

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**  
Навчально-наукового інституту атомної та теплової енергетики  
Кафедра інженерії програмного забезпечення в енергетиці

**МЕТОДИЧНІ ВКАЗІВКИ**  
**ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №3**  
**з дисципліни**

**«МЕТОДОЛОГІЯ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНИХ КОМП'ЮТЕРНИХ**  
**ПРОГРАМ»**

**Тема: «Оптимізація функції із застосуванням генетичних алгоритмів»**

**КИЇВ 2023**

**Мета:** Написати програму оптимізації функції з використанням генетичного алгоритму

### **Теоретичні відомості.**

На практиці часом складно, а часом і неможливо, зафіксувати властивості функціональної залежності вихідних параметрів від вхідних величин, ще складніше провести аналітичний опис такої залежності. Ця обставина значно ускладнює застосування класичних методів оптимізації, оскільки більшість із них ґрунтуються на використанні апріорної інформації про характер поведінки цільової функції, а задача визначення приналежності функції до того або іншого класу порівняна за складністю з початковою.

У зв'язку із цим виникає необхідність побудови таких методів оптимізації, які були б здатні відшукувати рішення при практично повній відсутності припущень про характер досліджуваної функції при рішенні цілочисельних або комбінаторних оптимізаційних задач. Цим вимогам у значній мірі задовольняють еволюційні обчислення, які являють собою алгоритми пошуку, оптимізації або навчання, заснованих на деяких формалізованих принципах природного еволюційного процесу розвитку живих організмів.

Парадигму **генетичних алгоритмів** запропонував Джон Холланд, що опублікував на початку 60-х років її основні положення. А загальне визнання вона одержала після виходу у світ в 1975 році його класичної праці «Адаптація в природних і штучних системах». Генетичний алгоритм був отриманий у процесі узагальнення й імітації в штучних системах таких властивостей живої природи, як природний відбір, пристосовність до мінливих умов середовища, спадкування нащадками життєво-важливих властивостей від батьків і т.д.

Тому що алгоритми в процесі пошуку використовують деяке кодування множини параметрів замість самих параметрів, то він ефективно застосовується для рішення дискретних і комбінаторних задач оптимізації, визначених як на числових множинах, так і на кінцевих множинах довільної природи. Оскільки для роботи алгоритму в якості інформації про оптимізуєму функцію використовуються лише її значення в розглянутих точках простору пошуку, і не потрібно обчислень ні похідних, ні яких-небудь інших характеристик, то даний алгоритм застосуємий до широкого класу функцій, зокрема, не маючих аналітичного опису.

Використання набору початкових точок дозволяє застосовувати для їхнього формування різні способи, що залежать від специфіки розв'язуваної оптимізаційної задачі, у тому числі можливе задання такого набору безпосередньо людиною. Таким чином, основна відмінність генетичних алгоритмів полягає в представленні будь-якої альтернативи рішення у вигляді бітового рядка фіксованої довжини, маніпуляції з якою здійснюються під час відсутності всякого зв'язку з її смисловою інтерпретацією. Тобто в цьому випадку застосовується єдине універсальне представлення будь-якої задачі.

### **Ідея генетичного алгоритму.**

Уявимо собі штучний світ, населений множиною істот (особин), причому кожна істота – це деяке рішення нашої задачі. Будемо вважати особину тим більше пристосованою, чим краще відповідне рішення (чим більше значення цільової функції воно дає). Тоді задача максимізації цільової функції зводиться до пошуку найбільш пристосованої істоти. Звичайно, ми не можемо оселити в наш віртуальний світ всі істоти відразу, тому що їхнє число визначається початковою множиною альтернатив. Замість цього ми будемо розглядати багато поколінь, що змінюють один одного.

Тепер, якщо ми зуміємо ввести в дію природний відбір і генетичне спадкування, то отриманий світ буде підкорятися законам еволюції. Помітимо, що, відповідно до нашого визначення пристосованості, метою цієї штучної еволюції буде саме створення найкращих рішень.

