

## ГЕНЕТИЧНІ АЛГОРИТМИ ТУРНІРНОГО ВИТИСНЕННЯ З ГАУСОВОЮ МУТАЦІЄЮ

**Анотація.** Для розв'язання задач багатоекстремальної оптимізації запропоновано новий генетичний алгоритм утворення ніш — генетичний алгоритм турнірного витиснення з гаусовою мутацією. Проведено порівняльний аналіз його з іншими алгоритмами витиснення та з паралельним алгоритмом сходження на вершини, який показав переваги розробленого алгоритму у багатьох випадках. Введено критерій оцінювання ступеня розпорошеності елементів популяції та показано, що обчислення цього критерію є доцільним для оцінювання якості роботи алгоритмів пошуку глобальних та локальних максимумів.

**Ключові слова:** задача багатоекстремальної оптимізації, генетичні алгоритми утворення ніш, алгоритми витиснення, паралельний алгоритм сходження на вершини, частка хибних піків.

### ВСТУП

Задача багатоекстремальної (мультимодальної) оптимізації полягає у знаходженні кількох екстремумів багатоекстремальної функції. Виділяють такі типи цих задач: знайти всі глобальні екстремуми; знайти всі глобальні та всі локальні екстремуми; знайти принаймні  $k$  глобальних екстремумів; знайти  $k$  найкращих екстремумів тощо.

Для розв'язання задач багатоекстремальної оптимізації розробляють спеціальні алгоритми. Серед них особливою популярністю користуються генетичні алгоритми утворення ніш, в основу яких покладено явище видоутворення та спеціалізації в природних екосистемах. Ці алгоритми сприяють формуванню стабільних субпопуляцій (видів) у просторі пошуку в такий спосіб, що кожна з субпопуляцій формується навколо одного з шуканих екстремумів. На сьогодні розроблено кілька десятків різних методів утворення ніш та відомо кілька варіантів класифікації цих методів [1, 2].

Зазначимо, що генетичні алгоритми знайшли також застосування для розв'язання багатьох задач дискретного програмування (наприклад, [3–5]). Хоча вони за швидкодією поступаються іншим відомим методам дискретної оптимізації, але продовжують активно розвиватися і ця ситуація може змінитися.

У цій роботі розглянуто задачі багатоекстремальної оптимізації, зокрема: задача знаходження всіх глобальних максимумів; задача пошуку всіх глобальних та всіх локальних максимумів. Для їхнього розв'язання запропоновано новий генетичний алгоритм утворення ніш, а саме генетичний алгоритм турнірного витиснення з гаусовою мутацією (збуренням), який класифікуватимемо як метод витиснення.

Під методами витиснення (crowding) розуміють генетичні алгоритми зі стійким типом репродукції, які у разі додавання нового елемента в популяцію виштовхують із популяції схожий на нього елемент [1, 6]. Найвідомішими серед цих методів є алгоритми стандартного, детермінованого, ймовірнісного, багатонішевого витиснення та їхні модифікації. Особливостями пропонованого алгоритму турнірного витиснення з гаусовою мутацією є представлення розв'язків дійсними змінними та використання для породження нащадків тільки оператора гаусової мутації (оператор кросинговеру — обміну різних розв'язків своїми частинами — не застосовується).

Для аналізу ефективності запропонованого алгоритму застосовано експериментальний підхід, суть якого полягає в оцінюванні даних, зібраних у результаті здійснення великої кількості незалежних прогонів алгоритму на заданому наборі тестових задач [6]. Розроблений алгоритм порівнюється з паралельним алгоритмом сходження на вершини, основна ідея якого є такою: кожна особина популяції намагається дістатися найближчого атрактора шляхом виконання невеличких кроків у різних напрямках. Цей алгоритм часто слугує еталоном для порівняння з іншими алгоритмами. Також запропонований алгоритм турнірного витиснення з гаусовою мутацією порівнюється з алгоритмами детермінованого та ймовірнісного витиснення.

Розглянемо оптимізаційну задачу такого вигляду: знайти

$$\max f(x), x \in X \subseteq R^n, \quad (1)$$

де  $R^n$  — множина  $n$ -вимірних дійсних векторів. Можна показати, що розв'язування довільної (в т.ч. багатоекстремальної) оптимізаційної задачі вигляду (1) зводиться до розв'язування генетичним алгоритмом оптимізаційної задачі такого вигляду: знайти

$$\max F(s), s \in S, \quad (2)$$

де  $S$  — множина закодованих значень елементів  $x \in X$ . При цьому будь-який допустимий розв'язок задачі (2) називається особиною популяції. Особини кодуються у вигляді хромосом та за аналогією з двоїстістю генотипу та фенотипу живих організмів говорять про генотип ( $s$ ) та фенотип ( $x$ ) особин популяції. Цільова функція  $F$  називається функцією пристосованості особин популяції, а значення функції пристосованості в точці  $s$  називають коефіцієнтом пристосованості відповідної особини.

#### АЛГОРИТМ ТУРНІРНОГО ВИТИСНЕННЯ З ГАУСОВОЮ МУТАЦІЄЮ

У пропонованому в цій роботі генетичному алгоритмі витиснення застосовується кодування в дійсних числах: у хромосомі  $\vec{X} = (x_1, x_2, \dots, x_n) \in R^n$  кожна дійсна змінна  $x_i \in R$  представляє відповідну цільову змінну. Використовується стійкий тип репродукції, причому до породження нащадків допускаються всі особини популяції (відбір батьків не проводиться), а для відбору особин у нове покоління застосовується детермінований турнірний відбір: проводяться турніри між кожним батьком та всіма його нащадками, найкраща особина переходить у наступне покоління. Іншими словами, в наступне покоління проходить найбільш пристосована особина — батько або його найкращий нащадок.

Для породження нащадків використовується гаусова мутація, яка змінює значення кожного гена кожної хромосоми на нормально розподілену випадкову величину

$$x_i = x_i + N_i(0, \sigma), \quad (3)$$

де  $N_i(0, \sigma)$  — стандартна нормально розподілена випадкова величина (очікування 0, стандартне відхилення  $\sigma$ ). Якщо після виконання оператора мутації результівне значення гена виходить за допустимі межі, його обрізують до допустимого. Кросингвер для породження нащадків не використовується.

