як:

$$I_{nl}(net) = \frac{2}{S(S-1)} \max_{i=1,2,\dots,N_M} \left\{ \sum_{s=1}^S \sum_{p=s+1}^S \left( \frac{\sum_{\ell=0}^S \left\{ \Psi^{(M,i)}(\ell x^p S^{-1} + (1-\ell S^{-1})x^s) - \Psi^{(M,i)}(x^p) \right\}}{\Psi_{\max}^{(M,i)} - \Psi_{\min}^{(M,i)}} \right)}{\sqrt{\sum_{j=1}^N (x_j^s - x_j^p)^2}} \right\}$$

Нелінійність навчальної вибірки  $\langle x, y \rangle$  визначимо як:

$$I_{nl}(\langle x, y \rangle) = \frac{2}{S(S-1)} \sum_{s=1}^S \sum_{p=s+1}^S \left( \exp \left( - \max_{i=1,2,\dots,N_M} \left\{ \frac{\sqrt{(y_i^s - y_i^p)^2}}{\max_{g=1,2,\dots,S} \{y_i^g\} - \min_{g=1,2,\dots,S} \{y_i^g\}} \right\} \right) \exp \left( \frac{1}{N} \sum_{j=1}^N \frac{(x_j^s - x_j^p)^2}{\left( \max_{g=1,2,\dots,S} \{x_j^g\} - \min_{g=1,2,\dots,S} \{x_j^g\} \right)^2} - 1 \right) \right)$$

На основі введених показників визначимо показник відповідності нелінійностей вибірки і нейромоделі:

$$\tilde{I}_{nl} = \frac{I_{nl}(\langle x, y \rangle)}{I_{nl}(net)}$$

Якщо даний показник буде дорівнювати одиниці, то можна вважати, що мережа відповідає вибірці за складністю. Якщо показник буде менше одиниці, то чим менше буде значення показника, тим більше буде виявлятися ефект перенавчання, а також це буде свідчити про можливу надмірність мережі. Якщо значення показника перевищить одиницю, то мережа є недостатньою для якісної апроксимації (потрібно донавчання або зміна структури мережі).

*Узагальнення* – здатність нейромережі після навчання на основі навчальної вибірки видавати відповіді для екземплярів тестової вибірки, подібних до екземплярів навчальної вибірки, але таких, що не входили до неї. Інакше, узагальнення – здатність мережі інтегрувати окремі дані для визначення закономірностей і пролонгації результатів, що дозволяє після навчання на одних даних застосовувати отримані знання для інших даних.

Визначимо показник узагальнення нейромоделі для навчальної  $\langle x, y \rangle$  і тестової  $\langle x^*, y^* \rangle$  вибірок як:

$$I_G = 1 - \exp \left( - \frac{1}{S_{\text{тест.}}} \sum_{p=1}^{S_{\text{тест.}}} \left\{ \frac{N_M \sum_{i=1}^{N_M} (\Psi^{(M,i)}(x^{p*}) - y_i^{p*})^2 \sum_{j=1}^N (x_j^s - x_j^{p*})^2}{N \sum_{i=1}^{N_M} (y_i^s - y_i^{p*})^2} \right\} \sum_{i=1}^{N_M} (y_i^s - y_i^{p*})^2 > 0 \right),$$

$$s = \arg \min_{t=1,2,\dots,S} \sum_{j=1}^N (x_j^t - x_j^{p*})^2,$$

де  $x^* = \{x^{p*}\}$ ,  $x^{p*} = \{x_j^{p*}\}$ ,  $y^* = \{y^{p*}\}$ ,  $y^{p*} = \{y_i^{p*}\}$ ,  $j = 1, 2, \dots, N$ ,  $i = 1, 2, \dots, N_M$ ,  $p = 1, 2, \dots, S_{\text{тест.}}$ ,  $S_{\text{тест.}}$  – кількість екземплярів у тестовій вибірці.

Показник узагальнення буде приймати значення в діапазоні від 0 до 1 і буде тим більше, чим менше помилка мережі при розпізнаванні екземпляра, а відмінність розпізнаваного екземпляра від найближчого за властивостями екземпляра навчальної вибірки – більше.

Коефіцієнт узагальнення навченої мережі визначимо як:

$$I_{\text{gen}} = \frac{NS}{N_w N_H} I_h \exp(-(E_{\text{об.}} - E_{\text{тест.}})^2),$$

де  $E_{\text{об.}}$ ,  $E_{\text{тест.}}$  – відповідно, помилка мережі для навчальної і тестової вибірок.

Якщо коефіцієнт узагальнення буде істотно більше одиниці, то мережа виявляє великі узагальнюючі здібності, якщо ж коефіцієнт узагальнення істотно менше одиниці, то мережа не виявляє узагальнюючих властивостей.

*Робастність* – властивість нейромережі надійно вирішувати задачу при одержанні неповних та / або ушкоджених даних. Крім того, результати повинні бути несуперечливими, навіть якщо деяка частина мережі ушкоджена.

Робастність нейромережі стосовно вхідних сигналів визначимо як нормовану найменшу зміну у вхідному сигналі, що приводить до істотного збільшення помилки мережі:

$$I_{Rb}^x = \min_{j=1,2,\dots,N} \left\{ \frac{\min(\tau_1, \tau_2) - \min_{s=1,2,\dots,S} \{x_j^s\}}{\max_{s=1,2,\dots,S} \{x_j^s\} - \min_{s=1,2,\dots,S} \{x_j^s\}} \right\},$$

$$\tau_1 = \frac{\min}{SN_M \sum_{s=1}^S \sum_{i=1}^{N_M} \left( \Psi^{(M,i)} \left( x^{s*} \left| \begin{array}{l} x_{i_0}^s = x_{i_0}^s, \forall b \neq j, b=1,2,\dots,N \\ x_j^s = x_j^s + \Delta x_j \end{array} \right. \right) - y_i^s \right)^2 > \epsilon} \{ \Delta x_j \},$$

$$\tau_2 = \frac{\min}{SN_M \sum_{s=1}^S \sum_{i=1}^{N_M} \left( \Psi^{(M,i)} \left( x^{s*} \left| \begin{array}{l} x_{i_0}^s = x_{i_0}^s, \forall b \neq j, b=1,2,\dots,N \\ x_j^s = x_j^s - \Delta x_j \end{array} \right. \right) - y_i^s \right)^2 > \epsilon} \{ \Delta x_j \},$$

де  $\Delta x_j = \min_{s=1,2,\dots,S} \{x_j^s\} + \delta_x \left( \max_{s=1,2,\dots,S} \{x_j^s\} - \min_{s=1,2,\dots,S} \{x_j^s\} \right), \dots, \max_{s=1,2,\dots,S} \{x_j^s\} -$   
 $-\delta_x \left( \max_{s=1,2,\dots,S} \{x_j^s\} - \min_{s=1,2,\dots,S} \{x_j^s\} \right), \delta_x \in (0, 1)$  – константа, що регулює  
 точність визначення показника робастності за входами мережі.

Робастність нейромережі стосовно її ваг визначимо як нормовану найменшу зміну значень ваг, що приводить до істотного збільшення помилки мережі:

$$I_{Rb}^w = \min_{j=1, 2, \dots, N_w} \left\{ \frac{\min(\tau_1, \tau_2) - w_{\min}}{w_{\max} - w_{\min}} \right\},$$

$$\tau_1 = \frac{1}{SN_M} \sum_{s=1}^S \sum_{i=1}^{N_M} \left\{ \min_{w_j = w_j + \Delta w} \left( \psi^{(M,i)}(x^s | w_j = w_j + \Delta w) - y_i^s \right)^2 \right\}_{>\varepsilon}$$

$$\tau_2 = \frac{1}{SN_M} \sum_{s=1}^S \sum_{i=1}^{N_M} \left\{ \min_{w_j = w_j - \Delta w} \left( \psi^{(M,i)}(x^s | w_j = w_j - \Delta w) - y_i^s \right)^2 \right\}_{>\varepsilon}$$

де  $\Delta w_j = w_{\min} + \delta_w (w_{\max} - w_{\min}), \dots, w_{\max} - \delta_w (w_{\max} - w_{\min}), \delta_w \in (0, 1)$  – константа, що регулює точність визначення показника робастності за вагами мережі.

Інтегральну оцінку робастності нейромоделі визначимо як:

$$I_{Rb} = I_{Rb}^x \cdot I_{Rb}^w$$

Інтегральний показник робастності нейромоделі буде приймати значення в діапазоні  $[0, 1]$ . Чим ближче буде значення інтегрального показника робастності до нуля, тим менше робастність мережі, тим чутливіша мережа до зміни вхідних сигналів або значень ваг. Чим ближче буде значення інтегрального показника робастності до одиниці, тим більше робастність мережі, тим менш чутливою буде мережа до зміни вхідних сигналів або значень ваг.

*Упевненість нейромоделі* в прийнятому рішенні в задачах розпізнавання образів визначимо як суб'єктивну оцінку нейронною мережею прийнятого рішення:

– щодо значення на  $i$ -му виході мережі для екземпляра  $x^s$ , поданого на її входи:

$$I_{cert}^i(x^s) = \min_{\substack{j=1,2,\dots,N_M, \\ j \neq i}} \left\{ \frac{\psi^{(M,i)}(x^s) - \psi_{\min}^{(M,i)}}{\psi_{\max}^{(M,i)} - \psi_{\min}^{(M,i)}}, \left( 1 - \frac{\psi^{(M,j)}(x^s) - \psi_{\min}^{(M,j)}}{\psi_{\max}^{(M,j)} - \psi_{\min}^{(M,j)}} \right) \right\},$$

де  $\psi_{\max}^{(M,i)}, \psi_{\min}^{(M,i)}$  – відповідно, максимальне і мінімальне граничні значення функції активації  $i$ -го нейрона останнього шару нейромережі;

– усереднену для вибірки  $x$ :

$$I_{cert}(x) = \frac{1}{SN_M} \sum_{s=1}^S \sum_{i=1}^{N_M} I_{cert}^i(x^s).$$

Показники суб'єктивної впевненості будуть приймати значення в інтервалі  $[0, 1]$ : чим вище буде значення показника, тим ближче по властивостях розпізнаваних екземпляр до сформованих мережею еталонам і тем більше мережа упевнена в прийнятому рішенні.

*Еквівалентність*: дві нейромережі еквівалентні, якщо вони мають однакові набори відповідей (однаково реагують на ті самі вхідні стимули).

Коефіцієнт еквівалентності навчених нейромоделей  $net_1$  та  $net_2$  для вибірки  $\langle x, y \rangle$  визначимо як:

$$I_{eq}(net_1, net_2) = \exp\left(-\frac{1}{SN_M} \sum_{s=1}^S \sum_{i=1}^{N_M} (\psi_1^{(M,i)}(x^s) - \psi_2^{(M,i)}(x^s))^2\right),$$

де  $\psi_k^{(N_M,i)}(x^s)$  – значення на виході  $i$ -го нейрона останнього шару  $k$ -ої мережі при подачі на входи мережі екземпляра  $x^s$ .

Значення коефіцієнта еквівалентності будуть знаходитися в діапазоні від 0 до 1: чим ближче будуть відповіді мереж при однакових вхідних впливах, тим більше буде значення коефіцієнта еквівалентності.

*Адаптивність* – властивість структур динамічно і самостійно змінити свою поведінку у відповідь на вхідний стимул.

Стосовно до нейромереж адаптивність визначається, насамперед, пластичністю – пластичність визначає ресурси для адаптації: чим більше пластичність, тим більше адаптивні властивості мережі. Проте, пластичність – це необхідна, але недостатня передумова адаптивності. Поряд із пластичністю на адаптивні властивості нейромоделі впливає чутливість нейромоделі, що визначає силу реакції мережі на мінімальну зміну значень її параметрів.

Таким чином, виділивши найважливіші фактори, що визначають адаптивні властивості нейромереж, визначимо показник адаптивності нейромоделі як:

$$I_{adapt} = I_{pl} \cdot I_{tol},$$

де  $I_{pl}$  – відносний коефіцієнт пластичності мережі,  $I_{tol}$  – показник чутливості нейромоделі до зміни вхідного сигналу.

Визначимо  $I_{pl}$  як:

$$I_{pl} = \frac{\left( \sum_{i=1}^{N_n} \vartheta_{np}(\varphi(i)) \vartheta_{np}(\psi(i)) \right) \sum_{i=1}^{N_n} \sum_{j=1}^{N_n} \vartheta_{sp}(i, j)}{N_n^{\max} \vartheta_{np}^{\max}(\varphi) \vartheta_{np}^{\max}(\psi) N_n^2 \text{round}\left(\frac{w_{\max} - w_{\min}}{\Delta w}\right)},$$

де  $N_{\text{н}}^{\max}$  – максимально можлива кількість нейронів,  $\mathfrak{Q}_{np}(element)$  – характеристика пластичності функціонального елемента  $element$  нейрона (будемо вважати  $\mathfrak{Q}_{np}(element)$  рівним числу можливих станів елемента  $element$ : для дискримінантної функції  $i$ -го нейрона  $\mathfrak{Q}_{np}(\varphi(i))$  буде дорівнювати числу можливих видів дискримінантних функцій, що можуть бути задані для даного нейрона; для функції активації  $i$ -го нейрона  $\mathfrak{Q}_{np}(\psi(i))$  буде дорівнювати числу можливих видів функцій активації, що можуть бути задані для даного нейрона),  $\mathfrak{Q}_{np}^{\max}(\varphi), \mathfrak{Q}_{np}^{\max}(\psi)$  – відповідно, максимальні характеристики пластичності дискримінантних і активаційних функцій серед усіх нейронів мережі,  $w_{\max}, w_{\min}$  – відповідно, максимальне і мінімальне можливі значення ваг мережі,  $\Delta w$  – мінімально можлива зміна ваги з урахуванням розміру розрядної сітки ЕОМ,  $round$  – функція округлення до найближчого цілого числа,  $\mathfrak{Q}_{sp}(i, j)$  – характеристика пластичності зв'язку від  $i$ -го нейрона до  $j$ -го:  $\mathfrak{Q}_{sp}(i, j) = 0$ , якщо зв'язок не може існувати, або зв'язок жорстко заданий і його вага не може змінюватися; в інших випадках:

$$\mathfrak{Q}_{sp}(i, j) = \text{round} \left( \frac{w_{i,j}^{\max} - w_{i,j}^{\min}}{\Delta w_{i,j}} \right),$$

де  $w_{i,j}^{\max}, w_{i,j}^{\min}$  – відповідно, максимальне і мінімальне можливі значення ваги зв'язку від  $i$ -го нейрона до  $j$ -го,  $\Delta w_{i,j}$  – мінімально можлива зміна ваги з урахуванням розміру розрядної сітки ЕОМ.

*Еквіфінальність* – закономірність функціонування і розвитку системи, що характеризує її граничні можливості.

Визначимо відносну еквіфінальність нейромоделі  $net$  як

$$I_{eqf}(net, \langle x, y \rangle) = \frac{NN_w}{N_{\text{н}}^{\max} (N_{\text{н}}^{\max 2} + N)} \exp \left( -\frac{1}{SN_M} \sum_{s=1}^S \sum_{i=1}^{N_M} (\psi^{(M,i)}(x^s) - y_i^s)^2 \right).$$

Відносна еквіфінальність буде приймати найбільше значення (у межі 1) для тих мереж, що досягли максимально можливих розмірів і звязності у процесі синтезу структури, а також найменшої помилки (у межі 0) у процесі навчання.

*Здатність до навчання* – властивість нейромереж удосконалювати свою роботу (навчатися, адаптуватися), використовуючи приклади для настроювання на вирішення визначеної задачі.

Визначимо показник здатності до навчання мережі як:

$$I_{lr} = \frac{I_{pl}(1-L)L_{net}}{NS},$$

де  $L_{net}$  – константа Ліпшица (складність) нейронної мережі *net* [47],  
 $L$  – модифікована константа Ліпшица для навчальної вибірки:

– для задач з дійсною вихідною змінною:

$$L(x, y) = \sqrt{\max_{\substack{s=1,2,\dots,S; \\ g=s+1,\dots,S}} \left\{ \frac{v_y (y^s - y^g)^2}{1 + \sum_{i=1}^N v_i (x_i^s - x_i^g)^2} \right\}},$$

$$v_i = \left( \frac{1}{\max_{s=1,2,\dots,S} \{x_i^s\} - \min_{s=1,2,\dots,S} \{x_i^s\}} \right)^2, \quad v_y = \left( \frac{1}{\max_{s=1,2,\dots,S} \{y^s\} - \min_{s=1,2,\dots,S} \{y^s\}} \right)^2;$$

– для задач з дискретною вихідною змінною:

$$L(x, y) = \left( 1 + \min_{\substack{s=1,2,\dots,S; \\ g=s+1,\dots,S}} \left\{ \sum_{i=1}^N v_i (x_i^s - x_i^g)^2 \right\} \right)^{-0,5}.$$

Чим більше буде значення показника здатності до навчання, тим більші можливості має нейронна мережа *net* для вирішення задачі апроксимації таблично заданої залежності  $y = f(x)$ .

*Надійність* і *стійкість* мережі до відмов окремих елементів, що її складають, виявляється в тому, що відмова одного чи декількох нейроелементів мережі не приводить до відмови всієї мережі та не може істотно впливати на роботу мережі в цілому.

Визначимо показники надійності ваг мережі щодо навчальної вибірки як:

$$I_{wre}^{avg} = 1 - \frac{\sum_{\eta=1}^M \sum_{i=1}^{N_{\eta}} \sum_{j=1}^{N_{\eta-1}} |E(w) - E(w | w_j^{(\eta,i)} = 0)|}{N_w \sum_{i=1}^{N_M} |\Psi_{\max}^{(M,i)} - \Psi_{\min}^{(M,i)}|}, \quad I_{wre}^{max} = 1 - \frac{\max_{\substack{\eta=1,\dots,M; \\ i=1,\dots,N_{\eta}; \\ j=1,\dots,N_{\eta-1}}} |E(w) - E(w | w_j^{(\eta,i)} = 0)|}{\max_{i=1,\dots,N_M} |\Psi_{\max}^{(M,i)} - \Psi_{\min}^{(M,i)}|},$$

де  $E(w) = \sum_{s=1}^S \sum_{i=1}^{N_M} (y_i^s - \Psi_s^{(M,i)})^2$  – помилка мережі при наборі вагових коефі-

цієнтів  $w$ ,  $\Psi_s^{(M,i)}$  – значення на  $i$ -му виході мережі для  $s$ -го екземпляра.

Визначимо показники надійності нейронів мережі щодо навчальної вибірки як:

$$I_{н.ге}^{avg} = 1 - \frac{\sum_{\eta=1}^M \sum_{i=1}^{N_{\eta}} \sum_{j=1}^{N_{\eta-1}} |E(w) - E(w | \forall p = 1, 2, \dots, N_{\eta+1} : w_i^{(\eta+1,p)} = 0)|}{N_w \sum_{i=1}^{N_M} |\Psi_{\max}^{(M,i)} - \Psi_{\min}^{(M,i)}|}$$

та

$$I_{н.ге}^{\max} = 1 - \frac{\max_{\substack{\eta=1, \dots, M; \\ i=1, \dots, N_{\eta}; \\ j=1, \dots, N_{\eta-1}}} |E(w) - E(w | \forall p = 1, 2, \dots, N_{\eta+1} : w_i^{(\eta+1,p)} = 0)|}{\max_{i=1, \dots, N_M} |\Psi_{\max}^{(M,i)} - \Psi_{\min}^{(M,i)}|}.$$

*Розподіленість пам'яті* – інформація в мережі зберігається за багатьма адресами, розподіленим образом, так що кожен елемент даних подається шаблоном активності, розподіленим за багатьма обчислювальними елементами, і кожен обчислювальний елемент бере участь у поданні багатьох різних елементів даних.

Розподіленість пам'яті мережі уявимо як усереднену частку ваг, що істотно впливають на вихід мережі, у загальному числі ваг для навчальної вибірки:

$$I_{md} = \frac{\sum_{s=1}^S \sum_{\eta=1}^M \sum_{i=1}^{N_{\eta}} \sum_{j=1}^{N_{\eta-1}} (1 - \exp(-(E(w, x^s) - E(w, x^s | w_j^{(\eta,i)} = 0))^2))}{SN_w}.$$

#### 4.9.4 Властивості структури і параметрів нейромоделей, що не залежать явно від навчальної вибірки

Дана група властивостей нейромоделей включає ті властивості, що хоча і можуть залежати від навчальної вибірки, проте, можуть бути оцінені і для ненавчених нейромереж. У цій групі властивостей можна виділити властивості структури, а також властивості параметрів нейромоделей.

До властивостей структури нейромоделей варто віднести пластичність, однорідність, паралелізм, ієрархічність, попередню організацію в навчанні, функціональну блочність, варіативність автономність, симетрію моделей.

До властивостей параметрів нейромоделей, що налагоджуються, можна віднести пластичність, попередню організацію в навчанні, інтерпретабельність, варіативність, емерджентність, автономність, симетрію нейромоделей.

*Пластичність* – обумовлює складність поведінки нейромережі, що розглядається як результат взаємодії багатьох елементів, кожний з яких обмежує дія інших і сам обмежується іншими на шляху до формування

глобальної поведінки, що спостерігається. Розрізняють нейронну пластичність (як пластичні елементи розглядаються нейрони), а також синаптичну пластичність (модифікація сили синаптичного зв'язку між нейронами).

Відносний коефіцієнт нейронної пластичності задамо як:

$$I_{np} = \frac{\sum_{i=1}^{N_n} \mathfrak{G}_{np}(\varphi(i)) \mathfrak{G}_{np}(\psi)}{N_{n.}^{\max} \mathfrak{G}_{np}^{\max}(\varphi) \mathfrak{G}_{np}^{\max}(\psi)},$$

де  $N_{n.}^{\max}$  – максимально можлива кількість нейронів,  $\mathfrak{G}_{np}(element)$  – характеристика пластичності функціонального елемента *element* нейрона (будемо брати  $\mathfrak{G}_{np}(element)$  рівним числу можливих станів елемента *element*: для дискримінантної функції *i*-го нейрона  $\mathfrak{G}_{np}(\varphi(i))$  буде дорівнювати числу можливих видів дискримінантних функцій, що можуть бути задані для даного нейрона; для функції активації *i*-го нейрона  $\mathfrak{G}_{np}(\psi(i))$  буде дорівнювати числу можливих видів функцій активації, що можуть бути задані для даного нейрона),  $\mathfrak{G}_{np}^{\max}(\varphi)$ ,  $\mathfrak{G}_{np}^{\max}(\psi)$  – відповідно, максимальні характеристики пластичності дискримінантних і активаційних функцій серед усіх нейронів мережі.

Відносний коефіцієнт нейронної пластичності мережі буде приймати значення в діапазоні від 0 до 1: чим більше буде його значення, тим вище рівень нейронної пластичності мережі.

Відносний коефіцієнт синаптичної пластичності мережі визначимо як:

$$I_{sp} = \frac{\sum_{i=1}^{N_n} \sum_{j=1}^{N_n} \mathfrak{G}_{sp}(i, j)}{N_{n.} \cdot 2 \cdot \text{round}\left(\frac{w_{\max} - w_{\min}}{\Delta w}\right)},$$

де  $w_{\max}$ ,  $w_{\min}$  – відповідно, максимальне і мінімальне можливі значення ваг мережі,  $\Delta w$  – мінімально можлива зміна ваги з урахуванням розрядності обчислювальної сітки ЕОМ, *round* – функція округлення до найближчого цілого числа,  $\mathfrak{G}_{sp}(i, j)$  – характеристика пластичності зв'язку від *i*-го нейрона до *j*-му ( $\mathfrak{G}_{sp}(i, j) = 0$ , якщо зв'язок не може існувати, або зв'язок жорстко заданий і її вага не може змінюватися; в інших випадках:

$$\mathfrak{G}_{sp}(i, j) = \text{round}\left(\frac{w_{i,j}^{\max} - w_{i,j}^{\min}}{\Delta w_{i,j}}\right),$$

де  $w_{i,j}^{\max}$ ,  $w_{i,j}^{\min}$  – відповідно, максимальне і мінімальне можливі значення ваги зв'язку від  $i$ -го нейрона до  $j$ -го,  $\Delta w_{i,j}$  – мінімально можлива зміна ваги з урахуванням розміру розрядної сітки ЕОМ.

Відносний коефіцієнт синаптичної пластичності мережі буде приймати значення в діапазоні від 0 до 1: чим більше буде його значення, тим вище рівень синаптичної пластичності мережі при фіксованій кількості нейронів.

Відносний коефіцієнт пластичності мережі визначимо на основі уведених відносних коефіцієнтів нейронної і синаптичної пластичностей:

$$I_{pl} = I_{np} I_{sp}.$$

Відносний коефіцієнт пластичності мережі буде приймати значення в діапазоні від 0 до 1: чим більше буде його значення, тим вище рівень пластичності мережі і, отже, мережа має великі адаптаційні здібності.

*Однорідність* нейроелементів мережі полягає в тому, що нейромережі будуються з множини простих уніфікованих типових елементів (нейронів), що виконують елементарні дії (множення, додавання, обчислення найпростішої нелінійної функції) і з'єднаних між собою різними зв'язками.

Однорідність дискримінантних функцій будемо оцінювати як:

$$I_{hd} = \frac{2 \sum_{i=1}^{N_n} \sum_{j=i}^{N_n} \{1 | \varphi(i) \equiv \varphi(j)\}}{N_n (N_n - 1)}.$$

Однорідність функцій активації пропонується визначати за формулою:

$$I_{ha} = \frac{2 \sum_{i=1}^{N_n} \sum_{j=i}^{N_n} \{1 | \psi(i) \equiv \psi(j)\}}{N_n (N_n - 1)}.$$

Однорідність нейронів мережі визначимо як:

$$I_{hn} = \frac{2 \sum_{i=1}^{N_n} \sum_{j=i}^{N_n} \{1 | (\psi(i) \equiv \psi(j)) \wedge (\varphi(i) \equiv \varphi(j))\}}{N_n (N_n - 1)}.$$

Показники однорідності будуть змінюватися від 0 до 1: чим більше будуть їхні значення, тим більш однорідні відповідні елементи мережі.

*Паралельна архітектура та розподіленість обчислень* – обробка інформації у нейромережах виконується декількома процесорними еле-

ментами, у той час, як у звичайних комп'ютерах, що мають тільки один центральний процесор, інформація обробляється послідовно, по кроках.

Паралелізм обчислень мережі прямого поширення без латеральних зв'язків будемо характеризувати за допомогою критерію:

$$I_{par} = \frac{\sum_{\eta=1}^M \frac{1}{\eta} \sum_{i=1}^{N_{\eta}} (N_{\eta-1} + 3)}{N_w + 2N_h}.$$

*Ієрархічна організація структури*, цілісність і подрібність елементів нейромережі: використання ієрархічних структур, дозволяє будувати моделі складних об'єктів з більш простих; робота ієрархічної структури вимагає, щоб інформаційний елемент у кожному ієрархічному рівні поведився як єдине ціле, але при переході з рівня на рівень допускав дроблення, причому при переході з верхнього ієрархічного рівня на нижній це дроблення відповідає виділенню складових його елементів, а при переході з нижнього рівня на верхній воно відповідає включенню визначеної частини цього елемента в більш складний об'єкт.

