

# ЧИСЕЛЬНІ МЕТОДИ В ЕКСТРЕМАЛЬНИХ ЗАДАЧАХ, МЕТОДИ НАБЛИЖЕННЯ ФУНКЦІЙ

---

УДК 519.712, 519.245 004.032.26

*Ю.А. Літвінчук*

## ПРО ОДНЕ УЗАГАЛЬНЕННЯ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

**Літвінчук Юлія Анатоліївна**

Чернівецький національний університет імені Юрія Федьковича,

orcid: 0000-0002-5277-0109

*julilitvintchuk@gmail.com*

У реальному світі існує безліч сценаріїв оптимізації з великою кількістю обмежень та цільових функцій, які за своєю природою є розривними, нелінійними, невиконуваними та мультимодальними. Крім того, задачі оптимізації є багатовимірними зі змішаними типами змінних, такими як цілі, дійсні, дискретні, двійкові, та мають різний діапазон значень, що вимагає нормалізації. Отже, простір пошуку задачі не завжди може бути гладким. Таким чином, еволюційні алгоритми почали використовуватися і розвиватися в обчислювальних процесах та вирішенні складних інженерних завдань, а також у методах штучного інтелекту. У запропонованій роботі розглянуто сукупність методів оцінки оптимальних параметрів тренувальних функцій з використанням еволюційних та генетичних алгоритмів пошуку в порівнянні з розширеним CMA-ES-алгоритмом. Проаналізовано метаевристичні алгоритми оптимізації ALO, ABCO, GA, PSO та класичний CMA-ES, які ґрунтуються на поведінці живих організмів в реальних природних середовищах. Для аналізу використано метод Монте-Карло, який дає можливість зробити висновки про розподіл кількості обчислень цільової функції. В роботі також запропоновано розширення алгоритму CMA-ES з використанням сумішей нормальних розподілів з невизначеною величиною розмірності суміші та з відомим базовим розподілом для оцінки оптимальних значень відомих тестових функцій. Розроблений алгоритм побудовано за припущенням багатопіковості розподілу параметрів складних систем, у тому числі гіперпараметрів нейронних мереж та параметрів стохастичних диференціальних рівнянь. Дослідження показують, що з ростом кількості піків для розширеного CMA-ES-алгоритму кількість звернень до цільової функції спадає і тим самим підтверджує ефективність запропонованого розширеного алгоритму CMA-ES. Однак для малої розмірності  $n$  вибір великої кількості піків є недоцільним. Дані властивості свідчать про існування взаємозалежності між розмірністю початкової задачі та кількістю вибраних піків.

**Ключові слова:** суміш розподілів, оптимізація гіперпараметрів, генетичні алгоритми, CMA-ES-алгоритм, метод Монте-Карло, стохастичні диференціальні рівняння.

## Вступ

Дослідженням методів оцінки параметрів чи гіперпараметрів складних систем присвячено велику кількість робіт, зокрема [1–8]. Основні проблеми у задачі оцінки пов'язані зі складністю алгоритмів, які використовуються у задачах оцінки параметрів стохастичних диференціальних рівнянь та інших складних систем. У даній роботі основну увагу сконцентровано на розширеному алгоритмі стратегії еволюційної адаптації на основі коваріаційної матриці (Covariance Matrix Adaptation Evolution Strategy — CMA-ES). Розглянемо більш детально класичний CMA-ES-алгоритм та його модифікації. Наприклад, у [9] представлено метод стратегії диференціальної еволюції (DES) — алгоритм, що є чимось середнім між диференціальною еволюцією (DE) та CMA-ES. DES використовує комбінації різницевих векторів між архівними індивідами і одновимірними гауссівськими випадковими векторами вздовж напрямків минулих зсувів середніх точок. DES-метод певною мірою відповідає CMA-ES, але обробляє точки, а не розподіли. Згідно з експериментальними результатами DES показує швидкість лінійної збіжності для квадратичних функцій у широкому спектрі чисел обумовленості матриці Гессе. Чисельні результати, представлені у статті, показують, що DES конкурентоспроможний порівняно з CMA-ES під час як локальної, так і глобальної оптимізації.