Наведемо загальну схему алгоритму турнірного витиснення з гаусовою мутацією. Вона містить такі кроки.

1. Генерація початкової популяції за рівномірним розподілом.
2. Обчислення коефіцієнта пристосованості всіх особин популяції.

3. Породження  $L$  нащадків кожною особиною шляхом застосування оператора гаусової мутації (3) до кожного гена особи.

4. Обчислення коефіцієнта пристосованості отриманих нащадків.

5. Формування нової популяції за допомогою турнірного відбору; група для кожного турніру формується з батька та усіх його нащадків.

6. Перехід на крок 3 або завершення роботи алгоритму.

Наведена схема потребує уточнення кроку 3: визначення параметра кроку мутації  $\sigma$ . Надалі залежно від способу його визначення розглядатимемо три варіанти алгоритму турнірного витиснення з гаусовою мутацією.

**S1:** алгоритм з обчисленням  $\sigma$  на основі аналізу початкової популяції. У цьому випадку значення  $\sigma$  визначається як  $1/k$  від середньої відстані між особинами початкової популяції,  $k > 1$  — ціле число.

**S2:** алгоритм з періодичним переобчисленням  $\sigma$ . У ньому значення  $\sigma$  визначається як  $1/k$  від середньої відстані між особинами попередньої популяції,  $k > 1$  — ціле число. Оскільки цей варіант є досить ресурсоємним, то переобчислення значення  $\sigma$  можна проводити кожні  $m$  ітерацій.

**S.evol:** алгоритм із самоадаптивним керуванням параметром  $\sigma$ . Ідею цієї мутації взято з еволюційних стратегій (некорельована мутація з  $n$  розмірами кроку мутації або мутація-2) [6]. У цьому випадку в хромосомі разом із значеннями цільових змінних  $x_i$  зберігаються значення стандартних відхилень  $\sigma_i$ , які також еволюціонують. Іншими словами, особина популяції представляється парою  $(\bar{X}, \bar{\Sigma})$ , де  $\bar{X} = (x_1, x_2, \dots, x_n) \in R^n$  — вектор параметрів об'єкта,  $\bar{\Sigma} = (\sigma_1, \sigma_2, \dots, \sigma_n) \in R_+^n$  — вектор параметрів стратегії. Механізм дії оператора мутації задається у такий спосіб:

$$\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1)),$$

$$x'_i = x_i + \sigma'_i \cdot N_i(0, 1).$$

Тут  $N(0, 1)$  — стандартна нормально розподілена випадкова величина, генерується одна для кожного нащадка;  $N_i(0, 1)$  — стандартна нормально розподілена випадкова величина, генерується для кожної змінної кожного нащадка;  $\tau'$  та  $\tau$  — відповідно спільний та покоординатний коефіцієнти навчання.

#### АЛГОРИТМИ, ОБРАНІ ДЛЯ ПОРІВНЯННЯ

Алгоритм детермінованого витиснення (deterministic crowding) [7] є одним з найкращих у класі алгоритмів витиснення [8–10] та часто використовується для порівняння з іншими алгоритмами утворення ніш.

Алгоритм ймовірнісного витиснення (probabilistic crowding) є модифікацією алгоритму детермінованого витиснення [11]. Його суть полягає у запобіганні втраті видів, сформованих навколо нижчих піків.

Паралельний алгоритм сходження на вершини (parallel hill-climbing) будемо розглядати у варіанті, поданому в [7].

Вибір параметрів є однією з найскладніших проблем практичного застосування генетичних алгоритмів. Від їхнього вдалого вибору залежить якість знайдених алгоритмом розв'язків та швидкість роботи алгоритму. Зазвичай, частину параметрів обирають, керуючись певними рекомендаціями або власним досвідом, а частину — на основі порівняльного аналізу результатів роботи алгоритму за різних значень параметрів (для цього збирають статистику, ґрунтуючись на численних незалежних прогонах алгоритму).

**Параметри алгоритмів.** Параметри алгоритму турнірного витиснення з гаусовою мутацією є такими.

$L$  — кількість нащадків особини, породжених у результаті застосування оператора мутації; підбирається експериментально (розглянуто значення 1; 2; 3).

$\sigma$  — визначений вище параметр мутації. Значення  $1/k$  підбираються експериментально (розглянуто значення  $1/2; 1/4; 1/8; 1/16$ ); при цьому в алгоритмі S2 значення  $\sigma$  переобчислюються кожні 60 ітерацій. Початкові значення  $\sigma_i$  в алгоритмі S.evol обчислюються як  $N_i(0, \sigma)$ , де  $\sigma = 1/k$  від середньої відстані між особинами початкової популяції (розглянуто значення  $1/2; 1/4; 1/8; 1/16$ ). Значення коефіцієнтів навчання в алгоритмі S.evol зафіксовано так:  $\tau = (\sqrt{2N})^{-1}$ ,  $\tau' = (\sqrt{2\sqrt{N}})^{-1}$ , де  $N$  — розмір популяції.

Вибір значень  $L$  та  $\sigma$  обґрунтовується намаганням віднайти баланс між швидкістю роботи алгоритму та його ефективністю.

В алгоритмах детермінованого та ймовірнісного витиснення використано: бінарне кодування (код Грея) вузлами дискретизації з точністю три знаки після коми; відстань Геммінга для визначення схожості особин; одноточковий кросинговер. Додатково після застосування оператора кросинговеру до нащадків застосовується оператор мутації з параметрами  $P_m$  та  $R_m$  — ймовірностями застосування оператора мутації до заданої хромосоми та до кожного гена заданої хромосоми відповідно; значення  $P_m$  та  $R_m$  підбирають експериментально.

Параметри паралельного алгоритму сходження на вершини: початкове значення радіуса околу 0.05; мінімальне значення радіуса околу 0.0001.

Як відомо, експериментальний аналіз стохастичних (у тому числі генетичних) алгоритмів ґрунтується на аналізі статистичних даних, отриманих в результаті багатьох незалежних прогонів алгоритмів на однакових вхідних даних. Визначені з цією метою набір тестових функцій та критерії оцінювання алгоритмів описано далі.