Визначимо ієрархічність мережі як:

$$I_h = \frac{\sum_{\eta=1}^M \eta N_{\eta}}{MN_h}.$$

*Попередня організація в навчанні* – властивість мережі, що відбиває її придатність для вирішення задачі на початку процесу структурно-параметричної ідентифікації. Дану властивість будемо характеризувати набором таких показників.

Відносний показник попередньої організації в навчанні визначимо як:

$$I_{po} = \frac{1}{2N_w} \sum_{\eta=1}^M \sum_{i=1}^{N_{\eta}} \sum_{j=1}^{N_{\eta-1}} \mathfrak{G}_{po}(w_j^{(\eta,i)}) + \frac{1}{4N_h} \sum_{\eta=1}^M \sum_{i=1}^{N_{\eta}} \mathfrak{G}_{po}(\varphi^{(\eta,i)}) + \frac{1}{4N_h} \sum_{\eta=1}^M \sum_{i=1}^{N_{\eta}} \mathfrak{G}_{po}(\psi^{(\eta,i)}),$$

де  $\mathfrak{G}_{po}$  – евристична функція, що показує ступінь не випадковості задання значення параметра мережі у попередній організації її навчання:

$$\mathfrak{G}_{po}(element) = \begin{cases} 0, & element = rand; \\ 1, & element \neq rand, \end{cases}$$

або

$$\mathfrak{G}_{po}(element) = 1 - \frac{\max(rand) - \min(rand)}{\max(element) - \min(element)},$$

де *element* – умовне позначення параметра функції, *rand* – випадковий компонент елемента.

Відносний показник автоматизації попередньої організації в навчанні задамо як:

$$I_{apo} = \frac{\mathfrak{G}_L + \mathfrak{G}_n}{4} + \frac{1}{4N_w} \sum_{\eta=1}^M \sum_{i=1}^{N_n} \sum_{j=1}^{N_{\eta-1}} (\mathfrak{G}_a(w_j^{(\eta,i)}) \mathfrak{G}_{po}(w_j^{(\eta,i)})) + \\ + \frac{1}{8N_n} \sum_{\eta=1}^M \sum_{i=1}^{N_n} (\mathfrak{G}_a(w_j^{(\eta,i)}) \mathfrak{G}_{po}(\varphi^{(\eta,i)})) + \frac{1}{8N_n} \sum_{\eta=1}^M \sum_{i=1}^{N_n} (\mathfrak{G}_a(w_j^{(\eta,i)}) \mathfrak{G}_{po}(\psi^{(\eta,i)})),$$

де  $\mathfrak{G}_L$  – евристична міра, що показує ступінь автоматизації задавання числа шарів у попередній організації навчання мережі ( $\mathfrak{G}_L = 1$ , якщо число шарів визначається автоматично; у протилежному випадку – приймають:  $\mathfrak{G}_L = 0$ ),  $\mathfrak{G}_n$  – евристична міра, що показує ступінь автоматизації задавання числа нейронів у попередній організації навчання мережі ( $\mathfrak{G}_n = 1$ , якщо число нейронів визначається автоматично; у протилежному випадку – приймають  $\mathfrak{G}_n = 0$ ),  $\mathfrak{G}_a(element)$  – евристична функція, що показує ступінь автоматизації задавання значення параметра *element* мережі в попередній організації її навчання ( $\mathfrak{G}_a(element) = 0$ , якщо значення параметра *element* задає користувач; у протилежному випадку –  $\mathfrak{G}_a(element) = 1$ ).

*Функціональна блочність* – властивість побудови архітектури мережі зі стандартизованих функціональних блоків, внутрішня організація і властивості яких визначені заздалегідь і відомі розробнику.

Оскільки всі нейронні і нейро-нечёткие мережі по своїй природі складаються з нейронів, що мають однотипність виконуваних операцій, відносну функціональну блочність будемо розглядати на зовнішньому рівні стосовно рівня нейронів:

$$I_{fb} = 1 - \frac{2 \sum_{i=1}^{N_n} \sum_{j=i+1}^{N_n} \mathfrak{G}_{fb}(i, j)}{N_n(N_n - 1)},$$

де  $\mathfrak{G}_{fb}(i, j)$  – показник приналежності *i*-го та *j*-го нейронів мережі до одного й того ж функціонального блоку ( $\mathfrak{G}_{fb}(i, j) = 1$ , якщо нейрони належать до одного й того ж блоку,  $\mathfrak{G}_{fb}(i, j) = 0$ , у протилежному випадку).

Якщо рівень функціональної блочності дорівнює нулю, це означає, що розробнику мережі зрозумілі тільки перетворення на рівні конкрет-

них нейронів, однак немає таких груп нейронів, функції яких були б зрозумілі. Якщо ж рівень функціональної блочності дорівнює одиниці, то всі нейрони мережі об'єднані в систему блоків, функціонування яких зрозуміло на більш високому рівні, ніж рівень конкретного нейрона.

*Інтерпретабельність* (логічна прозорість) – властивість мережі бути зрозумілою для сприйняття й аналізу людиною. Очевидно, що мережа є більш інтерпретабельною, якщо вона є ієрархічною, функціонально блочною, а також її ваги є контрастними (близькими за значенням до нуля або одиниці), причому середнє число входів нейрона не перевищує 5-7 (це число обумовлене особливостями людської психіки).

Поряд із раніше запропонованими в [74, 75] характеристиками логічної прозорості можна евристично визначити інтерпретабельність через ієрархічність і функціональну блочність:

$$I_{\text{interp.}} = \frac{I_{fb} I_h N_n^2}{\sum_{\eta=1}^M \sum_{i=1}^{N_\eta} \sum_{j=1}^{N_{\eta-1}} \vartheta_i(w_j^{(\eta,i)})},$$

$$\vartheta_i(w_j^{(\eta,i)}) = \begin{cases} 1 - |\sin(\pi w_j^{(\eta,i)})|, & 0 \leq w_j^{(\eta,i)} \leq 1; \\ \frac{1}{w_j^{(\eta,i)}}, & w_j^{(\eta,i)} > 1; \\ \frac{1}{|w_j^{(\eta,i)} - 1|}, & w_j^{(\eta,i)} < 0. \end{cases}$$

*Варіативність* моделей апроксимованої залежності за однією й тією ж таблицею даних нейромережею полягає в тому, що при незмінній структурі, але різних початкових значеннях ваг або різних методах їхнього настроювання після навчання в загальному випадку може вийти кілька різних нейромоделей.

Визначимо абсолютний показник варіативності моделей для мережі:

$$I_v = \prod_{\eta=1}^M \prod_{i=1}^{N_\eta} \left( \vartheta_v(\varphi^{(\eta,i)}) \vartheta_v(\psi^{(\eta,i)}) \prod_{j=1}^{N_{\eta-1}} \vartheta_v(w_j^{(\eta,i)}) \right),$$

де  $\vartheta_v(\varphi^{(\eta,i)})$ ,  $\vartheta_v(\psi^{(\eta,i)})$ ,  $\vartheta_v(w_j^{(\eta,i)})$  – відповідно, показники варіативності типів дискримінантних і активаційних функцій нейронів, а також значень ваг у процесі навчання мережі: якщо відповідний елемент мережі в процесі навчання може приймати тільки один фіксований стан, то два-

жають  $\mathfrak{G}_v(element) = 1$ , варіативність дискримінантних і активаційних функцій визначається архітектурою мережі та методом навчання, для ваг варіативність пропонується оцінювати за формулою:  $\mathfrak{G}_v(w_j^{(\eta,i)}) = 1$ , якщо вага є незмінною у процесі навчання;

$$\mathfrak{G}_v(w_j^{(\eta,i)}) = \text{round} \left( \frac{w_{j \max}^{(\eta,i)} - w_{j \min}^{(\eta,i)}}{\Delta w_j^{(\eta,i)}} \right), \text{ якщо вага може змінюватися. Тут}$$

$w_{j \max}^{(\eta,i)}, w_{j \min}^{(\eta,i)}$  – відповідно, максимальне і мінімальне можливі значення ваги  $w_j^{(\eta,i)}$ ,  $\Delta w_j^{(\eta,i)}$  – мінімально можлива зміна ваги з урахуванням розміру розрядної сітки ЕОМ.

*Емерджентність* (цілісність) – закономірність, що виявляється в системі у виникненні, появи в неї нових властивостей, відсутніх в елементах. Властивість цілісності зв'язана з метою, для виконання якої створюється система.

Ступінь цілісності системи відповідно до [72] визначається як

$$I_\alpha = -\frac{C_v}{C_o},$$

де  $C_o$  – власна складність, що являє собою сумарну складність (зміст) елементів системи поза зв'язком їх між собою (у випадку прагматичної інформації – сумарну складність елементів, що впливають на досягнення мети);  $C_v$  – взаємна складність, що характеризує ступінь взаємозв'язку елементів у системі (тобто складність її устрою, схеми, структури).

Стосовно до нейромоделі визначимо емерджентність як

$$I_\alpha = -\frac{\sum_{i=1}^{N_n} \sum_{j=1}^{N_n} \mathfrak{G}_{\alpha v}(i, j)}{\sum_{j=1}^{N_n} \mathfrak{G}_{\alpha o}(j)},$$

де  $\mathfrak{G}_{\alpha v}(i, j)$  – характеризує наявність зв'язку між  $i$ -м і  $j$ -м нейронами мережі ( $\mathfrak{G}_{\alpha v}(i, j) = 1$ , якщо зв'язок є; у протилежному випадку –  $\mathfrak{G}_{\alpha v}(i, j) = 0$ ),  $\mathfrak{G}_{\alpha o}(j)$  – складність  $j$ -го нейрона, що може бути визначена подібно [75] в одиницях елементарних операцій додавання і множення.

*Автономність* – здатність агента діяти без безпосереднього втручання людини шляхом здійснення контролю за власними діями і внут-

рішнім станом. Також автономність припускає можливість навчання на основі досвіду.

Оскільки навчені штучні нейронні мережі, як правило, у процесі свого функціонування при прийнятті рішень не вимагають участі людини, то вони в однаковій мірі мають властивість автономності функціонування.

Однак у процесі навчання рівень автономності для різних нейромереж і різних методів навчання може істотно варіюватися. Тому будемо розглядати далі характеристики автономності нейромереж тільки стосовно до процесу їхнього навчання.

Оскільки здатність до навчання обумовлюється пластичністю, то автономність навчання (самоадаптивність) будемо характеризувати показником, що залежить від характеристик пластичності нейромоделі. З іншого боку залежність процесу навчання нейромоделі від людини будемо характеризувати його впливом (часткою) на формування структури і параметрів нейромоделі. Поєднуючи ці розуміння, одержимо показник автономності методу навчання нейромоделі:

$$I_{aut} = \frac{N_{met}^{aut}}{N_{met}} \cdot \frac{\sum_{i=1}^{N_n} \mathcal{G}_{aut}^i \sum_{i=1}^{N_n} \mathcal{G}_{aut}(w_{ij})}{N_n^2},$$

$$\mathcal{G}_{aut}^i = \frac{1}{3} (\mathcal{G}_{aut}^{i*} + \mathcal{G}_{aut}^i(\varphi(i)) + \mathcal{G}_{aut}^i(\psi(i))),$$

де  $\mathcal{G}_{aut}^i$  – характеристика автономності формування  $i$ -го елемента структури мережі,  $\mathcal{G}_{aut}^{i*} = 0$ , якщо включення (або невключення)  $i$ -го нейрона в мережу визначає тільки людина;  $\mathcal{G}_{aut}^{i*} = 1$ , якщо включення (або невключення)  $i$ -го нейрона в мережу визначається тільки автоматично методом навчання;  $\mathcal{G}_{aut}^{i*} = 0,5$ , якщо включення (або невключення)  $i$ -го нейрона в мережу може визначати як людина, так і метод навчання;  $\mathcal{G}_{aut}^i(\varphi(i))$ ,  $\mathcal{G}_{aut}^i(\psi(i))$  – характеристики автономності вибору параметрів структури  $i$ -го нейрона мережі: відповідно, дискримінантних і активаційних функцій (приймають рівними нулю, якщо відповідні параметри визначаються тільки людиною; рівними одиниці – якщо, параметри визначаються тільки методом; рівними 0,5 – якщо параметри можуть визначатися як людиною, так і методом);  $\mathcal{G}_{aut}(w_{ij})$  – характеристика автономності на-строювання значення вагового коефіцієнта зв'язку між  $i$ -м і  $j$ -м нейро-

нами ( $\mathcal{G}_{aut}(w_{ij}) = 0$ , якщо вага задається тільки людиною;  $\mathcal{G}_{aut}(w_{ij}) = 1$ , якщо вага визначається автоматично методом навчання;  $\mathcal{G}_{aut}(w_{ij}) = 0,5$ , якщо вага визначається як людиною, так і методом);  $N_{met}$  – кількість параметрів методу навчання,  $N_{met}^{aut}$  – кількість параметрів методу навчання, значення яких визначаються автоматично без участі людини.

*Симетрія* – пропорційність, домірність у розташуванні частин цілого в просторі, повна відповідність (за розташуванням, величиною) однієї половини цілого іншій половині [33].

У нейромережах будемо виділяти симетрію структури і симетрію зв'язків.

Показник симетрії зв'язків нейромоделі визначимо як:

$$I_{sym}^w = \frac{\sum_{i=1}^{N_n} \sum_{j=1}^{N_n} \{1 | w_{i,j} = w_{j,i}\}}{N_n^2}.$$

Показник симетрії структури нейромоделі визначимо як:

$$I_{sym}^h = \frac{2 \sum_{i=1}^{N_n} \sum_{j=i+1}^{N_n} \{1 | \varphi(i) \equiv \varphi(j), \psi(i) \equiv \psi(j)\}}{N_n^2 - N_n}.$$

Загальний показник симетрії структури і зв'язків нейромоделі визначимо як:

$$I_{sym} = I_{sym}^h \cdot I_{sym}^w.$$

Відповідно, визначимо показники асиметрії:

- структури:  $I_{asym}^h = 1 - I_{sym}^h$ ;
- зв'язків:  $I_{asym}^w = 1 - I_{sym}^w$ ;
- загальний показник:  $I_{asym} = 1 - I_{sym}$ .

#### 4.9.5 Інтегральні показники властивостей і критерії порівняння нейромоделей

Розглянуті вище критерії характеризують тільки окремі властивості нейромереж, а для вирішення задач порівняння і відбору нейромоделей необхідно враховувати їхні властивості у комплексі, для цього пропонується використовувати інтегральні критерії порівняння нейромереж.

Найбільш широко використовуваними інтегральними критеріями є інформаційних критеріїв для синтезу і порівняння моделей. Дані критерії залежать від суми квадратів миттєвих помилок  $E$ , числа екземплярів у вибірці  $S$ , а також кількості налагоджуваних параметрів моделі  $N_w$ . Стосовно нейронних і нейро-нечітких мереж ці критерії можуть бути визначені у такий спосіб.

– Інформаційний критерій  $HQC$  (Hannan-Quinn information criterion):

$$HQC = S \ln\left(\frac{E}{S}\right) + 2N_w \ln(\ln(S)) \cdot$$

– Інформаційний критерій  $BIC$  (Bayesian Information Criterion, Schwarz Criterion, SBC, SBIC):

$$BIC = S \ln\left(\frac{E}{S}\right) + N_w \ln(S) \cdot$$

– Інформаційний критерій  $AIC$  (Akaike's Information Criterion):

$$AIC = S \ln\left(\frac{E}{S}\right) + 2N_w \cdot$$

– Критерій  $AICc$  (скорегований AIC для вибірок малого розміру):

$$AICc = S \ln\left(\frac{E}{S}\right) + 2N_w + \frac{2N_w(N_w + 1)}{S - N_w - 1} \cdot$$

– Критерій  $AICu$ :

$$AICu = \ln\left(\frac{E}{S - N_w}\right) + \frac{S + N_w}{S - N_w - 2} \cdot$$

– Критерій  $CP$ :

$$CP = E \left( \frac{2N_w - S}{S - N_w^{\max}} + 1 \right) \cdot$$

– Критерій  $MDL$  (Minimum Description Length – мінімум довжини опису):

$$MDL = E \left( \frac{N_w \ln(S)}{2(S - N_w^{\max})} + 1 \right) \cdot$$

– Критерій  $SSD$  (Shortest Data Description – найкоротший опис даних):

$$SSD = E \left( \frac{N_w \ln\left(\frac{S+2}{24}\right) + 2 \ln(S+1)}{S - N_w^{\max}} + 1 \right) \cdot$$

– Критерій *CAIC* (Consistent AIC):

$$CAIC = E \left( \frac{N_w (\ln(S) + 1)}{S - N_w^{\max}} + 1 \right).$$

Особливістю розглянутих інформаційних критеріїв є те, що вони сильно залежать від числа екземплярів у вибірці. При побудові ж і порівнянні нейромоделей, як правило, передбачається однаковість обсягу вибірок. Тому обсяг вибірок доцільно виключити з критеріїв порівняння.

У той же час різні синтезовані моделі можуть використовувати не усі з наявних у вибірці ознак. Тому число ознак, використовуваних у моделі, варто розглядати як важливу особливість при їхньому порівнянні.

При визначенні числа параметрів нейромереж, що налагоджуються, доцільно узяти до уваги той факт, що зв'язок вважається відсутнім, якщо його вага дорівнює нулю, таким чином, можна з числа налагоджуваних параметрів моделі виключити ваги, що дорівнюють нулю. Більш того, якщо нейромоделі містить константні ваги, що незмінюються в процесі навчання і незалежать від вибірки, то їх також доцільно виключити з числа параметрів нейромоделі, що налагоджуються.

На основі цих розумінь визначимо інтегральний інформаційний критерій як:

$$IIC = \left( 1 - \frac{N'(N_w - N_{w=0} - N_{w=const})}{NN_w^{\max}} \right) e^{-E},$$

де  $N$ ,  $N'$  – відповідно, число ознак у первісному наборі й у скороченому наборі ознак;  $N_{w=0}$  – число ваг у мережі, рівних нулю (чи прийнятих як рівні нулю),  $N_{w=const}$  – число ваг у мережі, нерівних нулю, але при цьому заданих константно (таких, що не змінюються ітеративно у процесі навчання).

Цей критерій буде приймати значення в діапазоні від 0 до 1. Чим менше буде його значення, тим гірше нейромоделі, а чим більше – тим краще.

Для урахування більшої кількості властивостей нейромереж розглянемо поняття ефективності нейромоделі.

*Ефективність* (якість) вирішення задач нейромоделлю визначається точністю (помилкою) вирішення задачі для навчальних і тестових даних, простотою, логічною прозорістю і швидкістю отриманої нейромережевої моделі, а також витратами на побудову нейромережевої

моделі (вимоги до апаратних засобів, ітераційність і витрати часу методу навчання).

Узагальнений показник ефективності нейромоделі на основі запропонованих показників визначимо як

$$I_{ef} = I_{pl} I_{par} I_h I_{fb} I_{gen} I_{lr} I_{interp} (I_{wre}^{avg} + I_{n.re}^{avg}) \left( \frac{1 + 0,5(I_{po} + I_{apo})}{6} \right) e^{-E}.$$

Узагальнений показник ефективності може бути використаний як для порівняння нейромоделей і методів їхнього синтезу, так і для оптимізації процесу побудови нейромоделей.

#### **4.10 Автоматизована система еволюційного синтезу й оптимізації нейромережових моделей**

Запропоновані методи синтезу нейромоделей та відбору інформативних ознак [18–23, 50–69] були програмно реалізовані мовою пакету Matlab. Розроблена автоматизована система еволюційного синтезу й оптимізації нейромережових моделей [43, 57] 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. 4.4.

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 нейромережових моделей – призначений для побудови нейромоделей за навчаючими вибірками даних;
- блок інтерфейсу користувача й візуалізації (форми, графіки).

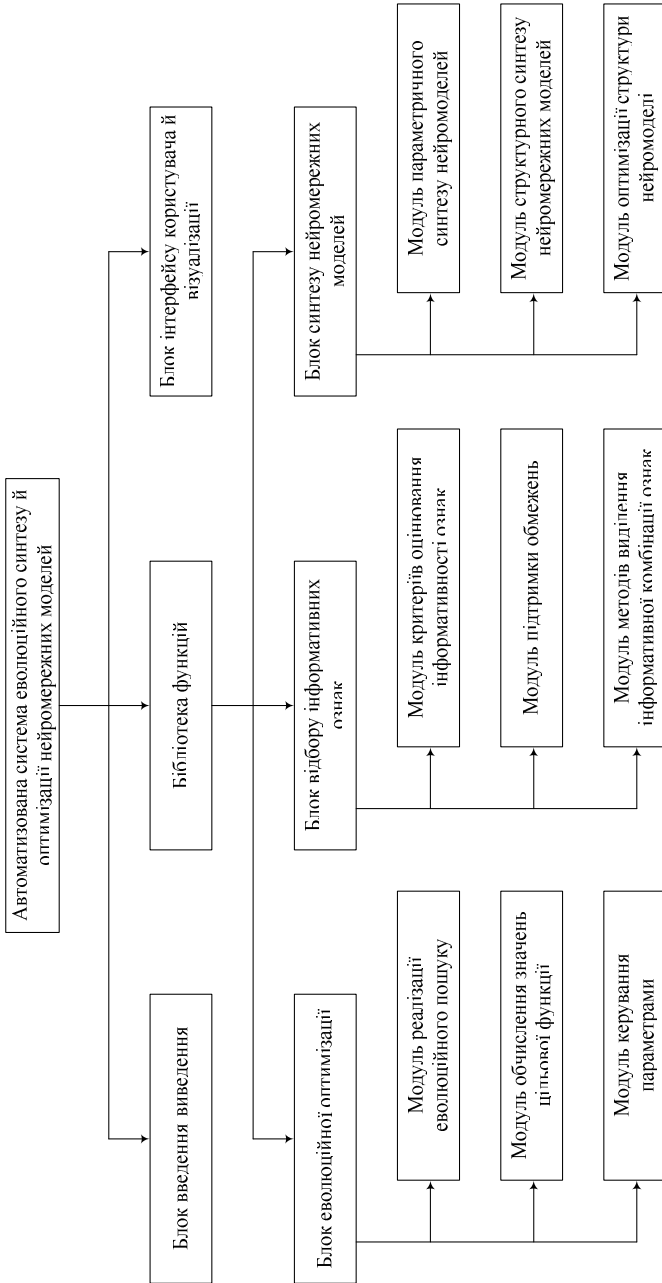


Рисунок 4.4 – Структурна схема автоматизованої системи еволюційного синтезу й оптимізації нейромережних моделей

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 геш-таблиці, 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 нейромережевої моделі можуть застосовуватися такі методи: метод зворотного поширення помилки, метод класичного еволюційного пошуку, запропонований метод з використанням апріорної інформації.

Модуль структурного синтезу нейромережевих моделей дозволяє синтезувати оптимальну структуру нейромережевої моделі одним з ме-

тодів: 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 нейромережєвих 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 підготовляти 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 нейромережєвих моделей відображено на рис. 4.5.

Схему функціонування автоматизованої системи еволюційного синтезу й оптимізації нейромережєвих моделей зображено на рис. 4.6.

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.

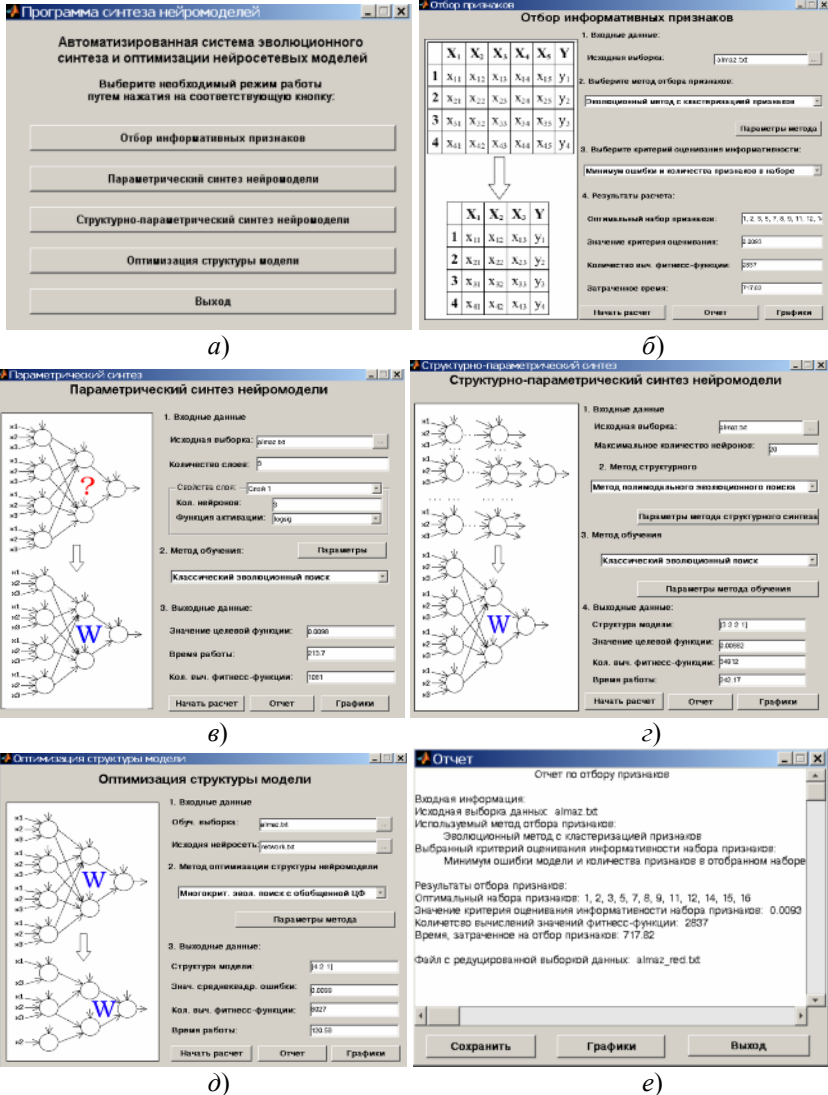


Рисунок 4.5 – Основні діалогові форми системи:  
 а) головна форма; б) форма відбору інформативних ознак;  
 в) форма параметричного синтезу моделей;  
 г) форма структурно-параметричного синтезу моделей;  
 д) форма оптимізації структури моделі; е) приклад звіту

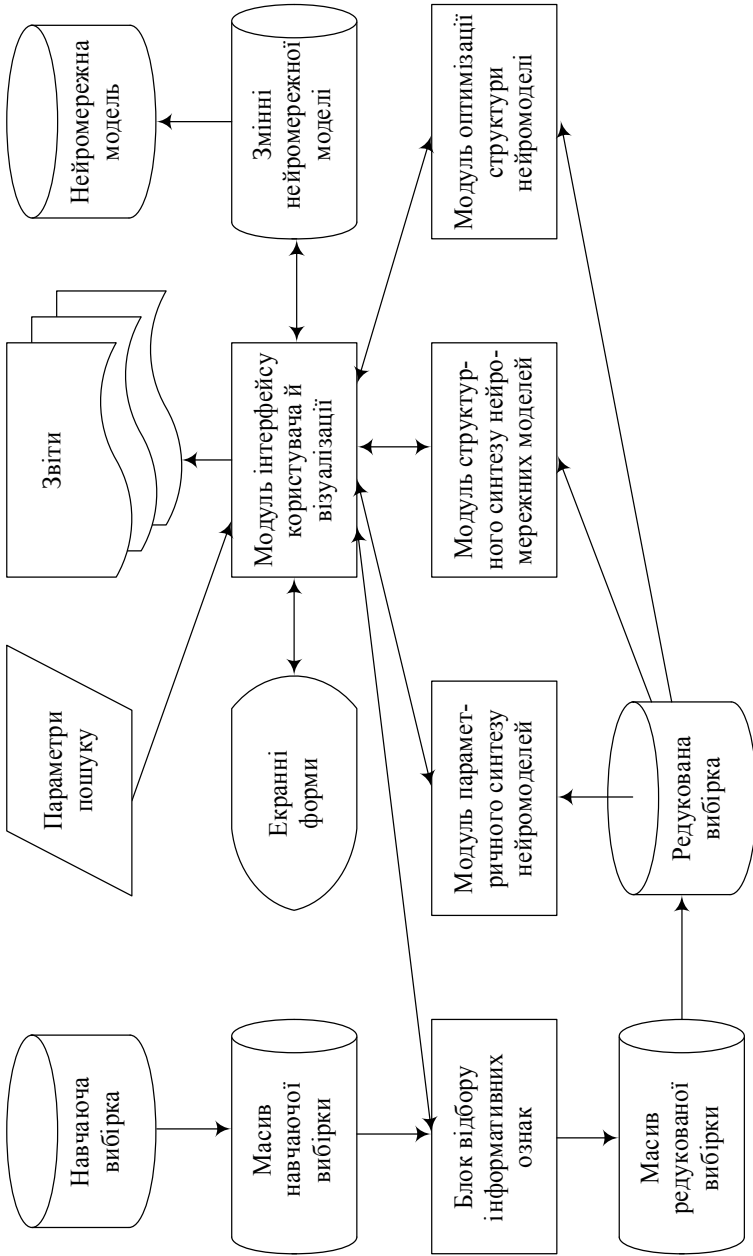


Рисунок 4.6 – Схема функціонування автоматизованої системи еволюційного синтезу й оптимізації нейромережкових моделей

З метою перевірки ефективності еволюційних методів побудови нейромережових моделей вирішувався ряд завдань технічної й біомедичної діагностики, автоматичної класифікації образів [6, 51, 58, 61, 68, 71, 85]. 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.