У [10] пропонується новий варіант CMA-ES, названий AEALSCE, для задач оптимізації в неперервній області. AEALSCE отримано шляхом інтеграції CMA-ES з двома стратегіями, які можуть коригувати еволюційні напрямки та збагачувати різноманітність популяції індивідів. Автори пропонують нову стратегію адаптації анізотропного власного значення (AEA), що адаптує область пошуку до оптимальних еволюційних напрямків та анізотропно масштабує власні значення коваріаційної матриці на основі виявлення локального ландшафту (local fitness landscape) відповідності:

$$\begin{cases} x_L = m_t + \delta v_j, \\ x_R = m_t - \delta v_j, \end{cases}$$

де  $\delta \sim N(0, \lambda_j)$ ,  $v_j$  вказує на  $j$ -й власний вектор коваріаційної матриці,  $\lambda_j$  —  $j$ -те власне значення коваріаційної матриці відповідно.

Інша стратегія називається стратегією локального пошуку (LS), яка виконується у власній системі координат і як локальний метод експлуатації може збагатити різноманітність популяції. Згідно з статистичними результатами експериментів запропонований авторами алгоритм AEALSCE не поступається іншим алгоритмам ефективної точності.

В [11] запропоновано модифікований метод обробки обмеження блоку (box constraint) для CMA-ES. Також, як і в [10], ідея полягає в оптимізації без обмежень завдяки введенню штучного фітнес-ландшафту (artificial fitness landscape), де функція штрафу додається до значень функції в найближчих допустимих розв'язках, адаптуючи штрафні коефіцієнти, що визначають чутливість обмежень до значення цільової функції, він створює «розумний» ландшафт віртуальної функції за межами допустимої області. Оптимізація працює доти, доки не буде досягнуто розумних штрафних коефіцієнтів. Експериментальні результати показують, що запропонований алгоритм може не збігатися з функціями показникового множення, тоді як запропонований у роботі алгоритм демонструє збіжність.

Застосування гібридизованої стратегії еволюції методу адаптації коваріаційної матриці крос-ентропії (CE-CMAES) запропоновано в [12]. Представлений CE-CMA-ES використовує перехресну ентропію для глобального дослідження

простору пошуку та стратегію адаптації коваріаційної матриці для локальної експлуатації. Основна новизна алгоритму CE-CMA-ES — це адаптивний механізм зміни розміру кроку (стандартного відхилення), що контролюється шляхом еволюції і є сумою послідовних кроків у кожному вимірі. Порівняльний аналіз показує, що CE-CMA-ES досягає кращих результатів порівняно з сучасними алгоритмами.

У [13] розроблено метод самоадаптації коваріаційної матриці з відштовхувальними субпопуляціями (RS-CMSA), який гібридує та об'єднує кілька концепцій і технік з різних існуючих методів, таких як точки табу в пошуку табу, нормалізована відстань Махаланобіса тощо. RS-CMSA в основному складається зі стратегії еволюції методу адаптації коваріаційної матриці (CMSA-ES), підсиленої елітарністю, як основної пошукової системи кількох субпопуляцій однакового розміру. RS-CMSA показав значну перевагу над іншими методами (NMMSO та NEA2), які також тестувалися на більш складних композитних функціях.

У [14] автор розглядає непараметричне моделювання вхідних просторів в оптимізації чорної скриньки в просторах Гільберта відтворюючого ядра (RKHS). Це моделювання призводить до проблеми функціональної оптимізації, предметною областю якої є функціональний простір RKHS, що дозволяє проводити оптимізацію дуже великого класу функцій. Автор пропонує CMA-ES-RKHS, узагальнений алгоритм CMA-ES, здатний виконувати оптимізацію функцій чорної скриньки в RKHS. Результати CMA-ES-RKHS показують явне виконання функціональної оптимізації та подолання проблеми підбору значень початкових параметрів.