**Критерії порівняння.** Перш ніж вводити критерії оцінювання якості роботи алгоритмів, слід визначити умови зупинки алгоритмів та варіанти розташування елементів фінальної популяції, за яких будемо вважати, що шуканий максимум знайдено.

Паралельний алгоритм сходження на вершини зупиняє роботу, якщо значення радіуса околу стає меншим за 0.0001 або виконано 20000000 обчислень функції пристосованості. Генетичний алгоритм зупиняє роботу в разі встановлення його збіжності (середнє значення коефіцієнта пристосованості особин популяції не змінюється останні п'ять поколінь більше ніж на 0.0001) або здійснення 20000000 обчислень функції пристосованості. Іншими словами, якщо збіжність не досягнута, роботу алгоритму зупиняють після виконання заданої кількості обчислень функції пристосованості.

Максимум знайдено, якщо встановлено збіжність алгоритму та знайдеться хоча б одна особина фінальної популяції, коефіцієнт пристосованості якої є не меншим за  $\delta$  від значення цього максимуму, та яка розташовується на відстані не більше ніж  $\sigma$  від цього максимуму. Покладемо  $\delta = 0.01$ ,  $\sigma = 0.01$  для всіх наведених нижче функцій, крім функції  $F_7$ , та  $\sigma = 0.5$  для функції  $F_7$ .

Усі відомі авторам критерії оцінювання якості знайдених алгоритмом розв'язків [6] визначені на основі порівняння значень та розташування реальних максимумів із знайденими алгоритмом максимумами. У деяких критеріях додатково вимагається концентрація певного відсотку особин популяції в околах шуканих максимумів. На нашу думку, такий підхід є прийнятним лише для задач пошуку глобальних максимумів. Під час аналізу алгоритмів утворення ніш їх слід порівнювати не лише за якістю знайдених популяцією відомих розв'язків,

але й за способами взаємного розташування особин фінальної популяції. Алгоритми з великою кількістю хибних видів (ніш) — скупчень особин не в достатньо малих околах реальних максимумів — слід вважати найгіршими через неможливість їхнього практичного використання. У зв'язку з викладеним вище нами запропоновано явно визначати множину видів, утворених фінальною популяцією, та обчислювати додаткові критерії оцінювання кількості знайдених алгоритмом хибних піків.

Для визначення множини сформованих алгоритмом видів, якою визначається множина ніш або можливих піків функції пристосованості, аналізувати мемо фінальну популяцію за таким алгоритмом:

1. Покласти  $Seeds = \emptyset$ .
  2. Вибрати найкращу необроблену особину популяції  $x$ ; позначити її як оброблену  $x^*$ .
  3. Покласти  $found = FALSE$ .
  4. Якщо знайдеться особина  $s \in Seeds$  така, що  $d(x^*, s) \leq \varepsilon$ , то покласти  $found = TRUE$ ; інакше додати особину  $x^*$  до множини піків: покласти  $Seeds = Seeds \cup \{x^*\}$ .
  5. Якщо в популяції залишилися необроблені особини, то перейти на п. 2.
- Тут  $Seeds$  — множина найкращих особин кожного виду;  $d(x, y)$  — евклідова відстань між особинами  $x$  та  $y$ ; параметр  $\varepsilon = 0.01$ .

Очевидно, що в загальному випадку множина  $Seeds$  може містити як реальні, так і хибні піки; для визначення реальних піків елементи з  $Seeds$  порівнюють із реальними піками з використанням параметрів  $\delta$  та  $\sigma$ .

Для кожного прогону алгоритму обчислюватимемо такі критерії:  
 $NFE$  (number of fitness function evaluations) — кількість обчислень функції пристосованості;

$NSeeds$  (number of seeds) — кількість піків (реальних та хибних);

$NP$  (number of peaks) — кількість реальних піків;

$GP$  (number of global peaks) — кількість реальних глобальних піків;

$LP$  (number of local peaks) — кількість реальних локальних піків.

Очевидно, що  $NP = GP + LP$ .

$PR$  (peak ratio) — відношення кількості знайдених алгоритмом реальних піків до загальної кількості піків, які необхідно локалізувати.

$GPR$  (global peak ratio) — відношення кількості знайдених алгоритмом реальних глобальних піків до загальної кількості глобальних піків, які потрібно локалізувати.

$LPR$  (local peak ratio) — відношення кількості знайдених алгоритмом реальних локальних піків до загальної кількості локальних піків, які необхідно локалізувати.

Оскільки в загальному випадку  $NSeeds \geq NP$  (алгоритм знаходить хибні піки), введемо також критерій оцінювання кількості хибних піків.

$FPR$  (fake peak ratio) — частка хибних піків — відношення кількості хибних піків до загальної кількості сформованих алгоритмом ніш:  $FPR = \frac{NSeeds - NP}{NSeeds}$ .

Очевидно,  $0 \leq PR, FPR \leq 1$ ; алгоритм є кращим за більших значень  $PR$  та за менших значень  $FPR$ .

Для аналізу алгоритмів обчислюватимемо усереднені за всіма прогонами значення наведених критеріїв.

**Тестові функції.** Для експериментального аналізу алгоритмів обрано такі тестові функції [6].

1. Функція однакових максимумів (тестова функція Деба 1)

$$F_1(x_1, x_2, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n \sin^6(5\pi x_i), \quad 0 \leq x_i \leq 1, \quad i = \overline{1, n},$$

має  $5^n$  глобальних максимумів, що знаходяться на однаковій відстані один від одного.

2. Функція спадаючих максимумів (тестова функція Деба 2)

$$F_2(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \exp \left\{ -2(\ln 2) \left( \frac{x_i - 0.1}{0.8} \right)^2 \right\} \sin^6(5\pi x_i), \quad 0 \leq x_i \leq 1, \quad i = \overline{1, n},$$

має  $5^n$  максимумів різної висоти, розташованих на однаковій відстані один від одного.

3. Функція нерівномірних максимумів (тестова функція Деба 3)

$$F_3(x_1, x_2, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n \sin^6(5\pi(x_i^{0.75} - 0.05)), \quad 0 \leq x_i \leq 1, \quad i = \overline{1, n},$$

має  $5^n$  глобальних максимумів, що знаходяться на різних відстанях один від одного, причому відстані між точками максимумів зростають із збільшенням значення аргументу.