Очевидно, еволюція – нескінченний процес, у ході якого пристосованість особин поступово підвищується. Примусово зупинивши цей процес через досить довгий час після його початку й вибравши найбільш пристосовану особину в поточному поколінні, одержимо якщо не абсолютно точну, то близьку до оптимальної відповідь. В цьому й полягає основна ідея генетичного алгоритму. Перейдемо тепер до точних визначень і опишемо роботу генетичного алгоритму більш детально.

### **Основний зміст генетичного пошуку.**

Генетичні алгоритми, будучи однією з парадигм еволюційних обчислень, являють собою алгоритми пошуку, побудовані на принципах, подібних із принципами природного відбору й генетики. Якщо говорити узагальнено, вони поєднують у собі принцип виживання найбільш перспективних особин-рішень і структурований випадково-детермінований обмін інформацією, у якому є присутнім елемент випадковості, що моделює природні процеси спадкування й мутації. Додатковою властивістю цих алгоритмів є невтручання людини в розвиваючийся процес пошуку. Людина може впливати на нього лише опосередковано, задаючи певні параметри.

Чим обумовлена популярність генетичних алгоритмів? Як було вже відзначено, ГА дозволяють знайти більш гарні або «раціональні» рішення NP-повних практичних оптимізаційних задач за менший час, ніж інші методи, звичайно застосовувані в цих випадках. Звичайно, термін «гарні» або «раціональні» не строгий у математичному змісті.

Під «раціональними» розуміються рішення, які задовольняють дослідника. Адже в більшості реальних задач немає необхідності знаходити саме глобальний оптимум. Частіше всього метою пошуку є рішення, що задовольняють певним обмеженням. Наприклад, час випробування устаткування не повинний перевищувати певної заданої величини. У цьому сенсі досить знайти саме «раціональне», тобто розумне рішення. Друга важлива причина росту популярності ГА полягає в стрімкому зростанні продуктивності сучасних комп'ютерів.

Переваги **генетичних алгоритмів** стають більш очевидними, якщо розглянути основні їхні відмінності від традиційних методів. Основних відмінностей п'ять:

1. Генетичні алгоритми працюють із кодами, у яких представлений набір параметрів, що прямо залежать від аргументів цільової функції.

2. Для пошуку генетичний алгоритм використовує декілька точок пошукового простору одночасно (розпаралелення), а не переходить від точки до точки, як це робиться в традиційних методах. Тобто ГА оперують одночасно з усією сукупністю припустимих рішень.

3. Генетичні алгоритми в процесі роботи не використовують ніякої додаткової інформації, що підвищує швидкість його роботи.

4. Генетичний алгоритм використовує як ймовірнісні правила для породження нових точок пошуку, так і детерміновані правила для переходу від одних точок до інших.

5. Генетичні алгоритми здійснюють пошук оптимального рішення за однією й тою ж стратегією, як для унімодальних, так і для багатоекстремальних функцій.

Генетичний алгоритм працює з **кодovими послідовностями** (КП) – кодами безвідносно їхньої значеннєвої інтерпретації. Тому сама КП і її структура описуються поняттям **генотип**, а його інтерпретація, з погляду розв'язуваної задачі, поняттям **фенотип**. Кожна КП представляє, по суті, точку простору пошуку. Екземпляр кодової послідовності називають хромосомою, особоною або індивідуумом.

У принципі, ГА не обмежені бінарним або цілочисельним поданням. Відомі й інші реалізації, побудовані винятково на векторах речовинних числах. Незважаючи на те, що для багатьох реальних завдань більше підходять рядки змінної довжини, у даний час структури фіксованої довжини найпоширеніші й вивчені.

На кожному кроці роботи генетичний алгоритм використовує декілька точок пошуку одночасно. Сукупність цих точок є набором кодових послідовностей (особин), які утворюють початкову множину рішень – **К** (популяцію). Кількість особин у популяції називають **розміром популяції**. На кожному кроці роботи генетичний алгоритм обновляє початкову множину **К** шляхом створення нових КП і знищення «безперспективних», не задовольняючих критерію цільової функції. Кожне відновлення інтерпретується як зміна поколінь і звичайно ідентифікується за заданим розміром.

