

Міністерство освіти і науки України  
Запорізький національний технічний університет

## **МЕТОДИЧНІ ВКАЗІВКИ**

до виконання лабораторних та практичних робіт  
з дисципліни

**“Еволюційні методи інженерії інтелектуальних систем”**

для студентів спеціальностей

121 “Інженерія програмного забезпечення” та

122 “Комп’ютерні науки та інформаційні технології”

(всіх форм навчання)

Методичні вказівки до виконання лабораторних та практичних робіт з дисципліни “Еволюційні методи інженерії інтелектуальних систем” для студентів спеціальностей 121 “Інженерія програмного забезпечення” та 122 “Комп’ютерні науки та інформаційні технології” (всіх форм навчання) / А. О. Олійник, С. О. Субботін, Є.М. Федорченко. – Запоріжжя: ЗНТУ, 2016. – 100 с.

Автори: А. О. Олійник, к.т.н., доцент  
С. О. Субботін, д.т.н., професор  
Є. М. Федорченко, ст. викладач

Рецензент: В. П. Рисіков, к.т.н., доцент

Відповідальний  
за випуск: С. О. Субботін, д.т.н., професор

Затверджено  
на засіданні кафедри  
програмних засобів

Протокол № 1  
від “16” серпня 2016 р.

## ЗМІСТ

<b>Вступ .....</b>	<b>6</b>
<b>1 Лабораторна робота № 1 Методи еволюційного пошуку.....</b>	<b>7</b>
1.1 Мета роботи .....	7
1.2 Основні теоретичні відомості.....	7
1.2.1 Генетичний пошук як метод оптимізації.....	7
1.2.2 Аналогія генетичних методів з поняттями генетики .....	8
1.2.3 Узагальнена схема роботи генетичних методів.....	9
1.2.4 Моделі генетичного пошуку.....	11
1.2.5 Ініціалізація та запуск генетичного пошуку .....	11
1.2.5.1 Кодування параметрів, що оптимізуються.....	11
1.2.5.2 Завдання цільової функції.....	15
1.2.5.3 Ініціалізація .....	16
1.2.6 Відбір .....	17
1.2.6.1 Пропорційний відбір .....	18
1.2.6.2 Відбір ранжируванням .....	19
1.2.6.3 Турнірний відбір.....	19
1.2.6.4 Відбір з використанням порогу .....	20
1.2.7 Схрещування.....	20
1.2.7.1 Вибір батьківської пари .....	21
1.2.7.2 Оператори схрещування .....	23
1.2.8 Мутація .....	27
1.2.8.1 Проста мутація.....	28
1.2.8.2 Мутація гомологічних числових хромосом .....	29
1.2.9 Формування нового покоління.....	31
1.2.10 Критерії зупину.....	32
1.2.11 Генетичний пошук в пакеті Matlab .....	33
1.3 Порядок виконання роботи.....	35
1.4 Індивідуальні завдання.....	36
1.5 Контрольні питання.....	55
<b>2 Лабораторна робота № 2 Дослідження еволюційних операторів .....</b>	<b>57</b>
2.1 Мета роботи .....	57
2.2 Завдання до роботи.....	57

2.3 Зміст звіту.....	58
2.4 Контрольні питання.....	59
<b>3 Лабораторна робота №3 Статистичний аналіз результатів еволюційної оптимізації .....</b>	<b>60</b>
3.1 Мета роботи .....	60
3.2 Основні теоретичні відомості.....	60
3.3 Завдання до роботи.....	62
3.4 Зміст звіту.....	64
3.5 Контрольні питання.....	65
<b>4 Лабораторна робота №4 Комбінаторна оптимізація за допомогою еволюційних методів .....</b>	<b>68</b>
4.1 Мета роботи .....	68
4.2 Основні теоретичні відомості.....	68
4.2.1 Оператори схрещування для негомологічних числових хромосом .....	69
4.2.1.1 Впорядковуючий оператор схрещування.....	69
4.2.1.2 Схрещування із частковим відображенням.....	70
4.2.1.3 Циклове схрещування .....	71
4.2.1.4 Жадібний оператор схрещування.....	71
4.2.1.5 Схрещування методом дихотомії.....	72
4.2.1.6 Оператор сегрегації .....	73
4.2.2 Оператори мутації .....	73
4.2.2.1 Мутація обміну .....	73
4.2.2.2 Інвертування.....	76
4.2.2.3 Транслокація, вставка та делеція .....	77
4.3 Завдання до роботи.....	78
4.4 Зміст звіту.....	79
4.5 Контрольні питання.....	79
<b>5 Лабораторна робота № 5 Бінарні асоціативні правила.....</b>	<b>82</b>
5.1 Мета роботи .....	82
5.2 Основні теоретичні відомості.....	82
5.3 Порядок виконання роботи.....	87
5.4 Зміст звіту.....	87
5.5 Контрольні питання.....	88
<b>6 Лабораторна робота № 6 Числові асоціативні правила.....</b>	<b>89</b>
6.1 Мета роботи .....	89

6.2 Основні теоретичні відомості.....	89
6.3 Порядок виконання роботи.....	91
6.4 Зміст звіту.....	91
6.5 Контрольні питання.....	92
<b>7 Лабораторна робота № 7 Дерева розв'язків .....</b>	<b>93</b>
7.1 Мета роботи .....	93
7.2 Основні теоретичні відомості.....	93
7.3 Порядок виконання роботи.....	93
7.4 Зміст звіту.....	94
7.5 Контрольні питання.....	94
<b>8 Лабораторна робота №8 Еволюційний синтез нейромережевих моделей .....</b>	<b>97</b>
8.1 Мета роботи .....	97
8.2 Порядок виконання роботи.....	97
8.3 Зміст звіту.....	97
8.4 Контрольні питання.....	98
<b>Література.....</b>	<b>100</b>

## ВСТУП

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання основ еволюційного програмування.

Відповідно до графіка студенти перед виконанням лабораторної або самостійної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою.

Для одержання заліку по кожній роботі студент здає викладачу цілком оформлений звіт, а також тексти розроблених програм, файлів програм, що виконуються, файли даних і текст звіту.

Звіт має містити:

- титульний аркуш;
- мету, варіант і завдання роботи;
- лаконічний опис теоретичних відомостей;
- текст програми, що обов'язково містить коментарі;
- вхідні та вихідні дані програми;
- змістовний аналіз отриманих результатів та висновки.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок. Для набору тексту звіту використовують редактор MS Word 97: шрифт Times New Roman, 12 пунктів. Міжрядковий інтервал: полуторний – для тексту звіту, одинарний – для листингів програм, таблиць і роздруківок даних.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, про методи виконання кожного етапу роботи, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах. Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і здачі роботи студент повинний відповісти на контрольні питання, приведені наприкінці опису відповідної роботи. Загальний залік студент одержує після виконання і здачі останньої роботи.

# 1 ЛАБОРАТОРНА РОБОТА № 1

## МЕТОДИ ЕВОЛЮЦІЙНОГО ПОШУКУ

### 1.1 Мета роботи

1.1.1 Вивчити основні методи еволюційного пошуку.

1.1.2 Навчитися використовувати еволюційні методи для розв'язку оптимізаційних задач.

### 1.2 Основні теоретичні відомості

У багатьох технічних задачах актуальною є проблема знаходження глобального оптимуму цільової функції в багатомірному просторі керованих змінних. Традиційні методи багатомірної оптимізації є методами локального пошуку та сильно залежать від вибору початкової точки пошуку. Для знаходження глобального оптимуму доцільно використовувати методи еволюційного пошуку.

Еволюційні методи засновані на аналогії з природними процесами селекції та генетичними перетвореннями, і поєднують комп'ютерні методи моделювання еволюційних процесів у природних і штучних системах.

Традиційно до еволюційних методів відносять генетичні алгоритми, еволюційні стратегії, генетичне програмування та еволюційне програмування.

#### 1.2.1 Генетичний пошук як метод оптимізації

Генетичний пошук містить у собі групу багатомірних, стохастичних, евристичних оптимізаційних методів, вперше запропонованих Д. Холландом у 1975 р. і заснованих на ідеї еволюції за допомогою природного відбору. Генетичні методи були отримані в процесі узагальнення й імітації в штучних системах таких властивостей живої природи, як природний відбір, пристосованість до змінюваних умов середовища, спадкування нащадками життєво важливих властивостей від батьків і т.ін.

Під стандартним генетичним методом розуміють метод для вирішення оптимізаційних задач вигляду:

$$f(H) \rightarrow \min,$$

де  $f$  – функція пристосованості (функція придатності, цільова функція, фітнесс-функція);

$H = \{0; 1\}^L$  – хромосома, що містить в закодованому вигляді параметри цільової функції;

$L$  – кількість розрядів у хромосомі.

Генетичні методи в процесі пошуку використовують деяке кодування множини параметрів замість самих параметрів, тому вони можуть ефективно застосовуватися для рішення задач оптимізації, визначених як на числових множинах, так і на кінцевих множинах довільної природи.

### 1.2.2 Аналогія генетичних методів з поняттями генетики

Основна концепція класичної генетики – *ген* (реально існуюча, незалежна, комбінуюча та розщеплюєча при схрещуваннях одиниця спадковості) була введена І.Г. Менделем з метою пояснення спостережуваної статистики спадкування. Носіями генів у клітинному ядрі особини є нитковидні тіла – *хромосоми* (структурні елементи клітинного ядра біологічних організмів, що є носіями генів). У генетичних методах терміни “хромосома” і “особина” використовуються як синоніми. Місце, що займає ген у хромосомі, називається *локусом*. Схематично можна уявити собі хромосому як прямолінійний відрізок, а локуси – як послідовні ділянки, на які цей відрізок розбитий. Гени приймають значення, які називаються *алелями*.

Дії генів проявляються в досить великих співтовариствах організмів, що схрещуються між собою. Такі співтовариства називають *популяціями*. Популяції характеризуються набором хромосом кожного з об'єктів, сукупність яких визначає *генофонд популяції*.

Таким чином, генетичні методи запозичили з біології понятійний апарат, ідею колективного пошуку екстремуму, способи представлення генетичної інформації, способи передачі генетичної інформації в послідовності поколінь (генетичні оператори), ідею про переважне розмноження найбільш пристосованих особин.



### 1.2.3 Узагальнена схема роботи генетичних методів

Суть генетичного пошуку полягає в циклічній заміні однієї популяції наступною, більш пристосованою. Таким чином, популяція існує не тільки в просторі, але й у часі. Початкова популяція  $P_0$  створюється на етапі ініціалізації генетичного пошуку.

Подальша робота генетичного методу представляє собою ітераційний процес виконання генетичних операторів відбору, схрещування й мутації. Генетичні оператори необхідні для того, щоб застосувати принципи спадковості й мінливості до популяції. Генетичні оператори мають властивість імовірності, тобто вони не обов'язково застосовуються до всіх рішень, що вносить додатковий елемент невизначеності в процес пошуку рішення.

Кожне рішення (хромосома) оцінюється мірою пристосованості. Пристосованість хромосоми визначається як обчислена цільова функція. Правила відбору прагнуть залишити тільки ті рішення, де досягається оптимум цільової функції. Найбільш пристосовані хромосоми одержують можливість відтворювати нащадків за допомогою схрещування з іншими хромосомами популяції. Це призводить до появи нових хромосом, які сполучають у собі деякі характеристики, наслідувані ними від батьків. Найменш пристосовані рішення з меншою ймовірністю зможуть відтворити нащадків, у результаті чого властивості, якими вони володіли, будуть поступово зникати з популяції в процесі еволюції.

Схрещування найбільш пристосованих хромосом приводить до того, що досліджуються найбільш перспективні ділянки простору пошуку. В остаточному підсумку популяція буде сходитися до оптимального рішення задачі.

Після схрещування іноді відбуваються мутації – спонтанні зміни в генах, які випадковим чином розкидають рішення по всьому простору пошуку.

У результаті схрещування й мутації розмір популяції збільшується. Однак для наступних перетворень необхідно скоротити число хромосом поточної популяції. Як правило, наступна популяція формується з нащадків, отриманих у поточній популяції в результаті схрещування й мутації, а також елітних хромосом, що володіють найкращою пристосованістю.

Узагальнений метод генетичного пошуку можна записати в такий спосіб.

Крок 1. Встановити лічильник ітерацій (часу):  $t = 0$ .

Крок 2. Згенерувати початкову популяцію хромосом  $P(t)$ .

Крок 3. Обчислити функцію пристосованості для всіх хромосом у популяції  $f(P(t))$ .

Крок 4. Перевірити умови закінчення пошуку (час, число ітерацій, значення функції пристосованості і т.ін.). Якщо критерії зупину задоволені, перейти до кроку 12.

Крок 5. Збільшити лічильник ітерацій (часу):  $t = t + 1$ .

Крок 6. Вибрати частину популяції (батьківські хромосоми) для схрещування  $P'$ .

Крок 7. Схрестити обрані батьківські хромосоми  $P'(t)$ .

Крок 8. Застосувати оператор мутації до хромосом  $P'(t)$ .

Крок 9. Обчислити нову функцію пристосованості популяції  $f(P'(t))$ .

Крок 10. Вибрати хромосоми, що вижили, виходячи з рівня пристосованості.

Крок 11. Перейти на крок 4.

Крок 12. Кінець.

У наш час запропоновано багато різних генетичних методів, і в більшості випадків вони мало схожі на наведений генетичний метод. Із цієї причини під терміном “генетичні методи” мається на увазі досить широкий клас методів, часом мало схожих один на одного.

Використання генетичного пошуку для рішення практичних задач передбачає:

– вибір методу представлення вхідних даних для генетичного пошуку (кодування параметрів, що оптимізуються);

– визначення цільової функції, що використовується для оцінки хромосом;

– вибір оператора відбору хромосом, що будуть використані для генерації нових рішень за допомогою схрещування й мутації;

– вибір методів одержання нових рішень (операторів схрещування й мутації);

– завдання параметрів пошуку, таких як кількість особин у популяції, імовірнісні характеристики генетичних операторів, максимально припустима кількість ітерацій генетичного пошуку, кількість елітних особин при використанні стратегії елітизму.

### 1.2.4 Моделі генетичного пошуку

Крім узагальненої схеми функціонування генетичного методу, розглянутої вище, використовують також інші технології генетичного пошуку. Вибір моделі генетичного методу залежить від типу розв'язуваної задачі.

Виділяють наступні методи й моделі генетичного пошуку:

- канонічні моделі (репродуктивний план Холланда, генетичний метод Девиса, генетичний метод Гольдберга);
- модель Genitor (Д. Уїтлі);
- гібридні генетичні методи;
- модель СНС;
- генетичний метод зі змінним часом життя особин;
- мобільний генетичний метод;
- паралельні й багаторівневі генетичні методи (однопопуляційні генетичні методи, острівна модель, дрібноструктурні генетичні методи, ієрархічні гібриди);
- генетичний пошук зі зменшенням розміру популяції.

### 1.2.5 Ініціалізація та запуск генетичного пошуку

#### 1.2.5.1 Кодування параметрів, що оптимізуються

Будь-який організм може бути представлений своїм *фенотипом*, що фактично визначає, чим є об'єкт у реальному світі, і *генотипом*, що містить всю інформацію про об'єкт на рівні хромосомного набору. При цьому кожний ген, тобто елемент інформації генотипу, має своє відображення у фенотипі. Таким чином, для розв'язку задач необхідно представити кожен ознаку об'єкта у формі, що підходить для використання в генетичному методі. Все подальше функціонування механізмів генетичного методу відбувається на рівні генотипу, що дозволяє обходитися без інформації про внутрішню структуру об'єкта, що й обумовлює широке застосування генетичного пошуку в самих різних задачах.

За методами представлення генів хромосоми можна умовно розділити 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 розділити на гомологічні та негомологічні.