4. Функція нерівномірних спадаючих максимумів (тестова функція Деба 4)

$$F_4(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \exp \left\{ -2(\ln 2) \left( \frac{x_i - 0.08}{0.854} \right)^2 \right\} \sin^6(5\pi(x_i^{0.75} - 0.05)),$$

$$0 \leq x_i \leq 1, \quad i = \overline{1, n},$$

має  $5^n$  максимумів різної висоти, розташованих на різних відстанях один від одного, причому відстані між точками максимумів зростають із збільшенням значення аргументу.

Для всіх чотирьох функцій відомими є розташування та значення всіх максимумів.

Набір тестових функцій Деба є одним з найпоширеніших для аналізу алгоритмів розв'язання задач багатоекстремальної оптимізації. Для порівняльного аналізу роботи алгоритмів використовуватимемо також подані нижче функції.

5. Функція Растрігіна

$$F_5(x_1, x_2, \dots, x_n) = \sum_{i=1}^n (10 \cos(2\pi x_i) - x_i^2) - 10n, \quad -5.12 \leq x_i \leq 5.12, \quad i = \overline{1, n}.$$

Її контур складається з великої кількості локальних максимумів, значення яких спадають із збільшенням відстані від глобального максимуму. Така функція має  $10^n$  локальних та один глобальний максимум.

6. Функція Гриванга

$$F_6(x_1, x_2, \dots, x_n) = n - \left( \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \right), \quad -600 \leq x_i \leq 600, \quad i = \overline{1, n}.$$

Вона має один глобальний та багато локальних максимумів, причому локальні максимуми є регулярно розподіленими та їхні значення спадають із збільшеною відстані від глобального максимуму.

7. Функція спини шестигорбого верблюда (тестова функція Урсема 3)

$$F_7(x_1, x_2) = - \left( \left( 4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1x_2 + 4(x_2^2 - 1)x_2^2 \right),$$
$$-3 \leq x_1 \leq 3, \quad -2 \leq x_2 \leq 2,$$

має два глобальних та чотири локальних максимуми.

8. Функція Хін-Ши Янга 2

$$F_8(x_1, x_2) = (|x_1| + |x_2|) \exp(-x_1^2 - x_2^2), \quad -10 \leq x_1, x_2 \leq 10,$$

має чотири глобальні максимуми, розташовані на незначній відстані один від одного.

#### ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ

Аналіз алгоритмів проведено окремо для задачі пошуку всіх локальних та глобальних максимумів (задача 1) та задачі пошуку всіх глобальних максимумів (задача 2), причому розглянуто окремо випадки наявності в цільовій функції багатьох глобальних та, можливо, локальних максимумів (задача 2.1) та наявності в цільовій функції тільки одного глобального та багатьох локальних максимумів (задача 2.2).

Усі розглянуті алгоритми реалізовано програмно; як платформу розроблення програмного застосування обрано MATLAB.

Для оцінювання ефективності алгоритмів для кожного набору вхідних даних виконано по 10 прогонів всіх алгоритмів. Відзначимо, що  $i$ -й прогін кожного алгоритму має спільну стартову точку: випадково згенерована початкова популяція є однаковою для всіх алгоритмів на відповідному прогоні. Розмір популяції  $N = 500$ .

Для визначення оптимального набору параметрів кожного з алгоритмів сформовано такі тестові набори функцій залежно від класу задачі.

Задача 1, тестовий набір T1 включає функції Деба 1, Деба 2, Деба 3, Деба 4.

Задача 2.1, тестовий набір T21 містить функції Деба 1, Деба 3.

Задача 2.2, тестовий набір T22 включає функції Деба 2, Деба 4.

Експерименти проводились для тестових функцій розмірностей  $n = 1, 2, 3$ .

Усі три варіанти S1, S2, S3 алгоритму турнірного витиснення з гаусовою мутацією показали найкращі результати для  $L=3$ ,  $1/k=1/16$ . Загалом спостерігається тенденція до покращення роботи цього алгоритму (збільшення  $PR$ , зменшення  $FPR$ ) як у разі збільшення кількості нащадків  $L$ , так і в разі зменшення значення  $\sigma$ . Також, як і очікувалося, спостерігається погіршення роботи алгоритму із збільшенням розмірності задачі: зменшується кількість знайдених алгоритмом реальних піків, зростає кількість хибних піків.

Надалі посилатимемось на розглянуті алгоритми за першими буквами їхніх назв англійською мовою: TCGM\_S1, TCGM\_S2, TCGM\_S3 — відповідно варіанти S1, S2, S3 алгоритмів турнірного витиснення з гаусовою мутацією, PNC — алгоритм паралельного сходження на вершини, DC — алгоритм детермінованого витиснення, PC — алгоритм ймовірнісного витиснення.

Таблиця 1. Результати експериментів: алгоритм TCGM\_S2, тестовий набір T1

Параметри		Критерії				
$L$	$1/k$	$NFE$	$PR$	$GPR$	$LPR$	$FPR$
1	0.5	436792	0.5860	0.6919	0.2289	0.5420
2	0.5	601813	0.5493	0.7395	0.1649	0.4286
3	0.5	786983	0.5347	0.8099	0.1313	0.3548
1	0.25	470917	0.6611	0.7180	0.2992	0.5266
2	0.25	694388	0.6787	0.7967	0.2956	0.3948
3	0.25	823850	0.6884	0.8507	0.2924	0.3292
1	0.125	360767	0.7503	0.7304	0.3784	0.4261
2	0.125	594838	0.8017	0.7566	0.4038	0.3344
3	0.125	825733	0.8314	0.8321	0.4159	0.2896
1	0.0625	249708	0.8939	0.8373	0.4405	0.2636
2	0.0625	314463	0.9361	0.8461	0.4658	0.1989
3	0.0625	380167	0.9563	0.8908	0.4744	0.1538

Таблиця 2. Результати експериментів: алгоритм TCGM\_S2, функція  $F_2$ ,  $n = 1$ 