### Дві основні задачі

**Задача 1.** У роботах [1, 2] розглянуто проблему мінімізації дисперсії випадкового процесу, що описує динаміку розвитку сукупності клітин  $\{x_1, x_2, x_3\}$  системою стохастичних диференціальних рівнянь

$$\begin{aligned} d \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} = & \begin{pmatrix} -q - \alpha(t) & 0 & 2\gamma \\ \alpha(t) & -q - \beta & 0 \\ 0 & \beta & -q - \gamma \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} dt + \\ & + \begin{pmatrix} -\sqrt{\alpha(t)x_1(t)} & 0 & 2\sqrt{\gamma x_3(t)} \\ \sqrt{\alpha(t)x_1(t)} & -\sqrt{\beta x_2(t)} & 0 \\ 0 & -\sqrt{\beta x_2(t)} & -\sqrt{\gamma x_3(t)} \end{pmatrix} \begin{pmatrix} dW_1(t) \\ dW_2(t) \\ dW_3(t) \end{pmatrix} + \\ & + \begin{pmatrix} \sqrt{qx_1(t)} & 0 & 0 \\ 0 & \sqrt{qx_2(t)} & 0 \\ 0 & 0 & \sqrt{qx_3(t)} \end{pmatrix} \begin{pmatrix} dW_4(t) \\ dW_5(t) \\ dW_6(t) \end{pmatrix}, \end{aligned} \quad (1)$$

де  $W_i(t), t \geq 0$  — стандартні вінеровські процеси, що визначаються кореляційною матрицею

$$R = (R_{ij})_{ij=1}^6;$$

$\beta, \gamma, q$  — невідомі константи системи, які необхідно оцінити;  $\alpha(t)$  — невідома функція, яка зазвичай належить деякому класу. Наприклад, у роботах [15, 16] розглянуто клас поліноміальних функцій, тобто

$$\alpha(t) \in A = \left\{ \alpha_0 + \dots + \alpha_k t^k \right\}, k \in N.$$

Більш складні простори функцій розглянуто у роботі [17], де як клас функцій  $\alpha(t)$  розглянуто раціональні функції

$$\alpha(t) \in A = \left\{ \frac{\alpha_0^1 + \dots + \alpha_k^1 t^k}{\alpha_0^2 + \dots + \alpha_m^2 t^k} \right\}, \quad k, m \in N.$$

Основна проблема при аналізі системи (1) полягає в оцінці невідомих параметрів

$$(\beta, \gamma, q, \alpha, R) \in R_+^3 \times A \times \Xi,$$

де  $\Xi$  — простір додатно визначених квадратних матриць розмірності  $6 \times 6$  з одиничними діагональними елементами, а критерій оптимальності визначається за допомогою наступного функціоналу якості:

$$J(\beta, \gamma, q, \alpha, R) = \sum_{i=1}^n c_i E |x(t_i) - x_i|^2. \quad (2)$$

У формулі (2)  $x(t_i)$  позначає розв'язок системи (1) у моменти часу  $t_i$ ,  $x_i$  — відомі значення процесу, який потрібно оцінити,  $c_i$  — нормуючі константи, характер яких диктується оптимізаційною задачею. У формулі (2) важливими складовими є математичні сподівання відхилень  $E |x(t_i) - x_i|^2$ . Слід зауважити, що із-за складного вигляду правої частини системи (1) у загальному випадку неможливо знайти аналітичний вигляд  $E |x(t_i) - x_i|^2$  як функції від невідомих параметрів  $(\beta, \gamma, q, \alpha, R)$ . Тому для оцінки  $E |x(t_i) - x_i|^2$  найчастіше використовується метод Монте-Карло з модифікаціями [18–22]. Одним з недоліків методів Монте-Карло є неперервна лінійна залежність складності алгоритму від кількості симуляцій  $M$ , тобто складність методів Монте-Карло рівнятиметься  $O(M)$ . З іншого боку, для пошуку оптимального набору параметрів

$$(\hat{\beta}, \hat{\gamma}, \hat{q}, \hat{\alpha}, \hat{R}) = \arg \min_{(\beta, \gamma, q, \alpha, R) \in R_+^3 \times A \times \Xi} J(\beta, \gamma, q, \alpha, R) \quad (3)$$

варто визначити метод з невисокою складністю. Оскільки досліджувані параметри розглядаються в неперервній області, для пошуку оптимальних параметрів найчастіше використовують еволюційні та генетичні алгоритми пошуку [23–26], які в свою чергу неперервно лінійно залежні від кількості частинок (особин)  $P$  в даному алгоритмі. Якщо узагальнити, то складність алгоритму пошуку оптимальних параметрів  $(\hat{\beta}, \hat{\gamma}, \hat{q}, \hat{\alpha}, \hat{R})$  рівняння (1) та відповідної оптимізаційної задачі (3) з генетичним алгоритмом у ролі оновлення параметрів та алгоритму Монте-Карло для обчислення  $E |x(t_i) - x_i|^2$  буде мати складність  $O(M * P * T)$ , де  $T$  — величина розбиття на інтервалі  $[0, \max_{1 \leq i \leq n} t_i]$ . Також дану методику можна використовувати для задач з точками концентрації, модель яких розглянута в [27].