*Гомологічними* називають хромосоми, що мають загальне походження, морфологічно та генетично подібні, і тому не утворюють неприпустимих рішень при застосуванні стандартних генетичних операторів. У гомологічних числових хромосомах кожний ген може приймати цілі значення в заданому інтервалі. Для різних генів можуть бути задані різні інтервали. Бінарна хромосома є гомологічною числовою хромосоною, кожний 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. Негомологічні хромосоми, як правило, застосовуються при розв'язку задач комбінаторної оптимізації.

3. *Векторні хромосоми* – хромосоми, гени яких представляють собою вектор 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 негомологічні, самі векторні хромосоми є гомологічними.

*Процес кодування параметрів*, що оптимізуються, можна виконати в наступній послідовності кроків.

Крок 1. Визначення параметрів, що оптимізуються, – генів.

Крок 2. Вибір числа розрядів у кожному гені.

Крок 3. Вибір методу кодування.

Варто врахувати, що занадто велика довжина кодування прискорює процес збіжності всіх членів популяції до кращого знайденого рішення. Часто такий ефект є небажаним, оскільки при цьому більша 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 буде мати 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 досить великою, щоб все-таки забезпечити швидкий пошук, з іншого боку – по можливості малою, щоб не допускати передчасної збіжності й залишити методу шанс відшукати кілька оптимальних значень.

Наведемо варіанти кодування генів у деяких задачах, розв'язуваних за допомогою генетичних методів:

– оптимізація функцій: гени – незалежні змінні;

– апроксимація: гени – параметри-константи апроксимуючих функцій;

– задача відбору інформативних ознак: гени ідентифікують значимість відповідних ним ознак (наприклад, якщо значення гену дорівнює одиниці, то відповідна йому ознака вважається інформативною);

– настроювання 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 (рослина або тварина);

– задача про найкоротший шлях: гени – пункти пересування. Вся хромосома представляє собою маршрут з початкової точки в кінцеву, причому не завжди існуючий.

Невдалий вибір упорядкування та кодування бітів у хромосомі може викликати передчасну збіжність до локального оптимуму. Для подолання цього недоліку можна вибрати спосіб кодування, ґрунтуючись на додатковій інформації про задачу.

Варто відзначити, що використання різних варіантів кодування розподіляє точки в просторі пошуку по-різному. У найбільш розповсюдженій різновиді генетичного пошуку для представлення генотипу об'єкту використовуються *бітові рядки*. При цьому кожному атрибуту об'єкту у фенотипі відповідає один ген у генотипі об'єкта.

Ген є бітовим рядком, найчастіше фіксованої довжини, що являє собою значення цієї ознаки.

Кількість розрядів  $r$  у 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;

$W_{\max}$  і  $W_{\min}$  – максимально й мінімально можливі значення ознаки (параметра, незалежної змінної);

$\varepsilon$  – задана похибка визначення оптимального значення ознаки.

Розрядність хромосоми  $L$  визначається як сума розрядностей генів. У випадку, якщо задані однакові значення  $W_{\max}$ ,  $W_{\min}$  і  $\varepsilon$  для всіх  $n$  генів, розрядність хромосоми може бути обчислена за формулою:

$$L = n \cdot r.$$

Після того, як обрані параметри, їх кількість та розрядність, необхідно вирішити, як безпосередньо записувати дані, тобто вибрати метод кодування. Можна використати *звичайне кодування* або *коди Грея*. Незважаючи на те, що використання кодів Грея призводить до кодування/декодування даних, вони дозволяють уникнути деяких проблем, які з'являються в результаті звичайного кодування.

Перевага коду Грея в тому, що якщо два числа відрізняються на 1, то і їхні двійкові коди відрізняються тільки на один розряд, а у двійкових кодах не все так просто. Так, наприклад, числа 7 і 8 у бітовому поданні відрізняються в чотирьох позиціях ( $7_{10}=0111_2$ ,  $8_{10}=1000_2$ ), що затрудняє функціонування генетичного методу й збільшує час, необхідний для його збіжності, а в кодї Грея ці числа відрізняються всього на одну позицію ( $7_{10}=0100_G$ ,  $8_{10}=1100_G$ ).

Варто відзначити, що кодувати й декодувати у кодї Грея досить зручно. *Кодування/декодування з бінарного коду в код Грея* можна виконати в такий спосіб.

Крок 1. Скопіювати старший розряд кодуемого (декодуемого) числа в старший розряд декодуемого (кодуемого) числа.

Крок 2. Виконати перетворення за формулами:

– із двійкового коду в код Грея:  $G[i] = \text{XOR}(B[i+1], B[i]);$

– з коду Грея у двійковий:  $B[i] = \text{XOR}(B[i+1], G[i]),$

де  $G[i]$  –  $i$ -ий розряд коду Грея;

$B[i]$ –  $i$ -ий розряд бінарного коду.

Наприклад, послідовність чисел від 0 до 15 у двійковому кодї: {0000, 0001 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001 1010, 1011, 1100, 1101, 1110, 1111}, а в кодах Грея: {0000, 0001 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101 1111, 1110, 1010, 1011, 1001, 1000}.

*Кодування ознак, яким відповідають числа із плаваючою комою.*

Найпростіший спосіб кодування – використання бітового представлення. Однак такий варіант має ті ж недоліки, що й при кодуванні цілих чисел. Тому на практиці застосовується наступна послідовність дій.

Крок 1. Розбивається весь інтервал допустимих значень ознаки на ділянки з необхідною точністю.

Крок 2. Приймається значення гена як ціле число, що визначає номер інтервалу (використовуючи код Грея).

Крок 3. В якості значення параметра приймається число, що є серединою цього інтервалу.

### 1.2.5.2 Завдання цільової функції

*Цільова функція* – це функція, оптимум якої необхідно знайти. Генетичний метод вимагає, щоб хромосоми оцінювалися за допомогою цільової функції (фітнесс-функцій, функції пристосованості, функції оцінки) задачі.

Наприклад, для задачі апроксимації, цільовою функцією може бути середнє відхилення значень вихідного параметру, розрахованих за утвореною моделі, від його реальних значень.

Можна відзначити, що обчислення фітнесс-функції – один з найбільш важливих етапів генетичного пошуку.

Тому при виборі цільової функції потрібно враховувати наступне.

1. Функція пристосованості повинна бути адекватна задачі. Це означає, що для успішного пошуку необхідно, щоб розподіл значень фітнесс-функцій збігався з розподілом реальної якості рішень (не завжди “якість” рішення еквівалентна його оцінці за фітнесс-функцією).

2. Фітнесс-функція повинна мати рельєф. Крім того, рельєф повинен бути різноманітним. Це означає, що генетичний метод має мало шансів на успіх, якщо на поверхні фітнесс-функції є величезні

“плоскі” ділянки, тому що це приводить до того, що більшість рішень (хромосом) у популяції при різних генотипах не будуть відрізнятися фенотипом. Тобто, незважаючи на те, що рішення розрізняються, вони мають однакову оцінку, а значить метод не має можливості вибрати краще рішення та вибрати напрямок подальшого розвитку.

3. Фітнесс-функція повинна вимагати мінімум ресурсів, тому що її обчислення є найбільш часто виконуваним етапом методу, і тому складність обчислення фітнесс-функції має істотний вплив на швидкість роботи методу.

4. У випадку, якщо цільова функція містить ділянки, що представляють собою так зване “вузьке горло” (різкий стрибок або спад), необхідно врахувати, що генетичний пошук може не знайти глобального екстремуму, що розташований у вузькому горлі. Для підвищення якості генетичного пошуку при такій цільовій функції можна рівномірно формувати початкову популяцію на всьому інтервалі припустимих значень змінних.

### 1.2.5.3 Ініціалізація

Стандартні генетичні методи починають свою роботу з ініціалізації, тобто формування *початкової популяції*  $P_0$  – кінцевого набору допустимих рішень задачі:

$$P_0 = \{H_1, H_2, \dots, H_N\},$$

де  $N$  – розмір популяції;

$H_j = \{h_{1j}, h_{2j}, \dots, h_{Lj}\}$  – хромосома, що складається з  $L$  генів;

$\min_i \leq h_{ij} \leq \max_i$ ,  $\min_i$  і  $\max_i$  – мінімальне й максимальне значення  $i$ -го параметра в розв'язуваній за допомогою генетичного методу задачі.

Ці рішення можуть бути обрані випадковим образом або введені користувачем. Вибір початкової популяції не має значення для збіжності процесу в асимптотиці, однак, формування гарної початкової популяції (наприклад, із множини локальних оптимумів) може помітно скоротити час досягнення глобального оптимуму. Таким чином, при наявності необхідної інформації завдання початкової популяції користувачем є кращим.

Найчастіше розмір початкової популяції вибирається в інтервалі 20–100 особин.



**Стратегія створення початкової популяції** може бути різною. Відомі наступні стратегії.

1. *Стратегія “ковдри”* – вихідна множина містить всі можливі варіанти рішень. Стратегія “ковдри” має наступні недоліки:

- у багатьох випадках неможливо здійснити повний перебір;
- з погляду адаптивного розвитку не представляє цінності, тому що часто вже в першому поколінні рішень відшукується оптимальне рішення.

2. *Стратегія “фокусування”* – стартова множина рішень включає різновиди одного рішення.

Стратегія “фокусування” застосовується в тих випадках, коли є припущення, що деяке рішення є різновидом відомого субоптимального. Тоді шляхом поступових незначних змін існуючого рішення можна одержати більш якісне субоптимальне рішення.

Для більшості задач оптимізації неприйнятні стратегії “ковдри” (внаслідок проблематичності повного перебору) і фокусування (відсутня чітка залежність якості рішення від параметрів рішення).

3. *Стратегія “дробовика”* – генерується досить велика множина рішень. Дана стратегія може бути реалізована одним з трьох способів.

3.1. Рівномірне формування початкової популяції.

3.2. Випадкове формування початкової популяції.

3.3. Комплементарне формування початкової популяції, яке виконується за два кроки.

Крок 1. Сформувати випадковим чином першу половину початкової популяції.

Крок 2. Сформувати другу половину початкової популяції шляхом додавання хромосом, протилежних (одиниці замінюються нулями) хромосомам в першій половині популяції.

### 1.2.6 Відбір

Оператор відбору (вибору, селекції) представляє собою оператор, що на основі значення цільової функції вибирає хромосоми таким чином, щоб з ненульовою ймовірністю будь-який елемент популяції міг би бути обраний у якості одного з батьків при схрещуванні.

Найпоширенішими є наступні оператори відбору:

- пропорційний відбір (пропорційно-імовірнісний відбір);

- відбір ранжируванням;
- турнірний відбір;
- відбір з використанням порогу.

### 1.2.6.1 Пропорційний відбір

Даний вид відбору складається з наступної послідовності кроків.

Крок 1. Обчислити пристосованість кожної особини  $f_j$ .

Крок 2. Знайти середню пристосованість у популяції  $f_{cp}$  як середнє арифметичне значень пристосованості 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 шляхами:

Перший шлях (стохастичний залишковий відбір): якщо  $P_s(j) > 1$ , то особина вважається добре пристосованою та допускається до схрещування.

Наприклад, якщо дріб  $P_s(j) = 2,36$ , то дана особина має подвійний шанс на схрещування й буде мати ймовірність рівну 0,36 третього схрещування. Якщо ж пристосованість дорівнює 0,54, то особина візьме участь у єдиному схрещуванні з імовірністю 0,54.

Другий шлях: після знаходження відносини  $P_s(j)$  відбувається відбір (із заміщенням) всіх  $N$  особин для подальшої генетичної обробки, відповідно до величини  $P_s(j)$ .

Найпростіший пропорційний відбір – *рулетка* – відбирає особини за допомогою  $N$  запусків рулетки.

Колесо рулетки містить по одному сектору для кожного члена популяції. Розмір  $j$ -го сектору пропорційний відповідній величині  $P_s(j)$ . Особина одержує можливість створення нащадків, якщо випадково згенероване число в межах від 0 до  $2\pi$  попадає в сектор, що відповідає цій особини. При такому відборі члени популяції з більш

високою пристосованістю з більшою ймовірністю будуть частіше вибиратися, чим особини з низькою пристосованістю.

При реалізації відбору рулеткою доцільно замінити колесо рулетки інтервалом  $[0;1]$  у зв'язку з тим, що в такому випадку немає необхідності обчислювати ширину кожного сектора – у цьому випадку кожній особині ставиться у відповідність напівінтервал  $[x_{j-1}; x_j]$ , де  $x_{j-1} - x_j = P_s(j)$ , а  $x_0 = 0$  (при цьому  $x_N = 1$ ). У наступне покоління переходить особина з номером  $j$ , де  $j: x_{\text{rnd}} \in [x_{j-1}; x_j]$ , а число  $x_{\text{rnd}}$  повертається щораз випадковою функцією з рівномірним розподілом щільності ймовірності на відрізку  $[0;1]$ .

Схема рулетки може давати дуже великі помилки, у тому розумінні, що кінцеве число нащадків даної особини може сильно відрізнятись від очікуваного. Кінцеве число наближається до очікуваного тільки в популяціях дуже більших розмірів.

### 1.2.6.2 Відбір ранжируванням

Відбір ранжируванням виконується за 4 кроки.

Крок 1. Обчислити пристосованість кожної особини  $f_j$ .

Крок 2. Відсортувати (ранжирувати) популяцію по зростанню пристосованості особин.

Крок 3. Для кожної особини обчислити величину  $P_s(j)$ . Для цього використати один із 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  $\eta_{\max} \in [1; 2]$ ,  $\eta_{\min} = 2 - \eta_{\max}$ .

$$\text{б) рівномірне ранжирування: } P_s(j) = \begin{cases} \frac{1}{\mu}, & 1 \leq i \leq \mu, \\ 0, & \mu < i \leq N, \end{cases}$$

де  $\mu$  – деяке фіксоване число перших членів популяції.

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

### 1.2.6.3 Турнірний відбір

Турнірний відбір реалізує  $k$  турнірів, щоб вибрати  $k$  особин. Кожний турнір складається із двох етапів.

Етап 1. Вибір  $m$  елементів з популяції.

Етап 2. Вибір кращої особини серед особин, відібраних на попередньому етапі.

Розмір групи особин, що відбираються для турніру, часто дорівнює 2. У цьому випадку говорять про *парний турнір*. Взагалі ж  $m$  називається *чисельністю турніру*.

Турнірний відбір має певні переваги перед пропорційним, тому що не втрачає своєї вибірковості, коли в ході еволюції всі елементи популяції стають приблизно рівними за значенням цільової функції.

#### 1.2.6.4 Відбір з використанням порогу

Відбір з використанням порогу (відбір усіканням) виконується в наступній послідовності.

Крок 1. Обчислити пристосованість кожної особини  $f_j$ .

Крок 2. Відсортувати популяцію по зростанню пристосованості особин.

Крок 3. Задати поріг  $P \in [0;1]$ . Поріг визначає, яка частка особин, починаючи з найпершої (самої пристосованої), буде брати участь у відборі. В принципі, поріг можна задати й числом, більшим за одиницю, тоді він буде просто дорівнює числу особин з поточної популяції, допущених до відбору.

Крок 4. Серед особин, що потрапили під значення порога, випадковим образом  $N$  раз вибирати саму везучу й записувати 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 більшим для популяцій великого розміру й залежати також від алгоритму сортування.

#### 1.2.7 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 (кроссинговер, кроссовер, рекомбінація). Цей оператор моделює процес схрещування особин і визначає передачу ознак батьків нащадкам.

Метою оператору схрещування є породження з наявної множини рішень нової, у якій кожна хромосома буде нащадком деяких двох елементів попередньої популяції, тобто нести в собі частково інформацію кожного батька. Допускається ситуація, коли обидва батьки представлені одним елементом популяції.

### 1.2.7.1 Вибір батьківської пари

Вибираючи щораз для схрещування найбільш пристосовані особини, можна з певним ступенем впевненості стверджувати, що нащадки будуть або не набагато гіршими, ніж батьки, або кращими за них.

Існує кілька способів вибору батьківської пари.

*Випадковий вибір батьківської пари (панмісія)* – це найпростіший підхід, коли обидві особини, які утворюють батьківську пару, випадковим чином вибираються із всієї популяції, причому будь-яка особина може стати членом декількох пар.

