

*Леві Л. І., д.т.н., професор,  
Носко Н.В., магістрант,  
Полтавський національний технічний університет  
імені Юрія Кондратюка*

## **ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ ДЛЯ ОПТИМІЗАЦІЇ РОЗМІЩЕННЯ ТЕХНОЛОГІЧНИХ ОБ'ЄКТІВ**

*Стаття присвячена генетичним алгоритмам та застосуванню генетичних алгоритмів для оптимізації задач розміщення технологічних об'єктів. Як відомо, оптимізаційні завдання полягають в знаходженні мінімуму, або максимуму заданої функції. З їх допомогою можна знайти екстремальне значення цільової функції, але не завжди можна бути впевненим, що отримано значення глобального екстремуму. Знаходження локального екстремуму замість глобального називається передчасною збіжністю. Крім проблеми передчасної збіжності існує інша проблема - час процесу обчислень. Найчастіше точніші оптимізаційні методи працюють дуже довго.*

*Для вирішення поставлених проблем і проводиться пошук нових оптимізаційних алгоритмів. Запропоновані порівняно недавно - в 1975 році - Джоном Холландом генетичні алгоритми (ГА) засновані на принципах природного відбору Ч. Дарвіна. ГА відносяться до стохастичних методів. Ці алгоритми успішно застосовуються в різних областях діяльності економіка, фізика, технічні науки.*

**Ключові слова:** *генетичний алгоритм, застосування генетичних алгоритмів, збіжність, популяція*

### **Вступ**

Однією з серйозних проблем, що виникають при використанні генетичних алгоритмів, є передчасна збіжність. Не рекомендується використовувати

класичні ГА на маленьких популяціях, оскільки в популяціях з малим розміром гени поширюються надто швидко всі особини стають схожими (популяція вироджується) ще до того, як знайдено рішення задачі. Тобто новий генотип з найкращою оцінкою швидко витісняє менш хороші комбінації генів, виключаючи тим самим можливість отримання кращого рішення на їх базі. Можна запропонувати три основні шляхи усунення передчасної збіжності: збільшення розміру популяції, застосування генетичних операторів, що адаптуються самостійно, і створення «банку» замічних особин.

У першому випадку, збільшуючи розмір популяції, можна сподіватися на досягнення різноманіття генотипу в популяції. Але з іншого боку, збільшення числа особин веде до збільшення займаної пам'яті і часу роботи алгоритму. Даний підхід може бути ефективний або при паралельних обчисленнях, або при наявності досить простої цільової функції.

Другий, і найпоширеніший спосіб, - використання алгоритмів, що адаптуються самостійно - є більш ефективним. Самоадаптація полягає в застосуванні динамічних мутацій. Динамічні мутації в залежності від перехресних особин змінюють значення ймовірності мутації, тим самим стає можливим самоврядування алгоритму. В таких випадках вибирається малий розмір популяції.

У третьому підході створюється масив для збереження особин, генотип яких було втрачено під час формування нових поколінь, і часом ці особини.

### **Самоадаптуючі алгоритми. Неоднорідна мутація**

Одним із способів організації самоадаптуючого алгоритму являється застосування неоднорідною мутації. Якщо мутує ген  $y_i$ , то нове значення  $y'_i$  випадково генерується на відрізку  $[\min_i, \max_i]$

$$y'_i = \begin{cases} y_i + (\max_i - y_i) \left( 1 - T \left( 1 - \frac{t}{T} \right)^b \right) & \text{якщо } q = 0 \\ y_i - (y_i - \min_i) \left( 1 - T \left( 1 - \frac{t}{T} \right)^b \right) & \text{якщо } q = 1 \end{cases}$$

де  $q$  випадковим чином приймає значення 0 або 1;  $r$  – випадкове число, що приймає значення з діапазону  $[0;1]$ ;  $t$  - номер покоління;  $T$ - максимальне число поколінь;  $b$  - деякий параметр, обумовлений природою завдання;  $min_i$  та  $max_i$  - верхня і нижня межі для величини  $y_i$ .