**Задача 2.** Друга задача, яка потребує покращень алгоритмів пошуку, — це оцінка параметрів нейронних мереж [4, 28–31]. Для нейронних мереж невеликої розмірності надають перевагу алгоритмам «грубої сили», а саме, алгоритмам пошуку по сітці, алгоритмам випадкового пошуку по сітці та їх модифікаціям. Проте для нейронних мереж великої розмірності дані тривіальні підходи вимагають великої кількості обчислень, тому використовуються байєсівські, градієнтні та гене-

тичні алгоритми. З оптимізаційних алгоритмів оцінки гіперпараметрів нейронних мереж особливу увагу слід приділити стратегії еволюційної адаптації на основі коваріаційної матриці [3–8]. Основна ідея СМА-ES-стратегії полягає у виборі нових особин у генетичному алгоритмі на основі деякого базового розподілу (зазвичай нормального закону розподілу  $N(a, \Sigma)$  при середньому значенні  $a$ ), дисперсія  $\Sigma$  якого оновлюється на кожній ітерації (епосі) на основі нових обчислених особин з найоптимальнішими значеннями на основі критерію якості (2) чи точності нейронної мережі для оптимізації гіперпараметрів нейронних мереж.

Алгоритм СМА-ES або його розширення можуть використовуватися як для задачі оптимізації параметрів системи стохастичних диференціальних рівнянь (1), так і для пошуку оптимальних гіперпараметрів нейронних мереж. Основний недолік алгоритму СМА-ES — однопіковість базового розподілу та, як наслідок, велика дисперсія при оцінці параметрів. Дану проблему можна проілюструвати наступним простим прикладом багатопікової функції:

$$f_{T1}(x) = \frac{\sin(\sum_{i=1}^n x_i^2)}{\sum_{i=1}^n x_i^2}, \quad (4)$$

де  $x = (x_1, \dots, x_n) \in R^n$ . На рис. 1 зображено функцію  $f(x)$  (4) у двовимірному просторі, тобто при  $n = 2$  для  $x \in [-20, 20]^2$ . Проблема використання еволюційних чи генетичних алгоритмів, зокрема, полягає у визначенні регіону, в якому знаходиться шуканий максимум чи мінімум. Крім того, на основі методу СМА-ES коваріаційна матриця при невдалому початковому значенні не буде збігатися зі значенням, яке дозволяло б концентруватися навколо оптимального значення функції (4).