Параметри		Критерії								
$L$	$1/k$	$NFE$	$Seeds$	$NP$	$GP$	$LP$	$PR$	$GPR$	$LPR$	$FPR$
1	0.5	205600	4.3	4.3	1	3.3	0.86	1	0.825	0
2	0.5	160700	2.9	2.9	1	1.9	0.58	1	0.475	0
3	0.5	157100	2.2	2.2	1	1.2	0.44	1	0.3	0
1	0.25	301200	5.9	5	1	4	1	1	1	0.1405
2	0.25	317150	5.1	5	1	4	1	1	1	0.0167
3	0.25	251900	5	5	1	4	1	1	1	0
1	0.125	112300	5	5	1	4	1	1	1	0
2	0.125	115700	5	5	1	4	1	1	1	0
3	0.125	117300	5	5	1	4	1	1	1	0
1	0.0625	77800	5	5	1	4	1	1	1	0
2	0.0625	77000	5	5	1	4	1	1	1	0
3	0.0625	82700	5	5	1	4	1	1	1	0

Таблиця 3. Результати експериментів: алгоритм TCGM\_S2, функція  $F_2$ ,  $n = 3$ 

Параметри		Критерії								
$L$	$1/k$	$NFE$	$Seeds$	$NP$	$GP$	$LP$	$PR$	$GPR$	$LPR$	$FPR$
1	0.5	668400	381.1	2.5	0	2.5	0.02	0	0.0202	0.9935
2	0.5	1065500	277.4	4.8	0.2	4.6	0.0384	0.2	0.0379	0.9825
3	0.5	1703700	179.4	8.2	0.7	7.5	0.0656	0.7	0.0605	0.9527
1	0.25	696500	364.5	7.2	0.2	7	0.0576	0.2	0.0565	0.9795
2	0.25	1362200	229.6	14	0.6	13.4	0.112	0.6	0.1081	0.9372
3	0.25	1742700	181.3	14.8	0.7	14.1	0.1184	0.7	0.1137	0.9144
1	0.125	719800	375.8	19.8	0.1	19.7	0.1584	0.1	0.1589	0.9467
2	0.125	1210550	295	34.6	0.2	34.4	0.2768	0.2	0.2774	0.8777
3	0.125	1875100	221.7	47.9	0.8	47.1	0.3832	0.8	0.3798	0.7789
1	0.0625	443200	262.8	67.7	0.2	67.5	0.5416	0.2	0.5446	0.7397
2	0.0625	615500	198.5	90.9	0.3	90.6	0.7272	0.3	0.7307	0.5358
3	0.0625	734700	175.3	99.3	0.5	98.8	0.7944	0.5	0.7968	0.4299

Таблиця 4. Результати експериментів: алгоритм РС, тестовий набір Т1

Параметри		Критерії				
$P_m$	$R_m$	$NFE$	$PR$	$GPR$	$LPR$	$FPR$
0.1	0.1	1402625	0.1737	0.4865	0.0336	0.8950
0.1	0.2	1381592	0.2135	0.5115	0.0372	0.9052
0.1	0.3	1423467	0.2502	0.5766	0.0309	0.9139
0.3	0.1	1457358	0.2430	0.5797	0.0505	0.9192
0.3	0.2	2172325	0.3216	0.6117	0.0676	0.9327
0.3	0.3	2574675	0.4101	0.6283	0.1031	0.9343
0.5	0.1	2051800	0.2861	0.5422	0.0653	0.9324
0.5	0.2	2163658	0.3809	0.5111	0.1017	0.9441
0.5	0.3	1986567	0.4354	0.5097	0.1461	0.9478
0.7	0.1	1796850	0.2894	0.4778	0.0658	0.9418
0.7	0.2	2111642	0.4128	0.4821	0.1364	0.9522
0.7	0.3	2160133	0.4123	0.4548	0.1362	0.9553

Зазначимо, що у табл. 1–11 для всіх критеріїв наведено усереднені (середні) значення по всіх прогонах. У табл. 1 подано результати обчислення цих значень на різних наборах параметрів для всіх функцій з тестового набору Т1 для алгоритму TCGM\_S2 турнірного витиснення з гаусовою мутацією. Табл. 2 та 3 ілюструють роботу цього алгоритму для функцій  $F_2$  розмірностей  $n = 1$  та  $n = 3$  відповідно. Із табл. 3 випливає, що алгоритм турнірного витиснення з гаусовою мутацією не завжди надає перевагу глобальним оптимумам (можуть бути сформовані ніші навколо локальних оптимумів, при цьому глобальний оптимум знайдений не буде).

Алгоритм детермінованого витиснення показав найкращі результати за таких значень параметрів:  $P_m = 0.7$ ,  $R_m = 0.3$ . В цілому алгоритм характеризується дуже низьким відсотком хибних піків, що є безсумнівною його перевагою. Водночас за наявності локальних піків він у більшості випадків знаходить лише один локальний пік або не знаходить їх взагалі. Алгоритм завжди віддає перевагу глобальному піку: на відміну від алгоритму турнірного витиснення з гаусовою мутацією він завжди формує нішу навколо хоча б одного глобального піку.

Алгоритм ймовірнісного витиснення не дав прийнятних результатів на жодному наборі параметрів. Хоча він і знаходить близько 40% піків на своїх найкращих конфігураціях, але характеризується надзвичайно великою (понад 90%)

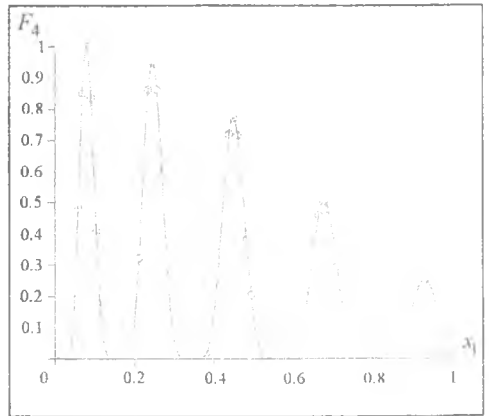
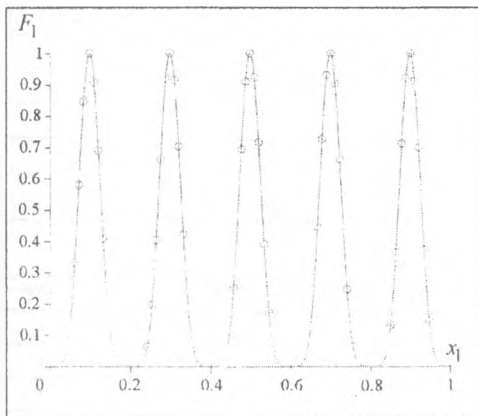