### Критерій відстані

Як відстань в разі подання особини у вигляді бінарної рядки (код Грея) береться відстань Хеммінга. Перьякс і інші запропонували інший критерій відстані, заснований на критерії мінімальної площі. І замість використання абсолютної відстані вони взяли відносну відстань, яку в розрахунках кордонів для кожної змінної індивідуально. Нехай  $A = (a_1, \dots, a_i, \dots, a_n)$  і  $B = (b_1, \dots, b_i, \dots, b_n)$  - два батька,  $min_i$  та  $max_i$  - верхня і нижня межі для  $i$ -ї змінної:

$$\text{dist}(A, B) = \frac{1}{n} \sum_{i=1}^n \left| \frac{a_i - b_i}{\max_i - \min_i} \right|^d,$$

де  $d$  - параметр, який використовується для обчислення відстані.

Нехай  $M(A, B) = 1 - \text{dist}(A, B)$  - ймовірність мутації, пов'язана з відстанню. Тоді  $\text{dist}(A, B) \in [0, 1]$  та  $M(A, B) \in [0, 1]$ .  $M(A, B)$  досягається максимуму при  $\text{dist}(A, B) = 0$ ; при ідентичних батьків  $A$  та  $B$  отримуємо  $M(A, B) = 1$ . Кросинговер двох ідентичних індивідуумів при  $M(A, B) = 1$ , викликає обов'язкову мутацію для кожної змінної нащадка. Очевидно, що така мутація є дуже великою, і тому введемо множник  $M_m$ , що дозволяє скорегувати значення  $M(A, B)$ . Потім визначають ефективну ймовірність мутації  $P_r : P_r(A, B) = M(A, B)M_m = (1 - \text{dist}(A, B))M_m$ , де  $M_m$  - фіксований параметр. В даному класі задач оптимальними є наступні значення параметрів:  $d = 0,2$  і  $M_m = 0,9$ .

### Параметри обчислень

Для обчислень Перьякс і інших використовували класичні тестові функції Де Йонга, широко застосовуються для аналізу ефективності ГА. Використовувалися наступні параметри:

- розмір популяції  $N = 20$ ;
- ймовірність кросинговеру  $P_c = 0,8$ ;
- ймовірність мутації відтворення  $P_m^{rep} = 0,01$ ;

- максимальна ймовірність мутації  $M_m = 0,9$ .

Для кожної функції алгоритм запускався 100 разів. Процес роботи генетичного алгоритму представлений на Рис.1.

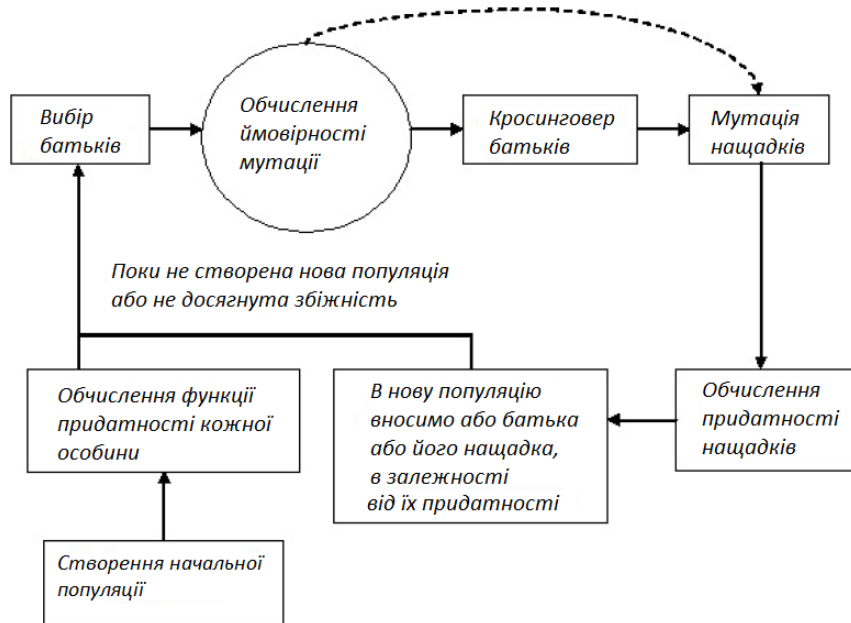


Рис.1 ГА в роботах Перьякса

## Переваги та недоліки генетичних алгоритмів

### Генетичні алгоритми мають ряд переваг

- не вимагають ніякої інформації про поведінку функції (наприклад, диференціальність і безперервність);

- стійкі до потрапляння в локальні оптимуми;
- придатні для вирішення багатомасштабних проблем оптимізації;
- можуть бути використані для широкого класу задач;
- прості в реалізації;
- можуть бути використані в задачах з середовищем, що змінюється.