Крок 1. Для вибору пари батьків задається ймовірність схрещування  $P_c$ .

Крок 2. Довільним чином нумеруються всі представники вихідної популяції.

Крок 3. Вибір першого батька: починаючи з першого рішення, проглядається популяція доти, поки випадково обране число з інтервалу  $[0, 1]$  не буде меншим, ніж  $P_c$ . Елемент, для якого виконується така умова, стає першим батьком.

Крок 4. Відбувається перегляд популяції, починаючи з наступному після першого батька рішення, поки знову випадково обране число не буде меншим, ніж  $P_c$ . Елемент, для якого виконується така умова, стає другим батьком.

Описаним способом складаються пари доти, поки не вибереться потрібна кількість пар батьків.

Конкретне значення  $P_c$  залежить від розв'язуваної задачі, і в загальному випадку лежить в інтервалі  $[0,6; 0,99]$ .

Незважаючи на простоту, такий підхід універсальний для розв'язування різних класів задач. Однак він досить критичний до чисельності популяції, оскільки ефективність методу, що реалізує такий підхід, знижується з ростом чисельності популяції.

Інший метод випадкового вибору батьківської пари може бути представлений у вигляді наступної 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) у підпопуляціях.

Крок 4. Допустити до схрещування пари, для яких випадково згенероване в інтервалі  $[0;1]$  число буде перевищувати задану ймовірність схрещування.

*Селективний спосіб* вибору особин у батьківську пару полягає в тому, що батьками можуть стати тільки ті особини, значення пристосованості яких не менше середнього значення пристосованості по популяції, при рівній ймовірності таких кандидатів утворити батьківську пару.

Такий підхід забезпечує більш швидку збіжність генетичного пошуку. Однак через швидку збіжність селективний вибір батьківської пари не підходить тоді, коли ставиться задача визначення декількох екстремумів, оскільки для таких задач метод, як правило, швидко збігається до одного з рішень.

Крім того, для деякого класу задач зі складним ландшафтом фітнес-функції швидка збіжність може перетворитися в передчасну збіжність до квазіоптимального розв'язку. Цей недолік може бути частково компенсований використанням відповідного механізму відбору, який би “гальмував” занадто швидку збіжність методу.

Інші два способи формування батьківської пари – це *інбридинг* та *аутбридинг*. Обоє ці методи побудовані на формуванні пари на основі близького й далекого “споріднення”, відповідно. Під “спорідненням” тут розуміється відстань між членами популяції як у сенсі евклідової (геометричної) відстані особин у просторі параметрів (для фенотипів), так і у сенсі відстані Хеммінгу між хромосомними наборами особин (для генотипів).

*Евклідова відстань*  $R^{(jk)}$  між  $j$ -ою та  $k$ -ою особинами популяції визначається за формулою:

$$R^{(jk)} = \sqrt{\sum_{i=1}^p (x_i^{(j)} - x_i^{(k)})^2},$$

де  $p$  – кількість параметрів (генів) особини;

$x_i^{(j)}$  –  $i$ -ий параметр у незакодованому вигляді  $j$ -ої особини.

*Відстань Хеммінгу*  $H^{(jk)}$  між  $j$ -ою та  $k$ -ою особинами популяції визначається як кількість різних бітів в однакових позиціях  $j$ -ої та  $k$ -ої хромосом.

*Інбридинг* складається з двох етапів:

1. Перший член пари вибирається випадково.
2. Другим батьком з більшою ймовірністю буде максимально близька до першого особина.

Один з варіантів процедури інбридингу може бути реалізований у такий спосіб.

Крок 1. Вибрати випадковим чином першого батька.

Крок 2. Вибрати з поточної популяції випадковим чином групу з  $C$  хромосом ( $C = 1\%–15\%$  від розміру популяції).

Крок 3. Розрахувати *Евклідову* відстань від хромосоми, отриманої на першому кроці, до кожної із  $C$  відібраних на другому кроці хромосом.

Крок 4. В якості другого батька вибрати найближчу до першого батька хромосому.

*Аутбридинг* формує батьківські пари з максимально далеких особин.

Використання генетичних інбридингу й аутбридингу є ефективним для багато екстремальних задач. Однак два цих способи по-різному впливають на поводження 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. Аутбридинг, навпаки, спрямований на попередження збіжності методу до вже знайдених рішень, змушуючи метод переглядати нові, недосліджені області.

### 1.2.7.2 Оператори схрещування

При *точковому* схрещуванні:

1. Випадково обираються  $n$  точок розриву, що приводить до розбиття вихідних векторів на  $n + 1$  частин різної довжини.

2. Обмінюються у вихідних хромосомах ділянки з парними номерами, а ділянки з непарними залишаються без змін:

Точкове схрещування може застосовуватися для бінарних, векторних і гомологічних числових хромосом.

Класичним варіантом такого схрещування є одноточечне схрещування, при якому:

1. Випадковим чином визначається точка в середині хромосоми. Ця точка називається точкою розриву (точкою *схрещування*, crossover point).

2. В обраній точці обидві хромосоми діляться на дві частини й обмінюються ними. У результаті утворюються два нащадки.

Даний тип схрещування називається одноточечним, тому що при ньому батьківські хромосоми розрізаються тільки в одній випадковій точці.

При двохточечному схрещуванні в хромосомі випадково вибираються вже дві точки схрещування. Ліву точку будемо вважати першою, а праву – другою. Перший нащадок формується із частин першого батька, розташованих лівіше від першої точки схрещування й правіше від другої точки, і частини другого батька, розташованої між першою й другою точками схрещування. Другий нащадок формується із лівої та правої частин другого батька й центральної частини першого батька.

Обчислювальні експерименти показали, що навіть для простих функцій не можна говорити про перевагу того або іншого оператора. Більше того, використання механізму випадкового вибору одно- або двохточечного схрещування для кожної конкретної батьківської пари часом виявляється більше ефективним, ніж детермінований підхід до його вибору, оскільки досить важко апріорно визначити, який із двох операторів більше підходить для кожного конкретного виду функції пристосованості.

**Однорідне схрещування** (uniform crossover) генерує нащадка шляхом випадкової передачі йому генетичної інформації від батьків. Генерація нащадка виконується в такий спосіб.

Крок 1. Встановити лічильник бітів (генів) нащадка:  $j = 1$ .

Крок 2. Визначити хромосому з батьківської пари, що передасть значення свого  $j$ -го гена нащадкові:  $n = \text{round}(\text{rand}[0;1])$ , де  $\text{rand}[0;1]$  – випадково згенероване число в інтервалі  $[0;1]$ ;  $\text{round}(A)$  – округлене значення числа  $A$ .

Крок 3. Виконати:  $h_{jn} = h_{jn}$ , де  $h_{jn}$  – 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 на крок 6. У противному випадку перейти на крок 2.

Крок 6. Кінець.

**Рівномірне схрещування** може застосовуватися для бінарних, гомологічних числових і векторних хромосом. Таке схрещування зручно застосовувати в тому випадку, коли вже отримані індивіди з гарними спадкоємними ознаками, і їх необхідно закріпити в поточній популяції. Рівномірне схрещування виконується в наступній послідовності кроків.

Крок 1. Випадковим чином задається маска схрещування, що представляє собою рядок з нулів та одиниць, довжини, що дорівнює довжині хромосом.

Крок 2. Формування першого нащадка.

Одиниця на конкретній позиції в масці схрещування означає, що елемент, що знаходиться на тому ж місці в першого батька, необхідно помістити на це місце в першій дитині. Нуль на цій позиції в масці схрещування означає, що елемент, що знаходиться на тому ж місці в другого батька, необхідно помістити на це місце першій дитині.

Крок 3. Формування другого нащадка.

Якщо першого батька вважати другим, а другого – першим, то аналогічно кроку 2 можна одержати другого нащадка.

Варто відзначити, що маска схрещування може бути одна для всіх хромосом, або своя для кожної пари батьків.

При **порівняльному схрещуванні** в хромосомах батьків порівнюються всі біти. Якщо на однакових позиціях обох батьків знаходяться однакові біти (0 і 0 або 1 і 1), то нащадкам присвоюються ті ж значення відповідних бітів. Якщо на однакових позиціях батьків розташовані гени із різними значеннями, тоді значення відповідних генів нащадків визначають за допомогою генератору випадкових чисел.

**Арифметичне (диференціальне) схрещування** є найбільш вдалим для пошуку оптимуму функції багатьох дійсних змінних.

Нехай  $H_1$  і  $H_2$  – це два індивідууми в популяції, тобто дійсні вектори, від яких залежить цільова функція. Тоді нащадок  $H_n$  обчислюється за формулою  $H_n = H_1 + k \cdot (H_1 - H_2)$ , де  $k$  – це деякий дійсний коефіцієнт (який може залежати від  $\|H_1 - H_2\|$  – відстані між векторами). У цій моделі мутація – це додавання до індивідуума

випадкового вектора малої довжини. Якщо вихідна функція неперервна, то ця модель працює добре, а якщо вона ще й гладка, те – відмінно.

При *діагональному схрещуванні* для схрещування  $R$  батьків випадковим чином вибирається  $(R - 1)$  однакових точок схрещування в кожному з них.  $R$  нащадків отримують шляхом комбінування відповідних елементів батьків по діагоналі.

Крок 1. Вибрати випадковим чином  $R$  батьківських хромосом для схрещування.

Крок 2. Вибрати випадковим чином  $(R - 1)$  точок схрещування в хромосомах, розділивши тим самим кожну з них на  $R$  сегментів.

Крок 3. Встановити:  $r = 1$ .

Крок 4. Сформувати  $r$ -го нащадка.

Крок 4.1 Встановити:  $k = 1$ .

Крок 4.2. Сформувати  $k$ -ий сегмент  $r$ -го нащадка, взявши  $g$ -ий сегмент із  $k$ -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  $k \leq R$ , виконати перехід до кроку 4.2.

Крок 5. Виконати:  $r = r + 1$ .

Крок 6. Якщо  $r \leq R$ , виконати перехід до кроку 4.

Крок 7. Кінець.

Слід зазначити, що різні типи схрещування мають загальну позитивну властивість: вони контролюють баланс між подальшим використанням уже знайдених гарних підобластей простору пошуку та дослідженням нових підобластей. Це досягається за рахунок неруйнування загальних блоків усередині хромосом-батьків і одночасного дослідження нових областей в результаті обміну частинами хромосом.

Спільне використання операторів відбору та схрещування призводить до того, що області простору з кращою в середньому оптимальністю, вмщують більше членів популяції, чим інші. Тому, оператор схрещування є найбільш критичним із всіх операторів генетичних методів з поглядом одержання глобальних результатів.

У малих популяціях краще застосовувати більш руйнівні варіанти схрещування (багатоточечне та однорідне), а в великих популяціях краще працює двохточечне.

### 1.2.8 Мутація

Оператор мутації полягає в зміні генів у випадково обраних позиціях. На відміну від операторів відбору та схрещування, які використовуються для поліпшення структури хромосом, метою оператора мутації є диверсифікація, тобто підвищення розмаїтості пошуку й введення нових хромосом у популяцію для того, щоб більш повно досліджувати простір пошуку. Оскільки число членів популяції  $P$  звичайно вибирається значно меншим у порівнянні із загальним числом ( $2^L$ ) можливих хромосом у просторі пошуку, то в силу цього вдається досліджувати лише його частину. Отже, мутація ініціює розмаїтість у популяції, дозволяючи переглядати більше рішень у просторі пошуку й виходити в такий спосіб з локальні екстремумів в процесі пошуку.

В результаті виконання мутації з ненульовою ймовірністю чергове рішення може перейти в будь-яке інше рішення.

Вибір хромосом для мутації відбувається в такий спосіб.

Крок 1. Нумеруються довільним чином всі хромосоми  $H_j$  вихідної популяції.

Крок 2. Починаючи з першої хромосоми, проглядається 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 з 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 й рідко перевищує 1%.

Необхідно відзначити, що оператор мутації є основним пошуковим оператором і відомі такі методи, що не використають інших операторів крім мутації.

### 1.2.8.1 Проста мутація

Проста мутація використовується для бінарних, гомологічних числових і векторних хромосом. Суть її полягає у внутригенній мутації. При цьому в хромосомі випадковим чином вибирається ген, а потім відбувається його випадкова зміна.

*Алгоритм простої мутації для бінарних і гомологічних числових хромосом* може складатися з наступних кроків.

Крок 1. Скопіювати батьківську хромосому в хромосому-нащадка.

Крок 2. Вибрати випадковим чином ген для мутації.

Крок 3. У заданому інтервалі припустимих значень гена вибрати нове значення гена, що не дорівнює поточному.

Для *векторних хромосом* проста мутація відбувається шляхом внесення змін у 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 мутуючого гена.

Крок 4. Зробити обмін значеннями між розрядами мутуючого гена, що перебувають безпосередньо ліворуч і праворуч від точки мутації.

Відомий також модифікований оператор простої мутації, що називається *точковою мутацією*, який відрізняється тим, що в хромосомі мутує не один, а кілька генів із заданою ймовірністю. Такий оператор використовується для бінарних, гомологічних числових і векторних хромосом. Алгоритм даного оператора наведений 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 меншим ймовірності мутації гену  $P_{mg}$ , то виконати мутацію гена  $h_i$ .

Крок 5. Встановити  $i = i + 1$ .

Крок 6. Якщо  $i < (L + 1)$ , де  $L$  – довжина хромосоми, то виконати перехід на крок 3.

Крок 7. Кінець.

Крім того, відома множина різних варіантів операторів мутації. Більшість із них використовують комбінації наступних ідей.

1. Так само як і схрещування, мутація може проводитися не тільки по одній випадковій точці. Можна вибирати деяку кількість точок у хромосомі для інверсії, причому їхнє число також може бути випадковим.

2. Піддається мутації відразу деяка група послідовних генів.

3. Спільне застосування випадкової мутації та часткової перебудови рішення алгоритмами локального пошуку. Застосування локального спуску дозволяє генетичному методу зосередитися тільки на локальних оптимумах. Множина локальних оптимумів може виявитися експоненціально більшим і на перший погляд здається, що такий варіант методу не буде мати великих переваг. Однак експериментальні дослідження розподілу локальних оптимумів свідчать про високу концентрацію їх у 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.

### 1.2.8.2 Мутація гомологічних числових хромосом

Такі види мутацій полягають в зміні обраного для мутації гена  $h_{ji}$  (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 обраного для мутації  $i$ -го гену  $h_{ji}$  хромосоми  $H_j$  застосовується за формулою:

$$h_{ji}' = \begin{cases} h_{ji} + \Delta(t, \max_i - h_{ji}) & \text{з ймовірністю } 0,5, \\ h_{ji} - \Delta(t, h_{ji} - \min_i) & \text{з ймовірністю } 0,5, \end{cases}$$

$$\text{де } \Delta(t, y) = y \cdot \left( 1 - r^{(1-t/T)^k} \right);$$

$r = \text{rand}[0; 1]$  – випадково згенероване число в інтервалі  $[0; 1]$ ;

$t$  – номер поточної ітерації;

$T$  – максимальна кількість ітерацій;

$k$  – параметр, що визначає ступінь однорідності (рівномірності);

$\min_i$  і  $\max_i$  – мінімальне й максимальне значення  $i$ -го параметру

в розв'язуваній за допомогою генетичного методу задачі.

Крім того, нерівномірна мутація  $i$ -го гену  $j$ -ої хромосоми  $h_{ji}$  може бути виконана за формулою:

$$h_{ji}' = \begin{cases} h_{ji} + \Delta(t, \max_i - h_{ji}), & \text{якщо } f(H_j + \Delta(t, \max_i - h_{ji})) \geq f(H_j - \Delta(t, h_{ji} - \min_i)), \\ h_{ji} - \Delta(t, h_{ji} - \min_i), & \text{якщо } f(H_j + \Delta(t, \max_i - h_{ji})) < f(H_j - \Delta(t, h_{ji} - \min_i)), \end{cases}$$

де  $\Delta(t, y) = y \cdot w_i(t)$ ;  $w_i(t)$  – коефіцієнт, що залежить від

відношення  $t/T$ .