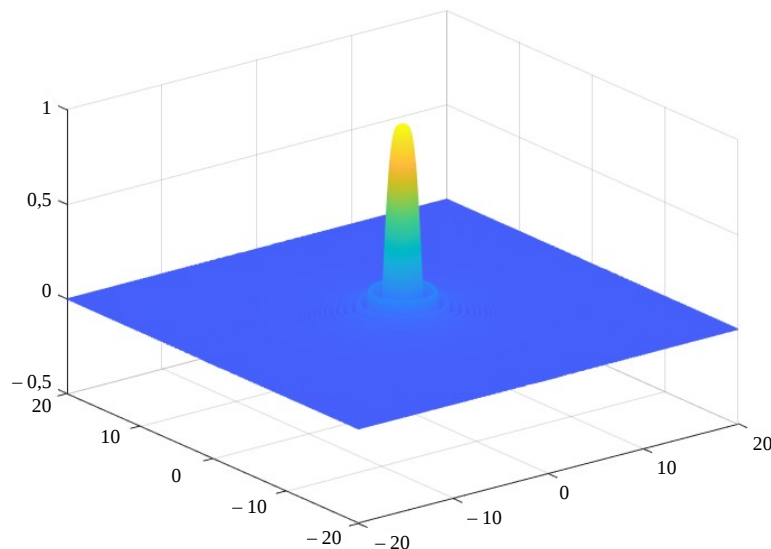


Рис. 1

Як зазначалося вище, у даній роботі проаналізовані деякі класичні генетичні алгоритми та розширений СМА-ES-алгоритм [32]. Основною новизною даного алгоритму є заміна одновікових класичних розподілів СМА-ES на суміші розподілів з невизначеними коефіцієнтами. В роботі [32] наведено покрокову реалізацію СМА-ES-алгоритму, причому оцінку параметрів нормальної суміші пропонується оцінювати, використовуючи ЕМ-алгоритм.

### Аналіз на основі методу Монте–Карло

Розглянемо аналіз декількох класичних еволюційних алгоритмів пошуку оптимального значення для відомих функцій і порівняємо отримані результати з відповідними результатами для розширеного алгоритму CMA-ES, описаного вище. Як класичні алгоритми використовуватимемо метаевристичні алгоритми оптимізації.

- ALO (Ant Lion Optimization) — це метаевристичний алгоритм, що імітує механізм полювання мурашиних левів у природі. Він має п'ять основних процесів полювання на здобич: установка пасток, ловля мурах, очікування, виявлення та затримання здобичі. Математична модель ALO включає такі кроки: ініціалізація, фаза руху, фаза оновлення, завершення [33]. Рух мурашиних левів моделюється рівнянням

$$x_{i,j}^{t+1} = x_{i,j}^t + \frac{rand()}{|D_{i,j}|} \times (x_{best,j}^t - x_{i,j}^t),$$

де  $x_{i,j}^t$  — положення  $i$ -го мурашиного лева в  $j$ -му вимірі в момент часу  $t$ ,  $x_{best,j}^t$  — найкраща позиція рою в  $j$ -му вимірі,  $D_{i,j}$  — діапазон  $j$ -го виміру,  $rand()$  — випадкове число від 0 до 1. Алгоритм повторюється доти, доки не буде виконано критерій зупинки.

- ABCO (Artificial Bee Colony Optimization) — алгоритм, заснований на поведінці медоносних бджіл, які шукають джерело їжі та повідомляють про її місцезнаходження іншим бджолам. Алгоритм використовує набір правил для моделювання поведінки медоносних бджіл. Модель складається з трьох основних компонентів: зайнятих, безробітних бджіл-годувальниць та джерел їжі [34]. Алгоритм ABCO включає такі етапи: ініціалізація, фаза працевлаштування бджіл, фаза бджіл-спостерігачів, фаза бджіл-розвідників, завершення. Бджола оновлює свій стан на основі рівняння

$$x_{i,j}^{t+1} = x_{i,j}^t + \phi_{ij} \times (x_{i,j}^t - x_{k,j}^t),$$

де  $x_{i,j}^t$  — положення  $i$ -ї бджоли-глядача в  $j$ -му вимірі в момент часу  $t$ ,  $\phi_{ij}$  — випадкове число між  $-1$  і  $1$ , а  $x_{k,j}^t$  — положення випадково вибраної бджоли, що працює, відмінної від  $i$ , в  $j$ -му вимірі в момент часу  $t$ . Алгоритм завершує роботу при виконанні критерію зупинки, наприклад, при досягненні максимальної кількості ітерацій або пошуку задовільного рішення.

- GA (Genetic Algorithm) — алгоритм, заснований на принципах генетики та еволюції, використовує набір правил для моделювання процесу природного відбору, який підтримує популяцію потенційних рішень (окремих особин) та розвиває їх протягом поколінь. Оператори відбору, схрещування та мутації використовуються для відтворення та генетичного розмаїття популяції  $P$ , що складається з  $N$  особин, кожна з яких є потенційним розв'язком задачі оптимізації. Населення можна уявити як  $P = \{x_1, x_2, \dots, x_N\}$ -представлення  $x_i$  особи в популяції, що часто кодується як хромосома. Надалі особини відбираються для відтворення з урахуванням їх показників пристосованості, і алгоритм повторюється доти, доки не буде виконано критерій зупинки [35].