Рис. 1. Ніші, знайдені алгоритмом РС, функція Деба 1,  $n = 1$

Рис. 2. Ніші, знайдені алгоритмом РС, функція Деба 4,  $n = 1$

кількістю хибних піків, до того ж має високу складність (високе значення показника  $NFE$ ). Як ілюстрацію наведено табл. 4 з результатами обчислення усереднених значень критеріїв на різних наборах параметрів для всіх функцій з тестового набору T1. Ще однією ілюстрацією є рис. 1 та рис. 2, на яких кружечками позначено знайдені алгоритмом ніші (*Seeds*) для функцій  $F_1$  та  $F_4$ ,  $n=1$ , за конфігурації  $P_m = 0.5$ ,  $R_m = 0.3$  (у середньому за всіма прогонами ця конфігурація дає найкраще значення критерію  $PR = 0.4354$ ).

Підсумовуючи сказане, випишемо рекомендовані набори параметрів алгоритмів для розглянутих задач. Усі три варіанти розробленого алгоритму витиснення з гаусовою мутацією мають такі параметри: для задач 1 і 2.1  $L=3$ ,  $1/k = 1/16$ ; для задачі 2.2  $L=3$ ,  $1/k = 1/16$ ,  $1/k = 1/4$ ,  $1/k = 1/8$  відповідно для варіантів S1, S2, S3 цього алгоритму. Параметри алгоритму детермінованого витиснення є однаковими для всіх трьох задач:  $P_m = 0.7$ ,  $R_m = 0.3$ . Алгоритм ймовірнісного витиснення має такі параметри:  $P_m = 0.5$ ,  $R_m = 0.3$  (для задачі 1);  $P_m = 0.3$ ,  $R_m = 0.3$  (для задачі 2.1);  $P_m = 0.3$ ,  $R_m = 0.1$  (для задачі 2.2).

### ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ

У табл. 5 подано усереднені значення критеріїв, обчислені за результатами прогонів всіх досліджуваних алгоритмів на тестовому наборі T1. Можна стверджувати, що під час розв'язування задачі пошуку всіх (глобальних, локальних) максимумів найкраще зарекомендували себе алгоритми PNC, TCGM\_S2 і TCGM\_S1. Алгоритм DC дає непогані результати пошуку глобальних максимумів, але відсоток знаходження локальних максимумів є недопустимо малим. PC є найгіршим серед розглянутих алгоритмів. Він характеризується досить невисоким значенням критерію  $PR$ , великою складністю та надзвичайно високою розпоршеністю особин фінальної популяції в пошуковому просторі: кількість хибних піків є надзвичайно великою ( $FPR > 0.9$ ). У зв'язку з цим алгоритм PC було вилучено з подальшого розгляду. Проте своєї мети — запобігання втраті нижчих піків на користь вищих — цей алгоритм досяг: значення критерію  $LPR$  для нього є вищим за відповідне значення для алгоритму DC.

Таблиця 5. Порівняння алгоритмів для задачі 1, тестовий набір T1

Алгоритм	Критерії				
	$NFE$	$PR$	$GPR$	$LPR$	$FPR$
PNC	39127	0.9881	0.9525	0.4942	0.1040
DC	77233	0.3915	0.8235	0.0323	0.0188
PC	1986567	0.4354	0.5097	0.1461	0.9478
TCGM_S1	401600	0.9559	0.9243	0.4735	0.1515
TCGM_S2	380167	0.9563	0.8908	0.4744	0.1538
TCGM_S3	564433	0.8054	0.7801	0.4005	0.9093

Таблиця 6. Порівняння алгоритмів для задачі 1, функція  $F_7$

Алгоритм	Критерії								
	$NFE$	$NSeeds$	$NP$	$GP$	$LP$	$PR$	$GPR$	$LPR$	$FPR$
PNC	63826	6	6	2	4	1	1	1	0
DC	73900	2	2	2	0	0.3333	1	0	0
TCGM_S1	541700	5.7	5.7	2	3.7	0.95	1	0.925	0
TCGM_S2	319300	6	6	2	4	1	1	1	0
TCGM_S3	3855700	8.1	6	2	4	1	1	1	0.2367

Таблиця 7. Порівняння алгоритмів для задачі 1, функція  $F_8$

Алгоритм	Критерій			
	$NFE$	$NSeeds$	$NP = GP$	$PR = GPR$
PHC	51800	1	1	0.25
TCGM_S1	86200	9.4	2.7	0.675
TCGM_S2	77900	9.3	2.9	0.725

Оскільки тестовий набір T1 використовувався для налаштування параметрів генетичних алгоритмів, їхню роботу було додатково перевірено на функції  $F_7$ . Ця функція має два глобальні і чотири локальні максимуми та є однією з найпоширеніших для порівняльного аналізу генетичних алгоритмів утворення ніш. Як видно з табл. 6, алгоритмами-переможцями знову стали алгоритми PHC та TCGM\_S2: вони знайшли всі шукані максимуми, причому всі особини фінальної популяції сконцентрувалися навколо піків функції ( $FPR = 0$ ); типовий результат прогону алгоритму TCGM\_S2 подано на рис. 3.

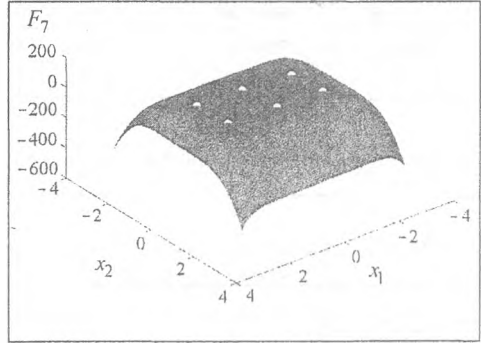


Рис. 3. Ніші, знайдені алгоритмом TCGM\_S2, функція  $F_7$

Відомо [7], що алгоритм PHC дає непогані результати на нескладних функціях. Було проведено його порівняння з алгоритмами TCGM\_S2 та TCGM\_S1 на складній функції  $F_8$ . Ця функція має чотири однакових максимуми, розташованих на незначній відстані один від одного. З табл. 7 видно, що алгоритм TCGM\_S2 дав найкращі результати: в більшості прогонів він знайшов три з чотирьох піків, тоді як алгоритм PHC групує популяцію навколо одного піку. Знайдені цими алгоритмами піки функції  $F_8$  зображено на рис. 4 та рис. 5.