Наприклад, коефіцієнт  $w_i(t)$  може бути заданий формулою:

$$w_i(t) = r \left( 1 - \frac{t}{T} \right)^{k_i},$$

де  $r = \text{rand}[0; 1]$  – випадково згенероване число в інтервалі  $[0; 1]$ ;

$k_i > 0$  – параметр, що задає користувач.

2. **Випадкова мутація** обраного гена  $h_{ji}$  полягає в зміні його значення на величину  $\Delta h_{ij}$ , розраховану за формулою:

$$\Delta h_{ij} = \text{rand}[\min_i \cdot r \cdot q(t); \max_i \cdot r \cdot q(t)],$$

де  $\text{rand}[a; b]$  – випадково згенероване число в інтервалі  $[a; b]$ ;

$\min_i$  і  $\max_i$  – мінімальне й максимальне значення  $i$ -го гену;

$r = \text{rand}[0; 1]$  – випадково згенероване число в інтервалі  $[0; 1]$ ;

$$q(t) = \frac{\ln T - \ln t}{\ln T}.$$

3. **Гауссовська (нормальна) мутація** до обраного для мутації  $i$ -го гену  $j$ -ої хромосоми  $h_{ji}$  застосовується за формулою:

$$h_{ji}' = h_{ji} + \varepsilon,$$

де  $\varepsilon$  – випадкове число, отримане за нормальним розподілом (Коші, або будь-якому іншому розподілу) з нульовим середнім і

$$\sigma_i = \frac{T - t}{T} \cdot \frac{\max_i - \min_i}{3}.$$

Число  $\varepsilon$  може бути додане до одного гена. Можливий варіант додавання випадкового вектора до всієї хромосоми.

4. **Мутація** обраного гена  $h_{ij}$  на основі **квадратичної апроксимації**.

Крок 1. Обчислити значення фітнесс-функції при  $h_{ij} + \Delta h_{ij}$  і при  $h_{ij} - \Delta h_{ij}$ :  $f(h_{ij} + \Delta h_{ij})$  і  $f(h_{ij} - \Delta h_{ij})$ . Значення  $\Delta h_{ij}$  можуть бути обчислені за формулами знаходження  $\Delta$  і  $\varepsilon$ , аналогічними нерівномірній та Гауссовській мутації.

Крок 2. Апроксимувати точки  $h_{ij}$ ,  $h_{ij} + \Delta h_{ij}$  і  $h_{ij} - \Delta h_{ij}$  у параболу.

Крок 3. Знайти мінімальне значення отриманої кривій  $f_{\text{параб min}}$  і відповідне значення точки в просторі ознак, що відповідає мінімальному значенню параболи  $h_{ij \text{ min}}$ .

Крок 4. Присвоїти:  $h_{ij} = h_{ij \text{ min}}$ .

Важливо відзначити, що описані оператори мутації можуть застосовуватися й для бінарних хромосом, попередньо перетворених до реальних числових значень із погляду розв'язуваної задачі. Після застосування описаних вище операторів мутації для таких хромосом їх необхідно знову перетворити до бінарного вигляду, застосувавши використовуваний метод кодування.

### 1.2.9 Формування нового покоління

Після схрещування та мутації необхідно створити нову популяцію. Види операторів формування нового покоління (репродукції, редукції) практично збігаються з видами операторів відбору батьків, що передбачають формування проміжного масиву особин, допущених до схрещування.

При формуванні нового покоління необхідно вирішити проблему: які з нових особин увійдуть у наступне покоління, а які – ні. Для цього застосовують один із двох способів.

Перший спосіб полягає в тому, що нові особини (нащадки) займають місця своїх батьків. Після чого починається наступний етап, у якому нащадки оцінюються, відбираються, дають потомство й поступаються місцем своїм нащадкам.

Недоліком даного способу є можливість втрати найбільш пристосованої особини попереднього покоління. Одним зі способів вирішення даної проблеми може бути використання *принципу "елітизму"*, що полягає в тому, що особини з найбільшою пристосованістю гарантовано переходять у нову популяцію. Їхнє число може бути від 1 і більше. Кількість елітних особин  $k_e$ , які гарантовано перейдуть у наступну популяцію, може бути обчислена за формулою:

$$k_e = (1 - S_0) \cdot N,$$

де  $S_0$  – ступінь відновлення популяції, що перебуває в діапазоні  $[0,95; 1,0]$ ;  $N$  – розмір популяції.

Використання принципу “елітизму” дозволяє прискорити збіжність генетичного методу. Недолік використання даної стратегії в тому, що підвищується ймовірність попадання методу в локальний мінімум.

Другий спосіб заснований на тому, що створюється проміжна популяція, яка містить у собі як батьків, так і їхніх нащадків. Члени цієї популяції оцінюються, а потім з них вибираються  $N$  найкращих, які й увійдуть у наступне покоління.

Другий варіант є більше оптимальним, але він вимагає сортування масиву розміром  $2N$ .

Другий варіант формування нового покоління можна реалізувати за допомогою *принципу витиснення*, що носить двохкритеріальний характер – те, чи буде особина з репродукційної групи заноситися в популяцію нового покоління, визначається не тільки величиною її пристосованості, але й тим, чи є вже у популяції наступного покоління особина з аналогічним хромосомним набором. Із всіх особин з однаковими генотипами перевага спочатку віддається тим, чия пристосованість вище. Таким чином, досягаються дві мети: по-перше, не губляться кращі знайдені рішення з різними хромосомними наборами, а по-друге, у популяції постійно підтримується достатня генетична розмаїтість. Витиснення в цьому випадку формує нову популяцію скоріше з далеко розташованих особин, замість особин, що групуються біля поточного знайденого рішення.

### 1.2.10 Критерії зупину

Одним з важливих етапів у генетичних методах є визначення критеріїв зупину. Очевидно, еволюція – нескінченний процес, у ході якого пристосованість особин поступово підвищується. Примусово зупинивши цей процес через досить довгий час після його початку й обравши найбільш пристосовану особину в поточному поколінні, можна одержати не абсолютно точну, але близьку до оптимальної відповідь.



Як правило, в якості критерію зупину застосовується *обмеження на максимальну кількість ітерацій* функціонування методу (тобто обмеження на кількість поколінь). Кількість популяцій може бути будь-якої, але частіше за все обирають 50-100 популяцій. Якщо в якості критерію зупину обирається максимальна кількість ітерацій, то задається кількість ітерацій  $T$ , в результаті чого цикл генетичного пошуку виконується  $T$  раз.

Зупин роботи генетичного методу може відбутися також у випадку, якщо *популяція вироджується*, тобто якщо практично немає розмаїтості в генах особин популяції. Виродження популяції називають передчасною збідністю.

Якщо в якості критерію зупину обирається виродження популяції, то задаються кількість ітерацій  $T$  і відсоток поліпшення значення кращої хромосоми  $F$ .

Починаючи із циклу  $T+1$ , у генетичних операторах обчислюються значення цільової функції для кожної хромосоми останньої популяції, і вибирається із цих значень найкраще значення цільової функції  $f_{best_{T+t}}$ . З попередніх  $T$  поколінь вибирається найкраще значення цільової функції  $f_{best_{T+t-1}}$ . Після чого обчислюється відсоток поліпшення  $F'$  по формулі:

$$F' = \frac{100 \cdot (f_{best_{T+t}} - f_{best_{T+t-1}})}{f_{best_{T+t-1}}}$$

Якщо значення  $F'$  виявляється меншим, ніж  $F$ , то критерій зупину вважається досягнутим, в протилежному випадку виконується наступний цикл генетичного пошуку.

*Прийнятне значення цільової функції  $f_n$*  також може використовуватися в якості критерію зупину. Якщо в процесі функціонування генетичного методу значення цільової функції  $f$  деякої особини досягло значення  $f_n$  з визначеною заздалегідь заданою точністю  $\varepsilon$ , то метод зупиняється. При цьому розв'язком задачі є отримане значення цільової функції  $f_n$ .

### 1.2.11 Генетичний пошук в пакеті Matlab

Для використання генетичних методів, у середовищі Matlab передбачена функція ga:

```
[x, Fmin] = ga(@fitnessfun, n, options)
```

Функція `ga` знаходить мінімум `Fmin` функції `fitnessfun`, що має `n` параметрів, а також вектор `x`, що мінімізує цільову функцію.

Параметри роботи функції `ga` задаються в змінній `options`, що представляє собою наступну структуру:

```
options =
    PopulationType: 'doubleVector'
    PopInitRange: [2x1 double]
    PopulationSize: 20
    EliteCount: 2
    CrossoverFraction: 0.8000
    MigrationDirection: 'forward'
    MigrationInterval: 20
    MigrationFraction: 0.2000
    Generations: 100
    TimeLimit: Inf
    FitnessLimit: -Inf
    StallLimitG: 50
    StallLimitS: 20
    InitialPopulation: []
    InitialScores: []
    PlotInterval: 1
    CreationFcn: @gacreationuniform
    FitnessScalingFcn: @fitscalingrank
    SelectionFcn: @selectionstochunif
    CrossoverFcn: @crossoverscattered
    MutationFcn: @mutationgaussian
    HybridFcn: []
    Display: 'final'
    PlotFcns: []
    OutputFcns: []
    Vectorized: 'off'
```

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 використати 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 використати таку команду:

```
scale = 0.5; shrink = 0.75;
options = gaoptimset('MutationFcn',{@mutationgaussian,scale,shrink})
```

Matlab, починаючи з версії 7.0, містить візуальний інтерфейсний модуль для роботи з генетичними методами – Genetic Algorithm Tool (GAT), що входить до складу бібліотеки Genetic Algorithm and Direct Search Toolbox. Запуск GAT відбувається за командою:

```
gatool
```

### 1.3 Порядок виконання роботи

1.3.1 Ознайомитися з основними теоретичними відомостями та рекомендованою літературою за темою роботи.

1.3.2 Вивчити роботу функції `ga` та візуального інтерфейсного модулю GAT пакету Matlab

1.3.3 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.

1.3.4 Виконати тестування розробленого 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.

1.3.5 Оформити звіт з роботи.

1.3.6 Відповісти на контрольні питання.

## 1.4 Індивідуальні завдання

### 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.

### 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.

### 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, мінімальне та максимальне значення отриманого оптимуму цільової функції та часу оптимізації.

### 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 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.

### 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  $M$ , 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:  $MI = 7$ ,  $T = 100$ ).

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.

### Варіант №6

1. Розробити за допомогою пакету Matlab програму, що оптимізує 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.

### Варіант №7

1. Розробити за допомогою пакету Matlab програму, що оптимізує 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.

### Варіант №8

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_8(x_1, x_2, x_3, x_4) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2 + 90 \cdot (x_4 - x_3^2)^2 + 10 \cdot (x_2 - 1)^2 + (x_4 - 1)^2$ , 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.

### Варіант №9

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_9(x_1, x_2) = (x_2 - x_1)^2 + (1 - x_1)^2$ , використовуючи еволюційний пошук. В якості еволюційних операторів обрати: відбір – пропорційний, схрещування – однокрестове, 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.

### 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.

### 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.

### 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:  $shrink_i = 0,05 + 0,065 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від параметру  $shrink$ . Вивести на екран графік синтезованої моделі.

### Варіант №13

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{13}(x_1, x_2) = 1,4388 + \frac{(\sin \sqrt{x_1^2 + x_2^2 + 1} \cdot \sin \sqrt{x_1^2 + 1} - 0,5)}{(1 + 0,001 \cdot (x_1^2 + x_2^2))^2}$ ,

використовуючи 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.

### 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 – одноточкове, мутація – гассовська; кількість елітних особин – 5. Використовувати острівну модель 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.

### 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.

### 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.

### Варіант №17

1. Розробити за допомогою пакету Matlab програму, що

оптимізує функцію  $f_{17}(x) = \begin{cases} x^2, & \text{якщо } x \leq -10, \\ x^2 \cdot |\sin x| + x, & \text{якщо } -10 < x \leq -5, \\ x^2 \cdot |\sin x|, & \text{якщо } -5 < x \leq 5, \\ x, & \text{якщо } x > 5, \end{cases}$

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 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.

### 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.

### 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 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 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.

### Варіант №20

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{20}(x_1, \dots, x_n) = n \cdot A + \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i))$ ,  $n = 7$ ,  $A = 10$ ,

використовуючи 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.

### Варіант №21

1. Розробити за допомогою пакету Matlab програму, що оптимізує функцію  $f_{21}(x) = |x| + \sin(x)$ , використовуючи еволюційний



пошук. В якості 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.

### 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.

### 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 еволюційний 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.

### 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 еволюційний 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.

### 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 – односточкове, 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 обґрунтувати.

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.

### 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 обґрунтувати.

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.

### 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:  $\text{shrink}_i = 0,05 + 0,065 \cdot i$ . Побудувати лінійну регресійну модель залежності середнього значення отриманого оптимуму цільової функції від параметру shrink. Вивести на екран графік синтезованої моделі.

### 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 – гауссовська. 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 обґрунтувати.

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.

### 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 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 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.

### 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 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 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.

## 1.5 Контрольні питання

1.5.1 Порівняйте методи еволюційного пошуку з іншими методами оптимізації.

1.5.2 Які методи відносять до еволюційних?

1.5.3 В чому переваги еволюційних методів?

1.5.4 Проаналізуйте умови ефективного використання методів еволюційного пошуку.

1.5.5 Назвіть особливості еволюційних методів.

1.5.6 Які недоліки еволюційного пошуку та в чому вони полягають?

1.5.7 Дайте визначення основних термінів, що відносяться до теорії еволюційного пошуку: популяція, розмір популяції, число поколінь, хромосома, ген, локус, алель, фенотип, генотип.

1.5.8 Проаналізуйте узагальнену схему роботи еволюційних методів.

1.5.9 Наведіть послідовність виконання узагальненого еволюційного пошуку.

1.5.10 Які параметри необхідно визначати для роботи еволюційних методів?

1.5.11 Виконайте порівняльний аналіз канонічних моделей еволюційного пошуку.

1.5.12 В чому полягають особливості моделі Genitor?

1.5.13 Що таке гібридний еволюційний метод? Які існують стратегії взаємодії класичних та еволюційних методів?

1.5.14 Назвіть відмінності моделі СНС від класичних еволюційних методів.

1.5.15 Які особливості еволюційного методу із змінним часом життя хромосом?

1.5.16 Порівняйте мобільний еволюційний метод з класичними методами еволюційного пошуку. Для чого призначені оператори CUT та SPLICE?

1.5.17 Проаналізуйте паралельні та багаторівневі еволюційні методи.

1.5.18 Наведіть послідовність виконання еволюційного пошуку із зменшенням розміру популяції.

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

1.5.20 Порівняйте еволюційні стратегії з генетичними алгоритмами та методом імітації відпалу.

1.5.21 Виконайте порівняльний аналіз генетичного та еволюційного програмування.

1.5.22 Які параметри можна використовувати в функції *ga*? Яким чином вони задаються? Як отримати поточні параметри функції *ga*?

1.5.23 Проаналізуйте візуальний модуль для роботи з методами еволюційного пошуку *gatool*: призначення, використання, параметри, візуальні компоненти, методи представлення результатів еволюційного пошуку.



## 2 ЛАБОРАТОРНА РОБОТА № 2 ДОСЛІДЖЕННЯ ЕВОЛЮЦІЙНИХ ОПЕРАТОРІВ

### 2.1 Мета роботи

Вивчити основні оператори еволюційного пошуку.

### 2.2 Завдання до роботи

2.3.1 Ознайомитися з рекомендованою літературою за темою роботи.

2.3.2 Розробити за допомогою пакету Matlab програмне забезпечення, що реалізує 2 методи еволюційного пошуку з використання різних еволюційних операторів. Основні еволюційні оператори для реалізації еволюційних методів обрати з таблиці 2.1 відповідно до варіанту.