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.

## ? 4.11 Контрольні питання

1. Виконайте постановку задачі синтезу нейромережових моделей.
2. Які етапи виконуються при синтезі нейронних мереж?
3. В чому полягає задача відбору інформативних ознак?
4. Порівняйте структурний та параметричний синтез нейромереж.
5. З якою метою виконують оптимізацію побудованих нейромоделей?
6. Як виконується кодування хромосом для виділення найбільш значущої комбінації ознак?
7. Наведіть послідовність еволюційного пошуку відбору інформативних ознак.
8. Як відбувається керування параметрами еволюційного пошуку при відборі інформативних ознак?
9. У чому полягають особливості еволюційних методів відбору ознак з використанням апріорної інформації?
10. Проаналізуйте використання еволюційного пошуку для параметричного синтезу нейромереж.
11. У чому полягають переваги та недоліки еволюційного пошуку при параметричному синтезі нейромоделей?
12. Подання інформації, фітнес-функція та еволюційні оператори при структурному синтезі нейронних мереж.

13. Наведіть послідовність виконання структурно-параметричного синтезу нейромоделей на основі еволюційного пошуку.

14. Поясніть переваги та недоліки використання еволюційних методів для синтезу нейромережових моделей.

15. Як виконується оптимізація побудованих нейромоделей при використанні еволюційного підходу?

16. Проаналізуйте критерії порівняння еволюційних методів синтезу нейронних мереж.


17. Які критерії якості навчальних множин використовують традиційно при формуванні навчальних вибірок?

18. Які критерії можна використовувати для формування навчальної множини у задачах розпізнавання образів за допомогою еволюційного пошуку?

19. Проаналізуйте критерії порівняння нейромоделей, що традиційно використовують при вирішенні задач структурно-параметричного синтезу нейромереж?

20. Які критерії можна використовувати для побудови нейромереж за допомогою еволюційного пошуку?

## 5 ПРИКЛАДИ ТА ІЛЮСТРАЦІЇ


 **Приклад 1.** Визначити параметри, їхню кількість й розрядність для вирішення задачі апроксимації набору точкових даних у вигляді функції  $f(x) = A \cdot e^{kx}$  (де  $A$ ,  $k$  – константи,  $x$  – незалежна змінна) за допомогою еволюційного пошуку.

Для вирішення задачі за допомогою еволюційного пошуку її дані необхідно подати у вигляді генів особини. Для цього, насамперед, потрібно визначити, які параметри задачі необхідно настроїти.

Як параметри даної задачі будуть виступати  $A$  і  $k$ , оскільки від їхнього значення залежать властивості апроксимуючої функції. Таким чином, одержуємо два параметри й, отже, два гени.

Наступний крок – це вибір кількості розрядів у кожному гені. Якщо взяти для задачі апроксимації 16-розрядні гени, то при цьому параметри будуть змінюватися від  $-32,768$  до  $32,767$  із кроком в  $0,001$ . Дані числа отримані, виходячи з того, що 16-розрядне число може приймати одне з  $2^{16} = 65536$  значень від  $0$  до  $65535$ . Якщо припустити, що  $0$  буде в середині цього інтервалу, причому кожне значення розділиться потім на  $1000$ , то одержимо інтервал зміни параметрів і крок зміни.

Таким чином, при розв'язанні задачі апроксимації параметри  $A$  та  $k$  у десятковій системі числення являють собою фенотип, а в закодованому виді – генотип.


 **Приклад 2.** Для задачі синтезу математичної моделі, що описує деякий масив даних, визначити незалежні змінні (параметри, що оптимізуються за допомогою еволюційного пошуку) і цільову функцію для її вирішення за допомогою еволюційних методів.

*Розв'язок:*

Незалежними змінними є параметри моделі, обраної для опису заданого масиву даних. При цьому необхідно заздалегідь вибрати тип синтезованої моделі (регресійна, нейромережева, нечіткологічна, нейро-нечітка, і т.ін.).

Як цільову функцію доцільно обрати суму квадратів відхилень між реальними значеннями вихідного параметра й значеннями, отриманими за допомогою синтезованої моделі (квадратична похибка побудованої моделі). Мета оптимізації: мінімізація цільової функції. Простір значень цільової функції (область припустимих значень): додатні дійсні числа.


Крім визначення параметрів, що оптимізуються, і цільової функції необхідно також вибрати спосіб подання параметрів у хромосомі (спосіб кодування), визначити правила ініціалізації початкової популяції, вибрати оператори відбору, схрещування й мутації, а також задати їхні параметри, визначити критерії зупинення еволюційного пошуку.

 **Приклад 3.** Перевести десяткові числа  $47_{10}$  та  $48_{10}$  у код Грея, а число  $101101_{\Gamma}$  – з коду Грея до десяткового подання.

$$3.1. 47_{10} = 101111_2 = (1 \text{ XOR}(10111, 01111))_{\Gamma} = 111000_{\Gamma}.$$

$$3.2. 48_{10} = 110000_2 = (1 \text{ XOR}(11000, 10000))_{\Gamma} = 101000_{\Gamma}.$$

$$3.3. 101101_{\Gamma} = \begin{array}{l} 1 \Rightarrow 1 \\ 0 \Rightarrow 0 \text{ XOR } 1 = 1 \\ 1 \Rightarrow 1 \text{ XOR } 1 = 0 \\ 1 \Rightarrow 1 \text{ XOR } 0 = 1 \\ 0 \Rightarrow 0 \text{ XOR } 1 = 1 \\ 1 \Rightarrow 1 \text{ XOR } 1 = 0 \end{array} = 110110_2 = 54_{10}.$$


 **Приклад 4.** Змінна  $w_1$  подається за допомогою 8 біт,  $w_2$  – за допомогою 16 біт. Генетичний пошук відбувається в області  $w_1 \in [-2; 2]$ ,  $w_2 \in [-5; 5]$ . Визначити значення незалежних змінних  $w_1$  та  $w_2$  для хромосоми  $01100111111100000111110_2$ . Для зберігання незалежних змінних у хромосомах використовується двійкове подання.

Оскільки для подання змінної  $w_1$  використовується 8 біт, а для  $w_2$  – 16 біт, то хромосома  $01100111111100000111110_2$  розбивається на гени зі значеннями  $01100111_2$  (103 у десятковому виді) і  $1111100000111110_2$  (63550 у десятковому виді).

Таким чином, після декодування маємо:

$$w_1 = -2 + \frac{2 - (-2)}{2^8 - 1} 103 \approx -0,3843;$$

$$w_2 = -5 + \frac{5 - (-5)}{2^{16} - 1} 63550 \approx 4,6971.$$

 **Приклад 5.** Перевести в десятковий формат число  $00101101_{\Gamma}$ , що являє собою значення незалежної змінної, розташованої в інтервалі


[0; 1]. Визначити точність кодування. Для кодування незалежних змінних у хромосомах використовується код Грея.

Для кодування ознаки використовується 8 біт, тому інтервал зміни значень незалежної змінної розбивається на  $2^8 = 256$  підінтервалів. Таким чином, забезпечується точність  $1/256 \approx 0,0039$ .


Використовуючи код Грея, знаходимо відповідний значенню гена номер інтервалу:  $00101101_{\Gamma} = 00110110_2 = 54_{10}$ . Далі визначаємо, який

інтервал йому відповідає:  $\left[ \frac{54-1}{256}, \frac{54}{256} \right] = [0,2070312, 0,2109375]$ .


Значення ознаки розраховуємо як середнє значення знайденого інтервалу:  $(0,20703125 + 0,2109375) / 2 = 0,208984375$ .

 **Приклад 6.** Визначити відстань Хемінгу  $H^{(12)}$  між хромосомами  $H_1 = 0011010011$  та  $H_2 = 1010110010$ .


$H_1 = 0011010011$   
 $H_2 = 1010110010$   $\Rightarrow$  відстань Хемінгу (кількість різних генів в однакових позиціях хромосом  $H_1$  і  $H_2$ ) дорівнює 4.

 **Приклад 7.** Виконати одноточкове схрещування хромосом 0000000000 і 1111111111. Точка розриву – після 3-го гену.


0000000000 = **000**~0000000  $\Rightarrow$  **000**~11111111 = **000**1111111  
 1111111111 = **111**~1111111  $\Rightarrow$  **111**~0000000 = **111**0000000

 **Приклад 8.** Виконати п'ятиточкове схрещування хромосом 10010100000 і 01011011111. Точки розриву – після 1-го, 3-го, 5-го, 7-го й 9-го генів.


10010100000 = 1~**00**~10~**10**~00~**00**  $\Rightarrow$  1~**10**~10~**01**~00~**11** = **11010010011**  
 01011011111 = 0~**10**~11~**01**~11~**11**  $\Rightarrow$  0~**00**~11~**10**~11~**00** = **00011101100**

 **Приклад 9.** Виконати рівномірне схрещування хромосом 1100110 і 0110001 при масці схрещування (100101).


**1100110** + **0110001** = 1110010  
**0110001** + **1100110** = 0100101

 **Приклад 10.** Виконати одноточкове впорядковуюче схрещування хромосом (7 8 9 3 4 6 2 5 1) та (3 5 8 2 1 7 9 4 6). Точка розриву – після 6-го гену.


$$\begin{aligned} (7\ 8\ 9\ 3\ 4\ 6\ 2\ 5\ 1) &= 7\ 8\ 9\ 3\ 4\ 6 \sim \mathbf{2\ 5\ 1} \Rightarrow 7\ 8\ 9\ 3\ 4\ 6 \sim \mathbf{5\ 2\ 1} = (7\ 8\ 9\ 3\ 4\ 6\ 5\ 2\ 1) \\ (3\ 5\ 8\ 2\ 1\ 7\ 9\ 4\ 6) &= 3\ 5\ 8\ 2\ 1\ 7 \sim \mathbf{9\ 4\ 6} \Rightarrow 3\ 5\ 8\ 2\ 1\ 7 \sim \mathbf{9\ 4\ 6} = (3\ 5\ 8\ 2\ 1\ 7\ 9\ 4\ 6) \end{aligned}$$

 **Приклад 11.** Виконати двоточкове впорядковуюче схрещування хромосом (7 8 9 3 4 6 2 5 1) та (3 5 8 2 1 7 9 4 6). Точки розриву – після 2-го й 6-го генів.

$$\begin{aligned} (7\ 8\ 9\ 3\ 4\ 6\ 2\ 5\ 1) &= 7\ 8 \sim \mathbf{9\ 3\ 4\ 6} \sim 2\ 5\ 1 \Rightarrow 7\ 8 \sim \mathbf{3\ 9\ 4\ 6} \sim 2\ 5\ 1 = (7\ 8\ 3\ 9\ 4\ 6\ 2\ 5\ 1) \\ (3\ 5\ 8\ 2\ 1\ 7\ 9\ 4\ 6) &= 3\ 5 \sim \mathbf{8\ 2\ 1\ 7} \sim 9\ 4\ 6 \Rightarrow 3\ 5 \sim \mathbf{7\ 8\ 2\ 1} \sim 9\ 4\ 6 = (3\ 5\ 7\ 8\ 2\ 1\ 9\ 4\ 6) \end{aligned}$$

 **Приклад 12.** Виконати позиційно впорядковуюче схрещування хромосом (7 8 9 3 4 6 2 5 1) та (3 5 8 2 1 7 9 4 6). Обрані гени для безпосереднього копіювання в нащадків – 1-ий, 3-ій, 6- ой і 7- ой.

$$\begin{aligned} (7\ 8\ 9\ 3\ 4\ 6\ 2\ 5\ 1) &= \mathbf{7\ 8\ 9\ 3\ 4\ 6\ 2\ 5\ 1} \Rightarrow \mathbf{7\ 3\ 9\ 5\ 8\ 6\ 2\ 1\ 4} = (7\ 3\ 9\ 5\ 8\ 6\ 2\ 1\ 4) \\ (3\ 5\ 8\ 2\ 1\ 7\ 9\ 4\ 6) &= \mathbf{3\ 5\ 8\ 2\ 1\ 7\ 9\ 4\ 6} \Rightarrow \mathbf{3\ 4\ 8\ 6\ 2\ 7\ 9\ 5\ 1} = (3\ 4\ 8\ 6\ 2\ 7\ 9\ 5\ 1) \end{aligned}$$

 **Приклад 13.** Виконати РМХ-схрещування. Дано:  
 $P_1 = (5\ 1\ 3\ 2\ 6\ 4\ 7\ 8\ 9)$  та  $P_2 = (2\ 7\ 6\ 5\ 1\ 4\ 3\ 8\ 9)$ ,  $L = 9$ ,  $m_1 = 3$ ,  $m_2 = 6$ .

$$k = m_2 - m_1 + 1 = 6 - 3 + 1 = 4.$$

Визначимо транспозиції, необхідні для знаходження  $C_1$ . Виходячи з їхнього визначення, одержимо:

$$\begin{aligned} t_1 &= (m_1, P_1^{-1}(P_2(m_1))) = (3, P_1^{-1}(P_2(3))) = (3, P_1^{-1}(6)) = (3, 5) \Rightarrow \\ & P_1 \circ t_1 = (5\ 1\ 6\ 2\ 3\ 4\ 7\ 8\ 9) \\ t_2 &= (4, 1) \Rightarrow P_1 \circ t_1 \circ t_2 = (2\ 1\ 6\ 5\ 3\ 4\ 7\ 8\ 9) \\ t_3 &= (5, 2) \Rightarrow P_1 \circ t_1 \circ t_2 \circ t_3 = (2\ 3\ 6\ 5\ 1\ 4\ 7\ 8\ 9) \\ t_4 &= (6, 6) \Rightarrow P_1 \circ t_1 \circ t_2 \circ t_3 \circ t_4 = (2\ 3\ 6\ 5\ 1\ 4\ 7\ 8\ 9) \end{aligned}$$

Таким чином:  $C_1 = (2\ 3\ 6\ 5\ 1\ 4\ 7\ 8\ 9)$ .


Аналогічно знайдемо  $C_2$ :

$$\begin{aligned}
 t_1 &= (m_1, P_2^{-1}(P_1(m_1))) = (3, P_2^{-1}(P_1(3))) = (3, P_2^{-1}(3)) = (3, 7) \Rightarrow \\
 & P_2 \circ t_1 = (2 \ 7 \ 3 \ 5 \ 1 \ 4 \ 6 \ 8 \ 9) \\
 t_2 &= (4, 1) \Rightarrow P_2 \circ t_1 \circ t_2 = (5 \ 7 \ 3 \ 2 \ 1 \ 4 \ 6 \ 8 \ 9) \\
 t_3 &= (5, 7) \Rightarrow P_2 \circ t_1 \circ t_2 \circ t_3 = (5 \ 7 \ 3 \ 2 \ 6 \ 4 \ 1 \ 8 \ 9) \\
 t_4 &= (6, 6) \Rightarrow P_2 \circ t_1 \circ t_2 \circ t_3 \circ t_4 = (5 \ 7 \ 3 \ 2 \ 6 \ 4 \ 1 \ 8 \ 9)
 \end{aligned}$$

Таким чином  $C_2 = (5 \ 7 \ 3 \ 2 \ 6 \ 4 \ 1 \ 8 \ 9)$ .

Результат застосування РМХ-схрещування:

$$\begin{array}{ccc}
 P_1 = (5 \ 1 \ 3 \ 2 \ 6 \ 4 \ 7 \ 8 \ 9) & (2 \ 3 \ 6 \ 5 \ 1 \ 4 \ 7 \ 8 \ 9) = C_1 & \\
 | & | & \Rightarrow \\
 P_2 = (2 \ 7 \ 6 \ 5 \ 1 \ 4 \ 3 \ 8 \ 9) & (5 \ 7 \ 3 \ 2 \ 6 \ 4 \ 1 \ 8 \ 9) = C_2 &
 \end{array}$$

 **Приклад 14.** Виконати циклове схрещування.  $P_1$  та  $P_2$  взяти з попереднього приклада,  $s_0 = 2$ .

З визначення циклового схрещування знайдемо підстановку  $(s_0, s_1, \dots, s_k)$ . Одержуємо:

$$\left. \begin{array}{l}
 s_0 = 2 \\
 s_1 = P_2^{-1}(P_1(s_0)) = P_2^{-1}(1) = 5 \\
 s_2 = P_2^{-1}(P_1(s_1)) = P_2^{-1}(6) = 3 \\
 s_3 = P_2^{-1}(P_1(s_2)) = P_2^{-1}(3) = 7 \\
 s_4 = P_2^{-1}(P_1(s_3)) = P_2^{-1}(7) = 2
 \end{array} \right\} \Rightarrow (s_0, s_1, s_2, s_3) = (2, 5, 3, 7)$$

Знаходимо  $C_1$ :

$$\begin{aligned}
 C_1 &= P_2 \circ (2, 5, 3, 7) = (2 \ 7 \ 6 \ 5 \ 1 \ 4 \ 3 \ 8 \ 9) \circ (2, 5, 3, 7) = (2 \ 7 \ 6 \ 5 \ 1 \ 4 \ 3 \ 8 \ 9) \circ (2, 5) \circ (5, 3) \circ (3, 7) = \\
 &= (2 \ 1 \ 6 \ 5 \ 7 \ 4 \ 3 \ 8 \ 9) \circ (5, 3) \circ (3, 7) = (2 \ 1 \ 7 \ 5 \ 6 \ 4 \ 3 \ 8 \ 9) \circ (3, 7) = (2 \ 1 \ 3 \ 5 \ 6 \ 4 \ 7 \ 8 \ 9).
 \end{aligned}$$

У цьому випадку  $C_1$  успадковує елементи 1, 3, 6 і 7 з  $P_1$ , а елементи 2, 5, 4, 8 і 9 – з  $P_2$ .

Аналогічно отримаємо  $C_2$ :

$$\left. \begin{array}{l}
 s_0 = 2 \\
 s_1 = P_1^{-1}(P_2(s_0)) = P_1^{-1}(7) = 7 \\
 s_2 = P_1^{-1}(P_2(s_1)) = P_1^{-1}(3) = 3 \\
 s_3 = P_1^{-1}(P_2(s_2)) = P_1^{-1}(6) = 5 \\
 s_4 = P_1^{-1}(P_2(s_3)) = P_1^{-1}(1) = 2
 \end{array} \right\} \Rightarrow (s_0, s_1, s_2, s_3) = (2, 7, 3, 5)$$



4. Оскільки обране найкоротше ребро приводить до передчасного замикання циклу, то шлях збільшується за рахунок включення пункту, обраного випадковим чином з числа ще не пройдених:

$$\begin{aligned} \text{1-й нащадок} \quad & 4 \rightarrow 2(7); \\ & 4 \rightarrow 3(6); \quad \rightarrow 1(1); \rightarrow 4(\text{цикл}); \quad \rightarrow 2(2) \\ & \quad \quad \quad \rightarrow 2(2). \end{aligned}$$

5. Процес триває доти, поки не буде побудований повний шлях комівояжера. У нашому прикладі такий шлях уже побудований:


1-й нащадок  $4 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 4$ ;  $f(1\pi) = 6 + 1 + 2 + 3 = 12$ .

Аналогічним чином одержуємо другого нащадка:


В якості початкової точки за умовою заданий пункт 2.

$$\begin{aligned} \text{2-й нащадок:} \quad & 2 \rightarrow 3(2); \\ & 2 \rightarrow 1(1); \quad \rightarrow 4(4); \rightarrow 2(\text{цикл}); \\ & \quad \quad \quad \rightarrow 4(4); \rightarrow 3(6). \end{aligned}$$

2-й нащадок  $2 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 2$ ;  $f(2\pi) = 1 + 4 + 6 + 2 = 13$ .


 **Приклад 16.** Виконати просту мутацію третього біта бінарної хромосоми  $H = \{0101001\}$ .

Результатом простої мутації хромосоми  $H$  є хромосома  $H' = \{0111001\}$ .


 **Приклад 17.** Виконати просту мутацію третього гену векторної хромосоми  $H = \{\{1,2,3\}, \{4,5,6\}, \{7,8,9\}, \{10,11,12\}\}$ . Точку мутації в гені, що піддається мутації, вибрати після другої (між другою та третьою) позиції гену.

Для виконання мутації необхідно поміняти місцями числа, що знаходяться у другій і третій позиціях третього гену заданої векторної хромосоми.


$$\begin{aligned} H &= \{\{1,2,3\}, \{4,5,6\}, \{7,8 \sim 9\}, \{10,11,12\}\} \Rightarrow \\ H' &= \{\{1,2,3\}, \{4,5,6\}, \{7,9,8\}, \{10,11,12\}\}. \end{aligned}$$

 **Приклад 18.** Виконати мутацію обміну хромосоми  $H = \{0101001\}$  при  $y_1=2$  і  $y_2=5$ .

Результатом мутації обміну хромосоми  $H$  є хромосома  $H' = \{0001101\}$ .


 **Приклад 19.** Виконати одноточкову мутацію обміну хромосоми  $H = \{0101001\}$ . Точку мутації в хромосомі вибрати після другого гена.

Результатом одноточкової мутації обміну хромосоми  $H$  є хромосома  $H' = \{0011001\}$ .

 **Приклад 20.** Виконати одноточкову мутацію обміну хромосоми  $H = \{0101001\}$ . Точку мутації в хромосомі визначити за допомогою методу золотого розтину.

Розрахуємо точку мутації  $D = \text{Ціле}(\tau \cdot L) = \text{Ціле}(0,61803 \cdot 7) = 4$ . Таким чином, точка мутації знаходиться між 4-им та 5-им генами.

Результатом одноточкової мутації обміну хромосоми  $H$  є хромосома  $H' = \{0100101\}$ .

 **Приклад 21.** Виконати мутацію 3-го гена векторної хромосоми  $H = \{\{1,2,3,4,5\}, \{6,7,8,9,10,11\}, \{12,13,14,15,16,17,18,19,20\}, \{21,22,23\}\}$  на основі чисел Фібоначчі. Обмін числових значень між мутуючими розрядами гену зробити циклічно зі зсувом праворуч.

До третього гена застосовується мутація, причому розряди для мутації обираються відповідно до чисел Фібоначчі, тобто № 1, 2, 3, 5, 8. Зробимо обмін числових значень між мутуючими розрядами циклічно зі зсувом праворуч:


$$h_3 = \{12,13,14,15,16,17,18,19,20\} \Rightarrow \{19,12,13,15,14,17,18,16,20\} = h'_3.$$

У результаті одержуємо:  $H' = \{\{1,2,3,4,5\}, \{6,7,8,9,10,11\}, \{19,12,13,15,14,17,18,16,20\}, \{21,22,23\}\}$ .

 **Приклад 22.** Виконати класичне інвертування:

- хромосоми  $H_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$  при  $y_1=2$  і  $y_2=7$ ;
- хромосоми  $H_2 = \{0101101\}$  при  $y_1=2$  і  $y_2=6$ .

Результатом інвертування хромосоми  $H_1$  є хромосома  $H_1' = \{1, 2, 6, 5, 4, 3, 7, 8\}$ , а хромосоми  $H_2$  – хромосома  $H_2' = \{0111001\}$ .

 **Приклад 23.** Виконати інвертування зі зсувом підстроки (1 3 2) хромосоми (5 1 3 2 6 4 7 8 9) у позицію 6.

$$(5\ 1\ 3\ 2\ 6\ 4\ 7\ 8\ 9) \Rightarrow (7\ 8\ 9\ 5\ 6\ 1\ 3\ 2\ 4)$$



 **Приклад 24.** Визначити для схеми  $H=\{10^{**}1\}$  усю множину хромосом, а також порядок  $O(H)$  і визначену довжину  $L(H)$ .

Схема  $10^{**}1$  визначає собою множину з чотирьох п'ятибітових хромосом:  $\{10001; 10011; 10101; 10111\}$ .

Порядок схеми  $O(H) = 3$  – кількість визначених бітів (0 или 1).

Визначена довжина  $L(H) = 5 - 1 = 4$  – відстань між крайніми визначеними бітами.

 **Приклад 25.** Визначити для схеми  $H=\{**1^{***}101^{**}\}$  порядок  $O(H)$ , визначену довжину  $L(H)$ , кількість представлених нею хромосом  $k(H)$ , імовірність виживання після односточкового схрещування  $P_s(H)$ , імовірність знищення після односточкового схрещування  $P_d(H)$ , імовірність виживання після застосування оператора мутації  $P_{sm}(H)$  з імовірністю  $P_m = 0,1$ .

Порядок схеми  $O(H) = 4$ .


Визначена довжина  $L(H) = 9 - 3 = 6$ .

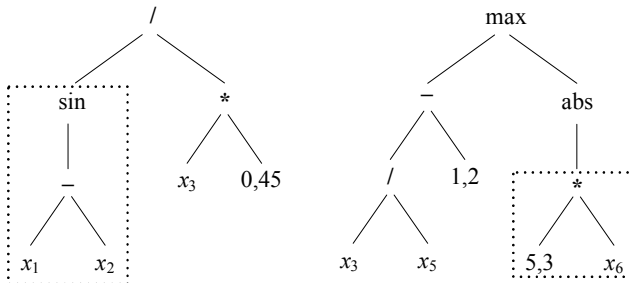
Кількість хромосом  $k(H) = 2^{L-O(H)} = 2^{11-4} = 2^7$ .