У процесі роботи алгоритму генерація нових особин відбувається на основі моделювання процесу розмноження. При цьому, природно, що породжуючі особини називаються батьками, а породжені – нащадками. Батьківська пара, як правило, породжує пари нащадків. Безпосередня генерація нових рядків із двох обраних відбувається за рахунок роботи **оператора схрещування** (випадково-детермінованого обміну), що у процесі роботи алгоритму може застосовуватися не до всіх пар батьків.

Частина цих пар може переходити в популяцію наступного покоління безпосередньо. Наскільки часто буде виникати така ситуація, залежить від ймовірності застосування оператора схрещування, що є одним з параметрів генетичного алгоритму.

Моделювання процесу генерації нових точок пошуку здійснюється за рахунок роботи **оператора мутації**, що задається певною ймовірністю. Оскільки еволюційний процес біологічних видів супроводжується загибеллю останніх, то породження нащадків повинне супроводжуватися знищенням інших безперспективних особин. Вибір пар батьків з популяції для породження нащадків робить **оператор відбору**, а вибір особин для знищення - **оператор редукції**.

Характеристики генетичного алгоритму вибираються таким чином, щоб забезпечити малий час роботи, з одного боку, і пошук якомога кращого рішення, з іншого.

### Загальний вид генетичного алгоритму

Незважаючи на те, що теорія генетичних алгоритмів в інформатиці з'явилася порівняно недавно, цей напрямок було швидко підхоплено вченими. І до теперішнього часу існує досить велика кількість їхніх різновидів еволюційних алгоритмів. Проте їхню основу становить базова модель, представлена на рис. 1.



Рисунок 1. Базовий генетичний алгоритм

Різниця полягає лише в генетичних операціях, представлених усередині зазначеного алгоритму. Виключення становлять гібридні алгоритми, у яких до зазначених блоків можуть додаватися елементи класичних алгоритмів оптимізації. Далі розглянемо детально методологію побудови генетичного алгоритму.

### **Створення хромосом**

Хромосома представляє собою набір зістикованих значень змінних, представлений у двійковому коді. Розглянемо метод кодування:

Нехай  $X$  описується діапазоном значень  $[X_{\min}...X_{\max}]$ , з точністю  $E$ .

Спочатку обчислюється кількість значень  $N$ , яких необхідно закодувати, за формулою:

$$N = \frac{X_{\max} - X_{\min}}{E} . \quad (1)$$

Потім підбирається таке значення кількості бітів  $NB$ , що задовольняє умові:

$$2^{NB} \geq N. \quad (2)$$

Далі обчислюється крок дискретизації заданого діапазону:

$$D = \frac{X_{\max} - X_{\min}}{2^{NB}} \quad (3)$$

Знаючи крок дискретизації ми можемо закодувати значення  $X$  натуральним числом  $Nx$  за формулою:

$$Nx = \frac{X}{D} . \quad (4)$$

Після чого  $Nx$  кодується у двійковій системі числення.

У випадку мінімізації функції багатьох змінних двійкові коди кожної змінної зістиковуються в певній послідовності й зберігається інформація про кількість бітів, якою кодується кожна змінна.

### **Функція пристосованості.**

Під функцією пристосованості розуміється деяка цільова функція, екстремум якої й потрібно відшукати. Однак, у деяких літературних джерелах вказується, що ця функція є зворотною від цільової функції у випадках коли потрібно відшукати мінімум. Але це вносить певні незручності в перевірку рішення. Тому під функцією пристосованості краще розуміти саме цільову функцію. Якщо доводиться використовувати генетичний алгоритм для навчання нейронних, нейронечітких мереж або ж добірки коефіцієнтів ПД- регуляторів за зразком (навчальною вибіркою), то як функція пристосованості звичайно виступає класична функція суми квадратів помилки  $F$ :

$$F = \sum_i (Y_i - Y'_i)^2, \quad (5)$$

де  $i$  - номер елемента навчальної вибірки;  $Y$  - вектор значень вихідний змінної;  $Y'$  - вектор відповідних значень мережі (регулятора).

### Ініціалізація початкової популяції

На цьому етапі визначається кількість хромосом (особин) у популяції  $K$ . У різних джерелах літератури вказується, що воно залежить від кількості вхідних змінних і звичайно вибирається за правилом:

$$K \geq 2M, \quad (6)$$

де  $M$  – кількість вхідних змінних.