Таблиця 2.1 – 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

Продовження таблиці 2.1

№ варіанту	№ завдання	Еволюційні оператори		
		Відбір	Схрещування	Мутація
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.

2.3.4 Виконати тестування розробленого 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.

2.3.5 Порівняти одержані результати оптимізації різних функцій за допомогою обох реалізованих еволюційних методів. Результати порівняльного аналізу звести до таблиці, попередньо розробивши систему критеріїв порівняння методів еволюційного пошуку.

2.3.6 Оформити звіт з роботи.

2.3.7 Відповісти на контрольні питання.

## 2.3 Зміст звіту

2.4.1 Тема та мета роботи.

2.4.2 Короткі теоретичні відомості.

2.4.3 Текст розробленого програмного забезпечення з коментарями (включно з усіма модулями, що реалізують основні етапи еволюційного пошуку та еволюційні оператори), а також текст програми для тестування розроблених еволюційних методів.

2.4.4 Графіки та аналітичні вирази обраних тестових функцій.

2.4.5 Результати роботи програмного забезпечення (таблиця порівняльного аналізу розроблених еволюційних методів).

2.4.6 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

## 2.4 Контрольні питання

2.4.1 Які існують способи кодування параметрів, що оптимізуються, при використанні еволюційних методів?

2.4.2 Що таке фітнес-функція?

2.4.3 Порівняйте стратегії створення початкової популяції.

2.4.4 Виконайте порівняльний аналіз операторів відбору (пропорційний відбір, відбір за допомогою ранжирування, турнірний відбір та відбір з використанням порогу).

2.4.5 Які способи формування батьківської пари використовуються в еволюційних методах?

2.4.6 Проаналізуйте оператори схрещування ( $n$ -точкове, рівномірне, порівняльне, арифметичне, діагональне).

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

2.4.8 Яким чином відбувається формування нового покоління?

2.4.9 Які критерії зупину використовуються при еволюційному пошуку?

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

## **3 ЛАБОРАТОРНА РОБОТА №3 СТАТИСТИЧНИЙ АНАЛІЗ РЕЗУЛЬТАТІВ ЕВОЛЮЦІЙНОЇ ОПТИМІЗАЦІЇ**

### **3.1 Мета роботи**

3.1.1 Навчитися обирати оптимальні параметри для роботи еволюційних методів.

3.1.2 Ознайомитися із статистичними методами дослідження результатів роботи методів еволюційного пошуку.

### **3.2 Основні теоретичні відомості**

Більшість оптимізаційних методів, зокрема і методи еволюційного пошуку, є евристичними, тобто вони не мають чіткого математичного обґрунтування, що викликає необхідність підбору різних параметрів еволюційних методів для досягнення оптимальних результатів з урахуванням особливостей розв'язуваної задачі.

Вибір оптимальних параметрів методів еволюційного пошуку доцільно проводити за допомогою використання методів математичної статистики шляхом виконання статистичного аналізу даних, отриманих при багаторазовому застосуванні еволюційних методів.

В результаті багаторазового використання еволюційних методів отримується певна вибірка даних, що містить відомості про досліджувані параметри еволюційного пошуку (тип та параметри еволюційних операторів, кількість елітних особин, максимальна кількість ітерацій, і т.ін.) та їх вплив на отримувані результати пошуку (досягнута точність, час виконання методу, і т.ін.).

Вихідні параметри і більшість вхідних параметрів у вибірці є випадковими величинами, тобто величинами, що приймають заздалегідь невідомі значення. Випадкові величини можуть бути дискретними або неперервними. Основною характеристикою випадкової величини є розподіл ймовірностей.

Числові характеристики розподілу ймовірностей допомагають скласти наочне представлення про цей розподіл. Основними

характеристиками розподілу ймовірностей випадкової величини служать моменти (математичне очікування, дисперсія, асиметрія, ексцес) та квантілі.

Часто дані групують, розбиваючи числову вісь на інтервали та вказуючи для кожного інтервалу число вибірки, які у нього потрапили. Графічне зображення залежності частоти влучення елементів вибірки від відповідного інтервалу групування називається гістограмою вибірки.

Найбільш розповсюдженим є нормальний розподіл ймовірностей, він часто використовується для наближеного опису багатьох випадкових явищ. Повнота теоретичних дослідів, а також порівняно прості математичні властивості роблять нормальний закон найбільш привабливим та зручним у використанні, тому що: по-перше, можна використовувати нормальний закон у якості першого наближення; по-друге, легко підібрати таке перетворення досліджуваної величини, яке видозмінить вихідний “ненормальний” закон розподілу, перетворюючи його у нормальний.

Часто на практиці на основі вибірових даних необхідно перевірити, чи є вірною певна гіпотеза. В якості гіпотез можуть бути гіпотеза про нормальність розподілу випадкової величини, гіпотеза про рівність дисперсій двох генеральних сукупностей, і т.ін. Для перевірки гіпотез застосовують різні критерії, зокрема критерій Пірсона, Комогорова, Фішера, Стьюдента, Кохрена.

З метою вибору оптимальних параметрів методів еволюційного пошуку доцільно перевіряти їх вплив на вихідний параметр.

Для дослідження залежностей впливу одного фактору (параметру) на кінцевий результат можна застосовувати однофакторний дисперсійний аналіз. При цьому фактор повинен приймати лише кінцеве число значень (рівнів). Для оцінки впливу факторів на відгук при використанні однофакторного дисперсійного аналізу експериментальні статистичні дані отримують наступним способом: кожний з  $K$  рівнів фактора застосовують кілька разів до об'єкту, який досліджується, і реєструють результати. Підсумком випробувань є  $K$  вибірок. При однофакторному дисперсійному аналізі використовують нульову гіпотезу, яка припускає, що всі дані належать до одного розподілу, тобто впливу фактору на результуючий параметр не існує.

Статистична перевірка взаємозв'язку між двома випадковими величинами також може бути здійснена за допомогою методів кореляційного та регресійного аналізів. Кореляційний аналіз виконується за допомогою обчислення одного з відомих коефіцієнтів кореляції: коефіцієнт парної кореляції, коефіцієнт кореляції Пірсона, коефіцієнт кореляції Спірмена. Використання регресійного аналізу дозволяє побудувати функціональну залежність між двома групами числових змінних  $X$  та  $Y$ .

### 3.3 Завдання до роботи

3.3.1 Вивчити основні теоретичні відомості. Відновити знання, вміння та навички, отримані при вивченні дисциплін, пов'язаних із статистичною обробкою експериментальних даних.

3.3.2 За допомогою пакету Matlab розробити програмне забезпечення, що реалізує метод еволюційного пошуку. Параметри еволюційного методу обрати з табл. 3.1 відповідно до варіанту.

Таблиця 3.1 – 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.

3.3.3 Отримати масив даних  $F = \{y_{1min,j}\}$ ,  $y = 1, 2, \dots, 100$ , для статистичного аналізу. Для цього провести серію зі 100 експериментів з фіксованими параметрами еволюційного пошуку для вирішення задачі оптимізації тестової функції  $y_1$ , обраної в попередній лабораторній роботі.

3.3.4 На основі отриманого масиву даних  $F$  побудувати гістограму, таблицю частот та одержати числові характеристики розподілу ймовірностей (математичне очікування, дисперсію, асиметрію, ексцес).

3.3.5 Перевірити гіпотезу про нормальний розподіл вибірки  $F$ , використовуючи критерій Пірсона  $\chi^2$  і критерій Колмогорова.

3.3.6 Аналогічно п.3.3.3, 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. 3.2).

Таблиця 3.2 – 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

3.3.7 За допомогою однофакторного дисперсійного аналізу дослідити вплив параметру (фактору), що регулювався в п.3.3.6, на досягнуте оптимальне значення функції  $y_1$ .

3.3.8 Побудувати таблицю  $(X, Z)$ , де  $x_i$  – значення параметру, що регулювався в п.3.3.6 ( $i = 1, 2, \dots, 10$ ),  $z_i$  – середнє значення досягнутого оптимального значення функції  $y_1$  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 аналізу.

3.3.9 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.

3.3.10 Оформити звіт з роботи.

3.3.11 Відповісти на контрольні питання.

## 3.4 Зміст звіту

3.4.1 Тема та мета роботи.

3.4.2 Короткі теоретичні відомості.

3.4.3 Текст розробленого програмного забезпечення з коментарями.

3.4.4 Отриманий набір даних  $F$  для статистичного аналізу (якщо він великий – навести фрагмент). Побудовані гістограми, таблиці частот та одержані числові характеристики розподілу ймовірностей (математичне очікування, дисперсію, асиметрію, ексцес).

3.4.5 Результати (з поясненнями) перевірки гіпотези про нормальність розподілу вибірки  $F$ , отримані за допомогою використання критерію Пірсона  $\chi^2$  та критерію Колмогорова.

3.4.6 Отримані входні дані для виконання однофакторного дисперсійного аналізу. Результати (з поясненнями) однофакторного дисперсійного аналізу щодо дослідження впливу параметру, що регулювався в п.3.3.6, на досягнуте оптимальне значення функції  $y_1$ .



3.4.7 Отримані вхідні дані та результати з поясненнями статистичної перевірки взаємозв'язку між  $X$  та  $Z$ , отримані за допомогою використання методів кореляційного та лінійного регресійного аналізу.

3.4.8 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

### 3.5 Контрольні питання

3.5.1 З якою метою виконують статистичний аналіз результатів еволюційної оптимізації?

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

3.5.3 Дайте визначення випадкової величини. Чим відрізняється дискретна випадкова величина від неперервної?

3.5.4 Що таке ряд розподілу випадкової величини? Як визначається кількість та ширина інтервалів при побудові ряду розподілу?

3.5.5 Як будується гістограма? Як змінюється вигляд гістограми при зміні величини інтервалу?

3.5.6 Що таке таблиця частот?

3.5.7 Яким чином густина розподілу пов'язана з інтегральною функцією розподілу?

3.5.8 Проаналізуйте закони розподілу випадкових величин.

3.5.9 Які є основні характеристики розподілу ймовірностей.

3.5.10 Як визначити знаки коефіцієнтів асиметрії та ексцесу з вигляду графіка щільності ймовірності?

3.5.11 Що таке квантіль, квартиль, медіана?

3.5.12 У чому полягає основний принцип перевірки статистичних гіпотез?

3.5.13 У чому полягає суть критерію Пірсона, Колмогорова, Фішера, Стьюдента, Кохрена?

3.5.14 Яким критерієм визначається закон розподілу? Записати аналітичний вираз.

3.5.15 Як використовується правило трьох сігм для визначення закону розподілу випадкової величини?

3.5.16 Що таке нормальний закон розподілу випадкових величин? Які параметри використовуються в ньому? Запишіть аналітичний вираз та накресліть функцію густини ймовірності для нормального закону розподілу. Чому дорівнюють коефіцієнт асиметрії та коефіцієнт ексцесу для нормального закону?

3.5.17 В чому полягає постановка задачі однофакторного дисперсійного аналізу?

3.5.18 Поясніть терміни і визначення, які використовуються при однофакторному дисперсійному аналізі.

3.5.19 Яким чином отримують експериментальні дані для виконання дисперсійного аналізу? Яку форму має таблиця вхідних даних?

3.5.20 Що таке адитивна модель? Запишіть її аналітичний вираз.

3.5.21 За якими формулами обчислюються середні та дисперсії по стовпцям та по всій таблиці вихідних даних при дисперсійному аналізі?

3.5.22 Який аналітичний вираз використовується для обчислення критерію Фішера?

3.5.23 В яких випадках нульова гіпотеза при дисперсійному аналізі відкидається?

3.5.24 Проаналізуйте зв'язок задач двохфакторного та однофакторного дисперсійного аналізу.

3.5.25 В чому полягають основні завдання статистичного вимірювання взаємозв'язків?

3.5.26 Який зв'язок називають функціональним, стохастичним, кореляційним?

3.5.27 Які форми кореляційної залежності ви знаєте?

3.5.28 З якою метою використовується кореляційний аналіз?

3.5.29 Як визначається індекс кореляції і що він показує? Як визначається коваріація і коефіцієнт кореляції? Як визначити вибіркове значення коефіцієнта кореляції?

3.5.30 Як визначається кореляційне відношення? Який його фізичний зміст?

3.5.31 Яким чином перевіряється гіпотеза про відсутність кореляційного зв'язку?

3.5.32 Що таке рангова кореляція? Які ранги називаються об'єднаними? Що таке ранжирування? Які є основні задачі статистичного зв'язку між ранжируваннями?

3.5.33 Як розраховуються рангові коефіцієнти кореляції Спірмена та Кендалла?

3.5.34 Як розраховується узагальнений коефіцієнт кореляції?

3.5.35 У чому виражається відмінність між кореляційним і регресійним аналізом?

3.5.36 Яка мета регресійного аналізу?

3.5.37 За допомогою якого методу оцінюють параметри регресійної моделі?

3.5.38 Яким чином відбувається перевірка значимості оцінок коефіцієнтів рівняння регресії? Як перевіряється адекватність рівняння регресії?

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

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

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

3.5.42 Проаналізуйте засоби пакету Matlab, що можуть бути використані для виконання статистичного аналізу результатів еволюційної оптимізації.

## **4 ЛАБОРАТОРНА РОБОТА №4 КОМБІНАТОРНА ОПТИМІЗАЦІЯ ЗА ДОПОМОГОЮ ЕВОЛЮЦІЙНИХ МЕТОДІВ**

### **4.1 Мета роботи**

4.1.1 Вивчити основні еволюційні оператори, що призначені для вирішення комбінаторних задач.

4.1.2 Навчитися розв'язувати задачі комбінаторної оптимізації за допомогою методів еволюційного пошуку.

### **4.2 Основні теоретичні відомості**

При використанні методів еволюційного пошуку для розв'язку задач комбінаторної оптимізації, як правило, застосовуються негомологічні числові хромосоми, тобто такі хромосоми, гени яких можуть приймати значення в заданому інтервалі. При цьому інтервал однаковий для всіх генів, але в хромосомі не може бути двох генів з однаковим значенням.

Комбінаторні задачі оперують із дискретними структурами або розміщенням об'єктів, незначні зміни яких часто викликають стрибкоподібну зміну показників якості (фітнесс-функції).

Традиційні оператори еволюційні оператори, що генерують нових нащадків, не можуть бути застосовані при використанні негомологічних хромосом, оскільки внаслідок виконання таких операторів генеруються нащадки, що містять однакові гени і тому не можуть бути інтерпретовані при розв'язку комбінаторної задачі. Тому для розв'язку задач комбінаторної оптимізації були розроблені спеціальні генетичні оператори, що не створюють неприпустимих рішень.

## 4.2.1 Оператори схрещування для негомологічних числових хромосом

### 4.2.1.1 Впорядковуючий оператор схрещування

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

При одноточечному схрещуванні в хромосому першого нащадка копіюється хромосома першого батька, а потім гени нащадка, що розташовані правіше точки схрещування, перевпорядковуються у послідовність, що відповідає другому батькові. При цьому другий батько переглядається від початку до кінця, зліва направо, і елементи, яких не вистачає у нащадку, додаються, починаючи від точки схрещування, один по одному.

Крок 1. Випадковим чином вибрати точку схрещування.

Крок 2. Скопіювати в хромосому першого нащадка сегмент хромосоми першого батька, що розташований лівіше точки схрещування.

Крок 3. Інші гени в нащадку копіюються із другого батька в упорядкованому вигляді зліва направо, крім елементів, які вже увійшли до нащадка.

Для створення другого нащадка застосовується аналогічний порядок дій.

При двохточечному впорядковуючому схрещуванні змінюється частина хромосоми, що перебуває між точками схрещування.

Також застосовується позиційно впорядковуюче схрещування, що виконується в наступній послідовності.

Крок 1. Випадковим чином вибрати деяку кількість позицій (генів) у першій батьківській хромосомі.

Крок 2. Скопіювати в хромосому першого нащадка обрані на попередньому кроці гени першої батьківської хромосоми. При цьому копіювання генів відбувається в ті ж позиції, на яких вони розташовувалися в першій батьківській хромосомі.