$P_s(H) = 1 - L(H) / (L - 1) = 1 - 6 / (11 - 1) = 0,4$ .

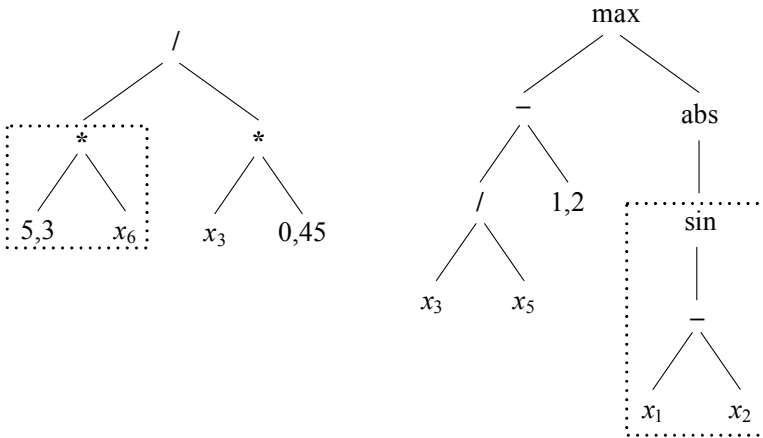
$P_d(H) = L(H) / (L - 1) = 6 / (11 - 1) = 0,6$ .


$P_{sm}(H) = (1 - P_m)^{O(H)} = (1 - 0,1)^4 = 0,6561$ .

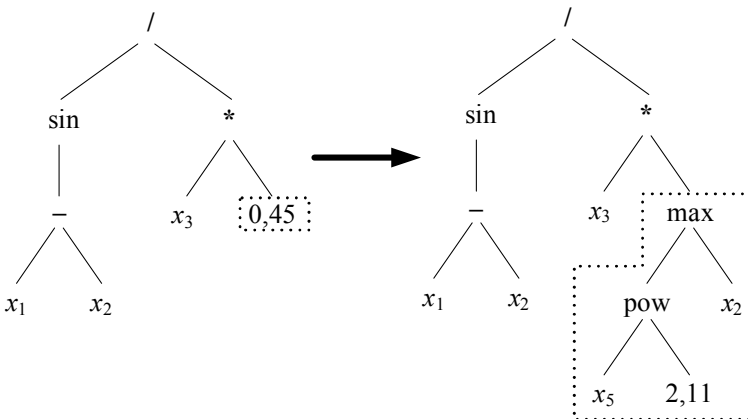
 **Приклад 26.** Виконати оператор схрещування над заданими деревами подання рішень при використанні генетичного програмування. Точки розриву – виділено прямокутниками.




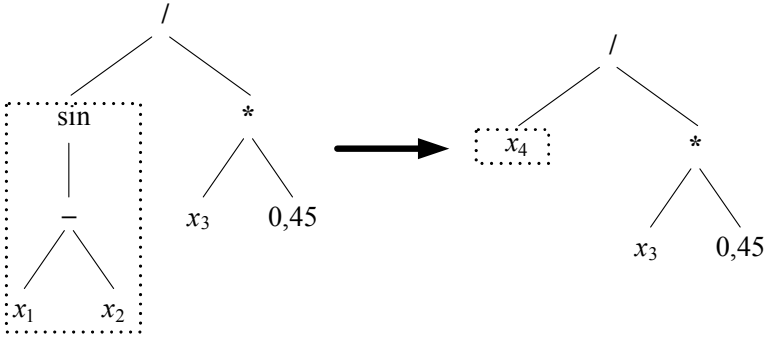
В результаті застосування оператора схрещування отримаємо дерева:




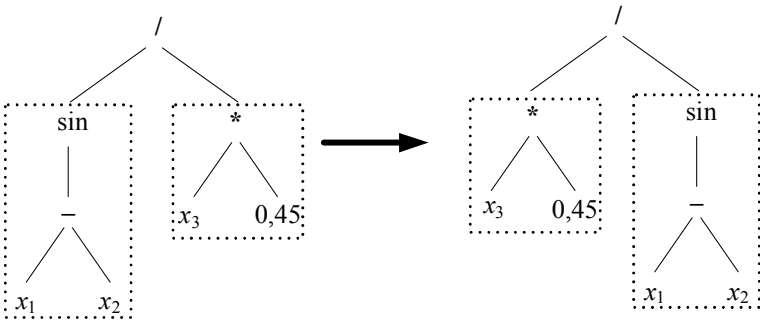
 **Приклад 27.** Виконати оператор мутації зі збільшенням над заданим деревом подання рішень при використанні генетичного програмування.




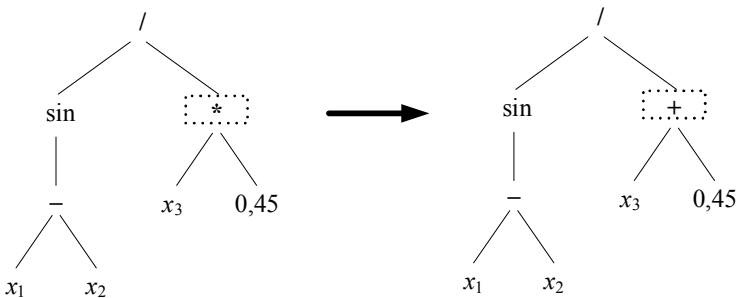
 **Приклад 28.** Виконати оператор мутації зі стисненням над заданим деревом подання рішень при використанні генетичного програмування.



 **Приклад 29.** Виконати оператор мутації із заміною над заданим деревом подання рішень при використанні генетичного програмування.



 **Приклад 30.** Виконати оператор циклічної мутації над заданим деревом подання рішень при використанні генетичного програмування.





**Приклад 31. Пошук мінімуму функції.** За допомогою пакету Matlab знайти мінімум функції  $f(x) = x_1^2 + x_2^2 + 5$ , використовуючи методи генетичного пошуку. Вивчити використання різних еволюційних операторів. Застосувати острівну модель еволюційного пошуку, що використовує 20 особин на 1 острові та 30 – на другому

*Розв'язок:* функція  $f(x)$  оформляється в окремому файлі F.m:

```
function mean1 = F(x)
mean1=x(1)*x(1)+x(2)*x(2)+5;
```

Знайдемо мінімум функції  $f(x)$  за допомогою функції `ga` пакета Matlab, формат виклику якої наведений нижче:

```
[x, Fmin] = ga(@fitnessfun, n, options)
```

Опис полів структури `options` наведено у табл. 5.1.

Таблиця 5.1 – Опис полів структури `options`

№ п/п	Назва поля	Значення за замовчуванням	Інші можливі значення	Опис
1	2	3	4	5
1	PopulationType	'doubleVector' (параметри функції, що оптимізується, є дійсними)	'bitstring'  'custom'	Тип даних популяції (тип параметрів функції, що оптимізується): бітовий рядок;  користувальницький (для його використання треба реалізувати власні CreationFcn, MutationFcn і CrossoverFcn).
2	PopInitRange	[0;1]	Матриця, вектор	Діапазон значень хромосом (параметрів, змінних) у початковій популяції.
3	PopulationSize	20	Додатне ціле Вектор вигляду [a <sub>1</sub> a <sub>2</sub> ... a <sub>k</sub> ]	Кількість особин у популяції. Кількість особин у кожній з <i>k</i> підпопуляцій.
4	EliteCount	2	Додатне ціле	Кількість елітних особин.
5	Crossover-Fraction	0.8000	0 ... 1	Частина популяції наступного покоління (не включаючи елітні особини), що створюється за допомогою схрещування.
6	MigrationDirection	'forward' (міграція з <i>n</i> -ої підпопуляції в ( <i>n</i> +1)-у)	'both' (міграція з <i>n</i> -ої підпопуляції в ( <i>n</i> +1)-у й ( <i>n</i> -1)-у)	Напрямок міграції.
7	MigrationInterval	20	Додатне ціле	Кількість поколінь, після яких відбуваються міграції.

Продовження таблиці 5.1

1	2	3	4	5
8	MigrationFraction	0.2000	0 ... 1	Частина особин у кожній підпопуляції, що мігрує в іншій підпопуляції.
9	Generations	100	Додатне ціле	Обмеження на максимальну кількість ітерацій.
10	TimeLimit	Inf	Скаляр	Обмеження часу на виконання алгоритму.
11	FitnessLimit	-Inf	Скаляр	Обмеження на значення цільової функції.
12	StallGenLimit	50	Додатне ціле	Метод зупиниться, якщо не буде поліпшення цільової функції протягом StallGenLimit поколінь.
13	StallTimeLimit	20	Додатне ціле	Метод зупиниться, якщо не буде поліпшення цільової функції протягом StallTimeLimit секунд.
14	InitialPopulation	[]	Матриця розміром PopulationSize * (кількість параметрів функції, що оптимізується)	Початкова популяція. Якщо InitialPopulation = [], то початкова популяція створюється за допомогою CreationFcn
15	InitialScores	[]	Вектор-стовпець	Значення цільової функції особин початкової популяції.
16	CreationFcn	@gacreationuniform	Заголовок функції користувача	Заголовок функції, що створює початкову популяцію.
17	FitnessScalingFcn	@fitscalingrank (перетворення значень цільової функції на підставі рангу, тобто позиції у відсортованому масиві)	@fitscalingshiftlinea r	Функція, що перетворює (масштабує) значення цільової функції до значень, використовуваних функцією відбору: значенню цільової функції кращого члена популяції привласнюється значення MaximumSurvivalRate, помножене на середнє значення цільової функції;
			@fitscalingprop	пропорційно значенням цільової функції;
			@fitscalingtop	значенням цільової функції кращих quantity членів популяції привласнюється значення 1/quantity, іншим - 0;
			функція користувача	користувальницька функція перетворення.

## Продовження таблиці 5.1

1	2	3	4	5
18	SelectionFcn	@selectionstochastic unif (пропорційний відбір)	@selectionremainder	Функція, що реалізує відбір: відбір пропорційно цілої частини цільової функції, а потім відбір рулеткою дробової частини, що залишилася, пропорційно до цільової функції;
			@selectionrandom	випадковий відбір;
			@selectionroulette	відбір з використанням рулетки;
			@selectiontourname nt	турнірний відбір;
			функція користувача	користувальницький оператор відбору.
19	CrossoverFcn	@crossoverscattered (однорідне схрещування)	@crossoverheuristic	Функція, що реалізує схрещування: евристичне схрещування;
			@crossoverintermediate	арифметичне схрещування;
			@crossoveringlepoint	одноточкове схрещування;
			@crossovertwooint	двоточкове схрещування;
			функція користувача	користувальницький оператор схрещування.
20	MutationFcn	@mutationgaussian (гауссовська мутація)	@mutationuniform	Функція, що реалізує мутацію: проста мутація;
			функція користувача	користувальницький оператор мутації.
21	HybridFcn	[]	@fminsearch @patternsearch @fminunc	Функція, що продовжує оптимізацію після завершення роботи генетичного методу.
22	Display	'final' (відобразити результати після завершення роботи ga)	'off'	Період відображення результатів роботи ga: не відобразити
			'iter'	відобразити на кожній ітерації
			'diagnose'	Відобразити діагностичну інформацію, а також результати роботи на кожній ітерації

Продовження таблиці 5.1

1	2	3	4	5
23	PlotFcns	[]	@gaplotbestf	Масив заголовків функцій, що відображають результати: графік кращих і середніх значень цільової функції;
			@gaplotbestindiv	значення цільової функції кращих членів поточної популяції;
			@gaplotdistance	середня відстань між членами кожної популяції;
			@gaplotexpectation	очікувана кількість нащадків у порівнянні з перетвореними значеннями цільової функції;
			@gaplotsselection	гістограма хромосом-батьків;
			@gaplotrange	мінімальні, максимальні й середні значення цільової функції в кожній популяції;
			@gaplotscorediversity	гістограма значень цільової функції на поточній популяції;
			@gaplotscores	значення цільової функції всіх членів поточної популяції;
			@gaplotstopping	досягнення критеріїв зупинення на поточній популяції;
	функція користувача		користувальницька функція висновку	
24	PlotInterval	1	Додатне ціле	Кількість ітерацій, між яким викликаються функції виведення.
25	OutputFcns	[]	Масив заголовків функцій користувача	Масив заголовків функцій, які викликає ga на кожній ітерації.
26	Vectorized	'off'	'on'	Рядок, що визначає, чи є обчислення цільової функції векторизованим.

Для зміни значень параметрів функції `ga` використовується команда `gaoptimset`, а для одержання поточних параметрів – функція `gaoptimget`.

Наприклад, для встановлення розміру популяції в 100 особин необхідно використовувати наступну команду:

```
options = gaoptimset('PopulationSize', 100)
```

Функція `gaoptimset` дозволяє також передавати параметри в обрані функції, що реалізують основні генетичні оператори. Наприклад, для використання функції мутації `mutationgaussian` з параметрами

scale = 0.5 (середньоквадратичне відхилення) і shrink = 0.75 (параметр скорочення) можна використовувати таку команду:

```
scale = 0.5; shrink = 0.75;
options = gaoptimset('MutationFcn',
{@mutationgaussian, scale, shrink})
```

Таким чином, для пошуку мінімуму функції  $f(x) = x_1^2 + x_2^2 + 5$  за допомогою острівної моделі еволюційного пошуку, що використовує 20 особин на 1 острові та 30 – на другому, у командному рядку пакета Matlab необхідно прописати наступні команди:

```
options = gaoptimset('PopulationSize', [20 30])
[x Fmin] = ga(@F, 2, options)
Результат: x = 0.0004    0.0020
Fmin = 5.0000
```



### Приклад 32. Розв'язання задачі лінійного програмування

Задачу лінійного програмування в матричному вигляді можна подати так:  $F = Cx \rightarrow \min$  при обмеженнях  $Ax \leq B$ , де  $B$  і  $C$  – вектори,  $A$  – матриця.

*Завдання:* написати функцію, що реалізує розв'язання задачі лінійного програмування. Вхідними параметрами функції є матриця  $A$  і вектори  $B$ ,  $C$ . У випадку, якщо не заданий хоча б один параметр  $A$ ,  $B$  або  $C$ , розв'язання таку задачу лінійного програмування (тестовий варіант роботи функції):

$$F = 5x_1 + 10x_2 \rightarrow \max;$$

$$\begin{cases} 8x_1 - 5x_2 \leq 40, \\ 2x_1 + 5x_2 \geq 10, \\ -6x_1 + 5x_2 \leq 60, \\ 2x_1 + x_2 \leq 14, \\ x_1, x_2 \geq 0. \end{cases}$$

Розв'язок оформимо у вигляді двох файлів GA\_LP.m і F.m.

Файл GA\_LP.m

```
function mean = GA_LP(A1, B1, C1)
global A B C
```

```
if (nargin~=3) % задані не всі параметри => A, B, C
                задаються за замовченням
    A=[8 -5; -2 -5; -6 5; 2 1];
```

```

    B=[40 -10 60 14];
    C=[-5 -10];
end;
if (nargin==3)
    A=A1;
    B=B1;
    C=C1;
end;
[X Fmin] = ga(@F, size(C,2))
mean = X;

```

### Файл F.m

```

function mean1 = F(X)
global A B C
X=X';
if (length(find((A*X)>B'))>0) %умова AX<=B не вико-
                               нується
    mean1=inf;
end;
if (length(find((A*X)>B'))==0)
    mean1=C*X;
end;

```

Виклик функції GA\_LP у середовищі MatLab і результати її роботи:

- GA\_LP  
 $X = 0.6251 \quad 12.7477$   
 $Fmin = -130.6022$
- GA\_LP([2 1; -1 -3; -3 -1], [8 -6 -3], [-2 1])  
 $X = 3.5964 \quad 0.8018$   
 $Fmin = -6.3911$



### Приклад 33. Розв'язання задачі нелінійного програмування

Формально загальна задача нелінійного програмування (нелінійної оптимізації) може бути сформульована в такий спосіб: мінімізувати цільову функцію  $f(w) \rightarrow \min$ , де  $w = \{w_1, w_2, \dots, w_N\}$  – набір змінних параметрів, від яких залежить значення цільової функції при  $m$  лінійних обмежень  $c_j(w) = 0; j = 1, \dots, m$  і  $p$  нелінійних обмежень  $c_k(w) \geq 0$  (або  $\leq, <, >$ );  $k = m + 1, \dots, m + p$ .

*Завдання:* написати функцію, що реалізує розв'язання задачі нелінійного програмування при обмеженнях  $\leq, \geq, <, >$ . Вхідними параметрами функції є StrFunc і StrC, що містять цільову функцію й обмеження відповідно, а також  $n$  – кількість незалежних змінних функції StrFunc.

Якщо обмеження StrC не задані, знайти мінімум функції StrFunc без обмежень. У випадку, якщо не задані параметри StrFunc і StrC, розв'язати таку задачу нелінійного програмування:

$$F = x_1^2 + x_2^2 + x_3 + 25 \rightarrow \min, \quad x_1 > 2, x_2 > -7, x_2 \cdot x_3 + 5 > 0.$$

Розв'язок оформимо у вигляді двох файлів GA\_NLP.m і F.m.

Файл GA\_NLP.m

```
function mean = GA_NLP(strFunc, n, StrC)
global strF C      % strF - функція, що оптимізується
                  % C - обмеження
                  % n - кількість змінних функції,
                  %   що оптимізується
if (nargin==0) || (nargin==1) %функція за замовченням
    strF='X(1)*X(1)+X(2)*X(2)+X(3)+25';
    C='((X(1)>2) && (X(2)>-7) &&
      ((X(3)*X(2)+5)>0))';
    n=3;
end;
if (nargin==2) % функція, що оптимізується, задана
               % як параметр
    strF=strFunc;
    C='1==1'; % обмеження не задані => встановлюємо
end;           % завжди істинну умову
if (nargin==3) % функція, що оптимізується, задана
               % в якості параметра
    strF=strFunc;
    C=StrC;
end;
[X Fmin] = ga(@F, n, gaoptimset('PopulationType',
'doubleVector'));
Fmin
mean = X;
```

Файл F.m

```
function mean1 = F(X)
global strF C
if (eval(C)~=1) % умова C не виконується
    mean1=inf;
end;
if (eval(C)==1)
    mean1=eval(strF);
end;
```

Виклик функції GA\_NLP у середовищі MatLab і результати її роботи:

```

1. GA_NLP
      Fmin = 16.7829
      ans = 2.0002    -0.0153    -12.2183
2. GA_NLP(' (1-X(1)) * (1-X(1)) + (2-X(2)) * (2-X(2)) + 7', 2)
      Fmin = 7.0000
      ans = 1.0009    2.0003
3. GA_NLP(' (1-X(1)) * (1-X(1)) + (2-X(2)) * (3-X(2)) + 7', 2, '(X(1) <= 0) && (X(2) > 1)')
      Fmin = 7.7500
      ans = -0.0000    2.4982

```

*Примітка:* для розв'язання задачі нелінійного програмування, що включає всі види обмежень ( $\leq$ ,  $\geq$ ,  $<$ ,  $>$  і  $=$ ), необхідно додатково розробити процедуру пошуку залежності між параметрами функції із заданих обмежень типу  $=$ . У такий спосіб кількість незалежних змінних функції StrFunc буде дорівнює  $n = K - K_*$ , де  $K$  – кількість параметрів функції StrFunc,  $K_*$  – кількість обмежень типу  $=$ .



### Приклад 34. Апроксимація

*Завдання:* написати функцію, що апроксимує набір точок, що заданий масивами  $X$  і  $Y$ . Також задається вид апроксимуючої функціональної залежності в рядку StrFunc і кількість параметрів, що підбираються,  $n$ . У випадку, якщо не заданий хоча б один параметр  $X$ ,  $Y$ , StrFunc або  $n$ , вирішити таку задачу апроксимації: апроксимувати набір точок  $x = [6.268 \ 9.344 \ 3.701 \ 5.445 \ 2.432 \ 5.569 \ 2.657 \ 2.332 \ 3.508 \ 1.465 \ 4.670 \ 4.351 \ 9.216 \ 9.471 \ 3.725 \ 8.494 \ 1.212 \ 2.611 \ 8.022]$ ;  $y = [7.412 \ 13.102 \ 4.607 \ 6.364 \ 3.642 \ 6.512 \ 3.797 \ 3.575 \ 4.445 \ 3.045 \ 5.513 \ 5.197 \ 12.795 \ 13.413 \ 4.628 \ 11.194 \ 2.905 \ 3.765 \ 10.256]$  функцією  $y = w_1 \cdot e^{w_2 x}$ , де  $w_1$  й  $w_2$  – параметри, що підбираються.

*Розв'язок* оформимо у вигляді двох файлів Approxim.m і ErrorPodborAppr.m.

Файл Approxim.m

```

function mean = Approxim(strFunc, n, x1, y1)
global x y strF % strF – апроксимуюча функція
                % n – кількість змінних апроксимуючої функції

```

```

if (nargin~=4) % апроксимуюча функція за замовчен-
ням
    strF='w(1)*exp(w(2)*x)';
    n=2;
    x=[6.268 9.344 3.701 5.445 2.432 5.569 2.657
2.332 3.508 1.465 4.670 4.351 9.216 9.471 3.725
8.494 1.212 2.611 8.022]';
    y=[7.412 13.102 4.607 6.364 3.642 6.512 3.797
3.575 4.445 3.045 5.513 5.197 12.795 13.413 4.628
11.194 2.905 3.765 10.256]';
end;
if (nargin==4) % всі параметри задані
    strF=strFunc;
    x=x1;
    y=y1;
end;
[W error] = ga(@ErrorPodborAppr, 2)
mean = W;

```

#### Файл ErrorPodborAppr.m

```

function mean1 = ErrorPodborAppr(w)
global x y strF
a=eval(strF);
for i=1:1:size(y,1)
    E(i)=abs(y(i)-a(i))/(abs(y(i)));
end;
mean1=mean(E); %результат функції - середня похибка

```

Виклик функції Approxim у середовищі MatLab і результати її роботи:

#### 1. Approxim

```

W = 2.3228    0.1853
error = 0.0011

```

```

2. Approxim('w(1)*x.*x+w(2)*x',2, [2 34 54 24 342
62 63 62], [34 6018 14958 3048 587214 19654 20286
19654])

```

```

W = 8.4141    0.1719
error = 5.1137e-009

```



### **Приклад 35. Параметричний синтез нейронної мережі заданої структури**

**Завдання:** написати функцію, що налагоджує ваги двохшарової нейронної мережі, що містить  $k_1$  нейронів на першому шарі й  $k_2$  нейронів – на другому й навчається за вибіркою  $X$  (матриця незалежних змінних) і  $Y$  (вектор значень цільової функції). Точність апроксимації 0,01. Підрахувати кількість викликів функції помилки.

У випадку, якщо не задані параметри  $k_1$ ,  $k_2$ ,  $X$  и  $Y$ , настроїти нейронну мережу, що містить 4 нейрони на першому шарі й 1 нейрон на другому шарі.

**Розв'язок** оформимо у вигляді двох файлів NNweights.m і ErrorPodbor.m.

Файл NNweights.m

```
function mean1 = NNweights(k1_, k2_, x1, y1)
global Q k1 k2 net x y %net - двохшарова нейронна
                        мережа
                        %k1 - кількість нейронів на 1-ому шарі
                        %k2 - кількість нейронів на 2-ому шарі
if (nargin~=4) % параметри не задані
    k1=4;
    k2=1;
x=[елементи матриця X]';
y=[елементи.масиву Y];
end;
if (nargin==4) % всі параметри задані
    k1=k1_;
    k2=k2_;
    x=x1;
    y=y1;
end;
%нормалізація X и Y
x_min=min(x', [], 1);
x_max=max(x', [], 1);
for i=1:1:size(x, 1)
    for j=1:1:size(x, 2)
        x(i, j)=(x(i, j)-x_min(i))./abs(x_max(i)-
x_min(i));
    end;
end;
y_max=max(y);
y_min=min(y);
```

```

y=(y-y_min)/abs(y_max-y_min);
net=newff(repmat([0 1],
size(x,1),1),[k1,k2],{'logsig','logsig'},'trainlm')
;
Q=0;
[W error] = ga(@ErrorPodbor,
k1*(size(x,1)+1)+k2*(k1+1),gaoptimset('FitnessLimit',
0.01));
Q
error
net.IW{1,1}
net.b{1,1}
net.b{2,1}
net.LW{2,1}
mean1 = W';

```

#### Файл ErrorPodbor.m

```

function mean1 = ErrorPodbor(w)
global k1 k2 net x y Q %Q -номер поточної ітерації
net_IW=[];
for i=1:1:k1
    for j=1:1:size(x,1)
        net_IW(i,j)=w(j+size(x,1)*(i-1));
    end;
end;
net.IW{1,1}=net_IW;
net.IW{1,1};
net_b11=[];
for i=1:1:k1
    net_b11(i)=w(i+size(x,1)*k1);
end;
net.b{1,1} = net_b11';
net_b21=[];
for i=1:1:k2
    net_b21(i)=w(i+size(x,1)*k1+k1);
end;
net.b{2,1} = net_b21;
net_LW=[];
for i=1:1:k2
    for j=1:1:k1
        net_LW(i,j)=w(j+size(x,1)*(i-
1)+size(x,1)*k1+k1+k2);

```

```

end;
end;
net.LW{2,1} = net_LW;
a=sim(net,x);
for i=1:1:size(y,1)
    E(i)=abs(y(i)-a(i))/(abs(y(i)));
end;
Q=Q+1;
A=[Q mean(E)];
mean1=mean(E); % результат функції -середня помилка

```

Виклик функції NNweights у середовищі MatLab і результати її роботи:

```

1. NNweights
Q = 180           %кількість ітерацій
error = 0.0019   %похибка
net.IW{1,1}=
    Columns 1 through 9
-1.3283    2.5324   -1.3129   -0.2808    0.4730    -
0.0987    0.3018    1.1025    2.2214
 0.4990   -1.1696    1.4755    0.1822   -0.8203
1.6955   -1.9219   -1.1233    2.8977
-0.4352    1.2547    3.0918   -0.1316    1.6743    -
0.6672    2.0966   -2.4275    1.2555
 2.1228    1.0495    3.6832    1.3034   -0.1187
1.1635    0.2265   -0.6895   -2.3788
    Columns 10 through 11
    -3.3060    0.2481
     3.8343   -0.9004
     1.9228   -0.2934
     0.9605    3.0611

net.b{1,1}=
    0.8495
   -2.8727
   -2.7475
    1.7031

net.b{2,1} = -0.6223
net.LW{2,1}= -1.7539    2.4341    0.9476    -
2.3492
2. NNweights(5, 1, x, y)

```

Q = 160

error = 0.0026

net.IW{1,1}=

Columns 1 through 9

-0.9647	1.7090	-0.0529	2.1717	0.3155
1.0761	0.5945	-0.9221	0.0985	
1.4284	-0.6901	0.8076	1.5180	2.9064
2.2507	-0.2081	-0.3386	0.7981	
0.3559	-0.6540	0.3723	-1.5776	-0.2134
0.0923	-0.4807	2.2941	0.3423	
-0.9890	-0.1484	1.0102	1.1160	-0.9156
1.7254	-0.2030	0.9167	0.5958	
-0.4758	0.3028	-0.8149	-0.1103	-3.0851
0.8314	2.3185	0.8609	-0.9077	

Columns 10 through 11

-2.9822	2.0700
0.6837	-1.0005
-0.8365	0.6877
2.7636	1.6551
1.9377	0.0296

net.b{1,1}=

-0.5353
-0.5897
0.7673
0.3227
-1.9030

net.b{2,1} = -1.6559

net.LW{2,1}= -0.5934 -0.2919 -1.1430 0.5409 1.6481

## 6 ПРАКТИЧНІ ЗАВДАННЯ

### 6.1 Моделі та методи еволюційного пошуку



*Завдання 1.* Відповідно до номеру варіанту розробити за допомогою пакету Matlab програмне забезпечення, що реалізує заданий метод еволюційного пошуку.