Далі задаються початкові значення хромосомам. Якщо відомі деякі наближення до мінімумів, то деяким хромосомам привласнюються ці значення. Всім іншим значення задаються випадково. Потім обчислюються відповідні їм значення функції пристосованості.

### Селекція

Суть операції селекції полягає у відборі пар хромосом для рекомбінації. І може здійснюватися трьома способами:

1. Ранжирування.
2. Випадковий вибір.
3. Виважено випадковий вибір.

*При ранжируванні* всі хромосоми впорядковуються за убутанням їхньої функції пристосованості й пари розбиваються послідовно за принципом «краща із кращої».

*При випадковому виборі* пари утворюються випадковим чином.

*При зважено випадковому виборі* кожній хромосомі привласнюється певний бал залежний від функції пристосованості. Потім всі хромосоми розташовуються послідовно на шкалі суми балів з діапазоном свого бального значення. Пари формуються шляхом випадкового вибору числа на шкалі. Отже чим більше бал у хромосомі, тим більше в неї шансів. Даний спосіб іноді ще називають «принципом рулетки».

### Рекомбінація (кросовер)

На даному етапі відбувається обмін генетичною інформацією. Рекомбінація є найбільш важливим генетичним оператором. Вона генерує нові хромосоми, поєднуючи генетичний матеріал двох батьківських. Існує декілька варіантів рекомбінації. Найбільш простим є одноточковий. У цьому варіанті просто беруться дві хромосоми, і розрізаються у випадково обраній точці. Результуючі хромосоми виходять із початку однієї й кінця іншої батьківських хромосом.

001100101110010 11000	-->	001100101110010 <b>11100</b>
<b>110101101101000</b>  11100		<b>110101101101000</b> 11000

### Мутація.

Мутація являє собою випадкову зміну хромосоми (звичайно простою зміною стану одного з бітів на протилежний). Даний оператор дозволяє більш швидко знаходити локальні екстремуми з одного боку, і дозволяє "перескочити" на інший локальний екстремум з іншого.

00110010111001 <b>0</b> 11000	-->	00110010111001 <b>1</b> 11000
-------------------------------	-----	-------------------------------

У деяких генетичних алгоритмах до мутації може додаватися операція інверсії.

Інверсія інвертує (змінює) порядок бітів у хромосомі шляхом циклічної перестановки (випадкова кількість разів).

001100101110010 <b>11000</b>	-->	<b>11000</b> 001100101110010
------------------------------	-----	------------------------------

### Умова зупинки.

Умова зупинки залежить від того, де ми застосовуємо генетичний алгоритм. Залежно від цього можна виділити наступні умови:

– Цільова функція досягла деякої заданої точності або значення. Цей випадок підходить для алгоритмів навчання із учителем нейронних або нейронечітких мереж, настроювання коефіцієнтів ПД-регуляторів або інших математичних апаратів, пов'язаних з навчанням за зразком (навчальній вибірці).

– Досягнуто задане число поколінь. Цей випадок застосовується в тих випадках, коли система явно обмежена часом і обчислювальними ресурсами на пошук оптимального рішення. Звичайно це потрібно в завданнях адаптації систем у динамічному режимі.

– Протягом заданого числа поколінь оптимальне значення цільової функції не змінюється. Такий варіант можливий у випадках, коли алгоритм досяг якого-небудь глобального або сильно вираженого локального екстремума. Це умова необхідно, коли алгоритм використовується як для навчання із учителем (вихід із зациклення у випадку неможливості відшукати рішення із заданою точністю), так і при пошуку екстремума, значення неможливо визначити заздалегідь.

Умови, при виконанні яких задача вирішується ефективно генетичними алгоритмами:

- великий простір пошуку, ландшафт якого є негладким (містить трохи екстремумів);
- складність формалізації оцінки якості рішення функцією ступеня придатності;
- багатокритеріальність пошуку;
- пошук прийняттого рішення за заданими критеріями на відміну від пошуку єдиного оптимального.