Із сказаного випливає, що для розв'язування задачі пошуку всіх (глобальних, локальних) максимумів багатоекстремальної функції слід рекомендувати алгоритм TCGM\_S2, а для простих задач — алгоритм PHC як більш швидкий. Аналіз табл. 8 та відповідних критеріїв з табл. 6 та 7 показує, що ці алгоритми можна рекомендувати також для розв'язування задачі знаходження всіх глобальних максимумів, якщо цільова функція має їх декілька (та, можливо, має локальні максимуми).

Із сказаного випливає, що для розв'язування задачі пошуку всіх (глобальних, локальних) максимумів багатоекстремальної функції слід рекомендувати алгоритм TCGM\_S2, а для простих задач — алгоритм PHC як більш швидкий. Аналіз табл. 8 та відповідних критеріїв з табл. 6 та 7 показує, що ці алгоритми можна рекомендувати також для розв'язування задачі знаходження всіх глобальних максимумів, якщо цільова функція має їх декілька (та, можливо, має локальні максимуми).

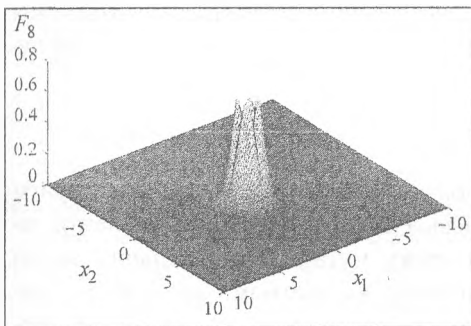


Рис. 4. Піки функції  $F_8$ , знайдені алгоритмом TCGM\_S2

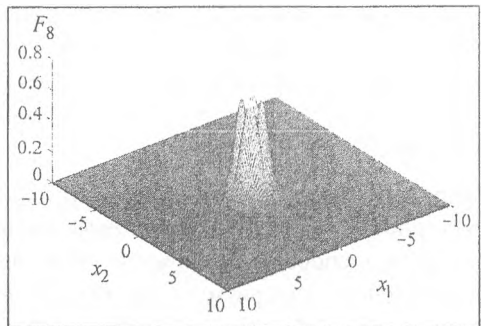


Рис. 5. Піки функції  $F_8$ , знайдені алгоритмом PHC

**Таблиця 8.** Порівняння алгоритмів для задачі 2.1, тестовий набір T21

Алгоритм	Критерії	
	<i>NFE</i>	<i>PR = GPR</i>
PHC	39138	0.9884
DC	75383	0.6469
PC	1763067	0.4732
TCGM_S1	404500	0.9653
TCGM_S2	384600	0.9649
TCGM_S3	537667	0.8103

**Таблиця 9.** Порівняння алгоритмів для задачі 2.2, тестовий набір T22

Алгоритм	Критерії	
	<i>NFE</i>	<i>PR = GPR</i>
PHC	39115	0.9167
DC	79083	1
PC	1971783	0.8833
TCGM_S1	398700	0.8833
TCGM_S2	1095066	0.9167
TCGM_S3	447250	0.8000

**Таблиця 10.** Порівняння алгоритмів для задачі 2.2, функція  $F_5$

Алгоритм	Критерії				
	<i>GP, n = 1</i>	<i>GP, n = 2</i>	<i>GP, n = 3</i>	<i>GPR</i>	<i>NFE</i>
PHC	1	1	0.3	0.7666	43178
DC	1	1	1	1	36566
TCGM_S1	1	1	0.3	0.7666	2159433
TCGM_S2	1	1	0.5	0.8333	1042233
TCGM_S3	1	1	0.3	0.7666	1510833

**Таблиця 11.** Порівняння алгоритмів для задачі 2.2, функція  $F_6$

Алгоритм	Критерії				
	<i>GP, n = 1</i>	<i>GP, n = 2</i>	<i>GP, n = 3</i>	<i>GPR</i>	<i>NFE</i>
PHC	1	0	0	0.3333	120438
DC	1	0.4	0	0.4667	48166
TCGM_S1	1	0	0	0.3333	793300
TCGM_S2	1	0.8	0	0.6	1011033
TCGM_S3	1	0	0	0.3333	1620103

З табл. 9 видно, що найкращим алгоритмом для розв'язування задачі пошуку всіх глобальних максимумів за умови наявності у заданій цільовій функції єдиного глобального та багатьох локальних максимумів є алгоритм DC; непогані результати продемонстрували також алгоритми TCGM\_S2 та PHC. Роботу алгоритмів було додатково перевірено на більш складних функціях  $F_5$  та  $F_6$ ; усереднені результати прогонів подано в табл. 10 та 11 відповідно. Як і очікувалося, на функціях  $F_5$  та  $F_6$  алгоритм PHC показав суттєво гірші результати, причому вірогідність знаходження глобального максимуму зменшується у разі збільшення розмірності задачі. Результати роботи алгоритмів TCGM\_S2 та DC є близькими.

## ВИСНОВКИ

Запропоновано новий генетичний алгоритм утворення ніш — алгоритм турнірного витиснення з гаусовою мутацією. Розглянуто три його варіанти залежно від способу визначення величини кроку мутації. Порівняльний аналіз розробленого алгоритму з іншими генетичними алгоритмами витиснення (детермінованим, ймовірнісним) та з паралельним алгоритмом сходження на вершини показав, що варіант S2 алгоритму турнірного витиснення з гаусовою мутацією є найкращим для задачі пошуку всіх (глобальних, локальних) макси-

мумів заданої функції, а також для задачі знаходження всіх глобальних максимумів за умови наявності кількох у цільовій функції. Цей алгоритм також продемонстрував непогані результати під час розв'язування задачі пошуку глобального максимуму функції, що характеризується наявністю єдиного глобального та багатьох локальних максимумів.