Програмне забезпечення повинно мати коментарі до тексту програми та складатися з трьох .m-файлів:

- funcXX.m (цільова функція оптимізації), де XX – номер індивідуального завдання студента;
- varXX.m – програма, що реалізує заданий еволюційний метод, використовуючи функцію ga пакету Matlab;
- testXX.m – програма тестування еволюційного пошуку.



*Завдання 2.* Виконати тестування розробленого у попередньому завданні програмного забезпечення. Отримати оптимальне значення цільової функції та відповідні значення незалежних змінних, побудувати графік цільової функції (у випадку, якщо задана цільова функція оптимізації  $f$  є функцією більш, ніж двох змінних, то побудувати графік функції  $f(x_1, x_2)$ ). Отримати інші параметри та побудувати графіки відповідно до індивідуального завдання.

#### ① Варіант №1

1. Розробити за допомогою пакету Matlab програму, яка приймає як аргументи назву функції, що необхідно оптимізувати, кількість її параметрів, а також розмір популяції хромосом для виконання еволюційного пошуку.

Програма повинна знаходити оптимум заданої функції за допомогою еволюційної оптимізації (гібридна модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `fminsearch`).

Параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації функції  $f_1(x) = |x| + \cos(x)$ . Для цього провести серію зі 100 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №2

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_2(x) = ((x - 5)^2 + 1) \cdot |x^3 - 3|$ , використовуючи острівну модель еволюційного пошуку (20 особин на першому острові, 80 особин на другому острові). Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

Як аргумент розроблювана програма приймає параметр  $A$ , що відображує необхідність виведення графіку цільової функції: якщо  $A = 1$ , то графік повинен бути виведений, в іншому випадку (включаючи ситуацію, коли розроблена програма завантажується без параметрів з командного рядку Matlab) виводити графік функції, що оптимізується, не потрібно.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього провести серію зі 100 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №3

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_3(x_1, x_2) = x_1 \sin(4x_1) + 1,1x_2 \sin(2x_2)$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – однорідне, мутація – гауссовська; розмір популяції – 200 особин; кількість елітних особин – 3.

Як аргумент  $P_c$  програма приймає ймовірність схрещування. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення. Передбачити ситуацію виклику розробленої програми без параметрів (встановити значення  $P_c$  за замовчанням 0,75).

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `patternsearch`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього провести серію зі 100 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мі-

німальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №4

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_4(x_1, \dots, x_n) = \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – арифметичне, мутація – проста; розмір популяції – 150 особин; максимальна кількість ітерацій еволюційного пошуку – 75.

Як аргументи програма приймає: кількість аргументів цільової функції, кількість елітних особин, розмір турніру для виконання турнірного відбору. Програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `fminunc`. Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обгрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього провести серію зі 100 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №5

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_5(x_1, x_2) = (1,5 - x_1 \cdot (1 - x_2))^2 + (2,25 - x_1 \cdot (1 - x_2^2))^2 + (2,625 - x_1 \cdot (1 - x_2^3))^2$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – за допомогою рулетки, схрещування – одноточкове, мутація – гауссовська. Використовувати острівну модель еволюційної оптимізації (30 особин на першому острові, 20 – на другому, 50 – на третьому).

Як аргументи програма приймає: інтервал міграції  $MI$ , максимальну кількість ітерацій еволюційного пошуку  $T$ . Програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення. Передбачити ситуацію виклику роз-

робленої програми без введення деяких параметрів (встановити значення за замовченням:  $MI = 7$ ,  $T = 100$ ).

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього провести серію зі 100 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №6

1. Розробити за допомогою пакету Matlab програму, що оптимізує

$$\text{функцію } f_6(x_1, x_2) = 1 - \left( \frac{(x_1 + 3)^2 + (x_2 + 3)^2}{200} - \cos(x_1 + 3) \cdot \cos\left(\frac{x_2 + 3}{\sqrt{2}}\right) + 2 \right)^{-1},$$

використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – арифметичне, мутація – проста; кількість елітних особин – 3.

Як аргумент  $N$  програма приймає розмір популяції. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `fminsearch`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив розміру популяції на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань розмір популяції встановлювати за формулою:  $N_i = 5 + 10 \cdot i$ . Побудувати графік залежності середнього значення часу оптимізації від розміру популяції.

### ① Варіант №7

1. Розробити за допомогою пакету Matlab програму, що оптимізує

$$\text{функцію } f_7(x_1, x_2) = \text{Ціле} \left[ 1 - \left( \frac{(x_1 + 3)^2 + (x_2 + 3)^2}{200} - \cos(x_1 + 3) \cdot \cos\left(\frac{x_2 + 3}{\sqrt{2}}\right) + 2 \right)^{-1} \right],$$

використовуючи еволюційний пошук. В якості еволюційних операторів

обрати: відбір – турнірний, схрещування – евристичне, мутація – проста; кількість елітних особин – 2. Використовувати острівну модель еволюційної оптимізації (27 особин на першому острові, 23 – на другому, 40 – на третьому).

Як аргумент  $m$  програма приймає розмір турніру для виконання турнірного відбору. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив розміру турніру на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань розмір турніру встановлювати за формулою:  $m_i = 2 + i$ . Побудувати графік залежності середнього значення отриманого оптимуму цільової функції від розміру турніру.

### ① Варіант №8

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_8(x_1, x_2, x_3, x_4) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2 + 90 \cdot (x_4 - x_3^2)^2 + 10,1 \cdot (x_2 - 1)^2 + (x_4 - 1)^2$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – за допомогою рулетки, схрещування – евристичне, мутація – гауссовська. Використовувати острівну модель еволюційної оптимізації (35 особин на першому острові, 65 – на другому).

Як аргумент  $MF$  програма приймає ймовірність міграції. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності міграції на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань ймовірність міграції встановлювати за формулою:  $MF_i = 0,05 + 0,08 \cdot i$ . Побудувати графік залежності середнього значення отриманого оптимуму цільової функції від ймовірності міграції.

### ① Варіант №9

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_9(x_1, x_2) = (x_2 - x_1^2)^2 + (1 - x_1)^2$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний,

схрещування – одноточкове, мутація – проста; кількість елітних особин – 5; розмір популяції – 200 особин.

Як аргумент  $T$  програма приймає максимальну кількість ітерацій еволюційного пошуку. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `patternsearch`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив максимальної кількості ітерацій еволюційного пошуку на його ефективність. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань максимальну кількість ітерацій еволюційного пошуку встановлювати за формулою:  $T_i = 10 + 10 \cdot i$ . Побудувати графік залежності середнього значення часу оптимізації від максимальної кількості ітерацій еволюційного пошуку.

### ① Варіант №10

1. Розробити за допомогою пакету Matlab програму, що оптимізує

функцію  $f_{10}(x_1, \dots, x_n) = 100 - \sum_{i=1}^n (10 \cos(2\pi x_i) - x_i^2)$ ,  $n = 10$ , використовуючи

еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – арифметичне, мутація – гауссовська. Використовувати острівну модель еволюційної оптимізації (19 особин на першому острові, 39 – на другому, 73 – на третьому).

Як аргумент  $P_c$  програма приймає ймовірність схрещування. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності схрещування на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань ймовірність схрещування встановлювати за формулою:  $P_{c,i} = 0,05 + 0,075 \cdot i$ . Побудувати графік залежності середнього значення отриманого оптимуму цільової функції від ймовірності схрещування.

### ① Варіант №11

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію

$$f_{11}(x_1, x_2) = 1 - \left( \frac{x_1^2 + x_2^2}{200} - \cos(x_1) \cdot \cos\left(\frac{x_2}{\sqrt{2}}\right) + 2 \right)^{-1},$$

використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – однорідне, мутація – гауссовська; кількість елітних особин – 4; розмір популяції – 100 особин.

Як аргумент  $S$  програма приймає параметр гауссовської мутації scale. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції fminunc.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив параметру scale на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань значення параметру scale встановлювати за формулою:  $scale_i = 0,05 + 0,07 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від параметру scale. Вивести на екран графік синтезованої моделі.

### ① Варіант №12

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію

$$f_{12}(x_1, x_2) = 0,391 - (\sin(\pi \cdot x_1) + \sin(\pi \cdot x_2)) \cdot 0,2 + 0,01 \cdot (0,4 \cdot (x_1 - 5,5)^2 + 0,5 \cdot (x_2 - 5,5)^2),$$

використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – евристичне, мутація – гауссовська; кількість елітних особин – 5. Використовувати острівну модель еволюційної оптимізації (33 особини на першому острові, 21 – на другому, 51 – на третьому).

Як аргумент  $S$  програма приймає параметр гауссовської мутації shrink. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив параметру *shrink* на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в *i*-й серії випробувань значення параметру *shrink* встановлювати за формулою:  $\text{shrink}_i = 0,05 + 0,065 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від параметру *shrink*. Вивести на екран графік синтезованої моделі.

### ① Варіант №13

1. Розробити за допомогою пакету Matlab програму, що оптимізує

функцію 
$$f_{13}(x_1, x_2) = 1,4388 + \frac{(\sin \sqrt{x_1^2 + x_2^2 + 1} \cdot \sin \sqrt{x_1^2 + 1} - 0,5)}{(1 + 0,001 \cdot (x_1^2 + x_2^2))^2}$$
, викори-

стовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – за допомогою рулетки, схрещування – арифметичне, мутація – проста. Використовувати острівну модель еволюційної оптимізації (45 особин на першому острові, 59 – на другому).

Як аргумент  $k_e$  програма приймає кількість елітних особин. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив кількості елітних особин на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в *i*-й серії випробувань значення кількості елітних особин встановлювати за формулою:  $k_{e,i} = i$ . Побудувати лінійну регресійну модель залежності середнього значення часу оптимізації від кількості елітних особин. Вивести на екран графік синтезованої моделі.

### ① Варіант №14

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію

$$f_{14}(x_1, x_2) = 8,1063 - 3 \cdot (1 - x_1)^2 \cdot e^{(-x_1^2 - (x_2 + 1)^2)} + 10 \cdot (0,2 \cdot x_1 - x_1^3 - x_2^5) \cdot e^{(-x_1^2 - x_2^2)} + \frac{1}{3} e^{-(x_1 + 1)^2 - x_2^2},$$

використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – однокриве, мутація – гассовська; кількість елітних особин – 5. Використовувати острівну мо-

дель еволюційної оптимізації (33 особини на першому острові, 67 – на другому).

Як аргумент  $P_m$  програма приймає ймовірність міграції. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності міграції на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань значення ймовірності міграції встановлювати за формулою:  $P_{m,i} = 0,07 + 0,09 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від ймовірності міграції. Вивести на екран графік синтезованої моделі.

### ① Варіант №15

1. Розробити за допомогою пакету Matlab програму, що оптимізує

функцію  $f_{15}(x_1, \dots, x_n) = \sum_{i=1}^n x_i^2$ ,  $n = 5$ , використовуючи еволюційний пошук.

В якості еволюційних операторів обрати: відбір – турнірний, схрещування – двоточкове, мутація – проста; кількість елітних особин – 5; розмір популяції – 175 особин.

Як аргумент  $P_m$  програма приймає ймовірність мутації генів хромосоми при простій мутації. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `patternsearch`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності мутації генів на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 100 експериментів кожна. При цьому в  $i$ -й серії випробувань ймовірність мутації генів встановлювати за формулою:  $P_{m,i} = 0,01 + 0,02 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення часу оптимізації від ймовірності мутації генів. Вивести на екран графік синтезованої моделі.

### ① Варіант №16

1. Розробити за допомогою пакету Matlab програму, яка приймає як аргументи назву функції, що необхідно оптимізувати, кількість її параметрів, а також кількість елітних хромосом для виконання еволюційного пошуку.

Програма повинна знаходити оптимум заданої функції за допомогою еволюційної оптимізації (гібридна модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції fminunc).

Параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації функції

$$f_{16}(x_1, \dots, x_n) = \sum_{i=1}^n (x_i^2 - 3 \cos(2\pi x_i) + 3), \quad n = 8. \text{ Для цього провести серію зі}$$

150 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №17

1. Розробити за допомогою пакету Matlab програму, що оптимізує

$$\text{функцію } f_{17}(x) = \begin{cases} x^2, & \text{якщо } x \leq -10, \\ x^2 \cdot |\sin x| + x, & \text{якщо } -10 < x \leq -5, \\ x^2 \cdot |\sin x|, & \text{якщо } -5 < x \leq 5, \\ x, & \text{якщо } x > 5, \end{cases} \quad \text{використовуючи}$$

острівну модель еволюційного пошуку (39 особин на першому острові, 82 особини на другому острові). Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

Як аргумент розроблювана програма приймає параметр  $Z$ , що відображує необхідність виведення графіку цільової функції: якщо  $Z = 1$ , то графік повинен бути виведений, в іншому випадку (включаючи ситуацію, коли розроблена програма завантажується без параметрів з командного рядку Matlab) виводити графік функції, що оптимізується, не потрібно.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього прове-

сти серію зі 150 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №18

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{18}(x) = \sin^6(5,1\pi x + 0,5)$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – однорідне, мутація – проста; розмір популяції – 120 особин; кількість елітних особин – 4.

Як аргумент  $P_c$  програма приймає ймовірність схрещування. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення. Передбачити ситуацію виклику розробленої програми без параметрів (встановити значення  $P_c$  за замовчанням 0,72).

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `fminsearch`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього провести серію зі 150 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №19

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{19}(x_1, \dots, x_n) = 1 + \sum_{i=1}^n \left( \frac{x_i^2}{4000} \right) - \prod_{i=1}^n \cos\left( \frac{x_i}{\sqrt{i}} \right)$ ,  $n = 10$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – арифметичне, мутація – проста; розмір популяції – 125 особин; максимальна кількість ітерацій еволюційного пошуку – 95.

Як аргументи програма приймає: кількість елітних особин, розмір турніру для виконання турнірного відбору. Програма повинна перевіря-

ти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `patternsearch`. Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього провести серію зі 150 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №20

1. Розробити за допомогою пакету Matlab програму, що оптимізує

функцію  $f_{20}(x_1, \dots, x_n) = n \cdot A + \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i))$ ,  $n = 7$ ,  $A = 10$ , використо-

вуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – за допомогою рулетки, схрещування – одноточкове, мутація – проста. Використовувати острівну модель еволюційної оптимізації (40 особин на першому острові, 30 – на другому, 80 – на третьому).

Як аргументи програма приймає: інтервал міграції  $MI$ , максимальну кількість ітерацій еволюційного пошуку  $T$ . Програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення. Передбачити ситуацію виклику розробленої програми без введення деяких параметрів (встановити значення за замовченням:  $MI = 10$ ,  $T = 200$ ).

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити ефективність еволюційного пошуку для оптимізації заданої функції. Для цього провести серію зі 150 експериментів з фіксованими параметрами еволюційної оптимізації та отримати середнє, середньоквадратичне відхилення, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### ① Варіант №21

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{21}(x) = |x| + \sin(x)$ , використовуючи еволюційний пошук. В яко-

сті еволюційних операторів обрати: відбір – пропорційний, схрещування – евристичне, мутація – проста; кількість елітних особин – 2.

Як аргумент  $N$  програма приймає розмір популяції. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `fminunc`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив розміру популяції на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань розмір популяції встановлювати за формулою:  $N_i = 7 + 11 \cdot i$ . Побудувати графік залежності середнього значення часу оптимізації від розміру популяції.

### ① Варіант №22

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{22}(x) = (x^2 + x) + \cos(x)$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – арифметичне, мутація – проста; кількість елітних особин – 3. Використовувати острівну модель еволюційної оптимізації (38 особин на першому острові, 29 – на другому, 50 – на третьому).

Як аргумент  $m$  програма приймає розмір турніру для виконання турнірного відбору. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив розміру турніру на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань розмір турніру встановлювати за формулою:  $m_i = 3 + i$ . Побудувати графік залежності середнього значення отриманого оптимуму цільової функції від розміру турніру.

### ① Варіант №23

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{23}(x_1, x_2) = x_2 \sin(4x_1) + 1,1x_1 \sin(2x_2)$ , використовуючи еволю-

ційний пошук. В якості еволюційних операторів обрати: відбір – за допомогою рулетки, схрещування – арифметичне, мутація – гауссовська. Використовувати острівну модель еволюційної оптимізації (100 особин на першому острові, 30 – на другому).

Як аргумент  $MF$  програма приймає ймовірність міграції. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності міграції на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань ймовірність міграції встановлювати за формулою:  $MF_i = 0,06 + 0,085 \cdot i$ . Побудувати графік залежності середнього значення отриманого оптимуму цільової функції від ймовірності міграції.

### ① Варіант №24

1. Розробити за допомогою пакету Matlab програму, що оптимізує

функцію  $f_{24}(x_1, x_2) = 0,5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0,5}{1 + 0,1(x_1^2 + x_2^2)}$ , використовуючи еволю-

ційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – евристичне, мутація – проста; кількість елітних особин – 2; розмір популяції – 150 особин.

Як аргумент  $T$  програма приймає максимальну кількість ітерацій еволюційного пошуку. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `fminsearch`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив максимальної кількості ітерацій еволюційного пошуку на його ефективність. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань максимальну кількість ітерацій еволюційного пошуку встановлювати за формулою:  $T_i = 10 + 15 \cdot i$ . Побудувати графік залежності середнього значення часу оптимізації від максимальної кількості ітерацій еволюційного пошуку.

### ① Варіант №25

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{25}(x_1, x_2) = (x_1^2 + x_2^2)^{0,25} \sin\left(30((x_1 + 0,5)^2 + x_2^2)^{0,1}\right) + |x_1| + |x_2|$ ,

використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – одноточкове, мутація – гауссовська. Використовувати острівну модель еволюційної оптимізації (29 особин на першому острові, 42 – на другому, 61 – на третьому).

Як аргумент  $P_c$  програма приймає ймовірність схрещування. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обгрунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності схрещування на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань ймовірність схрещування встановлювати за формулою:  $P_{c,i} = 0,055 + 0,085 \cdot i$ . Побудувати графік залежності середнього значення отриманого оптимуму цільової функції від ймовірності схрещування.

### ① Варіант №26

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{26}(x_1, x_2) = -e^{\sqrt{x_1^2 + x_2^2} + 3(\cos 2x_1 + \sin 2x_2)}$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – однорідне, мутація – гауссовська; кількість елітних особин – 8; розмір популяції – 170 особин.

Як аргумент  $S$  програма приймає параметр гауссовської мутації scale. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції patternsearch.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обгрунтувати.

2. За допомогою розробленої програми дослідити вплив параметру *scale* на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в *i*-й серії випробувань значення параметру *scale* встановлювати за формулою:  $scale_i = 0,06 + 0,075 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від параметру *scale*. Вивести на екран графік синтезованої моделі.

### ① Варіант №27

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{27}(x_1, x_2) = -x_1 \sin \sqrt{|x_1 - x_2 - 9|} - (x_2 + 9) \sin \sqrt{|x_2 + 0,5x_1 + 9|}$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – евристичне, мутація – гауссовська; кількість елітних особин – 5. Використовувати острівну модель еволюційної оптимізації (43 особини на першому острові, 41 – на другому, 55 – на третьому).

Як аргумент *S* програма приймає параметр гауссовської мутації *shrink*. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив параметру *shrink* на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в *i*-й серії випробувань значення параметру *shrink* встановлювати за формулою:  $shrink_i = 0,05 + 0,065 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від параметру *shrink*. Вивести на екран графік синтезованої моделі.

### ① Варіант №28

1. Розробити за допомогою пакету Matlab програму, що оптимізує

функцію  $f_{28}(x_1, \dots, x_n) = 1 + \sum_{i=1}^n \frac{x_i^2}{1000} - \prod_{i=1}^n \cos(x_i)$ ,  $n = 5$ , використовуючи

еволюційний пошук. В якості еволюційних операторів обрати: відбір – за допомогою рулетки, схрещування – арифметичне, мутація – гауссов-

ська. Використовувати острівну модель еволюційної оптимізації (55 особин на першому острові, 61 – на другому).

Як аргумент  $k_e$  програма приймає кількість елітних особин. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обгунтувати.

2. За допомогою розробленої програми дослідити вплив кількості елітних особин на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань значення кількості елітних особин встановлювати за формулою:  $k_{e,i} = i$ . Побудувати лінійну регресійну модель залежності середнього значення часу оптимізації від кількості елітних особин. Вивести на екран графік синтезованої моделі.

### ① Варіант №29

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{29}(x_1, \dots, x_n) = 20 + e - 20 \exp\left(-0,2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$ ,

$n = 8$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – одноточкове, мутація – проста; кількість елітних особин – 2. Використовувати острівну модель еволюційної оптимізації (53 особини на першому острові, 77 – на другому).

Як аргумент  $P_m$  програма приймає ймовірність міграції. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно виводити відповідне повідомлення.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обгунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності міграції на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань значення ймовірності міграції встановлювати за формулою:  $P_{m,i} = 0,07 + 0,09 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від ймовірності міграції. Вивести на екран графік синтезованої моделі.

### ① Варіант №30

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{30}(x_1, x_2) = (x_1^2 + x_2^2)/2 - \cos(20\pi x_1)\cos(20\pi x_2) + 2$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – турнірний, схрещування – одноточкове, мутація – проста; кількість елітних особин – 5; розмір популяції – 145 особин.

Як аргумент  $P_m$  програма приймає ймовірність мутації генів хромосоми при простій мутації. При цьому програма повинна перевіряти коректність введених даних, у випадку некоректності яких повинно вивести відповідне повідомлення.

Застосувати гібридну модель, що використовує результат еволюційного пошуку як початкову точку для оптимізації за допомогою функції `fminsearch`.

Інші параметри еволюційного пошуку обрати самостійно. Вибір параметрів обґрунтувати.

2. За допомогою розробленої програми дослідити вплив ймовірності мутації генів на ефективність еволюційного пошуку. Для цього послідовно провести 10 серій зі 150 експериментів кожна. При цьому в  $i$ -й серії випробувань ймовірність мутації генів встановлювати за формулою:  $P_{m,i} = 0,009 + 0,015 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення часу оптимізації від ймовірності мутації генів. Вивести на екран графік синтезованої моделі.



*Завдання 3.* На алгоритмічній мові програмування високого рівня (наприклад, Сі, Паскалі, мові пакету Matlab) створити програму, що реалізує одну з моделей генетичного пошуку.

## 6.2 Дослідження еволюційних операторів



*Завдання 4.* Дослідити еволюційні оператори схрещування та мутації. Отримати нові хромосоми, застосувавши еволюційні оператори схрещування (одноточкове, двоточкове, однорідне, рівномірне, порівняльне, просте, дискретне, плоске, арифметичне, лінійне арифметичне, геометричне, змішане, евристичне) та мутації (проста мутація, випадкова мутація) до заданих хромосом (табл. 6.1). Завдання виконати вручну. Описати хід розв'язання.

① Таблиця 6.1 – Значення хромосом для дослідження еволюційних операторів

№ варіанту	Схрещування				Мутація
	одноточкове, двоточкове, однорідне, рівномірне, порівняльне		просте, дискретне, плоске, арифметичне, лінійне арифметичне, геометричне, змішане, евристичне		
	$H_1$	$H_2$	$H_1$	$H_2$	$H$
1	00001000	01011011	57358612	12053461	67545338
2	11001001	10011100	86157584	51967592	62423846
3	00001010	01011101	62264572	15198476	65434724
4	11001011	10011110	14377256	72205273	56745121
5	00001100	01011111	23483445	38314635	24556343
6	11001101	10111111	72598642	24424175	97367252
7	00001110	01111110	51603457	32536371	46978363
8	11001111	10111101	27715721	13642673	85289361
9	00011000	01111100	83822574	75755243	34590314
10	11011001	10111011	55935813	37865312	65413452
11	00011010	01111010	40164735	82975372	48834136
12	11011011	10111001	75275932	83108371	95545461
13	00011100	01111000	68387905	18215167	47756352
14	11011101	10101111	85499053	94328362	42267125
15	00011110	01011011	32507452	82431636	37878432
16	11011111	10011100	15615832	52544326	51489264
17	01111111	00101000	64724831	78654714	48390573
18	10111110	11101001	53831357	37763715	46814878
19	01111101	00101010	11947532	18874732	52525987
20	10111100	11101011	35175625	18986836	72536085
21	01111011	00101100	64284271	15115278	58947874
22	10111010	11101101	75391378	18229254	24858674
23	01111001	00101110	85405216	18330860	57469573
24	10111000	11101111	56516424	38440978	47270584
25	01101111	00111000	76624624	42550978	18481573
26	10101110	11111001	57734318	38660867	46192468
27	01101101	00111010	90849853	47770864	49715635
28	10101100	11111011	89954678	62885724	88426583
29	01101011	00111100	76186453	53994615	45732567
30	10101010	11111101	65294574	54127245	54841862

Вважати, що гени числових хромосом приймають значення в інтервалі [0; 9]. Інші параметри, необхідні для виконання еволюційних операторів, обрати самостійно. Вибір параметрів обґрунтувати.



*Завдання 5.* Розробити за допомогою пакету Matlab програмне забезпечення, що реалізує 2 методи еволюційного пошуку з використанням різних еволюційних операторів. Еволюційні оператори програмно реалізувати самостійно (без використання вбудованих функцій). Основні еволюційні оператори для реалізації еволюційних методів обрати з таблиці 6.2 відповідно до варіанту.

① Таблиця 6.2 – Еволюційні оператори для виконання завдання

№ варіанту	№ завдання	Еволюційні оператори		
		Відбір	Схрещування	Мутація
1	1	пропорційний	однорідне	гауссовська
	2	ранжирування	рівномірне	нерівномірна
2	1	рулетка	арифметичне	проста
	2	пороговий	порівняльне	випадкова
3	1	турнірний	одноточечне	гауссовська
	2	ранжирування	діагональне	нерівномірна
4	1	пропорційний	двохточечне	проста
	2	пороговий	рівномірне	нерівномірна
5	1	рулетка	однорідне	гауссовська
	2	ранжирування	порівняльне	нерівномірна
6	1	турнірний	арифметичне	проста
	2	пороговий	діагональне	випадкова
7	1	пропорційний	одноточечне	гауссовська
	2	ранжирування	рівномірне	випадкова
8	1	рулетка	двохточечне	проста
	2	пороговий	порівняльне	нерівномірна
9	1	турнірний	однорідне	гауссовська
	2	ранжирування	діагональне	випадкова
10	1	пропорційний	арифметичне	проста
	2	пороговий	рівномірне	випадкова

Інші параметри, необхідні для еволюційного пошуку, обрати самостійно. Вибір параметрів обґрунтувати.