### У той же час існує ряд труднощів в практичному використанні ГА

- за допомогою проблематично знайти точний глобальний оптимум;
- неефективно застосовувати в разі оптимізації функції, що потребує великого часу на обчислення;

- непросто змоделювати для знаходження всіх рішень задачі;

- не для всіх задач вдається знайти оптимальне кодування параметрів;
- в багатоекстремальних завданнях ГА стикається з безліччю множин: на графіку функції Растрігіна від однієї змінної (Рис.2) видно, що істинний мінімум досягається при  $x=0$ ;

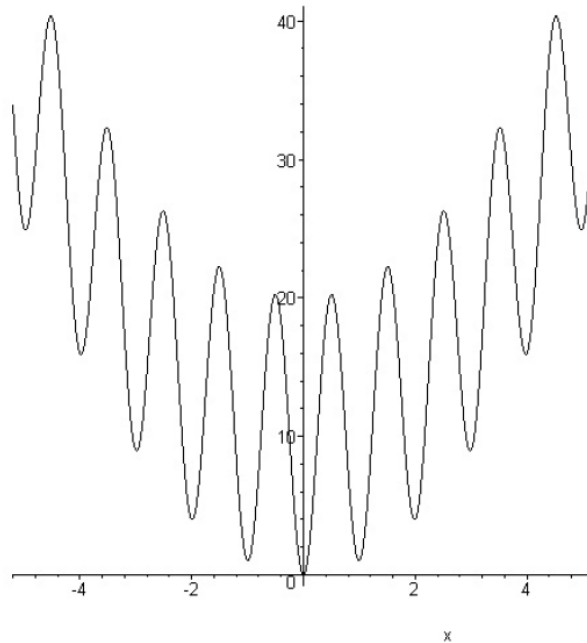


Рис.2 Функція Растрігіна

- ГА важко застосувати для ізолюваних функцій. Ізолюваність («Пошук голки в копиці сіна») - проблема для будь-якого методу оптимізації, оскільки функція не надає ніякої інформації, підказує, в якій області шукати максимум (Рис. 3).

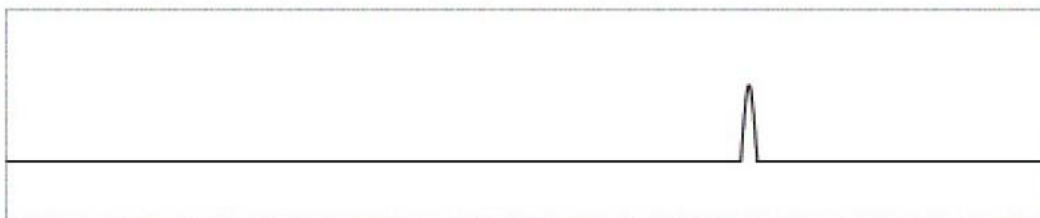


Рис.3. Ізолювана функція

- додатковий шум (noise) розкидає значення пристосованості шим, тому часто навіть хороші шими малого порядку не проходять відбір, що уповільнює пошук рішення ГА (Рис.4).

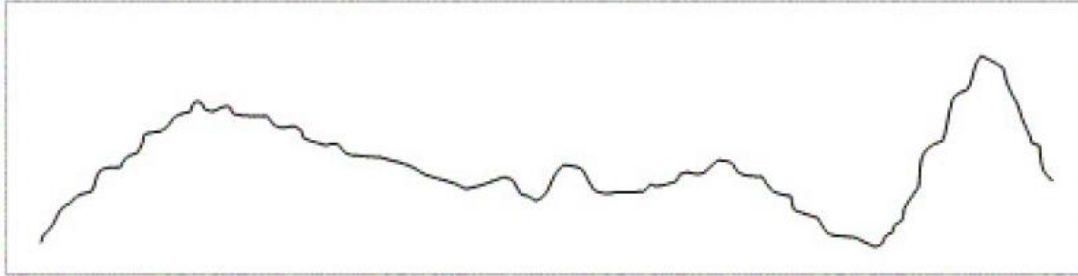


Рис.4 Зашумлена функція

- для деяких функцій шими малого порядку відводять популяцію до локального оптимуму. Таку характеристику функції називають фальшивою (deception). Наприклад, нехай рядок складається з 10 чотирьохбітних рядків. Нехай  $u_i$  дорівнює кількості одиниць в  $i$ -му підрядку. Задамо функцію  $g(u)$  наступною таблицею:

$u$	0	1	2	3	4
$g(u)$	3	2	1	0	4

і нехай функція пристосованості дорівнює сумі  $g(u_i)$  по всім  $i=1..10$ .

$$f = \sum_{i=1}^{10} g(u_i)$$

Локальний максимум досягається при всіх бітах, рівних 0, глобальний - при всіх 1. У більшості випадків при додаванні одиниці в підрядок пристосованість особини буде падати (за виключенням випадку, коли всі інші біти підрядка вже рівні 1). При заміні 1 на 0 вона буде рости. Тому з великою ймовірністю популяція зійдеться до вирішення, при якому більшість підрядків будуть складатися з усіх нулів, і лише деякі з усіх одиниць. Однак це не буде глобальним максимумом. З цього рішення потрапити в глобальний максимум, тобто замінити всі нулі одиницями для ГА буде важко. Розглянутий приклад сходження до помилкового оптиму демонструє явище епістазу.

На даний момент ведуться пошуки генетичних стратегій, здатних усунути ці недоліки. Усунення негативного впливу епістазу і усунення помилкового оптимуму із застосуванням різного роду мутацій, дозволяють вибрати популяцію з глухого кута. У статтях А. Ісаєва йдеться про можливість пошуку всіх рішень. Він пише: «. . . існує, принаймі, три класи завдань, які можуть бути вирішені представленим алгоритмом:

- *задача швидкої локалізації одного оптимального значення,*
  - *задача визначення декількох (або всіх) глобальних екстремумів,*
  - *задача опису ландшафту досліджуваної функції, яка може супроводжуватись виділенням не тільки глобальних, а й локальних максимумів*
- ».

Простір пошуку вирішення другого завдання досягається за рахунок деякого поєднання параметрів і досить великої чисельності популяції. При цьому ГА зможе виділити кілька (або навіть всі) глобальні екстремуми. Для виділення декількох глобальних максимумів, як пишуть дослідники, найбільше підходить використання таких параметрів, як, аутбридінг в поєднанні з інбридінгом, або елітний відбір або відбір з витісненням (останній більш надійний) для відбору особин в нову популяцію. Максимальна ефективність алгоритму досягається в поєднанні аутбридінга на початку пошуку для отримання максимально широкого «дослідження» і інбридінгу в завершенні пошуку з метою уточнення рішення в локальних групах.

### **Висновки**

Генетичні алгоритми – потужна і перспективна технологія, яка у своїй роботі використовує імітацію еволюції. Генетичні алгоритми як ефективні метаевристичні алгоритми наближеного розв'язку і оптимізації багатокритерійних задач відіграють вагомую роль у сучасній науці.

Постійно створюються нові програмні рішення, що реалізують застосування ГА у різноманітних сферах діяльності людини. Ці системи полегшують обчислення, уникаючи повного перебору варіантів рішень, значно скорочують часові та машинні витрати на розв'язання поставлених задач.

Генетичні алгоритми надають можливість швидкої генерації прийнятних розв'язків задач, які неможливо розв'язати іншими традиційними аналітичними методами.

*Джерела*

1. Foster, I. (1995). *Designing and Building Parallel Programs. 2.7 Case Study: Floorplan Optimization*. Addison-Wesley, 1995.
2. Курейчик В.М., Родзин С.И. (2003). *Эволюционные вычисления: генетическое и эволюционное программирование. "Новости Искусственного Интеллекта", №5(59), РАИИ, Москва, 2003, стр. 13 – 20.*
3. Гладков Л.А., Курейчик В.В., Курейчик В.М. *Генетические алгоритмы [Текст]/ Под ред. В.М. Курейчика. — 2-е изд., испр. и доп. — М.: ФИЗМАТЛИТ, 2006. — 320 с. — ISBN 5-9221-0510-8.*
4. Fogel D. B. *Evolutionary computation: towards a new philosophy of machine intelligence [Текст]/D. B. Fogel. — Piscataway: IEEE Press, 2000. — ISBN 0-7803-3481-7*