Для експериментального аналізу алгоритмів пошуку всіх (глобальних, локальних) максимумів запропоновано здійснювати явний розподіл особин фінальної популяції за нішами та обчислювати додатковий критерій — частку хибних піків (*FPR*). Без такого розподілу практичне використання будь-якого алгоритму пошуку всіх максимумів, на думку авторів, є неможливим. Обчислення критеріїв *PR* та *FPR* надає змогу оцінити практичну цінність алгоритмів: який відсоток реальних піків знайдено алгоритмом та який відсоток локалізованих алгоритмом ніш насправді є хибними піками. Врахування обох критеріїв, зокрема, дало змогу зробити висновок щодо неможливості практичного використання алгоритму ймовірнісного витиснення для пошуку і глобальних, і локальних максимумів через надвелику кількість угруповань особин у різних точках пошукового простору (як на піках, так і на схилах цільової функції). Водночас алгоритм детермінованого витиснення характеризується надзвичайно низьким показником *FPR* (менше двох відсотків), але й кількість знайдених ним локальних піків є неприпустимо малою. Пропонований алгоритм турнірного витиснення з гаусовою мутацією у варіанті S2 для великої кількості знайдених глобальних та локальних піків дає в середньому до 20 % хибних піків, що не є надвисоким показником.

#### СПИСОК ЛІТЕРАТУРИ

1. Glybovets M.M., Gulayeva N.M. Evolutionary multimodal optimization. In: *Optimization Methods and Applications: In Honor of Ivan V. Sergienko's 80th Birthday*. Butenko S., Pardalos P.M., Shylo V. (Eds.). Springer International Publishing, 2017. P. 129–173.
2. Preuss M. Multimodal optimization by means of evolutionary algorithms. Berlin: Springer, 2015. 175 p.
3. Reeves C. (ed.). *Modern heuristic techniques for combinatorial problems*. New York: Halsted Press, 1993.
4. Aickelin U. An indirect genetic algorithm for set covering problems. *J. of the Oper. Res. Society*. 2002. Vol. 53. P. 1118–1126.
5. Сергиенко И.В., Шило В.П. Задачи дискретной оптимизации: проблемы, методы решения, исследования. Киев: Наук. думка, 2003. 264 с.
6. Глибовець М.М., Гулаєва Н.М. Еволюційні алгоритми. Підручник. Київ: НАУКМА, 2013. 828 с.
7. Mahfoud S. Niching method for genetic algorithms. Ph.D. thesis. Urbana, 1995. 251 p.
8. Singh G., Deb K. Comparison of multi-modal optimization algorithms based on evolutionary algorithms. *Proc. 8th Ann. Conf. on Genetic and Evolutionary Computation (GECCO'06)* (July 8–12, 2006, Seattle, Washington, USA). Seattle, 2006. P. 1305–1312.
9. Friedrich T., Oliveto P.S., Sudholt D., Witt C. Analysis of diversity-preserving mechanisms for global exploration. *Evolutionary Computation*. 2009. Vol. 17, N 4. P. 455–476. <https://doi.org/10.1162/evco.2009.17.4.17401>.
10. Osuna E.C., Sudholt D. Runtime analysis of crowding mechanisms for multimodal optimization *IEEE Trans. Evol. Comp.* 2019. <https://doi.org/10.1109/TEVC.2019.2914606>.
11. Mengshoel O., Goldberg D. Probabilistic crowding: deterministic crowding with probabilistic replacement. *Proc. the Genetic and Evolutionary Computation Conference (GECCO-99)* (13–17 July, 1999, Orlando, FL, USA). Orlando, 1999. P. 409–416.

Надійшла до редакції 19.03.2019

**Аннотация.** Для решения задач многоэкстремальной оптимизации предложен новый генетический алгоритм образования ниш — генетический алгоритм турнирного вытеснения с гауссовой мутацией. Проведенный сравнительный анализ предложенного алгоритма с другими алгоритмами вытеснения и с параллельным алгоритмом поиска с восхождением к вершинам показал преимущества разработанного алгоритма во многих случаях. Введен критерий оценки степени разброса элементов популяции. Показано, что вычисление этого критерия является целесообразным для оценки качества работы алгоритмов поиска глобальных и локальных максимумов.

**Ключевые слова:** задача многоэкстремальной оптимизации, генетические алгоритмы образования ниш, алгоритмы вытеснения, параллельный алгоритм поиска с восхождением к вершинам, доля ложных пиков.

**V.P. Shylo, M.M. Glybovets, N.M. Gulayeva, K.V. Nikishchikhina**  
**TOURNAMENT CROWDING GENETIC ALGORITHMS BASED ON GAUSS MUTATION**

**Abstract.** To solve multimodal optimization problems, a new niching genetic algorithm named tournament crowding genetic algorithm based on Gauss mutation is proposed. A comparative analysis of this algorithm to other crowding algorithms and to parallel hill-climbing algorithm has shown the advantages of the proposed algorithm in many cases. The FPR criterion to estimate the distribution of population elements is proposed and it is shown that computation of this criterion is advisable to estimate algorithms solving multimodal problems of finding global and local maxima.

**Keywords:** multimodal optimization problem, niching genetic algorithms, crowding algorithms, parallel hill-climbing algorithm, fake peak ratio.

**Шило Володимир Петрович,**

доктор фіз.-мат. наук, професор, провідний науковий співробітник Інституту кібернетики ім. В.М. Глушкова НАН України, Київ, e-mail: v.shylo@gmail.com.

**Глибовець Микола Миколайович,**

доктор фіз.-мат. наук, професор, декан Національного університету «Києво-Могилянська академія», Київ, e-mail: glib@ukma.kiev.ua.

**Гулаєва Наталія Михайлівна,**

кандидат фіз.-мат. наук, доцент кафедри Національного університету «Києво-Могилянська академія», Київ, e-mail: ngulayeva@yahoo.com.

**Нікішихіна Катерина Вячеславівна,**

магістр факультету інформатики Національного університету «Києво-Могилянська академія», Київ, e-mail: katelyna.nikishchikhina@gmail.com.