**Завдання 6.** Виконати тестування розробленого у попередньому завданні програмного забезпечення за допомогою вирішення задач оптимізації тестових функцій. Тестові функції  $y_i$  (не менше п'яти) для виконання тестування програми обрати самостійно. Вибір тестових функцій обґрунтувати.



**Завдання 7.** Порівняти одержані результати оптимізації різних функцій за допомогою обох реалізованих у завданні № 5 еволюційних методів. Результати порівняльного аналізу звести до таблиці, попередньо розробивши систему критеріїв порівняння методів еволюційного пошуку.

### 6.3 Статистичний аналіз результатів еволюційної оптимізації



**Завдання 8.** За допомогою пакету Matlab розробити програмне забезпечення, що реалізує метод еволюційного пошуку. Параметри еволюційного методу обрати з табл. 6.3 відповідно до варіанту.

① Таблиця 6.3 – Параметри еволюційного пошуку для виконання завдання

№ варіанту	Еволюційні оператори		
	Відбір	Схрещування	Мутація
1	рулетка	арифметичне	проста
2	турнірний	одноточечне	гауссовська
3	пропорційний	двохточечне	проста
4	рулетка	однорідне	гауссовська
5	турнірний	арифметичне	проста
6	пропорційний	одноточечне	гауссовська
7	рулетка	двохточечне	проста
8	турнірний	однорідне	гауссовська
9	пороговий	арифметичне	проста
10	рулетка	одноточечне	нерівномірна

Інші параметри, необхідні для еволюційного пошуку, обрати самостійно. Вибір параметрів обґрунтувати.

Отримати масив даних  $F = \{y_{1\min,j}\}$ ,  $y = 1, 2, \dots, 100$ , для статистичного аналізу. Для цього провести серію зі 100 експериментів з фіксованими параметрами еволюційного пошуку для вирішення задачі оптимізації тестової функції  $y_1$ , обраної при виконанні завдання № 6.



**Завдання 9.** На основі отриманого при виконанні завдання № 8 масиву даних  $F$  побудувати гістограму, таблицю частот та одержати числові характеристики розподілу ймовірностей (математичне очікування, дисперсію, асиметрію, ексцес).



**Завдання 10.** Перевірити гіпотезу про нормальний розподіл вибірки  $F$  (див. завдання № 9), використовуючи критерій Пірсона  $\chi^2$  і критерій Колмогорова.



**Завдання 11.** Аналогічно завданню № 8, послідовно провести 10 серій зі 100 експериментів кожна. При цьому в кожній серії необхідно у відповідності із заданим кроком змінювати один із параметрів еволюційного пошуку (відповідно до табл. 6.4).

① Таблиця 6.4 – Параметри еволюційного пошуку для аналізу

№ варіанту	Параметр	Значення параметру для 1-ої серії випробувань	Крок зміни параметру
1	$T$	10	10
2	$P_{\text{схрещування}}$	0,05	0,1
3	$P_{\text{мутації гену}}$	0,05	0,1
4	scale (параметр гауссовської мутації)	0,05	0,1
5	$m$ (розмір турніру при турнірному відборі)	1	1
6	shrink (параметр гауссовської мутації)	0,05	0,1
7	$k_e$	1	1
8	$N$	10	10
9	$\Pi$ (поріг відбору)	0,05	0,1
10	$k$ (параметр нерівномірної мутації)	1	1

За допомогою однофакторного дисперсійного аналізу дослідити вплив параметру (фактору), що регулювався (табл. 6.4), на досягнуте оптимальне значення функції  $y_1$ .




**Завдання 12.** Побудувати таблицю  $(X,Z)$ , де  $x_i$  – значення параметру, що регулювався у завданні № 11 ( $i = 1, 2, \dots, 10$ ),  $z_i$  – середнє зна-

чення досягнутого оптимального значення функції  $u_1$  в  $i$ -ій серії експериментів.

Виконати статистичну перевірку взаємозв'язку між  $X$  та  $Z$ , використовуючи методи кореляційного (коефіцієнт парної кореляції, коефіцієнт кореляції Пірсона, коефіцієнт кореляції Спірмена) та лінійного регресійного аналізів.

Зробити висновки щодо одержаних результатів, зокрема про вплив досліджуваного параметру еволюційного пошуку на досягнення оптимального значення функції, що оптимізується, а також рекомендації щодо вибору значення досліджуваного параметру при еволюційному пошуку.


#### 6.4 Комбінаторна оптимізація за допомогою еволюційних методів

 *Завдання 13.* Розробити за допомогою пакету Matlab програмне забезпечення для вирішення задачі комівояжера. Параметри еволюційного методу обрати з табл. 6.5 відповідно до варіанту.

① Таблиця 6.5 – Параметри еволюційного пошуку для виконання завдання

№ варіанту	Еволюційні оператори	
	Схрещування	Мутація
1	одноточечне впорядковуюче	класична мутація обміну
2	PMX	одноточечна мутація обміну
3	циклове	мутація золотого перетину
4	жадібне	класичне інвертування
5	двохточечне впорядковуюче	інвертування із зсувом
6	позиційно впорядковуюче	класична мутація обміну
7	циклове	одноточечна мутація обміну
8	жадібне	мутація золотого перетину
9	PMX	класичне інвертування
10	позиційно впорядковуюче	інвертування із зсувом

Інші параметри, необхідні для еволюційного пошуку, обрати самостійно. Вибір параметрів обґрунтувати.

 *Завдання 14.* Виконати тестування розробленого у завданні № 13 програмного забезпечення за допомогою вирішення конкретних

прикладів задачі комівояжера. Задачі (не менше трьох) для виконання тестування програми сформулювати самостійно. Вибір тестових задач обґрунтувати.

Порівняти одержані результати вирішення різних прикладів задачі комівояжера. Результати порівняльного аналізу звести до таблиці, попередньо розробивши систему критеріїв порівняння результатів вирішення задачі комівояжера.

## 6.5 Генетичне програмування



*Завдання 15.* Розробити програму, що за допомогою методів генетичного програмування відновлює функціональну залежність за заданим набором вхідних даних. Генетичні оператори та тестові набори даних обрати самостійно. Протестувати розроблене програмне забезпечення та виконати аналіз отриманих результатів.

## 6.6 Багатокритеріальна еволюційна оптимізація



*Завдання 16.* На алгоритмічній мові програмування високого рівня (наприклад, Сі, Паскалі, мові пакету Matlab) створити програму, що реалізує один з методів багатокритеріальної еволюційної оптимізації.

Виконати тестування розробленої програми за допомогою обраних тестових функцій, розв'язавши задачі багатокритеріальної оптимізації.

Порівняти одержані результати вирішення різних задач багатокритеріальної оптимізації. Результати порівняльного аналізу звести до таблиці, попередньо розробивши відповідну систему критеріїв.

## 6.7 Еволюційний синтез нейромережових моделей



*Завдання 17.* Розробити програму, що за допомогою еволюційного пошуку реалізує один з етапів побудови нейромережових моделей:

- відбір інформативних ознак;
- параметричний синтез;
- структурний синтез;
- оптимізація структури.

Виконати тестування розробленої програми, побудувавши нейромережові моделі для різних вибірок даних.

Порівняти одержані результати синтезу нейронних мереж. Результати порівняльного аналізу звести до таблиці, попередньо розробивши відповідну систему критеріїв.

## 7 ТЕСТОВІ ЗАВДАННЯ

### 7.1 Завдання першого рівня



*Завдання 1.* До методів еволюційного пошуку не відносяться:

- А Генетичні алгоритми.
- Б Еволюційні стратегії.
- В Генетичне програмування.
- Г Еволюційне програмування.
- Д Генетичні стратегії.
- Е Серед відповідей А–Д правильної немає.



*Завдання 2.* Визначити, яка властивість не відноситься до методів еволюційного пошуку.

- А Багатовимірність.
- Б Стохастичність.
- В Евристичність.
- Г Випадкова направленість пошуку.
- Д Детермінованість.
- Е Всі перелічені відповіді А–Д є властивостями еволюційного пошуку.



*Завдання 3.* Який термін в теорії еволюційного пошуку є аналогом (синонімом) поняття “цільова функція”?

- А Фітнес-функція.
- Б Функція придатності.
- В Функція пристосованості.
- Г Функція оцінки.
- Д Фітнес.
- Е Всі з наведених термінів можуть застосовуватися як синоніми поняття “цільова функція”.



*Завдання 4.* Якого оператора відбору не існує?

- А Пропорційний.
- Б Одноточковий.
- В Турнірний.
- Г Відбір ранжируванням.
- Д Відбір з використанням порогу.
- Е Існують усі види операторів відбору, наведені у відповідях А–Д.



*Завдання 5.* Якщо  $N$  – кількість хромосом в популяції, а  $f_j$  – значення фітнес-функції  $j$ -ої хромосоми, то середня пристосованість популяції розраховується за формулою ...



**Е** Усі оператори відбору, наведені у відповідях А–Д, не використовують сортування популяції.



**Завдання 9.** При якому операторі відбору не виконується сортування популяції за значенням фітнес-функції?

**А** Відбір з використанням порогу.

**Б** Відбір лінійним ранжируванням.

**В** Турнірний відбір.

**Г** Відбір рівномірним ранжируванням.

**Д** Відбір усіканням.

**Е** Усі оператори відбору, наведені у відповідях А–Д, використовують сортування популяції.



**Завдання 10.** Відомо, що бінарна хромосома  $H = 011100110$  містить інформацію про ознаку, яка приймає значення в інтервалі  $[15; 23]$ . Визначте похибку кодування ознаки в хромосомі.

**А**  $2^{-7}$ .

**Б**  $2^{-8}$ .

**В**  $2^{-9}$ .

**Г**  $2^{-10}$ .

**Д**  $2^{-11}$ .

**Е** Інша відповідь.



**Завдання 11.** Розрахуйте кількість елітних особин для популяції з 150 хромосом, що має ступінь відновлення  $S_0 = 0,98$ .

**А** 1.

**Б** 2.

**В** 3.

**Г** 4.

**Д** 5.

**Е** Інша відповідь.



**Завдання 12.** Відомо, що на 82-й ітерації еволюційного пошуку максимуму певної функції найкраще значення фітнес-функції складало 0,3. При якому значенні фітнес-функції еволюційний метод не зупинить свою роботу, якщо одним з критеріїв зупину є необхідність покращення досягнутого на кожній ітерації результату не менше, ніж на 3% від результату, отриманого на попередній ітерації?

**А** Більше 0,291.

**Б** Менше 0,291.

**В** Більше 0,309.

**Г** Менше 0,309.

**Д** Більше 0,3.

**Е** Серед відповідей А–Д правильної немає.



**Завдання 13.** Обчислити величину  $P_s(j)$ , в залежності від якої буде формуватися масив допущених до схрещування особин, при відборі за допомогою лінійного ранжирування, якщо  $j = 11$ ,  $N = 101$ ,  $\eta_{\max} = 2$ ,

де  $j$  – номер хромосоми в відсортованій популяції,  $N$  – кількість хромосом популяції,  $\eta_{\max}$  – параметр лінійного ранжирування.

- А  $P_s(j) = \frac{0,9}{100}$ .       Б  $P_s(j) = \frac{3,6}{202}$ .       В  $P_s(j) = \frac{3,6}{102}$ .  
 Г  $P_s(j) = \frac{1,8}{101}$ .       Д  $P_s(j) = \frac{0,9}{101}$ .       Е Інша відповідь.



**Завдання 14.** Обчислити величину  $P_s(j)$ , в залежності від якої буде формуватися масив допущених до схрещування особин, при відборі за допомогою рівномірного ранжирування, якщо  $j = 20$ ,  $\mu = 10$ ,  $N = 40$ , де  $j$  – номер хромосоми в відсортованій популяції,  $\mu$  – параметр рівномірного ранжирування;  $N$  – кількість хромосом популяції.

- А 0,05.       Б 0,5.       В 0.       Г 0,25.  
 Д 0,1.       Е Інша відповідь.



**Завдання 15.** Яка з хромосом  $H_3 - H_9$  ( $H_3 = (10001100)$ ,  $H_4 = (11010110)$ ,  $H_5 = (10110010)$ ,  $H_6 = (10000000)$ ,  $H_7 = (01100000)$ ,  $H_8 = (10000011)$ ,  $H_9 = (01100011)$ ) є результатом односточкового схрещування хромосом  $H_1 = (10000010)$  та  $H_2 = (00110010)$ . Точка схрещування – після 2-го гену.

- А  $H_7$ .       Б  $H_3$ .       В  $H_4$ .       Г  $H_5$ .  
 Д  $H_8$ .       Е Серед відповідей А–Д вірної немає.



**Завдання 16.** Яка з хромосом  $H_3 - H_9$  ( $H_3 = (10000001)$ ,  $H_4 = (00111110)$ ,  $H_5 = (00000110)$ ,  $H_6 = (10000010)$ ,  $H_7 = (11111011)$ ,  $H_8 = (10100110)$ ,  $H_9 = (11111000)$ ) є результатом двоточкового схрещування хромосом  $H_1 = (10010110)$  та  $H_2 = (00100110)$ . Точки схрещування – після 2-го та 4-го генів.

- А  $H_5$ .       Б  $H_6$ .       В  $H_7$ .       Г  $H_8$ .  
 Д  $H_9$ .       Е Серед відповідей А–Д вірної немає.



**Завдання 17.** Які хромосоми серед  $H_3 - H_8$  ( $H_3 = (3,5 \ 6 \ 5,5 \ 3 \ 6,5 \ 6,5 \ 6 \ 6,5)$ ,  $H_4 = (2 \ 5 \ 7 \ 3 \ 6 \ 5 \ 7 \ 6)$ ,  $H_5 = (2 \ 5 \ 7 \ 3 \ 7 \ 5 \ 5 \ 7)$ ,  $H_6 = (2 \ 5 \ 7 \ 3 \ 7 \ 8 \ 7 \ 6)$ ,  $H_7 = (5 \ 7 \ 4 \ 4 \ 7 \ 8 \ 7 \ 6)$ ,  $H_8 = (5 \ 7 \ 4 \ 3 \ 6 \ 5 \ 5 \ 7)$ ) є результатом простого схрещування числових хромосом  $H_1 = (2 \ 5 \ 7 \ 3 \ 6 \ 5 \ 5 \ 7)$  та  $H_2 = (5 \ 7 \ 4 \ 3 \ 7 \ 8 \ 7 \ 6)$ . Точка схрещування – після 4-го гену.

- А  $H_3$  та  $H_5$ .       Б  $H_4$  та  $H_6$ .       В  $H_5$  та  $H_7$ .  
 Г  $H_6$  та  $H_8$ .       Д  $H_5$  та  $H_8$ .

Е Серед відповідей А–Д вірної немає.



*Завдання 18.* Які хромосоми серед  $H_2 - H_7$  ( $H_2 = (6\ 7\ 8\ 7\ 3\ 5\ 8\ 8)$ ,  $H_3 = (6\ 7\ 9\ 6\ 4\ 4\ 8\ 8)$ ,  $H_4 = (7\ 8\ 8\ 6\ 4\ 4\ 8\ 8)$ ,  $H_5 = (6\ 7\ 9\ 6\ 3\ 4\ 8\ 8)$ ,  $H_6 = (6\ 8\ 8\ 6\ 4\ 4\ 9\ 8)$ ,  $H_7 = (6\ 7\ 8\ 6\ 4\ 4\ 8\ 8)$ ) можуть бути результатом простої мутації хромосоми  $H_1 = (6\ 7\ 8\ 6\ 3\ 4\ 8\ 8)$ .

А  $H_2$  та  $H_4$ .  Б  $H_3$  та  $H_5$ .  В  $H_5$ .  Г  $H_5$  та  $H_7$ .

Д Усі наведені хромосоми ( $H_2 - H_7$ ) можуть бути результатом простої мутації хромосоми  $H_1$ .

Е Серед відповідей А–Д вірної немає.



*Завдання 19.* Яка команда середовища Matlab призначена для зміни значень параметрів генетичної оптимізації?

А gasetoptions.  Б gaoptimset.  В gaoptimget.

Г gatool.  Д gaset.

Е Серед відповідей А–Д вірної немає.



*Завдання 20.* За допомогою якого поля структури options задається ймовірність схрещування хромосом для генетичної оптимізації?

А CrossoverFcn.  Б EliteCount.  В CrossoverFraction.

Г PopInitType.  Д CrossoverProbability.

Е Серед відповідей А–Д вірної немає.

## 7.2 Завдання другого рівня



*Завдання 21.* Яке з наступних тверджень є невірним?

А Еволюційні методи є методами багатовимірного пошуку.

Б Еволюційні методи основані на аналогії з природними процесами селекції та генетичними перетвореннями.

В Еволюційні методи є методами локального пошуку.

Г Еволюційні методи можуть використовуватися для оптимізації одновимірних функцій.

Д Еволюційні методи не використовують значення похідних цільової функції.

Е Серед відповідей А–Д усі вірні.



*Завдання 22.* Визначте вірне твердження.

- А** Мутації – спонтанні зміни в генах, які випадковим чином розкидають рішення по всьому простору пошуку.
- Б** Мутації – спонтанні зміни в хромосомах, які чітко визначеним чином розкидають рішення по всьому простору пошуку.
- В** Мутації – спонтанні зміни в генах, які випадковим чином розкидають рішення по частині простору пошуку.
- Г** Мутації – спонтанні зміни в генах, які чітко визначеним чином розкидають рішення по частині простору пошуку.
- Д** Мутації – спонтанні зміни в хромосомах, які випадковим чином розкидають рішення по частині простору пошуку.
- Е** Серед відповідей А–Д вірної немає.



*Завдання 23.* Ген у векторних хромосомах має властивості ...

- А** негомологічної числової хромосоми.
- Б** гомологічної числової хромосоми.
- В** векторної хромосоми.
- Г** бінарної хромосоми.
- Д** матричної хромосоми.
- Е** серед відповідей А–Д вірної немає.

*Правильна відповідь: А*



*Завдання 24.* При розв’язанні задачі оптимізації функції за допомогою еволюційному пошуку в генах хромосоми представляється інформація про:

- А** незалежні змінні та параметри-константи функції, що оптимізується.
- Б** параметри-константи функції, що оптимізується.
- В** значення функції, що оптимізується.
- Г** незалежні змінні та значення функції, що оптимізується.
- Д** незалежні змінні функції, що оптимізується.
- Е** серед відповідей А–Д вірної немає.



*Завдання 25.* Визначте кількість генів хромосоми при розв’язанні задачі апроксимації з використанням функції вигляду

$f = ax_1 + bx_2x_3 + cx_4 + d^{x_5x_6} + e$ , де  $x_1 - x_6$  – незалежні змінні апроксимуючої функції;  $a, b, c, d, e$  – параметри апроксимуючої функції.

А  
 Д

7.  
11.

Б  
 Е

4.

Інша відповідь.

В

6.

Г

5.



**Завдання 26.** Ймовірність знаходження глобального оптимуму за допомогою еволюційного пошуку значно зменшується, якщо фітнес-функція ...

А  
 Б  
 В  
 Г  
 Д

не містить плоскі ділянки.

є адекватною задачі.

містить ділянки з різким стрибком або спадом.

вимагає мінімум апаратних ресурсів.

має значення, розподіл яких збігається з розподілом реальної якості рішень.

Е

серед відповідей А–Д правильної немає.



**Завдання 27.** Еволюційні методи починають свою роботу з ...

А  
 Б  
 В  
 Г  
 Д  
 Е

перетворення генофонду популяції у фенотип популяції.

генерації фенотипу популяції.

перетворення фенотипу популяції у генофонд популяції.

відбору хромосом для виконання еволюційних операторів.

формування початкової популяції.

серед відповідей А–Д вірної немає.



**Завдання 28.** При якій стратегії створення початкової популяції генерується множина розв'язків, рівно віддалених один від одного?

А  
 Б  
 В  
 Г  
 Д  
 Е

Випадкове формування початкової популяції.

Рівномірне формування початкової популяції.

Формування початкової популяції, засноване на знаннях.

Прямокутне формування початкової популяції.

Сіткова ініціалізація, заснована на знаннях.

Серед відповідей А–Д вірної немає.



**Завдання 29.** У чому полягає принцип елітизму?

А

Нова популяція формується з елітних хромосом.

- Б До нової популяції гарантовано переходять особини, що мають найбільшу кількість нащадків.
- В Особини з найбільшою пристосованістю гарантовано переходять в нову популяцію.
- Г В схрещуванні гарантовано беруть участь хромосоми з найбільшою пристосованістю.
- Д Найбільш пристосовані особини гарантовано беруть участь у схрещуванні і мутації.
- Е Серед відповідей А–Д правильної немає.



*Завдання 30.* Метою якого (яких) оператору (-ів) є породження нової множини рішень, де кожна хромосома є нащадком декількох елементів попередньої популяції?

- А Відбір.  Б Схрещування.  В Мутація.
- Г Схрещування та мутація.  Д Відбір та мутація.
- Е Серед відповідей А–Д вірної немає.



*Завдання 31.* При яких методах формування батьківської пари враховується відстань між хромосомами?

- А Селективний метод та інбридинг.
- Б Інбридинг та аутбридинг.
- В Селективний метод, інбридинг та аутбридинг.
- Г Аутбридинг та селективний метод.
- Д Випадковий метод, селективний метод, інбридинг та аутбридинг.
- Е Серед відповідей А–Д правильної немає.



*Завдання 32.* Які методи формування батьківської пари для схрещування можуть ефективно застосовуватись до розв'язку задач з декількома екстремумами?

- А Випадковий і селективний методи.
- Б Селективний метод і аутбридинг.
- В Аутбридинг і інбридинг.
- Г Аутбридинг, інбридинг, випадковий метод, селективний метод.
- Д Інбридинг і селективний метод.
- Е Серед відповідей А–Д правильної немає.



**Завдання 33.** Який оператор схрещування не може бути застосований до бінарних хромосом?

- А Двоточковий.     Б Рівномірний.     В Геометричний.  
 Д Порівняльний.     Е Усі наведені у відповідях А–Д оператори схрещування можуть застосовуватися до бінарних хромосом.



**Завдання 34.** Який оператор схрещування не може бути застосований до еволюційної оптимізації функції  $f(x_1, x_2) = x_1 \sin(4x_1) + 1,1x_2 \sin(2x_2)$ , якщо при оптимізації використовуються числові гомологічні хромосоми?

- А Триточковий.     Б Лінійний арифметичний.  
 В Нерівномірний арифметичний.  
 Г Одноточковий.     Д Геометричний.  
 Е Усі наведені у відповідях А–Д оператори схрещування можуть застосовуватися для еволюційної оптимізації заданої функції.



**Завдання 35.** Нехай випадковим чином згенеровано набір чисел (0,3 0,7 0,4 0,4 0,5 0,3 0,6 0,9). Яка з хромосом  $H_2 - H_8$  ( $H_2 = (00011011)$ ,  $H_3 = (00111101)$ ,  $H_4 = (11100001)$ ,  $H_5 = (10010011)$ ,  $H_6 = (11000011)$ ,  $H_7 = (10010101)$ ,  $H_8 = (10001110)$ ) є результатом однорідного схрещування хромосом  $H_0 = (10011011)$  та  $H_1 = (00000111)$ .

- А  $H_2$ .     Б  $H_3$ .     В  $H_4$ .     Г  $H_5$ .  
 Д  $H_6$ .     Е Серед відповідей А–Д вірної немає.



**Завдання 36.** Які хромосоми серед  $H_3 - H_9$  ( $H_3 = (10001000)$ ,  $H_4 = (10001010)$ ,  $H_5 = (00110011)$ ,  $H_6 = (00111001)$ ,  $H_7 = (10100011)$ ,  $H_8 = (00100111)$ ,  $H_9 = (10111101)$ ) є результатом рівномірного схрещування хромосом  $H_1 = (10110011)$  та  $H_2 = (00100011)$ . Маска схрещування MASK = (11001110).

- А  $H_4$  та  $H_6$ .     Б  $H_3$  та  $H_7$ .     В  $H_5$  та  $H_8$ .  
 Г  $H_4$  та  $H_8$ .     Д  $H_5$  та  $H_7$ .  
 Е Серед відповідей А–Д вірної немає.



**Завдання 37.** Яка з хромосом  $H_3 - H_9$  ( $H_3 = (10011001)$ ,  $H_4 = (00101010)$ ,  $H_5 = (00000100)$ ,  $H_6 = (10010001)$ ,  $H_7 = (11011000)$ ,  $H_8 = (00100111)$ ,  $H_9 = (10001011)$ ) є результатом порівняльного схрещу-

вання хромосом  $H_1 = (11101100)$  та  $H_2 = (00010101)$ . Вважати, що всі випадково згенеровані числа менше 0,5.

- А  $H_7$ .                       Б  $H_3$ .                       В  $H_4$ .                       Г  $H_5$ .  
 Д  $H_6$ .                       Е Серед відповідей А–Д вірної немає.



**Завдання 38.** Нехай випадковим чином згенеровано набір чисел  $(0,8\ 0,2\ 0,8\ 0,5\ 0,9\ 0,2\ 0,1\ 0,4)$ . Яка з хромосом  $H_2 - H_7$  ( $H_2 = (7\ 6\ 7\ 5\ 7\ 3\ 3\ 2)$ ,  $H_3 = (4\ 6\ 2\ 5\ 7\ 3\ 5\ 2)$ ,  $H_4 = (4\ 6\ 2\ 5\ 7\ 3\ 5\ 5)$ ,  $H_5 = (4\ 4\ 2\ 5\ 8\ 3\ 5\ 5)$ ,  $H_6 = (4\ 4\ 2\ 5\ 8\ 3\ 5\ 2)$ ,  $H_7 = (9\ 6\ 9\ 8\ 8\ 7\ 3\ 5)$ ) є результатом дискретного схрещування хромосом  $H_0 = (7\ 6\ 7\ 5\ 8\ 3\ 5\ 5)$  та  $H_1 = (4\ 4\ 2\ 5\ 7\ 3\ 3\ 2)$ .

- А  $H_2$ .                       Б  $H_3$ .                       В  $H_4$ .                       Г  $H_5$ .  
 Д  $H_6$ .                       Е Серед відповідей А–Д вірної немає.



**Завдання 39.** Нехай  $H_1 - H_6$  – векторні хромосоми, що складаються з трьох генів:

$$\begin{aligned} H_1 &= (\{3\ 2\ 4\ 1\ 5\} \{2\ 5\ 4\ 1\ 3\} \{2\ 5\ 3\ 1\ 4\}), \\ H_2 &= (\{2\ 5\ 4\ 1\ 3\} \{3\ 2\ 4\ 1\ 5\} \{2\ 5\ 3\ 1\ 4\}), \\ H_3 &= (\{3\ 2\ 4\ 1\ 5\} \{2\ 5\ 1\ 4\ 3\} \{2\ 5\ 3\ 1\ 4\}), \\ H_4 &= (\{3\ 4\ 2\ 1\ 5\} \{2\ 5\ 4\ 1\ 3\} \{2\ 5\ 3\ 1\ 4\}), \\ H_5 &= (\{3\ 2\ 4\ 1\ 5\} \{2\ 1\ 4\ 5\ 3\} \{2\ 5\ 3\ 1\ 4\}), \\ H_6 &= (\{3\ 2\ 4\ 1\ 5\} \{2\ 5\ 4\ 1\ 3\} \{2\ 4\ 3\ 1\ 5\}). \end{aligned}$$

Які хромосоми серед  $H_2 - H_6$  можуть бути результатом простої мутації векторної хромосоми  $H_1$ ?