Крок 3. Інші гени в нащадку копіюються із другого батька в упорядкованому вигляді зліва направо, крім елементів, що вже увійшли в нащадка.

Для створення другого нащадка застосовується аналогічна послідовність дій за винятком того, що перший батько тепер вважається другим, а другий - першим.

В цьому випадку кожна хромосома представляє собою впорядковану послідовність чисел і застосування стандартних операторів схрещування неминуче привело б до утворення нелегальних рішень.

#### 4.2.1.2 Схрещування із частковим відображенням

Схрещування із частковим відображенням (Partially Mapped Crossover, РМХ-схрещування – Д. Голдберг, Р. Лінгл) полягає в прямому відображенні частини батьківської інформації в нащадка. РМХ-схрещування можна здійснити шляхом композиції вихідної підстановки (хромосоми) й декількох транспозицій.

Сам метод полягає в наступному. Нехай  $P_1$  і  $P_2$  – рішення (підстановки), що беруть участь у схрещуванні.

Крок 1. Випадковим чином вибирається ділянка, що повинна бути відображена у нових хромосомах. Для цього породжуються два числа  $m_1$  і  $m_2$ , що представляють собою ліву й праву границі відображуваної ділянки.

Крок 2. Обчислюється  $k = m_2 - m_1 + 1$  – розмір обраної ділянки.

Крок 3. Визначається транспозиція.

Позначимо через  $(a, b)$  транспозицію, що здійснює перестановку елементів  $a$  та  $b$ , а інші залишає на своїх місцях. Розглянемо наступні транспозиції:

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

Крок 4. Отримання 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  $C_2$  отримується за допомогою заміни  $P_1$  на  $P_2$  у попередньому виразі (крок 4) і при визначенні транспозицій  $t_i$ .

У двохточечному схрещуванні із частковим відображенням обмін відбувається між центральними частинами хромосом.

### 4.2.1.3 Циклове схрещування

Циклове схрещування (Cycle crossover) одержує нове рішення  $C_1$  шляхом композиції вихідної підстановки  $P_2$  і деякої підстановки  $(s_0, s_1, \dots, s_k)$ , що представляє собою цикл довжини  $k + 1$ , тобто  $C_1 = P_2 \circ (s_0, s_1, \dots, s_k)$ , при цьому

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

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

$$\dots\dots\dots$$

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

Тут  $s_0$  – випадково обране число від 1 до  $L - 1$  і  $s_{k+1} = P_2^{-1}(P_1(s_k)) = s_0$ . Рішення  $C_2$  виходить аналогічним шляхом за допомогою заміни  $P_2$  на  $P_1$  у попередніх виразах.

### 4.2.1.4 Жадібний оператор схрещування

Жадібний оператор схрещування (поглинаюче схрещування, greedy crossover) був запропонований в 1985 році Д. Грефенстеттом у співавторстві з іншими вченими для розв'язку 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 із множини наявних, 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 їхнього наступного сполучення.

Схема роботи жадібного оператора схрещування може змінюватися залежно від характеру розв'язуваних задач.

Послідовність виконання жадібного оператора схрещування наведена нижче.

Крок 1. Обчислити значення цільової функції у відібраних для схрещування хромосом:  $f(H_1)$  і  $f(H_2)$ ,  $H_1 = \{h_{11}, h_{12}, \dots, h_{1n}\}$ ,  $H_2 = \{h_{21}, h_{22}, \dots, h_{2n}\}$ ,  $i, h_{vi} = 1, n, v = \{1, 2\}$ .

Крок 2. Встановити:  $j = 1$ . Випадковим чином вибрати початкову точку для генерації хромосоми-нащадка:  $p_j = \text{gand}(1, n)$ .

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 відповідає значенню  $\text{temp}$ .

Крок 5. У випадку, якщо хромосома-нащадок складена повністю ( $j = n$ ), перейти на крок 8.

Крок 6. Виконати перевірку на передчасне замикання циклу:  $p_j = p_d$ ,  $d = 1, j - 1$ . У випадку передчасного замикання циклу збільшити шлях за рахунок включення гена, обраного випадковим образом із числа ще не включених.

Крок 7. Виконати перехід на крок 3.

Крок 8. Кінець.

Практика показує, що застосування жадібного оператора схрещування підвищує швидкість збіжності розв'язку, але в той же час це сприяє зменшенню розмаїтості популяції, що веде до її швидкого виродження, а також зниженню можливостей виходу з локальних оптимумів.

#### 4.2.1.5 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 одноточечного схрещування одержати дві нових хромосоми-нащадка.

Крок 3. Кожну половину хромосоми нащадка знову розділити навпіл і процес розрахунку продовжити за вихідною 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  $bC_i$  –  $i$ -а частина  $b$ -ої хромосоми розміром  $(1/2^j)$  після  $j$ -го розбиття. Процес продовжувати доти, поки не буде отримана задана кількість хромосом нащадків або метод дихотомії завершиться.



При одержанні нелегальних хромосом з повторюваними генами останні змінюються на відсутні гени із хромосом батьків.

Крок 4. Кінець.

#### 4.2.1.6 Оператор сегрегації

В результаті застосування оператора сегрегації  $R$  батьківських хромосом створюють одного нащадка. При застосуванні оператору сегрегації повинні бути відкинуті повторювані рішення або рішення, що містять однакові елементи. Даний оператор можна реалізувати різними способами залежно від методу вибору генів із хромосом.

Крок 1. Вибрати випадковим чином  $R$  батьківських хромосом для схрещування.

Крок 2. Встановити:  $k = 1$ .

Крок 3. Сформувати  $k$ -ий елемент нащадка, взявши  $k$ -ий елемент (один або кілька генів) з  $k$ -ої хромосоми-батька.

У випадку, якщо даний елемент вже є в нащадку, взяти наступний елемент батька. Якщо ж і цей елемент уже включений у нащадка, повторювати дану процедуру доти, поки не буде знайдений елемент в  $k$ -ій хромосомі-батьку, що ще не включена у нащадка.

Крок 4. Виконати:  $k = k + 1$ .

Крок 5. Якщо  $k \leq R$ , виконати перехід на крок 3.

Крок 6. Кінець.

#### 4.2.2 Оператори мутації

##### 4.2.2.1 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 нащадка як копію батьківської хромосоми  $H = \{h_1, h_2, \dots, h_L\}$ .

Крок 2. Вибрати два числа  $y_1$  і  $y_2$  випадковим чином із множини  $Y = \{0, 1, 2, \dots, L+1\}$ , причому  $y_1 \neq y_2$ .

Крок 3. Сформувати нову хромосому  $H$  шляхом обміну елементів, розташованих 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. Одноточечна мутація обміну.** У даному операторі, на відміну від попереднього, обмінюються місцями тільки сусідні гени, і точка мутації вибирається між двома генами.

Крок 1. Створити хромосому нащадка як копію батьківської хромосоми  $H = \{h_1, h_2, \dots, h_L\}$ .

Крок 2. Вибрати точку мутації у випадковим чином із множини  $Y = \{1, 2, \dots, L-1\}$ .

Крок 3. Сформувати нову хромосому  $H$  шляхом обміну елементів, розташованих на позиціях  $y$  і  $y+1$ .

Таким чином, після застосування одноточечної мутації обміну одержуємо хромосому  $H'$ :

$$H' = \{h_1, h_2, \dots, h_{y+1}, h_y, \dots, h_L\}.$$

**3. Мутація золотого перетину.** У даному операторі вибір точки мутації здійснюється на основі правила “золотого перетину”, 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 перетину хромосома  $H = \{h_1, h_2, \dots, h_D, h_{D+1}, \dots, h_L\}$  перетворюється у 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 обираються на основі чисел Фібоначчі:  $\varphi_0 = \varphi_1 = 1$ ,  $\varphi_k = \varphi_{k-1} + \varphi_{k-2}$  для  $k = 2, 3, \dots$ . Тобто для мутації вибираються гени, позиції яких відповідають числам у числовій послідовності 1, 2, 3, 5, 8, 13, 21....

Крок 1. Створити хромосому нащадка як копію батьківської хромосоми.

Крок 2. Вибрати гени для мутації на основі чисел Фібоначчі.

Крок 3. Виконати обмін числових значень між мутуючими генами по колу із зсувом праворуч або ліворуч.

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 представлений нижче.

Крок 1. У заданій популяції хромосом на основі селекції вибрати батьківську хромосому довжини  $L$  з найменшим значенням цільової функції.

Крок 2. У даній хромосомі визначити точку розриву для реалізації оператора мутації методу Фібоначчі. Вона відповідає третьому числу ряду Фібоначчі. Установити  $i = 0$ .

Крок 3. За правилами побудови стандартного оператора мутації виконати зазначений оператор мутації й одержати нову хромосому нащадка.

Крок 4. Обчислити значення цільової функції нащадка. Якщо знайдено глобальний оптимум, тоді перейти на крок 9.

Крок 5. Встановити  $i = i + 1$ .

Крок 6. В якості точки мутації вибрати  $(3 + i)$ -ий член ряду Фібоначчі.

Крок 7. Якщо  $(3 + i) \leq L$ , виконати перехід на крок 3.

Крок 8. В якості хромосоми нащадка вибрати хромосому з оптимальним значенням цільової функції.

Крок 9. Кінець.

### 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 крапку розриву.

Крок 2. За правилами одноточечної мутації одержати нову хромосому нащадка.

Крок 3. Кожну половину хромосоми нащадка знову розділити навпіл і процес розрахунку продовжувати за вихідною схемою доти, поки не буде отримана задана кількість хромосом нащадків або оператор мутації методу дихотомії завершиться.

Крок 4. Кінець.

6. В інженерних задачах, де необхідно виконувати аналіз моделей з великим числом елементів (порядку 10000), використовують *оператор мутації Монте-Карло*. Цей оператор мутації полягає в єдиному обміні двох випадково обраних генів або ансамблів з декількох генів.

#### 4.2.2.2 Інвертування

Оператор інвертування (інверсії) змінює порядок генів у ділянках хромосом. Його мета полягає в тому, щоб спробувати знайти порядок генів, що має кращий еволюційний потенціал. Перевпорядкування також значно розширює область пошуку. Мало того, що генетичний метод намагається знаходити гарні множини значень генів, він також одночасно пробує знаходити гарне впорядкування генів.

Найчастіше використовуються наступні види інвертування:

1. **Класичне інвертування**, при якому вибираються 2 точки, між якими відбувається перерозподіл хромосоми.

Крок 1. Створити хромосому нащадка як копію батьківської хромосоми  $H = \{h_1, h_2, \dots, h_L\}$ .

Крок 2. Два числа  $y_1$  і  $y_2$  вибираються випадковим чином із множини  $Y = \{0, 1, 2, \dots, L+1\}$ , причому  $y_1 < y_2$ .

Крок 3. Нова хромосома формується з  $H$  шляхом інвертування сегменту, що 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, який переміщується на випадково обирану позицію, зсуваючи інші елементи вправо по циклу. Іншою формою такого виду інвертування є випадковий вибір підстроки й переніс її на випадково обирану позицію із зсувом елементів, що залишилися.

Як правило, задається ймовірність інвертування  $P_{\text{in}}$ , конкретне значення якої залежить від розв'язуваної задачі й у загальному випадку перебуває 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 оператору інвертування методу Фібоначчі. Вона відповідає третьому числу ряду Фібоначчі. Встановити  $i = 0$ .

Крок 3. За правилами побудови оператора інвертування, інвертуючи праву частину від точки інвертування з використанням методу Фібоначчі, одержати нову хромосому-нащадка.

Крок 4. Обчислити значення цільової функції нащадка.

Крок 5. Встановити  $i = i + 1$ .

Крок 6. В якості точки мутації вибрати  $(3 + i)$ -ий член ряду Фібоначчі.

Крок 7. Якщо  $(3 + i) \leq L$ , виконати перехід на крок 3.

Крок 8. В якості хромосоми-нащадка вибрати хромосому з оптимальним значенням цільової функції.

Крок 9. Кінець.

**4. Інвертування на основі методу золотого перетину** аналогічно оператору інвертування на основі методу Фібоначчі. Для оператора інвертування методу золотого перетину перша точка розриву визначається на відстані цілої частини  $0,618L$  від будь-якого краю хромосоми. Потім елементи частини хромосоми, які розташовані між точкою розриву й правим кінцем хромосоми, інвертуються. Друга точка розриву в новій хромосомі визначається як найближче ціле з виразу  $(L - 0,618L) \cdot 0,618$ . Далі процес триває аналогічно до закінчення можливості розбиття хромосоми.

#### 4.2.2.3 Транслокація, вставка та делеція

Застосування операторів транслокації, вставки й делеції може призвести до зміни довжини хромосом, внаслідок чого ці оператори мутації використовуються дуже рідко, хоча в деяких класах комбінаторних задач вони можуть призвести 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 точки (одну або декілька) вставки.

Крок 2. Проаналізувати гени хромосоми для визначення альтернативних вставок.

Крок 3. Зробити пробну вставку генів (праворуч від точки вставки або між двома точками оператора вставки) і обчислити нове значення фітнес-функції хромосоми. Виконати дану операцію декілька (задане число) раз.

Крок 4. В якості результату оператора вставки прийняти хромосому, отриману на кроці 3, при якій досягається оптимальне значення фітнес-функції.

### 3. Делеція (видалення).

Крок 1. Створити хромосому нащадка як копію батьківської хромосоми  $H = \{h_1, h_2, \dots, h_L\}$ .

Крок 2. На обраній хромосомі визначити точку зрізу.

Крок 3. Видалити праву або ліву (від точку зрізу) ділянку із хромосоми.

Оператори вставки й видалення змінюють розмір хромосоми. Якщо це неприпустимо при рішенні конкретної задачі, ці два оператори необхідно застосовувати спільно.

## 4.3 Завдання до роботи

4.3.1 Ознайомитися з основними теоретичними відомостями, вивчити еволюційні оператори схрещування та мутації, що використовуються при розв'язуванні задач комбінаторної оптимізації.

4.3.2 Розробити за допомогою пакету Matlab програмне забезпечення для вирішення задачі комівояжера. Параметри еволюційного методу обрати з табл. 4.1 відповідно до варіанту.

Таблиця 4.1 – 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 обгрунтувати.

4.3.3 Виконати тестування розробленого програмного забезпечення за допомогою вирішення конкретних прикладів задачі комівояжера. Задачі (не менше трьох) для виконання тестування програми сформулювати самостійно. Вибір тестових задач обгрунтувати.

4.3.4 Порівняти одержані результати вирішення різних прикладів задачі комівояжера. Результати порівняльного аналізу звести до таблиці, попередньо розробивши систему критеріїв порівняння результатів вирішення задачі комівояжера.

4.3.5 Оформити звіт з роботи.

4.3.6 Відповісти на контрольні питання.

#### **4.4 Зміст звіту**

4.4.1 Тема та мета роботи.

4.4.2 Короткі теоретичні відомості.

4.4.3 Текст розробленого програмного забезпечення з коментарями, а також текст програми для тестування розробленого еволюційного методу.

4.4.4 Результати роботи програмного забезпечення (таблиця порівняльного аналізу з поясненнями).

4.4.5 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

#### **4.5 Контрольні питання**

4.5.1 Який вид кодування хромосом використовується в еволюційних методах при розв'язку комбінаторних задач?

4.5.2 В чому полягає суть оператору впорядковуючого схрещування? Які відмінності одноточечного від двохточечного впорядковуючого схрещування?

4.5.3 Проаналізуйте схрещування із частковим відображенням.

4.5.4 Яким чином відбувається схрещування із частковим відображенням?

4.5.5 Наведіть послідовність виконання жадібного схрещування. Які переваги та недоліки такого оператору?

4.5.6 За рахунок чого реалізується схрещування методом дихотомії?

4.5.7 Чим відрізняється оператор сегрегації від інших операторів схрещування? Яким чином реалізується такий оператор?