Основні завдання, які можуть ефективно вирішувати генетичні алгоритми, можна звести до наступних класів:

- навчання нейронних і нейронечітких мереж із учителем;
- адаптація нейронних і нейронечітких мереж у динамічному режимі;
- настроювання коефіцієнтів ПД-регуляторів;
- рішення класичних оптимізаційних завдань (завдання лінійного програмування, транспортні завдання й т.д.);
- рішення комбінаторних завдань.



**Завдання:**

Вибрати функцію й діапазон відповідно до варіанта. Побудувати графік функції. Написати програму знаходження максимуму й мінімуму функції на заданому діапазоні. Проаналізувати отримані результати.

**Варіанти:**

№	Функція
1.	$Y(x)=x*\sin(5*x), x=[-2...5]$
2.	$Y(x)=1/x*\sin(5*x), x=[-5...5]$
3.	$Y(x)=2^x*\sin(10x), x=[-3...3]$
4.	$Y(x)=x^{(1/2)}*\sin(10*x), x=[0...5]$
5.	$Y(x)=15*\sin(10*x)*\cos(3*x), x=[-3...3]$
6.	$Y(x)=5*\sin(10*x)*\sin(3*x), x=[0...4]$
7.	$Y(x)=\sin(10*x)*\sin(3*x)/(x^2), x=[0...4]$
8.	$Y(x)=5*\sin(10*x)*\sin(3*x)/(x^{(1/2)}), x=[1...7]$
9.	$Y(x)=5*\cos(10*x)*\sin(3*x)/(x^{(1/2)}), x=[0...5]$
10.	$Y(x)=-5*\cos(10*x)*\sin(3*x)/(x^{(1/2)})x=[0...10]$
11.	$Y(x)=-5*\cos(10*x)*\sin(3*x)/(x^x), x=[0...5]$
12.	$Y(x)=5*\sin(10*x)*\sin(3*x)/(x^x), x=[0...8]$
13.	$Y(x)=x^{\sin(10*x)}, x=[1...10]$
14.	$Y(x)=-x^{\cos(5*x)}, x=[0...10]$
15.	$Y(x)=x^{\cos(x^2)}, x=[0...10]$
16.	$Y(x)=\cos(x^2)/x, x=[0...5]$
17.	$Y(x)=10*\cos(x^2)/x^2, x=[0...4]$
18.	$Y(x)=(1/x)*\cos(x^2+1/x), x=[1...10]$
19.	$Y(x)=\sin(x)*(1/x)*\cos(x^2+1/x), x=[-2...2]$
20.	$Y(x)=5*\sin(x)*\cos(x^2+1/x)^2, x=[1...10]$
21.	$Y(x)=5*\sin(1/x)*\cos(x^2+1/x)^2, x=[1...4]$
22.	$Y(x)=5*\sin(1/x)*\cos(x^2)^3, x=[-4...4]$
23.	$Y(x)=(x^3)*\cos(x^2), x=[-2...2]$
24.	$Y(x)=(x^3)+\cos(15*x), x=[-2...2]$
25.	$Y(x)=(3^x)+\cos(15*x), x=[-1...2]$

Перед захистом звіт з лабораторної роботи надсилається на пошту [pis2020@ukr.net](mailto:pis2020@ukr.net) Тема листа «Група\_Прізвище\_ЛР №» наприклад ТІ-01\_Петренко І.І.\_ЛР\_№1. Назва файлу «Група\_Прізвище\_ЛР\_№3» наприклад ТІ-01\_Петренко І.І.\_ЛР\_№3

Кінцевий термін захисту лабораторної роботи **24.04.2023.**

Якщо захист лабораторної роботи відбудеться пізніше **вказаної дати**, то оцінка знижується на 50 %.

**Контрольні питання:**

1. Що таке генетичні алгоритми? Для чого їх можна застосовувати?
2. З яких етапів складається генетичний алгоритм?
3. Яким чином кодується особина в генетичному алгоритмі?
4. Що таке функція пристосованості?
5. Як реалізується селекція?
6. Як реалізується кросовер?
7. Як реалізується мутація?
8. Якою може бути умова зупинки роботи генетичного алгоритму?
9. При виконанні яких умов задача вирішується ефективногенетичними алгоритмами?

Професор кафедри ІПЗЕ  
доктор технічних наук, доцент

Андрій МУСІЄНКО