- А  $H_2$  та  $H_3$ .                       Б  $H_3$ .                       В  $H_3$  та  $H_4$ .  
 Г  $H_4$  та  $H_5$ .                       Д Усі наведені хромосоми ( $H_2 - H_6$ ) можуть бути результатом простої мутації хромосоми  $H_1$ .  
 Е Серед відповідей А–Д вірної немає.



**Завдання 40.** Які параметри генетичного пошуку встановлені у змінній options?

```
options = gaoptimset('PopulationSize', [10 20], ...
                    'MigrationInterval', 7, ...
                    'MigrationFraction', 0,8, ...
                    'MutationFcn', '@mutationgaussian')
```

- А Кількість хромосом в популяції – від 10 до 20; інтервал міграції – 7 ітерацій; ймовірність міграції – 0,8; оператор мутації – гауссовська мутація.

- Б** Кількість хромосом в популяції – 10 на першому острові, 20 на другому острові; інтервал міграції – 7 ітерацій; ймовірність міграції – 0,8; оператор мутації – гауссовська мутація.
- В** Кількість хромосом в популяції – 10 на першому острові, 20 на другому острові; інтервал міграції – 7 ітерацій; ймовірність міграції – 0,8; оператор мутації – випадкова мутація.
- Г** Кількість хромосом в популяції – від 10 до 20; інтервал міграції – 7 ітерацій; ймовірність схрещування – 0,8; оператор мутації – гауссовська мутація.
- Д** Кількість хромосом в популяції – 10 на першому острові, 20 на другому острові; інтервал міграції – 7 ітерацій; ймовірність міграції – 0,8; оператор мутації – випадкова мутація.
- Е** Серед відповідей А–Д вірної немає.

### 7.3 Завдання третього рівня



*Завдання 41.* Серед відповідей А–Д виберіть вірну.

- А** Найбільш пристосовані гени одержують можливість відтворювати нащадків.
- Б** Найбільш пристосовані хромосоми одержують можливість відтворювати нащадків.
- В** Найменш пристосовані гени одержують можливість відтворювати нащадків.
- Г** Найбільш пристосовані хромосоми одержують можливість відтворювати нащадків.
- Д** Всі відповіді А–Г є вірними.
- Е** Серед відповідей А–Д вірної немає.



*Завдання 42.* Серед відповідей А–Д виберіть вірну.

- А** Фенотип містить інформацію на рівні хромосомного набору.
- Б** Генотип фактично визначає, чим є об'єкт у реальному світі.
- В** Кожний ген хромосоми має своє відображення у фенотипі.
- Г** Функціонування механізмів еволюційного пошуку відбувається на рівні фенотипу.
- Д** Відповіді А та Г є вірними.
- Е** Серед відповідей А–Д вірної немає.



*Завдання 43.* Визначте невірне твердження.

- А
- Б
- В
- Г
- Д
- Е

Гомологічні хромосоми мають загальне походження.

Гомологічні хромосоми морфологічно подібні.

Гомологічні хромосоми генетично подібні.

Гомологічні хромосоми не утворюють неприпустимих рішень.

У гомологічних хромосомах кожний ген приймає значення в заданому інтервалі.

Всі перелічені відповіді А–Д є вірними.



*Завдання 44.* Серед відповідей А–Д виберіть вірну.

- А
- Б
- В
- Г
- Д
- Е

Бінарна хромосома є негомологічною числовою хромосомою.

Кожен ген бінарної хромосоми приймає значення в інтервалі  $(0; 1)$ .

Гени бінарної хромосоми можуть приймати значення  $-1$  та  $1$ .

Числові хромосоми можна розділити на гомологічні і векторні.

Бінарна хромосома може складатися з генів, які приймають однакові значення.

Всі перелічені відповіді А–Д є невірними.



*Завдання 45.* Серед відповідей А–Д виберіть невірну.

- А
- Б
- В
- Г
- Д
- Е

Бінарна хромосома є гомологічною числовою хромосомою.

Кожен ген бінарної хромосоми приймає значення в інтервалі  $(0; 1)$ .

У негомологічних хромосомах гени можуть приймати значення в заданому інтервалі.

В негомологічних хромосомах інтервал значень однаковий для всіх генів.

Для негомологічних хромосом застосовуються еволюційні оператори, що не створюють неприпустимих рішень.

Всі відповіді А–Д є правильними.



*Завдання 46.* Визначте невірне твердження.

- А
- Б
- В
- Г

Негомологічні хромосоми, як правило, застосовуються для розв'язку задач комбінаторної оптимізації.

Гени векторних хромосом представляють собою вектор цілих чисел.

Ген векторної хромосоми не може містити двох однакових чисел.

Векторні хромосоми є негомологічними.

- Д В негомологічній числовій хромосомі не може бути двох генів з однаковим значенням.
- Е Всі відповіді А–Д є вірними.



*Завдання 47.* Виберіть вірне твердження.

- А В методах еволюційного пошуку ситуація, коли обидва батьки при схрещуванні представлені одним елементом популяції, є неможливою.
- Б В результаті виконання схрещування найбільш пристосованих особин нащадки з великою імовірністю будуть або не набагато гіршими, ніж батьки, або кращими за них.
- В В еволюційних методах за передачу ознак батьків нащадкам відповідають оператори відбору, схрещування та мутації.
- Г Селективний вибір батьківської пари називається панміксією.
- Д При випадковому виборі батьківської пари кожна хромосома може стати членом тільки однієї пари.
- Е Серед відповідей А–Д правильної немає.



*Завдання 48.* Виберіть вірне твердження.

- А Спільне використання операторів відбору та схрещування призводить до того, що області простору пошуку з кращою в середньому оптимальністю вміщують більше членів популяції, ніж інші.
- Б Спільне використання операторів відбору та схрещування призводить до того, що області простору пошуку з гіршою в середньому оптимальністю вміщують більше членів популяції, ніж інші.
- В Спільне використання операторів відбору та схрещування призводить до того, що області простору пошуку з кращою в середньому оптимальністю вміщують менше членів популяції, ніж інші.
- Г Спільне використання операторів схрещування та мутації призводить до того, що області простору пошуку з кращою в середньому оптимальністю вміщують більше членів популяції, ніж інші.
- Д Спільне використання операторів схрещування та мутації призводить до того, що області простору пошуку з гіршою в середньому оптимальністю вміщують більше членів популяції, ніж інші.
- Е Серед відповідей А–Д вірної немає.



*Завдання 49.* Виберіть невірне твердження.

- А Метою оператору мутації є диверсифікація.

- Б. Оператор мутації використовується для підвищення розмаїтості пошуку та введення нових хромосом в популяцію.
- В. Оператор мутації полягає в зміні генів у випадково обраних позиціях.
- Г. Використання оператору мутації допомагає виходити з локальних екстремумів в процесі пошуку.
- Д. В результаті виконання мутації з ненульовою ймовірністю чергове рішення може перейти в будь-яке інше рішення.
- Е. Всі відповіді А–Д є вірними.



**Завдання 50.** Визначте розрядність бінарної хромосоми, необхідну для кодування двох ознак, перша з яких приймає значення в інтервалі  $[0,095; 0,607]$ , а друга – в інтервалі  $[0,248; 1,272]$ . Похибка кодування ознак в хромосомі  $\varepsilon = 0,002$ .

- А. 14.                       Б. 15.                       В. 16.                       Г. 17.
- Д. 18.                       Е. Інша відповідь.



**Завдання 51.** При формуванні батьківської пари обрано хромосому  $H_1 = (5; 3; 1)$ . Яка з наведених хромосом  $H_2 = (3; 4; 1)$ ,  $H_3 = (2; 2; 2)$ ,  $H_4 = (6; 7; 1)$ ,  $H_5 = (5; 8; 5)$ ,  $H_6 = (4; 3; 2)$ ,  $H_7 = (1; 1; 1)$  буде обрана як другий член батьківської пари при використанні інбридингу?

- А.  $H_2$ .                       Б.  $H_3$ .                       В.  $H_4$ .                       Г.  $H_5$ .
- Д.  $H_6$ .                       Е. Інша відповідь.



**Завдання 52.** Які хромосоми серед  $H_3 - H_8$  ( $H_3 = (5,3 3,8 5,7 5,8 2,8 3 7,1 6,1)$ ,  $H_4 = (3 1,2 3 3,6 2,4 1,2 4,2 3)$ ,  $H_5 = (4,2 4,2 4,8 4,2 1,2 3 5,4 5,4)$ ,  $H_6 = (5,8 4 6,2 6,4 3,2 3,2 7,8 6,6)$ ,  $H_7 = (6,2 5 6,8 6,6 2,8 3,8 8,2 7,4)$ ,  $H_8 = (5,5 4,3 6 5,9 2,6 3,3 7,3 6,5)$ ) є результатом арифметичного схрещування хромосом  $H_1 = (5 2 5 6 4 2 7 5)$  та  $H_2 = (7 7 8 7 2 5 9 9)$ , якщо  $k = 0,6$ .

- А.  $H_6$  та  $H_8$ .                       Б.  $H_3$  та  $H_7$ .                       В.  $H_4$  та  $H_8$ .
- Г.  $H_5$  та  $H_7$ .                       Д.  $H_6$  та  $H_7$ .
- Е. Серед відповідей А–Д вірної немає.



**Завдання 53.** Які хромосоми серед  $H_3 - H_8$  ( $H_3 = (3,5 2 3 4 2,5 4 3,5 4)$ ,  $H_4 = (5 5,5 6,5 5,5 4 7,5 6,5 7)$ ,  $H_5 = (5,4 5,2 6,4 6 4,2 7,6 6,6 7,2)$ ,  $H_6 = (9 2,5 5,5 10,5 6 8,5 7,5 9)$ ,  $H_7 = (1 8,5 7,5 0,5 2 6,5 5,5 5)$ ,  $H_8 = (2,4 5,6 5,6 2,4 2,4 5,6 4,8 4,8)$ ) є ре-

зультатом лінійного арифметичного схрещування хромосом  
 $H_1 = (7\ 4\ 6\ 8\ 5\ 8\ 7\ 8)$  та  $H_2 = (3\ 7\ 7\ 3\ 3\ 7\ 6\ 6)$ .

А  $H_3, H_5, H_7$ .  Б  $H_4, H_6, H_8$ .  В  $H_3, H_8$ .

Г  $H_4, H_6, H_7$ .  Д  $H_5, H_6, H_8$ .

Е Серед відповідей А–Д вірної немає.



**Завдання 54** Яка хромосома (які хромосоми) серед  $H_3 - H_8$   
 $H_3 = (8,5\ 4,1\ 5,3\ 2,1\ 5,3\ 4,2\ 6,2\ 5,9)$ ,  $H_4 = (3,6\ 4,5\ 7,2\ 2,7\ 7,2\ 5,4\ 7,2\ 4,5)$ ,  
 $H_5 = (-0,5\ 5,9\ 10,7\ 3,9\ 10,7\ 7,8\ 9,8\ 4,1)$ ,  $H_6 = (0,4\ 0,5\ 0,8\ 0,3\ 0,8\ 0,6\ 0,8\ 0,5)$ ,  
 $H_7 = (0,9\ 0,4\ 0,5\ 0,2\ 0,5\ 0,4\ 0,6\ 0,6)$ ,  $H_8 = (4,5\ 4,9\ 7,7\ 2,9\ 7,7\ 5,8\ 7,8\ 5,1)$ ) є  
 результатом евристичного схрещування хромосом  $H_1 = (4\ 5\ 8\ 3\ 8\ 6\ 8\ 5)$   
 та  $H_2 = (9\ 4\ 5\ 2\ 5\ 4\ 6\ 6)$ , якщо  $k = 0,9$ .

А  $H_4$ .  Б  $H_5$  та  $H_6$ .  В  $H_7$  та  $H_8$ .

Г  $H_3$ .  Д  $H_5$ .  Е Серед відповідей А–Д вірної немає.



**Завдання 55** Яка хромосома (які хромосоми) серед  $H_3 - H_8$   
 $H_3 = (6\ 4\ 5\ 4\ 7\ 4\ 6\ 7)$ ,  $H_4 = (6\ 4\ 5\ 4\ 7\ 4\ 6\ 7)$ ,  $H_5 = (4\ 8\ 5\ 3\ 2\ 6\ 6\ 7)$ ,  
 $H_6 = (5\ 4\ 4\ 3\ 6\ 4\ 5\ 6)$ ,  $H_7 = (6\ 4\ 4\ 3\ 6\ 4\ 7\ 7)$ ,  $H_8 = (4\ 3\ 4\ 3\ 6\ 4\ 3\ 5)$ ) є резуль-  
 татом змішаного схрещування хромосом  $H_1 = (3\ 5\ 4\ 3\ 7\ 4\ 9\ 8)$  та  
 $H_2 = (3\ 6\ 5\ 3\ 5\ 2\ 7\ 2)$ , якщо  $\alpha = 0,5$ . Вважати, що  $\text{rand}(a; b) = \text{Ціле}\left(\frac{a+b}{2}\right)$ .

А  $H_4$  та  $H_8$ .  Б  $H_6$ .  В  $H_7$ .

Г  $H_8$ .  Д  $H_6$  та  $H_7$ .

Е Серед відповідей А–Д вірної немає.

#### 7.4 Відповіді до тестових завдань

- |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|
| 1. Д.  | 11. В. | 21. В. | 31. Е. | 41. Б. | 51. Д. |
| 2. Д.  | 12. В. | 22. А. | 32. В. | 42. В. | 52. Д. |
| 3. Е.  | 13. Г. | 23. А. | 33. В. | 43. Е. | 53. Г. |
| 4. Б.  | 14. В. | 24. Д. | 34. А. | 44. Д. | 54. Д. |
| 5. В.  | 15. Г. | 25. Г. | 35. Г. | 45. Б. | 55. Б. |
| 6. Д.  | 16. Г. | 26. В. | 36. Д. | 46. Г. |        |
| 7. А.  | 17. Г. | 27. Д. | 37. Г. | 47. Б. |        |
| 8. А.  | 18. Г. | 28. Б. | 38. В. | 48. А. |        |
| 9. В.  | 19. Б. | 29. В. | 39. В. | 49. Е. |        |
| 10. Е. | 20. В. | 30. Б. | 40. Б. | 50. Г. |        |

## 7.5 Приклад білету до контролю знань студентів

### Рівень1 (по 2 бали)

1. Відомо, що на 82-й ітерації еволюційного пошуку максимуму певної функції найкраще значення фітнес-функції складало 0,3. При якому значенні фітнес-функції еволюційний метод не зупинить свою роботу, якщо одним з критеріїв зупину є необхідність покращення досягнутого на кожній ітерації результату не менше, ніж на 3% від результату, отриманого на попередній ітерації?

- А Більше 0,291.       Б Менше 0,291.       В Більше 0,309.  
 Г Менше 0,309.       Д Більше 0,3.  
 Е Серед відповідей А–Д правильної немає.

2. Обчислити величину  $P_s(j)$ , в залежності від якої буде формуватися масив допущених до схрещування особин, при відборі за допомогою рівномірного ранжирування, якщо  $j = 20$ ,  $\mu = 10$ ,  $N = 40$ , де  $j$  – номер хромосоми в відсортованій популяції,  $\mu$  – параметр рівномірного ранжирування;  $N$  – кількість хромосом популяції.

- А 0,02.       Б 0,3.       В 0,15.  
 Г 0.       Д 0,5.       Е Інша відповідь.

3. Які хромосоми серед  $H_3 - H_9$  ( $H_3 = (10011000)$ ,  $H_4 = (11100000)$ ,  $H_5 = (01110000)$ ,  $H_6 = (01011011)$ ,  $H_7 = (10101110)$ ,  $H_8 = (10101001)$ ,  $H_9 = (10111001)$ ) є результатом односточкового схрещування хромосом  $H_1 = (10010000)$  та  $H_2 = (01111000)$ . Точка схрещування – після 4-го гену.

- А  $H_4$  та  $H_7$ .       Б  $H_5$  та  $H_8$ .       В  $H_4$  та  $H_5$ .  
 Г  $H_3$  та  $H_5$ .       Д  $H_6$  та  $H_8$ .  
 Е Серед відповідей А–Д вірної немає.

4. Які хромосоми серед  $H_3 - H_8$  ( $H_3 = (3\ 7\ 2\ 4\ 4\ 3\ 5\ 5)$ ,  $H_4 = (3\ 7\ 8\ 6\ 4\ 3\ 5\ 5)$ ,  $H_5 = (3\ 2\ 8\ 6\ 4\ 2\ 8\ 3)$ ,  $H_6 = (3\ 7\ 8\ 6\ 4\ 3\ 8\ 3)$ ,  $H_7 = (3\ 4,5\ 5\ 5\ 4\ 2,5\ 6,5\ 4)$ ,  $H_8 = (3\ 2\ 2\ 4\ 5\ 3\ 5\ 5)$ ) є результатом простого схрещування числових хромосом  $H_1 = (3\ 7\ 8\ 6\ 4\ 2\ 8\ 3)$  та  $H_2 = (3\ 2\ 2\ 4\ 4\ 3\ 5\ 5)$ . Точка схрещування – після 2-го гену.

- А  $H_4$  та  $H_8$ .       Б  $H_3$  та  $H_5$ .       В  $H_4$  та  $H_6$ .  
 Г  $H_5$  та  $H_7$ .       Д  $H_6$  та  $H_8$ .  
 Е Серед відповідей А–Д вірної немає.

5. Яка команда середовища Matlab призначена для зміни значень параметрів генетичної оптимізації?

- А gasetoptions.       Б gaoptimset.       В gaoptimget.  
 Г gatool.       Д gaset.  
 Е Серед відповідей А–Д вірної немає.

### Рівень2 (по 4 бали)

6. За методами представлення генів хромосоми можна розподілити на:

- А бінарні, числові, векторні.  
 Б бінарні, дискретні, неперервні.  
 В числові, векторні, матричні.  
 Г гомологічні, негомологічні, векторні.  
 Д бінарні, векторні, матричні.  
 Е серед відповідей А–Д правильної немає.

7. Який оператор схрещування гомологічних числових хромосом виконується аналогічно оператору одноточкового схрещування бінарних хромосом?

- А Плоский.       Б Арифметичний.       В Простий.  
 Г Дискретний.       Д Геометричний.  
 Е Серед відповідей А–Д вірної немає.

8. Які хромосоми серед  $H_3 - H_9$  ( $H_3 = (01011111)$ ,  $H_4 = (00010010)$ ,  $H_5 = (00110011)$ ,  $H_6 = (11100111)$ ,  $H_7 = (00101000)$ ,  $H_8 = (10100010)$ ,  $H_9 = (00100010)$ ) є результатом рівномірного схрещування хромосом  $H_1 = (01011110)$  та  $H_2 = (00101011)$ . Маска схрещування  $MASK = (01000111)$ .

- А  $H_4$  та  $H_7$ .       Б  $H_3$  та  $H_8$ .       В  $H_5$  та  $H_6$ .  
 Г  $H_4$  та  $H_9$ .       Д  $H_4$  та  $H_8$ .  
 Е Серед відповідей А–Д вірної немає.

9. Нехай  $H_1 - H_6$  – векторні хромосоми, що складаються з трьох генів:

$$\begin{aligned}
 H_1 &= (\{1\ 4\ 2\ 5\ 3\} \{2\ 1\ 3\ 5\ 4\} \{2\ 4\ 3\ 5\ 1\}), \\
 H_2 &= (\{2\ 1\ 3\ 5\ 4\} \{1\ 4\ 2\ 5\ 3\} \{2\ 4\ 3\ 5\ 1\}), \\
 H_3 &= (\{1\ 4\ 2\ 5\ 3\} \{2\ 1\ 5\ 3\ 4\} \{2\ 4\ 3\ 5\ 1\}), \\
 H_4 &= (\{3\ 1\ 4\ 2\ 5\} \{2\ 1\ 3\ 5\ 4\} \{2\ 4\ 3\ 5\ 1\}), \\
 H_5 &= (\{1\ 4\ 2\ 5\ 3\} \{2\ 5\ 3\ 1\ 4\} \{2\ 4\ 3\ 5\ 1\}),
 \end{aligned}$$

$H_6 = (\{1\ 2\ 4\ 5\ 3\} \ \{2\ 1\ 3\ 5\ 4\} \ \{2\ 4\ 3\ 5\ 1\})$ .

Які хромосоми серед  $H_2 - H_6$  можуть бути результатом простої мутації векторної хромосоми  $H_1$ ?

- А  $H_2$  та  $H_4$ .     Б  $H_3$ .     В  $H_3$  та  $H_6$ .     Г  $H_5$  та  $H_6$ .  
 Д Усі наведені хромосоми ( $H_2 - H_6$ ) можуть бути результатом простої мутації хромосоми  $H_1$ .     Е Серед відповідей А-Д вірної немає.

10. Які параметри генетичного пошуку встановлені у змінній options?

```
options = gaoptimset('EliteCount', 2, ...
                    'CrossoverFraction', 0.9, ...
                    'Generations', 150, ...
                    'CrossoverFcn', '@crossoversinglepoint')
```

- А Кількість елітних особин – 2; ймовірність схрещування – 0,9; максимальна кількість хромосом в популяції – 150; оператор схрещування – односточкове схрещування.  
 Б Кількість елітних особин – 2; ймовірність схрещування – 0,9; максимальна кількість ітерацій генетичного пошуку – 150; оператор схрещування – односточкове схрещування.  
 В Кількість елітних особин – 2; ймовірність міграції – 0,9; максимальна кількість ітерацій генетичного пошуку – 150; оператор схрещування – односточкове схрещування.  
 Г Кількість елітних популяцій – 2; ймовірність схрещування – 0,9; максимальна кількість ітерацій генетичного пошуку – 150; оператор схрещування – односточкове схрещування.  
 Д Кількість елітних особин – 2; ймовірність схрещування – 0,9; максимальна кількість ітерацій генетичного пошуку – 150; оператор схрещування – двоточкове схрещування.  
 Е Серед відповідей А-Д вірної немає.

### Рівень3 (по 6 балів)

11. Виберіть невірне твердження.

- А Метою оператору мутації є диверсифікація.  
 Б Оператор мутації використовується для підвищення розмаїтості пошуку та введення нових хромосом в популяцію.  
 В Оператор мутації полягає в зміні генів у випадково обраних позиціях.  
 Г Використання оператору мутації допомагає виходити з локальних екстремумів в процесі пошуку.

- Д В результаті виконання мутації з ненульовою ймовірністю чергове рішення може перейти в будь-яке інше рішення.
- Е Всі відповіді А–Д є вірними.

12. При формуванні батьківської пари обрано хромосому  $H_1 = (9; 1; 7)$ . Яка з наведених хромосом  $H_2 = (7; 1; 4)$ ,  $H_3 = (5; 1; 6)$ ,  $H_4 = (9; 3; 9)$ ,  $H_5 = (8; 4; 5)$ ,  $H_6 = (6; 4; 9)$ ,  $H_7 = (7; 3; 6)$  буде обрана як другий член батьківської пари при використанні аутбридингу?

- А  $H_2$ .                       Б  $H_3$ .                       В  $H_4$ .  
 Г  $H_5$ .                       Д  $H_6$ .                       Е Інша відповідь.

13. Які хромосоми серед  $H_3 - H_8$  ( $H_3 = (2,9 6,4 3,2 4,1 5,5 4,8 2,8 5,5)$ ,  $H_4 = (1 1,6 1,6 1,8 1,4 1,6 0,8 1,4)$ ,  $H_5 = (0,6 1,4 0,6 0,8 1,2 1 0,6 1,2)$ ,  $H_6 = (3,4 7,2 4 5 6,2 5,6 3,2 6,2)$ ,  $H_7 = (2,4 5,6 2,4 3,2 4,8 4 2,4 4,8)$ ,  $H_8 = (4,6 7,8 7 8 6,8 7,4 3,8 6,8)$ ) є результатом арифметичного схрещування хромосом  $H_1 = (5 8 8 9 7 8 4 7)$  та  $H_2 = (3 7 3 4 6 5 3 6)$ , якщо  $k = 0,2$ .

- А  $H_3$  та  $H_5$ .                       Б  $H_4$  та  $H_6$ .                       В  $H_5$  та  $H_7$ .  
 Г  $H_6$  та  $H_8$ .                       Д  $H_7$  та  $H_8$ .  
 Е Серед відповідей А–Д вірної немає.

14. Яка хромосома (які хромосоми) серед  $H_3 - H_8$  ( $H_3 = (4 8 5 2 4 3 6 8)$ ,  $H_4 = (4 8 5 2 4 3 6 8)$ ,  $H_5 = (3 5 3 4 7 8 8 4)$ ,  $H_6 = (4 7 5 2 3 2 5 7)$ ,  $H_7 = (4 7 6 2 3 2 6 7)$ ,  $H_8 = (4 7 3 2 3 2 4 7)$ ) є результатом змішаного схрещування хромосом  $H_1 = (4 8 3 2 3 2 7 7)$  та  $H_2 = (4 7 7 2 4 3 4 8)$ , якщо  $\alpha = 0,7$ . Вважати, що  $\text{rand}(a; b) = \text{Ціле} \left( \frac{a+b}{2} \right)$ .

- А  $H_3$  та  $H_6$ .                       Б  $H_4$ .                       В  $H_6$ .  
 Г  $H_7$ .                       Д  $H_4$  та  $H_7$ .  
 Е Серед відповідей А–Д вірної немає.

15. Яка хромосома (які хромосоми) серед  $H_3 - H_8$  ( $H_3 = (3 4 4,8 8,6 1,4 6,8 3,8 6,6)$ ,  $H_4 = (1,2 0,8 2,4 3,6 0,4 3,2 0,4 2)$ ,  $H_5 = (3 0 7,2 9,4 0,6 9,2 -1,8 3,4)$ ,  $H_6 = (1,8 1,2 3,6 5,4 0,6 4,8 0,6 3)$ ,  $H_7 = (1,8 4,2 1,8 4,8 1,2 3 4,8 5,4)$ ,  $H_8 = (3 5 4,2 8,4 1,6 6,2 5,2 7,4)$ ) є результатом евристичного схрещування хромосом  $H_1 = (3 2 6 9 1 8 1 5)$  та  $H_2 = (3 7 3 8 2 5 8 9)$ , якщо  $k = 0,4$ .

- А  $H_4$  та  $H_6$ .                       Б  $H_5$ .                       В  $H_6$  та  $H_7$ .  
 Г  $H_8$ .                       Д  $H_6$ .

Е) Серед відповідей А–Д вірної немає.

Правильні відповіді:

1. В. 2. А. 3. Г. 4. Б. 5. Б. 6. А. 7. В. 8. Д. 9. В. 10. Б. 11. Е. 12.  
Д. 13. Г. 14. В. 15. Б.