4.5.8 Дайте порівняльну характеристику операторів схрещування, яку використовуються методах при розв'язку задач комбінаторної оптимізації.

4.5.9 Порівняйте класичну із одноточечною мутацією обміну.

4.5.10 Яка особливість мутації золотого перетину?

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

4.5.12 В чому полягає нечітка мутація на основі методу дихотомії?

4.5.13 В яких випадках застосовується оператор мутації Монте-Карло?

4.5.14 Яка мета оператору інвертування?

4.5.15 Наведіть послідовність виконання класичного інвертування.

4.5.16 Проаналізуйте інвертування із зсувом.

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

4.5.18 Які особливості має інвертування на основі методу золотого перетину?

4.5.19 Порівняйте оператори мутації негомологічних числових хромосом.

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



4.5.21 Проаналізуйте складність еволюційних методів (теоретично).

4.5.22 Чим відрізняються задачі комбінаторної від задач неперервної оптимізації?

4.5.23 В чому полягає задача комівояжера. Які існують методи її вирішення?

4.5.24 Наведіть сутність задачі про Ханойську вежу та опишіть методи її вирішення.

4.5.25 Проаналізуйте задачу складання оптимального розкладу та методи її вирішення.

4.5.26 Метод гілок та меж. Задача про рюкзак.

4.5.27 Порівняйте транспортну задачу з іншими задачами комбінаторної оптимізації.

4.5.28 В чому полягає задача мінімізації перетинів в графі?

4.5.29 Задача розфарбовування графів.

4.5.30 Яким чином методи еволюційного пошуку можуть бути застосовані до розв'язку задачі відбору інформативних ознак?

## 5 ЛАБОРАТОРНА РОБОТА № 5 БІНАРНІ АСОЦІАТИВНІ ПРАВИЛА

### 5.1 Мета роботи

Вивчити основні поняття та методи видобування бінарних асоціативних правил.

### 5.2 Основні теоретичні відомості

Метою пошуку асоціативних правил (association rules) є знаходження закономірностей між пов'язаними подіями в базах даних.

Наведемо простий приклад асоціативного правила: покупець, що купує банку фарби, купить також пензлик для фарби з імовірністю 50%.

Уперше завдання пошуку асоціативних правил (association rule mining) було запропоновано для знаходження типових шаблонів покупок, виконаних у супермаркетах, тому іноді його ще називають **аналізом ринкового кошику** (market basket analysis).

**Ринковий кошик** – це набір товарів, придбаних покупцем у рамках однієї окремо взятої транзакції.

**Транзакції** є досить характерними операціями, ними, наприклад, можуть описуватися результати відвідувань різних магазинів. Таким чином, **транзакція** – це множина подій, які відбулися одночасно (наприклад, деякий покупець купив хліб, молоко, майонез, тобто кожна така транзакція являє собою набір товарів, куплених покупцем за один візит).

**Транзакційна база даних** (Transaction database) являє собою двовимірну таблицю, яка складається з номера транзакції (TID) і списку покупок, придбаних під час цієї транзакції.

Приклад транзакційної бази даних, що складається з купівельних транзакцій, наведено в таблиці 5.1.

Таблиця 5.1 – Транзакційна база даних

<b>TID</b>	<b>Придбані покупки</b>
100	хліб, молоко, печиво
200	Молоко, сметана
300	Молоко, хліб, сметана, печиво
400	Ковбаса, сметана
500	Хліб, молоко, печиво, сметана
600	Цукерки

Введемо позначення:

- транзакційна база даних =  $D$ ;
- хліб =  $a$ ;
- молоко =  $b$ ;
- печиво =  $c$ ;
- сметана =  $d$ ;
- ковбаса =  $e$ ;
- цукерки =  $f$ ;

Тоді табл. 5.1 може бути записана в такий спосіб (табл. 5.2).

Таблиця 5.2 – Транзакційна база даних  $D$ 

<b>TID</b>	<b>Придбані покупки</b>
100	$a, b, c$
200	$b, d$
300	$b, a, d, c$
400	$e, d$
500	$a, b, c, d$
600	$f$

Нехай задана транзакційна база даних  $D = \{T_1, T_2, \dots, T_{N_T}\}$ , у якій кожна транзакція  $T_j = \{t_{1j}, t_{2j}, \dots, t_{N_jj}\} \subseteq I$ ,  $j = 1, 2, \dots, N_T$  являє собою набір елементів з деякої множини  $I = \{\tau_1, \tau_2, \dots, \tau_{N_j}\}$ . При цьому  $i$ -й елемент  $j$ -ої транзакції  $T_j$  визначається в такий спосіб:

$$t_{ij} = \begin{cases} 1, & \text{якщо елемент } \tau_i \text{ присутній у транзакції } T_j; \\ 0, & \text{в іншому випадку,} \end{cases}$$

де  $N_T$  – кількість транзакцій у базі даних  $D$ ;  $N_I$  – кількість елементів множини  $I$ .

**Асоціативним правилом** називається імплікація  $X \rightarrow Y$ , де  $X \subset I$ ,  $Y \subset I$  і  $X \cap Y = \emptyset$ .

Тобто асоціативне правило має вигляд: "З події  $A$  випливає подія  $B$ " або «якщо  $A$  то  $B$ ». Наприклад, «якщо клієнт купив пиво, то він купить і чіпси».

У результаті аналізу й **пошуку асоціативних правил** встановлюються закономірності виду: "Якщо в транзакції зустрівся набір елементів  $X$ , то можна зробити висновок, що в цій же транзакції повинен з'явитися набір елементів  $Y$ ". Встановлення таких закономірностей дає можливість знаходити дуже прості й інтуїтивно зрозумілі правила.

**Завдання пошуку асоціативних правил** полягає в тому, щоб на основі наявної бази даних знайти закономірності між подіями (для прикладу з табл. 5.1 – необхідно знайти закономірності між покупками).

Основними характеристиками асоціативного правила є підтримка й вірогідність правила.

**Підтримка** (support). Існує поняття підтримки набору й підтримки правила.

**Підтримкою набору**  $\text{supp}(X)$  називають кількість або відсоток транзакцій, що містять певний набір даних.

Розглянемо набір товарів (табл. 5.1 і 5.2), що включає, наприклад, {Хліб, молоко, печиво}:  $X = \{a, b, c\}$ .

Цей набір товарів зустрічається в базі даних 3 рази (транзакції з TID = 100, 300, 500), тобто підтримка цього набору товарів рівна 3:

$$\text{supp}(X) = 3.$$

Для даного набору товарів підтримка, виражена у відсотковому відношенні, дорівнює:

$$\text{supp}(X) = \frac{3}{6} \cdot 100\% = 50\%.$$

Набір становить інтерес, якщо його підтримка вище певного користувачем мінімального значення (minsupport). Ці набори називають наборами, що часто зустрічаються (frequent).

Для прикладу 5.1 при мінімальному рівні підтримки, що дорівнює трьом, набір товарів  $X$  буде вважатися шаблоном, що часто зустрічається.

**Правило має підтримку**  $s$ , якщо  $s\%$  транзакцій із усього набору містять одночасно набори елементів  $A$  і  $B$  або, інакше кажучи, містять обидва товари.

**Правило**  $X \rightarrow Y$  має **підтримку**  $s$  (support), якщо  $s\%$  транзакцій з  $D$ , містять одночасно набори елементів  $X$  і  $Y$  ( $X \cup Y$ ):

$$\text{supp}(X \rightarrow Y) = \text{supp}(X \cup Y).$$

Тобто для обчислення підтримки правила  $X \rightarrow Y$  необхідно підрахувати кількість транзакцій в  $D$ , що містять обоє набору елементів  $X$  і  $Y$ ,  $X \cap Y = \emptyset$ .

Для прикладу 5.1: якщо молоко – це товар  $X$ , печиво – це товар  $Y$ . Підтримка правила "з покупки молока впливає покупка печива" дорівнює 3 (транзакції з TID = 100, 300, 500), або 50%.

**Вірогідність правила** показує, яка ймовірність того, що з  $X$  впливає  $Y$ . Правило  $X \rightarrow Y$  слухне з вірогідністю (confidence)  $c$ , якщо  $c\%$  транзакцій з  $D$ , що містять  $X$ , також містять  $Y$ :

$$\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}.$$

Для прикладу 5.1:

– число транзакцій, що містять молоко:  $\text{supp}(X) = 4$ ;

– число транзакцій, що містять молоко й печиво  $\text{supp}(X \cup Y) = 3$ .

Тому вірогідність правила рівна:

$$\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} = \frac{3}{4} \cdot 100\% = 75\%$$

Таким чином, вірогідність правила "з покупки молока впливає покупка печива" рівна 75%, тобто 75% транзакцій, що містять товар  $X$ , також містять товар  $Y$ .

Методи пошуку асоціативних правил призначені для знаходження всіх правил  $X \rightarrow Y$ , причому підтримка й вірогідність

цих правил повинні бути вище деяких наперед певних порогів, названих відповідно **мінімальною підтримкою** (minsupport) і **мінімальною вірогідністю** (minconfidence).

Якщо значення підтримки правила занадто велике, то в результаті роботи алгоритму будуть знайдені правила очевидні й добре відомі. Занадто низьке значення підтримки приведе до знаходження дуже великої кількості правил, які, можливо, будуть у більшій частині необгрунтованими, але не відомими й не очевидними для аналітика. Таким чином, необхідно визначити такий інтервал, який з однієї сторони забезпечить знаходження неочевидних правил, а з іншого – їх обгрунтованість.

Якщо рівень вірогідності занадто малий, то цінність правила викликає серйозні сумніви. Наприклад, правило з вірогідністю в 3% тільки умовно можна назвати правилом.

**Завдання пошуку асоціативних правил** розбивається на дві підзадачі:

1) пошук усіх наборів елементів, які задовольняють порогу minsupport. Такі набори елементів називаються, що часто зустрічаються;

2) генерація правил з наборів елементів, знайдених згідно п.1. с вірогідністю, що задовольняє порогу minconfidence.

Найбільш часто використовуваним методом пошуку асоціативних правил є **метод Apriori** (масштабований метод пошуку асоціативних правил), який використовує стратегію пошуку завширшки й здійснює пошук знизу-нагору.

Робота методу складається з декількох етапів, кожний з етапів складається з наступних кроків:

- формування кандидатів;
- підрахунок кандидатів.

**Формування кандидатів** (candidate generation) – етап, на якому сканується база даних і створюється множина  $i$ -елементних кандидатів ( $i$  – номер етапу). На цьому етапі підтримка кандидатів не розраховується.

**Підрахунок кандидатів** (candidate counting) – етап, на якому обчислюється підтримка кожного  $i$ -елементного кандидата. Тут же здійснюється відсікання кандидатів, підтримка яких менше мінімуму, встановленого користувачем (minsupport).  $i$ -елементні набори, що залишилися, будемо називати наборами, що **часто зустрічаються**.

Основна особливість методу – використання **властивості антимонотонності підтримки**: підтримка будь-якого набору елементів не може перевищувати мінімальної підтримки кожної з його підмножин. Наприклад, підтримка 3-елементного набору {Хліб, Молоко, Сметана} буде завжди менше або дорівнює підтримці 2-елементних наборів {Хліб, Сметана}, {Хліб, Молоко}, {Сметана, Молоко}, оскільки будь-яка транзакція, що містить {Хліб, Молоко, Сметана}, також повинна містити {Хліб, Сметана}, {Хліб, Молоко}, {Сметана, Молоко}, причому зворотне не вірно.

Завдяки цій властивості перебір не є «жадібним» і дозволяє обробляти більші масиви інформації за незначний час.

### 5.3 Порядок виконання роботи

5.3.1 Ознайомитися з основними теоретичними відомостями та рекомендованою літературою за темою роботи.

5.3.2 Розробити програмне забезпечення, що виконує пошук асоціативних правил за заданою базою транзакцій.

5.3.3 Виконати тестування розробленого програмного забезпечення. Для цього сформувані набір даних для обробки й аналізу, після цього здійснити обробку набору даних з метою виділення асоціативних правил.

5.3.4 Оформити звіт з роботи.

5.3.5 Відповісти на контрольні питання.

### 5.4 Зміст звіту

5.4.1 Тема та мета роботи.

5.4.2 Короткі теоретичні відомості.

5.4.3 Текст розробленого програмного забезпечення з коментарями.

5.4.4 Набір даних для обробки (якщо він великий – навести фрагмент).

5.4.5 Результати роботи програмного забезпечення (набір отриманих правил, інші характеристики).

5.4.6 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

## 5.5 Контрольні питання

5.5.1 Що таке асоціативне правило?

5.5.2 Для чого призначені асоціативні правила?

5.5.3 Дати визначення понять підтримки та достовірності правила.

5.5.4 Яке призначення алгоритмів пошуку асоціативних правил?

5.5.5 На які підзадачі розбивається задача знаходження асоціативних правил?

5.5.6 Які методи використовуються для знаходження асоціативних правил?

5.5.7 Яким чином обираються значення параметрів `minsupport` та `minconfidence`?

5.5.8 В чому полягає сутність масштабованого алгоритму пошуку асоціативних правил `Apriori`?

5.5.9 Яким чином перетворюються дані для можливості використання алгоритму `Apriori`?

5.5.10 Яка властивість використовується в алгоритмі `Apriori`? Для чого вона використовується?

5.5.11 Наведіть послідовність виконання алгоритму `Apriori`.

5.5.12 Опишіть функцію генерації кандидатів в алгоритмі `Apriori`.

5.5.13 Як відбувається підрахунок підтримки для кожного кандидату в алгоритмі `Apriori`? Для чого в цій процедурі використовують хеш-дерево?

5.5.14 Як здійснити добування правил з набору, що часто зустрічається?



## 6 ЛАБОРАТОРНА РОБОТА № 6 ЧИСЛОВІ АСОЦІАТИВНІ ПРАВИЛА

### 6.1 Мета роботи

Вивчити основні поняття та методи видобування числових асоціативних правил.

### 6.2 Основні теоретичні відомості

Часто ознаки  $\tau_a \in I$ ,  $a = 1, 2, \dots, N_I$  можуть приймати не тільки бінарні, але й чисельні значення з деякого діапазону значень  $\tau_a \in [\tau_{a\min}; \tau_{a\max}]$  або множини значень  $T(\tau_a) = \{\tau_{a1}, \tau_{a2}, \dots, \tau_{aN_{\tau_a}}\}$ . Тому актуальним є завдання виділення правил виду Якщо  $X \in [X_{\min}; X_{\max}]$ , то  $Y \in [Y_{\min}; Y_{\max}]$ .

Числовим асоціативним правилом (quantitative association rule) називається імплікація виду:

$$\langle X, v(X) \rangle \rightarrow \langle Y, v(Y) \rangle,$$

де  $v(X) \in T(X)$  й  $v(Y) \in T(Y)$  – значення змінних  $X$  і  $Y$ , відповідно, що належать множинам можливих значень  $T(X)$  і  $T(Y)$ .

Підтримка  $\text{supp}(X \rightarrow Y)$  числового асоціативного правила (АП)  $X \rightarrow Y$  визначається за формулою:

$$\text{supp}(X \rightarrow Y) = \frac{1}{|D|} \sum_{j=1}^{|D|} \prod_{i=1}^{N_X+N_Y} v_j(\tau_a), \quad \tau_a \in (X \cup Y),$$

де  $N_X$  й  $N_Y$  – кількість елементів  $\tau_a \in I$  у множинах  $X = \{\tau_1, \tau_2, \dots, \tau_{N_X}\}$  і  $Y = \{\tau_{N_X+1}, \tau_{N_X+2}, \dots, \tau_{N_X+N_Y}\}$ , відповідно;

$v_j(\tau_a)$  – значення  $a$ -ї ознаки  $\tau_a$  в  $j$ -й транзакції  $T_j$  бази даних  $D$ .