## ЛІТЕРАТУРА

### Основна література

1. Encyclopedia of artificial intelligence / Eds.: J. R. Dopico, J. D. de la Calle, A. P. Sierra. – New York : Information Science Reference, 2009. – Vol. 1-3. – 1677 p.
2. Gen M. Genetic algorithms and engineering design / M. Gen, R. Cheng. – New Jersey : John Wiley & Sons, 1997. – 352 p.
3. Haupt R. Practical genetic algorithms / R. Haupt, S. Haupt. – New Jersey : John Wiley & Sons, 2004. – 261 p.
4. Емельянов В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик. – М. : Физматлит, 2003. – 432 с.
5. Курейчик В. М. Генетические алгоритмы: монография / В. М. Курейчик. – Таганрог : ТРТУ, 1998. – 242 с.
6. Прогрессивные технологии моделирования, оптимизации и интеллектуальной автоматизации этапов жизненного цикла авиадвигателей : Монография / А. В. Богуслаев, Ал. А. Олейник, Ан. А. Олейник, Д. В. Павленко, С. А. Субботин ; под ред. Д. В. Павленко, С. А. Субботина. – Запорожье : ОАО «Мотор Сич», 2009. – 468 с.
7. Рассел С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М.: Вильямс, 2006. – 1408 с.
8. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети / А. П. Ротштейн. – Винница: Універсум-Вінниця, 1999. – 320 с.
9. Руденко О. Г. Штучні нейронні мережі / О. Г. Руденко, Є. В. Бодяньський. – Х.: Компанія СМІТ, 2006. – 404 с.
10. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский ; пер. с польск. И. Д. Рудинского. – М.: Горячая линия – Телеком, 2004. – 452 с.
11. Скобцов Ю. А. Основы эволюционных вычислений / Ю. А. Скобцов. – Донецк: ДонНТУ, 2008. – 330 с.
12. Субботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2009. – 375 с.
13. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – СПб : Издательский дом "Вильямс", 2005. – 1104 с.

14. Эволюционные методы компьютерного моделирования: монография / А. Ф. Верлань, В. Д. Дмитриенко, Н. И. Корсунов, В. А. Шорох. – К: Наукова думка, 1992. – 256 с.

### **Додаткова література**

15. Cantu-Paz E. Efficient and accurate parallel genetic algorithms / E. Cantu-Paz. – Massachusetts: Kluwer Academic Publishers, 2001. – 162 p.

16. Hoekstra A. Generalisation in feed forward neural classifiers: Proefschrift ter verkrijging van de graad van doctor. –Delft: Technische Universiteit Delft, 1998. – 136 p.

17. Holland J. H. Adaptation in natural and artificial systems / J. H. Holland. – Ann Arbor: The University of Michigan Press, 1975. – 97 p.

18. Oleynik A. Parametrical synthesis of neural network models based on the evolutionary optimization / A. Oleynik, S. Subbotin // The experience of designing and application of CAD systems in Microelectronics : X International Conference CADSM-2009, 24–28 February 2009 : proceedings of the conference. – Lviv, 2009. – P. 335–338.

19. Subbotin S.A. Evolutionary method with clustering for feature selection / S.A. Subbotin, An.A. Olejnik // Inductive Modelling : 2nd International Conference ICIM'2008, 15–19 September 2008 : proceedings of the conference. – Kyiv, 2008. – P. 213–216.

20. Subbotin S. Entropy based evolutionary search for feature selection / S. Subbotin, A. Oleynik // The experience of designing and application of CAD systems in Microelectronics : IX International Conference CADSM-2007, 20–24 February 2007 : proceedings of the conference. – Lviv, 2007. – P. 442–443.

21. Subbotin S. The feature selection method based on the evolutionary approach with a fixation of a search space / S. Subbotin, A. Oleynik // Modern problems of radio engineering, telecommunications and computer science : VIII International Conference TCSET'2006, 28 February – 4 March 2006 : proceedings of the conference. – Lviv, 2006. – P. 574–575.

22. Subbotin S. The multi objective evolutionary feature selection / S. Subbotin, A. Oleynik // Modern problems of radio engineering, telecommunications and computer science : IX International Conference TCSET'2008, 19–23 February 2008 : proceedings of the conference. – Lviv, 2008. – P. 115–116.

23. Subbotin S. A. Feature selection based on evolutionary approach / S. A. Subbotin, An. A. Oleynik, Al. A. Oleynik // Intelligent Systems Design: Neural Networks, Fuzzy Logic, Evolutionary Computation, Swarm Intelligence, and Complex Systems : 16-th International Conference ANNIE 2006,

5–8 November 2006 : proceedings of the conference. – Missouri-Rolla, 2006. – P. 125–130.

24. The practical handbook of genetic algorithms / ed. L. D. Chambers. – Florida : CRC Press, 2000. – Vol. I: Applications. – 520 p.

25. The practical handbook of genetic algorithms / ed. L. D. Chambers. – Florida : CRC Press, 2000. – Vol. II: New frontiers. – 421 p.

26. The practical handbook of genetic algorithms. / ed. L. D. Chambers. – Florida: CRC Press LLC, 2000. – Vol. III: Complex coding systems. – 659 p.

27. Айвазян С. А. Прикладная статистика: Исследование зависимостей / С. А. Айвазян, И. С. Енюков, Л. Д. Мешалкин. – М.: Финансы и статистика, 1985. – 487 с.

28. Биргер И.А. Техническая диагностика / И.А. Биргер.– М.: Машиностроение, 1978. – 240 с.

29. Борисов В. В. Нечёткие модели и сети / В. В. Борисов, В. В. Круглов, А. С. Федулов. – М.: Горячая линия-Телеком, 2007. – 284 с.

30. Букатова И. Л. Эвоинформатика: Теория и практика эволюционного моделирования / И. Л. Букатова, Ю. И. Михасев, А. М. Шаров. – М.: Наука, 1991. – 206 с.

31. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г. К. Вороновский, К. В. Махотило, С. Н. Петрашев, С. А. Сергеев. – Х.: Основа, 1997. – 111 с.

32. Головкин В. А. Нейронные сети: обучение, организация и применение: Учеб. пособие для вузов / Общ. ред. А. И. Галушкина. – М.: ИПРЖР, 2001. – 256 с.

33. Горбачев В.В. Концепции современного естествознания. В 2-х ч.: Учебное пособие / В.В. Горбачев. – М.: Издательство МГУП, 2000. – 274 с.

34. Горелик А. Л. Методы распознавания / А. Л. Горелик, В. А. Скрипкин. – М.: Высшая школа, 2004. – 261 с.

35. Дисперсионная идентификация / под ред. Н. С. Райбмана. – М.: Наука, 1981. – 336 с. – (Теоретические основы технической кибернетики).

36. Дли М. И. Нечеткая логика и искусственные нейронные сети / М. И. Дли. – М.: Физматлит, 2001. – 225 с.

37. Дубровін В. І. Методи оптимізації та їх застосування в задачах навчання нейронних мереж : навчальний посібник / В. І. Дубровін, С. О. Субботін. – Запоріжжя: ЗНТУ, 2003. – 136 с.

38. Дюк В. Data mining: учебный курс / В. Дюк, А. Самойленко. – СПб.: Питер, 2001. – 368 с.

39. Жамбю М. Иерархический кластер-анализ и соответствия / М. Жамбю. – М.: Финансы и статистика, 1988. – 342 с.
40. Замков О. О. Математические методы в экономике / О. О. Замков, А. В. Толстопятенко, Ю. Н. Черемных. – М.: «Дело и Сервис», 1999. – 366 с.
41. Ивахненко А. Г. Моделирование сложных систем по экспериментальным данным / А. Г. Ивахненко, Ю. П. Юрачковский. – М.: Радио и связь, 1987. – 118 с.
42. 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.
43. Комп'ютерна програма "Автоматизована система еволюційного синтезу та оптимізації діагностичних моделей" / С. О. Субботін, А. О. Олійник ; свідоцтво про реєстрацію авторського права на твір № 26729. – Держ. департамент інтелектуальної власності ; заявл. 09.06.08 ; зареєстр. 01.12.08.
44. Крисилов В. А. Представление исходных данных в задачах нейросетевого прогнозирования / В. А. Крисилов, К. В. Чумичкин, А. В. Кондратюк. – Нейроинформатика–2003. – М.: МИФИ, 2003. – Ч. 1. – С. 184–191.
45. Круглов В. В. Искусственные нейронные сети: Теория и практика / В. В. Круглов, В. В. Борисов. – М.: Горячая линия-Телеком, 2001. – 382 с.
46. Минаев Ю. Н. Методы и алгоритмы идентификации и прогнозирования в условиях неопределенности в нейросетевом логическом базисе / Ю. Н. Минаев, О. Ю. Филимонова, Б. Лиес. – М.: Горячая линия-Телеком, 2003. – 205 с.
47. Миркес Е. М. Нейроинформатика: Учеб. пособие для студентов / Е. М. Миркес. – Красноярск: ИПЦ КГТУ, 2002. – 347 с.
48. Нейрокомпьютер как основа мыслящих ЭВМ. – М.: Наука, 1993. – 239 с.
49. Нейросетевые системы управления / В. А. Терехов, Д. В. Ефимов, И. Ю. Тюкин, В. Н. Антонов. – СПб.: Изд-во С.-Петербургского ун-та, 1999. – 265 с.
50. Олейник А. А. Метод эволюционного поиска с сокращением размера популяции / А. А. Олейник, С. А. Субботин // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій : міжнародна наукова-практична конференція, 13–15 квітня 2006 р. : матеріали конференції. – Запоріжжя, 2006. – С. 179–181.

51. Олейник А. А. Определение влияния режимов высокоскоростного фрезерования на параметры нежестких деталей на основе эволюционного подхода / А. А. Олейник, Д. В. Павленко, С. А. Субботин // Вісник двигунобудування. – 2008. – № 1. – С. 84–90.

52. Олейник А. А. Параметрический синтез нейросетевых диагностических моделей на основе эволюционной оптимизации / А. А. Олейник, С. А. Субботин // Автоматизированные системы управления и приборы автоматики. – 2007. – № 141. – С. 73–81.

53. Олейник А. А. Упрощение структуры нейросетей на основе островной модели эволюционного поиска / А. А. Олейник, С. А. Субботин // Бионика интеллекта. – 2009. – № 1. – С. 107–112.

54. Олейник А. А. Эволюционный метод обучения нейромоделей / А. А. Олейник // Интеллектуальный анализ информации : восьмая международная конференция ИАИ-2008, 14–17 мая 2008 г. : сборник трудов. – К, 2008. – С. 351–361.

55. Олейник А. А. Эволюционный метод структурно-параметрического синтеза нейронных сетей / А. А. Олейник, С. А. Субботин // Адаптивні системи автоматичного управління. – 2008. – № 12 (32). – С. 96–103.

56. Олейник Ан .А. Эволюционный отбор заданного количества признаков / Ан .А. Олейник // Радіоелектроніка і молодь в ХХІ ст. : 11-ий міжнародний молодіжний форум, 10–12 квітня 2007 р. : матеріали форуму. – Х., 2007. – Ч. 2. – С. 142.

57. Олейник Ан. А. Автоматизированная система синтеза нейросетевых моделей на основе эволюционного подхода / Ан. А. Олейник, С. А. Субботин // Нейрокибернетика : XV международная конференция ICNC-09, 23–25 сентября 2009 г. : материалы конференции. – Ростов-на-Дону, 2009. – Т. 2. – С. 130–133.

58. Олейник Ан. А. Выбор системы информативных признаков для классификации транспортных средств на основе эволюционного поиска / Олейник Ан. А. // Комп'ютерне моделювання та інтелектуальні системи : збірник наукових праць / за ред. Д. М. Пізи, С. О. Субботіна. – Запоріжжя: ЗНТУ, 2007. – С. 134–146.

59. Олейник Ан. А. Метод полимодального эволюционного поиска для структурного синтеза нейромоделей / Ан. А. Олейник, С. А. Субботин // Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту : міжнародна наукова конференція ISDMCI'2008, 19–23 травня 2008 р. : матеріали конференції. – Херсон: ХНТУ, 2008. – Т. 3 (Ч. 2). – С. 37–40.

60. Олейник Ан. А. Многокритериальный эволюционный метод упрощения нейронных сетей / Ан. А. Олейник, С. А. Субботин // Авто-

матика-2008 : п'ятнадцята міжнародна науково-технічна конференція, 23–26 вересня 2008 р. : тези доповідей. – Одеса, 2008. – С. 392–395.

61. Олейник Ан. А. Построение нейросетевой модели коэффициента упрочнения деталей энергетических установок / Ан. А. Олейник, С. А. Субботин // Физика высоких энергий, ядерная физика и ускорители : 7-ая научно-техническая конференция, 23–27 февраля 2009 г. : материалы конференции. – Х., 2009. – С. 53–54.

62. Олейник Ан. А. Эволюционные методы синтеза нейронных сетей / Ан. А. Олейник // Радіоелектроніка і молодь в ХХІ сторіччі : 13-ий міжнародний молодіжний форум, 30 березня – 1 квітня 2009 р. : матеріали форуму. – Х.: ХНУРЕ, 2009. – Ч. 2. – С. 92.

63. Олейник Ан. А. Эволюционный синтез моделей сложных объектов и процессов / Ан. А. Олейник, Ал. А. Олейник // Радіоелектроніка і молодь в ХХІ ст. : 10-ий міжнародний молодіжний форум, 10–12 квітня 2006 р. : матеріали форуму. – Х., 2006. – С. 417.

64. Олейник Ан.А. Синтез нейросетевых моделей на основе методов эволюционной оптимизации / Ан.А. Олейник // Радіоелектроніка і молодь в ХХІ ст. : 12-ий міжнародний молодіжний форум, 1–3 квітня 2008 р. : матеріали форуму. – Х.: ХНУРЕ, 2008. – Ч. 2. – С. 316.

65. Олійник А. О. Виділення системи заданої кількості інформативних ознак на основі еволюційного пошуку / А. О. Олійник // Тиждень науки: збірник наукових праць. – Запоріжжя: ЗНТУ, 2007. – С. 59–61.

66. Олійник А. О. Еволюційні методи структурно-параметричного синтезу нейромоделей / А. О. Олійник // Тиждень науки: збірник наукових праць. – Запоріжжя: ЗНТУ, 2008. – Т. 2. – С. 61–63

67. Олійник А. О. Критерії та методика порівняння еволюційних методів побудови нейромережних моделей / А. О. Олійник, С. О. Субботін // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій : 4-а міжнародна науково-практична конференція, 24–26 вересня 2008 р. : матеріали конференції. – Запоріжжя, 2008. – С. 120–121.

68. Отбор геометрических параметров и синтез модели частотной характеристики лопаток компрессора на основе эволюционного поиска / А. В. Богуслаев, А. А. Олейник, Г. В. Пухальская, С. А. Субботин // Вісник двигунобудування. – 2006. – № 1. – С. 14–17.

69. Пат. 18294 Україна, МПК G06F 19/00. Спосіб відбору інформативних ознак для діагностики виробів / С. О. Субботін, А. О. Олійник (Україна); заявник Запорізький національний технічний університет. – № u200603087 ; заявл. 22.03.06 ; опубл. 15.11.06, Бюл. № 11.

70. Прикладная статистика: Классификация и снижение размерности / С. А. Айвазян, В. М. Бухштабер, И. С. Енюков, Л. Д. Мешалкин. – М.: Финансы и статистика, 1989. – 607 с.

71. Прогнозирование частот собственных колебаний лопаток компрессора высоких форма при регулировании частоты основного тона / В.А. Богуслаев, О.Н. Бабенко, А.А. Олейник, Д.В. Павленко, Е.Я. Корневский // Вісник двигунобудування. – 2009. – № 1. – С. 14–17.

72. Системный анализ и принятие решений: Словарь-справочник: Учеб. пособие для вузов / Под ред. В.Н. Волковой, В.Н. Козлова. – М.: Высшая школа, 2004. – 616 с.

73. Субботин С. А. Анализ свойств и критерии сравнения нейросетевых моделей для решения задач диагностики и распознавания образов / С. А. Субботин // Реєстрація, зберігання і обробка даних. – 2009. – Т. 11. – № 3. – С. 42–52.

74. Субботин С. А. О сравнении нейросетевых моделей / С. А. Субботин // Нейроинформатика и ее приложения: Материалы XI Всероссийского семинара, 3-5 октября 2003 г. / Под ред. А. Н. Горбаня, Е. М. Миркеса. – Красноярск: ИВМ СО РАН, 2003. – С. 152-153.

75. Субботин С. А. Методика и критерии сравнения моделей и алгоритмов синтеза искусственных нейронных сетей / С. А. Субботин // Радіоелектроніка. Інформатика. Управління. – 2003. – № 2. – С. 109-114.

76. Субботин С. А. Выбор набора информативных признаков для синтеза моделей объектов управления на основе эволюционного поиска с группировкой признаков / С. А. Субботин, А. А. Олейник // Штучний інтелект. – 2006. – № 4. – С. 488–494.

77. Субботин С. А. Выделение набора информативных признаков на основе эволюционного поиска с кластеризацией / С. А. Субботин, А. А. Олейник // Штучний інтелект. – 2008. – № 4. – С. 704–711.

78. Субботин С. А. Критерии сравнения эволюционных методов синтеза нейромоделей / С. А. Субботин, А. А. Олейник // Научная сессия МИФИ–2007. IX Всероссийская научно-техническая конференция “Нейроинформатика-2007”, 23–26 января 2007 г. : сборник научных трудов. – М., 2007. – Ч. 2. – С. 177–184.

79. Субботин С. А. Методы эволюционного отбора комбинаций признаков с использованием априорной информации об их индивидуальной значимости / С. А. Субботин, А. А. Олейник // Нейрокомпьютеры: разработка, применение. – 2007. – № 7. – С. 8–13.

80. Субботин С. А. Оптимизация нейросетевых моделей на основе многокритериального метода эволюционного поиска / С. А. Субботин, А. А. Олейник // Вестник НТУ “ХПИ”. – Харьков: НТУ “ХПИ”. – 2008.

– № 24. – С. 165–173. – (Тематический выпуск “Информатика и моделирование”).

81. Субботин С. А. Сравнительный анализ методов эволюционного поиска / С. А. Субботин, А. А. Олейник // Штучний інтелект. – 2008. – № 2. – С. 44–49.

82. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник / С. О. Субботін. – Запоріжжя: ЗНТУ, 2008. – 341 с.

83. Субботін С. О. Структурний синтез нейромоделей на основі полімодального еволюційного пошуку / С. О. Субботін, А. О. Олійник // Радіоелектроніка. Інформатика. Управління. – 2008. – № 1. – С. 111–117.

84. Холланд Х. Д. Генетические алгоритмы / Х. Д. Холланд // В мире науки. – 1992. – № 9. – С. 32–40.

85. Эволюционный синтез нейромоделей коэффициента упрочнения лопаток авиадвигателей / С. А. Субботин, В. К. Яценко, Ан. А. Олейник, Ал. А. Олейник // Научная сессия МИФИ–2006. VIII Всероссийская научно-техническая конференция “Нейроинформатика-2006”, 26–28 января 2006 г. : сборник научных трудов. – М., 2006. – Ч. 3. – С. 141–148.

86. Эшби У. Р. Введение в кибернетику / У. Р. Эшби. – М: Иностранная литература, 1959. – 432 с.

## АЛФАВІТНО-ПРЕДМЕТНИЙ ПОКАЖЧИК

### А

Алель, 14  
Аутбридинг, 46

### Б

Багатокритеріальна оптимізація, 117  
– з обмеженнями, 117, 135  
Багатокритеріальний еволюційний метод, 117, 121  
Будівельний блок, 77

### В

Вибір батьківської пари, 45  
– випадковий, 45  
– селективний, 46  
Визначена довжина схеми, 73  
Виродження популяції, 72  
Відбір, 42  
– з використанням порогу, 44  
– пропорційний, 42  
– ранжируванням, 44  
– рулеткою, 43  
– турнірний, 44  
Відстань, 47  
– Евклідова, 47  
– узагальнена, 133  
– Хеммінга, 47  
Вставка, 69

### Г

Ген, 14  
– активний, 26  
Генетичний метод, 11  
– гібридний, 23  
– з перевіркою на недомінантність в групі хромосом, 127  
– з сортуванням за принципом недомінантності, 126  
Генетичний оператор, 40  
– відбору, 42

– мутації, 62  
– схрещування, 45  
Генетичне програмування, 96  
Генотип, 14  
Генофонд популяції, 14

### Д

Делеція, 70  
Диверсифікація, 62  
Довжина хромосоми, 11

### Е

Еволюційна стратегія, 90  
–  $(\mu + \lambda)$ , 90  
–  $(\mu, \lambda)$ , 90  
– з архівуванням недомінантних хромосом, 135  
Еволюційне програмування, 98  
– із заміщенням особин, 112  
Еволюційний метод, 11  
– з архівуванням недомінантних хромосом, 128  
– з керуванням різноманітністю в популяції, 114  
Еволюційні методи, що розгалужуються, 109  
Елітизм, 70  
Епістазис, 13

### І

Інбридинг, 46  
Інвертування, 68  
– з використанням методу Фібоначчі, 69  
– із зсувом, 68  
– класичне, 68  
– на основі методу золотого перерізу, 69  
Ініціалізація, 39

**К**

Керування параметрами еволюційного пошуку, 78  
 – адаптивне, 80  
 – неадаптивне, 78  
 Код Грея, 36  
 Кодування параметрів, що оптимізуються, 35  
 Коефіцієнт помилковості одержаного фронту Парето, 133  
 Коефіцієнт розрядженості, 135  
 Критерії зупинення, 71

**Л**

Лексикографічне впорядкування, 119  
 Локус, 14

**М**

Маска схрещування, 49  
 Метод агрегувальних функцій, 118  
 Метод архівування недомінантних хромосом, 126  
 Метод групового врахування аргументів, 100  
 Метод досягнення заданого значення, 120  
 Метод імітації відпалу, 93  
 Метод просторової популяційної структури, 82  
 Метод розподілення фітнес-функції, 125  
 Метод сіткового розбиття простору цільових функцій, 126  
 Методи багатокритеріальної еволюційної оптимізації, 117  
 – апостеріорні, 118  
 – апріорні, 118  
 – прогресивні, 118  
 Методи випадкової міграції, 112

Методи відновлення, 111  
 Методи заміщення, 105  
 Методи множинного заміщення, 112  
 Методи рівноваги, що зміщується, 110  
 Методи розбиття простору пошуку, 109  
 Методи ухилення від передчасної збіжності, 103  
 Мікрогенетичний метод, 132  
 Мінімаксний метод, 119  
 Мобільний генетичний метод, 25  
 Модель генетичного пошуку, 21  
 – “ведучий-ведений”, 29  
 – дифузійна, 31  
 – ієрархічна, 31  
 – канонічна, 21  
 – мозаїчна, 83  
 – острівна, 29  
 Мутація, 62  
 – гомологічних числових хромосом, 63  
 – – випадкова, 63  
 – – Гауссова, 63  
 – – на основі квадратичної апроксимації, 64  
 – – нерівномірна, 63  
 – обміну, 66  
 – – золотого перерізу, 66  
 – – класична, 66  
 – – на основі чисел Фібоначчі, 67  
 – – нечітка, 67  
 – – одноточкова, 66  
 – проста, 62  
 – – бінарних і гомологічних числових хромосом, 63  
 – – векторних хромосом, 63  
 – – точкова, 63

**Н**

Настроювання параметрів, 79  
 Недомінантна хромосома, 121  
 Неявний паралелізм, 77

**О**

Оператор CUT, 26  
 Оператор SPLICE, 26  
 Оператор генетичного програмування, 97  
 – мутація, 98  
 – – із заміною, 98  
 – – із збільшенням, 98  
 – – із стисненням, 98  
 – – циклічна, 98  
 – схрещування, 97  
 Оператор еволюційного програмування, 100  
 – мутація, 100  
 Оптимізаційні методи, 10

**П**

Парето оптимальна хромосома, 121  
 Парето оптимальна множина, 121  
 Покоління, 15  
 Полімодальна еволюційна оптимізація, 103  
 Популяційний спосіб відтворення, 18  
 Популяція, 14  
 – початкова, 11  
 Порядок схеми, 73  
 Правило “20% успіху”, 91  
 Принципу витіснення, 71  
 Пристосованість, 37  
 – схеми, 73  
 Простий генетичний метод, 22  
 Процедура ущільнення, 124  
 Процедура експресії генів, 26

**Р**

Рекомбінація, 45  
 Релігійні еволюційні методи, 106  
 Репродуктивний план  
 Холланда, 21  
 Різноманітність популяції, 80  
 Розмір популяції, 11  
 Розрядженість, 135

**С**

СНС-метод, 113  
 Стационарний спосіб відтворення, 18  
 Схема, 73  
 – зруйнована, 75  
 Схрещування, 45  
 – бінарних хромосом, 48  
 – – *n*-точкове, 48  
 – – діагональне, 50  
 – – однорідне, 49  
 – – порівняльне, 49  
 – – рівномірне, 49  
 – числових гомологічних хромосом, 51  
 – – SBX, 55  
 – – арифметичне, 51  
 – – – лінійне, 53  
 – – – нерівномірне, 52  
 – – – розширене лінійчате, 53  
 – – – рівномірне, 52  
 – – геометричне, 53  
 – – дискретне, 51  
 – – евристичне, 54  
 – – змішане, 53  
 – – нечітке, 54  
 – – плоске, 51  
 – – просте, 51  
 – числових негомологічних хромосом, 56  
 – – впорядковуюче, 57  
 – – – двоточкове, 57

- одноточкове, 57
- позиційне, 57
- жадібне, 59
- з частковим відображенням, 58
- засноване на принципі золотого перерізу, 61
- методом дихотомії, 60
- побудоване на основі чисел Фібоначчі, 62
- сегрегації, 60
- циклічне, 59

## **Т**

- Теорема схем, 76
- Точка схрещування, 48
- Транслокація, 69
- Транспозиція, 58

## **Ф**

- Фаза попередня, 28
- Фаза процесингу, 28
- Фенотип, 34
- Фітнес-функція, 11, 37

- Формування початкової популяції, 39
- випадкове, 39
- засноване на знаннях, 39
- рівномірне, 39
- сіткове, засноване на знаннях, 39
- Фронт Парето, 121

## **Х**

- Хромосома, 14
- бінарна, 34
- векторна, 35
- числова, 34
- – гомологічна, 34
- – негомологічна, 34

## **Ц**

- Цільова функція, 11

## **Ш**

- Штрафна функція, 136
- адаптивна, 136
- динамічна, 136
- статична, 136

*Навчальне видання*

**ОЛІЙНИК Андрій Олександрович  
СУББОТІН Сергій Олександрович  
ОЛІЙНИК Олексій Олександрович**

**ЕВОЛЮЦІЙНІ ОБЧИСЛЕННЯ  
ТА ПРОГРАМУВАННЯ  
Навчальний посібник**

Надруковано у авторській редакції

Верстання

Олійник А. О.  
Новікова К. О.

Підп. до друку 07.09.2010. Формат 60×84/16.  
Бум. офс. Різогр. друк. Ум. друк. арк. 18,8.  
Тираж 300 прим. Зам. № 1476.

Запорізький національний технічний університет  
Україна, 69063 Запоріжжя, вул. Жуковського, 64  
Тел.: (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