Вірогідність  $\text{conf}(X \rightarrow Y)$  чисельного АП  $X \rightarrow Y$  визначається аналогічно бінарним АП (п. 5.2). При цьому підтримка  $\text{supp}(X)$  множини  $X$  обчислюється відповідно до формули:

$$\text{supp}(X) = \frac{1}{|D|} \sum_{j=1}^{|D|} \prod_{i=1}^{N_X} v_j(\tau_a), \tau_a \in X.$$

Процес пошуку чисельних АП по заданих наборах даних  $D$  пов'язаний з необхідністю розбиття на інтервали (дискретизації) діапазонів можливих значень елементів  $\tau_a \in I$ , що входять у транзакції  $T_j$ ,  $j = 1, 2, \dots, N_T$ . У результаті такого розбиття кожна  $j$ -а транзакція  $T_j = (\text{tid}_j, \text{item}_j)$  представляється списком елементів  $\text{item}_j = \{t_{1j}, t_{2j}, \dots, t_{N_{\text{item}_j}}\} \subseteq I$ , у якому кожний  $i$ -й елемент  $t_{ij}$  представляється у вигляді:

$$t_{ij} = (\text{елемент } \tau_a \in I; \text{діапазон значень елемента } \tau_a),$$

при цьому множина  $I = \{\tau_1, \tau_2, \dots, \tau_{N_I}\}$  можливих значень, які можуть входити в список елементів  $\text{item}_j$  кожної транзакції  $T_j$ , містить елементи  $\tau_a$ :

$$\tau_a \in [\tau_{a\min}; \tau_{a\max}] = \bigcup_{c=1}^{N_{\text{разб}\tau_a}} [\tau_{a\min c}; \tau_{a\max c}], a = 1, 2, \dots, N_I,$$

де  $\tau_{a\min}$  й  $\tau_{a\max}$  – мінімальне й максимальне значення, які може приймати  $a$ -й елемент  $\tau_a$  множини  $I$ ;

$\tau_{a\min c}$  і  $\tau_{a\max c}$  – мінімальне й максимальне значення  $c$ -го інтервалу розбиття значень  $a$ -го елемента  $\tau_a$  множини  $I$ ;

$N_{\text{разб}\tau_a}$  – кількість інтервалів розбивки  $a$ -го елемента  $\tau_a$ .

Після дискретизації значень чисельних змінних виконується пошук асоціативних правил  $X \rightarrow Y$ . При цьому використовуються методи видобування асоціативних правил, що задовольняють наведеним вище умовам до бінарних правил виду  $X \rightarrow Y$ , але при такому пошуку кожний діапазон дискретизації кожної змінної

вважається окремим елементом, який може бути використаний при побудові асоціативного правила.

Однак необхідність дискретизації діапазонів значень змінних для витягу чисельних АП суттєво збільшує простір пошуку й вимоги до обчислювальних ресурсів ЕОМ. Крім того, у деяких випадках дискретизація приводить до невдалих розбиттів діапазону значень змінних, у результаті чого не забезпечується прийнятна точність прогнозування або класифікації за синтезованою базою АП.

### **6.3 Порядок виконання роботи**

6.3.1 Ознайомитися з основними теоретичними відомостями та рекомендованою літературою за темою роботи.

6.3.2 Розробити програмне забезпечення, що виконує пошук числових асоціативних правил за заданою базою транзакцій.

6.3.3 Виконати тестування розробленого програмного забезпечення. Для цього сформувавши набір даних для обробки й аналізу, після цього здійснити обробку набору даних з метою виділення числових асоціативних правил.

6.3.4 Оформити звіт з роботи.

6.3.5 Відповісти на контрольні питання.

### **6.4 Зміст звіту**

6.4.1 Тема та мета роботи.

6.4.2 Короткі теоретичні відомості.

6.4.3 Текст розробленого програмного забезпечення з коментарями.

6.4.4 Набір даних для обробки (якщо він великий – навести фрагмент).

6.4.5 Результати роботи програмного забезпечення (набір отриманих правил, інші характеристики).

6.4.6 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

## 6.5 Контрольні питання

6.5.1 Що таке числові асоціативні правила?

6.5.2 Поясніть поняття “узагальнене асоціативне правило”.

6.5.3 Що називається ієрархією елементів?

6.5.4 Які переваги дає введення додаткової інформації про групування елементів?

6.5.5 Поясніть, які проблеми можуть виникнути при безпосередньому застосуванні алгоритмів знаходження асоціативних правил.

6.5.6 В чому полягає сутність виявлення узагальнених асоціативних правил?

6.5.7 Яким чином визначають “цікаві” правила? В чому полягає актуальність такого процесу?

6.5.8 Дати визначення понять батьківського правила (предка) та найближчого батьківського правила.

6.5.9 Порівняйте поняття цікавого та частково цікавого правила.

6.5.10 Які проблеми усуває алгоритм обчислення узагальнених асоціативних правил?

6.5.11 З яких етапів складається процес обчислення узагальнених асоціативних правил?

6.5.12 Проаналізуйте базовий алгоритм пошуку множин, що зустрічаються часто.

6.5.13 Опишіть алгоритм генерації кандидатів.

6.5.14 Яким чином використовується хеш-дерево для підрахунку підтримки кандидатів? Як відбувається процес побудови такого дерева?

6.5.15 Виконайте порівняльний аналіз базового та покращеного алгоритмів пошуку множин, що зустрічаються часто.

6.5.16 За рахунок яких оптимізацій відбувається покращення базового алгоритму пошуку множин, що зустрічаються часто?

## 7 ЛАБОРАТОРНА РОБОТА № 7 ДЕРЕВА РОЗВ'ЯЗКІВ

### 7.1 Мета роботи

Вивчити основні методи побудови дерев розв'язків.

### 7.2 Основні теоретичні відомості

Дерева розв'язків являють собою спадну систему, засновану на підході “розділай і пануй”, основною метою якої є розподіл дерева на взаємно непересічні підмножини. Кожна підмножина являє собою підзадачу класифікації.

Дерево розв'язків описує процедуру прийняття рішення про приналежність певного екземпляра до того або іншого класу.

Дерево розв'язків є деревоподібною структурою, що складається із внутрішніх і зовнішніх вузлів, зв'язаних ребрами. Внутрішні вузли – модулі, що приймають рішення, – розраховують значення функції розв'язку, на підставі чого визначають дочірній вузол, який буде відвідано далі. Зовнішні вузли (іноді називаються кінцевими вузлами), навпаки, не мають дочірніх вузлів і описують або мітку класу, або значення, що характеризує вхідні дані. У загальному випадку, дерева розв'язків використовуються в такий спосіб. Спочатку передаються дані (як правило, це вектор значень вхідних змінних) на кореневий вузол дерева розв'язків. В залежності від отриманого значення функції рішення, використовуваної у внутрішньому вузлі, відбувається перехід до одному з дочірніх вузлів. Такі переходи тривають доти, поки не буде відвідано кінцевий вузол, що описує або мітку класу, або значення, зв'язане з вхідним вектором значень ознак.

### 7.3 Порядок виконання роботи

7.3.1 Ознайомитися з основними теоретичними відомостями та рекомендованою літературою за темою роботи.

7.3.2 Розробити програмне забезпечення, що виконує побудову дерева розв'язків за заданою вибіркою даних.

7.3.3 Виконати тестування розробленого програмного забезпечення. Для цього сформувавши набір даних для обробки й аналізу, після цього здійснити обробку набору даних з метою побудови дерева розв'язків.

7.3.4 Оформити звіт з роботи.

7.3.5 Відповісти на контрольні питання.

## **7.4 Зміст звіту**

7.4.1 Тема та мета роботи.

7.4.2 Короткі теоретичні відомості.

7.4.3 Текст розробленого програмного забезпечення з коментарями.

7.4.4 Набір даних для обробки (якщо він великий – навести фрагмент).

7.4.5 Результати роботи програмного забезпечення (побудоване дерево розв'язків, інші характеристики).

7.4.6 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

## **7.5 Контрольні питання**

7.5.1 Що таке дерево розв'язків (рішаючих правил)? Який спосіб подання правил в них використовується?

7.5.2 Дати означення основних понять, що відносяться до теорії дерев рішаючих правил: об'єкт, атрибут, мітка класу, вузол, лист, перевірка.

7.5.3 Навести основні класи задач, до яких можуть бути застосовані дерева рішаючих правил.

7.5.4 Яким чином відбувається побудова дерева рішаючих правил? Який метод використовується для цього?

7.5.5 В чому полягає процес навчання з учителем?

7.5.6 Порівняйте методи, що реалізують дерева рішальних правил: CART та C4.5.

7.5.7 Поясніть принцип роботи “жадібних” алгоритмів.

7.5.8 Перелічіть основні аспекти, яким приділяється увага при побудові дерев рішальних правил.

7.5.9 В чому полягає правило відбору ознаки для розбиття? Сформулюйте загальне правило для відбору атрибуту.

7.5.10 Виконайте порівняльний аналіз критеріїв оцінки якості розбиття множини на класи.

7.5.11 Що визначає правило зупину? Дайте порівняльну характеристику відомих критеріїв зупину побудови дерева рішальних правил?

7.5.12 Для чого використовується правило відсіку?

7.5.13 Що розуміють під точністю та помилкою розпізнавання для дерева рішальних правил?

7.5.14 Що необхідно зробити для добування правил з дерева рішальних правил?

7.5.15 Які вимоги до структури та значень даних висуває метод C4.5?

7.5.16 Проаналізуйте алгоритм побудови дерева рішальних правил за допомогою методу C4.5.

7.5.17 Яким чином визначається критерій вибору атрибуту в методі C4.5?

7.5.18 В якому випадку в процесі роботи методу C4.5 вузол помічається як лист? Що обирається в якості розв’язку листа?

7.5.19 Коли ентропія досягає свого максимуму (мінімуму) при використанні методу C4.5?

7.5.20 В яких випадках необхідно обрати поріг для порівняння значень атрибуту?

7.5.21 В чому полягає класифікація нових об’єктів? Звідки починається обхід дерева?

7.5.22 Порівняйте покращений критерій розбиття з класичним.

7.5.23 Яке евристичне правило використовується для зменшення ймовірності створення вузлів та листя, які містять незначну кількість об’єктів?

7.5.24 Проаналізуйте процедуру роботи з пропущеними даними.

7.5.25 Яким чином відбувається класифікація нових об'єктів у випадку відсутності значення певного атрибуту об'єкту, що класифікується?

7.5.26 Які переваги використання дерев рішальчих правил?

7.5.27 Який напрямок побудови дерева рішальчих правил використовується при використанні методу ID3?

7.5.28 Наведіть послідовність побудови дерева рішальчих правил за допомогою методу ID3.

7.5.29 В чому полягає вибір властивості на основі теорії інформації?



## **8 ЛАБОРАТОРНА РОБОТА №8 ЕВОЛЮЦІЙНИЙ СИНТЕЗ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ**

### **8.1 Мета роботи**

Навчитися будувати нейромережеві моделі на основі еволюційного підходу.

### **8.2 Порядок виконання роботи**

8.2.1 Ознайомитися з рекомендованою літературою за темою роботи.

8.2.2 Розробити програмне забезпечення, що за допомогою еволюційного пошуку реалізує один з етапів побудови нейромережевих моделей:

- відбір інформативних ознак;
- параметричний синтез;
- структурний синтез;
- оптимізація структури.

8.2.3 Виконати тестування розробленої програми, побудувавши нейромережеві моделі для різних вибірок даних.

8.2.4 Порівняти одержані результати синтезу нейронних мереж. Результати порівняльного аналізу звести до таблиці, попередньо розробивши відповідну систему критеріїв.

8.2.5 Оформити звіт з роботи.

8.2.6 Відповісти на контрольні питання.

### **8.3 Зміст звіту**

8.3.1 Тема та мета роботи.

8.3.2 Короткі теоретичні відомості.

8.3.3 Текст розробленого програмного забезпечення з коментарями.

8.3.4 Набір даних для обробки (якщо він великий – навести фрагмент).

8.3.5 Результати роботи програмного забезпечення (побудована нейромережева модель, інші характеристики, порівняльні таблиці).

8.3.6 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

## **8.4 Контрольні питання**

8.4.1. Виконайте постановку задачі синтезу нейромережевих моделей.

8.4.2. Які етапи виконуються при синтезі нейронних мереж?

8.4.3. В чому полягає задача відбору інформативних ознак?

8.4.4. Порівняйте структурний та параметричний синтез нейромереж.

8.4.5. З якою метою виконують оптимізацію побудованих нейромоделей?

8.4.6. Як виконується кодування хромосом для виділення найбільш значущої комбінації ознак?

8.4.7. Наведіть послідовність еволюційного пошуку відбору інформативних ознак.

8.4.8. Як відбувається керування параметрами еволюційного пошуку при відборі інформативних ознак?

8.4.9. У чому полягають особливості еволюційних методів відбору ознак з використанням апріорної інформації?

8.4.10. Проаналізуйте використання еволюційного пошуку для параметричного синтезу нейромереж.

8.4.11. У чому полягають переваги та недоліки еволюційного пошуку при параметричному синтезі нейромоделей?

8.4.12. Подання інформації, фітнес-функція та еволюційні оператори при структурному синтезі нейронних мереж.

8.4.13. Наведіть послідовність виконання структурно-параметричного синтезу нейромоделей на основі еволюційного пошуку.

8.4.14. Поясніть переваги та недоліки використання еволюційних методів для синтезу нейромережевих моделей.

8.4.15. Як виконується оптимізація побудованих нейромоделей при використанні еволюційного підходу?

8.4.16. Проаналізуйте критерії порівняння еволюційних методів синтезу нейронних мереж.

8.4.17. Які критерії якості навчальних множин використовують традиційно при формуванні навчальних вибірок?

8.4.18. Які критерії можна використовувати для формування навчальної множини у задачах розпізнавання образів за допомогою еволюційного пошуку?

8.4.19. Проаналізуйте критерії порівняння нейромоделей, що традиційно використовують при вирішенні задач структурно-параметричного синтезу нейромереж?

8.4.20. Які критерії можна використовувати для побудови нейромереж за допомогою еволюційного пошуку?

## 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. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посібник / С. О. Субботін. – Запоріжжя: ЗНТУ, 2008. – 341 с.
5. Емельянов В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик. – М. : Физматлит, 2003. – 432 с.
6. Курейчик В. М. Генетические алгоритмы: монография / В. М. Курейчик. – Таганрог : ТРТУ, 1998. – 242 с.
7. 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», 2008. – 468 с.
8. Рассел С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М.: Вильямс, 2006. – 1408 с.
9. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети / А. П. Ротштейн. – Винница: Універсум-Вінниця, 1999. – 320 с.
10. Руденко О. Г. Штучні нейронні мережі / О. Г. Руденко, Є. В. Бодянський. – Х.: Компанія СМІТ, 2006. – 404 с.
11. 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.
12. Скобцов Ю. А. Основы эволюционных вычислений / Ю. А. Скобцов. – Донецк: ДонНТУ, 2008. – 330 с.

13. 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.

14. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – СПб : Издательский дом "Вильямс", 2005. – 1104 с.

15. Эволюционные методы компьютерного моделирования: монография / А. Ф. Верлань, В. Д. Дмитриенко, Н. И. Корсунов, В. А. Шорох. – К: Наукова думка, 1992. – 256 с.

16. Cantu-Paz E. Efficient and accurate parallel genetic algorithms / E. Cantu-Paz. – Massachusetts: Kluwer Academic Publishers, 2001. – 162 p.

17. 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.

18. The practical handbook of genetic algorithms / ed. L. D. Chambers. – Florida : CRC Press, 2000. – Vol. I: Applications. – 520 p.

19. The practical handbook of genetic algorithms / ed. L. D. Chambers. – Florida : CRC Press, 2000. – Vol. II: New frontiers. – 421 p.

20. The practical handbook of genetic algorithms. / ed. L. D. Chambers. – Florida: CRC Press LLC, 2000. – Vol. III: Complex coding systems. – 659 p.

21. 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.

22. 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.

23. 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.

24. 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.

25. Субботин С. А. Сравнительный анализ методов эволюционного поиска / С. А. Субботин, А. А. Олейник // Штучний інтелект. – 2008. – № 2. – С. 44–49.