- PSO (Particle Swarm Optimization) — це алгоритм ройового інтелекту, який моделює соціальну поведінку зграї птахів чи зграйної риби. Алгоритм починаєть-

ся із сукупності частинок, і кожна частинка — це потенційний розв'язок проблеми оптимізації. Частинки рухаються через простір пошуку залежно від власного кращого становища і кращого становища рою. Алгоритм має таку математичну модель: ініціалізація, оцінка, оновлення швидкості, оновлення позиції, завершення. Оновлення швидкості моделюється рівнянням

$$v_{i,j}^{t+1} = w \times v_{i,j}^t + c_1 \times rand() \times (pbest_{i,j} - x_{i,j}^t) + c_2 \times rand() \times (gbest_j - x_{i,j}^t),$$

де  $v_{i,j}^t$  — швидкість  $i$ -ї частки в  $j$ -му вимірі в момент часу  $t$ ,  $w$  — інерційна вага,  $c_1$  і  $c_2$  — коефіцієнти прискорення,  $rand()$  — випадкове число від 0 до 1,  $pbest_{i,j}$  — найкраще положення  $i$ -ї частки у  $j$ -му вимірі, а  $gbest_j$  — найкраща позиція рою в  $j$ -му вимірі. Алгоритм завершує роботу при виконанні критерію зупинки [36].

• СМА-ES — алгоритм з особливим видом стратегії чисельної оптимізації, що належить до сімейства стратегій еволюції (ES). Це стохастичний алгоритм оптимізації без похідних, який широко застосовується до неопуклих і нелінійних завдань оптимізації, особливо у безперервній галузі. Алгоритм має таку математичну модель: ініціалізація; оцінка рішень; вибір рішень-кандидатів; адаптація коваріаційної матриці обраних рішень-кандидатів; генерація нових рішень-кандидатів, що генеруються з адаптованого пошукового розподілу; завершення [37]. Процес адаптації моделюється рівнянням

$$C^{t+1} = (1 - c_1 - c_\mu)C^t + c_1 \times p_c \times (y^t - \bar{y}^t)(y^t - \bar{y}^t)^T + c_\mu \times \sum_{i=1}^{\mu} w_i^t (y_i^t - \bar{y}^t)(y_i^t - \bar{y}^t)^T,$$

де  $C^t$  — коваріаційна матриця в момент часу  $t$ ,  $c_1$  і  $c_\mu$  — швидкість навчання,  $p_c$  — параметр оновлення першого рангу,  $y^t$  — найкращий кандидат у момент часу  $t$ ,  $\bar{y}^t$  — середнє значення вибраних рішень-кандидатів у момент часу  $t$ ,  $\mu$  — кількість обраних рішень-кандидатів,  $w_i^t$  — вага  $i$ -го обраного кандидата в момент часу  $t$ .

Основна відмінність між алгоритмами ALO, ABCO, GA, PSO та СМА-ES полягає в тому, як вони моделюють поведінку тварин або природні процеси та способи пошуку оптимального рішення. Ще одна відмінність полягає в тому, що у PSO немає генетичних операторів, таких як схрещування та мутація, наявних у GA та СМА-ES. Кожен алгоритм має свої сильні і слабкі сторони, і вибір алгоритму залежить від конкретної задачі оптимізації.

Для аналізу використаємо метод Монте-Карло з  $I = 10^4$  симуляціями для наступних функцій [38]:

$$f_{T2}(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \quad x \in R^2,$$

$$f_{T3}(x) = \sum_{i=1}^3 (100(x_{i+1} - x_i^2)^2 - (1 - x_i)^2), \quad x \in R^4,$$

$$f_{T4}(x) = I_{\{A_1\}}(g(x_1) + h(x_1) + j(x_1)) + (1 - I_{\{A_1\}})x_2^2 + \\ + 1000I_{\{A_2\}}(g(x_2) + h(x_2) + j(x_2)), \quad x \in R^2,$$

де  $I_{\{A\}}$  — індикатор події  $A$ , множини  $A_1$  та  $A_2$  задаються співвідношенням

$$A_i = \left\{ 0, 2 \left\lceil \frac{x_i}{0,2} \right\rceil - 0,5 < x_i < 0, 2 \left\lceil \frac{x_i}{0,2} \right\rceil + 0,5 \right\},$$

функції  $g, h, j$  визначаються співвідношеннями

$$g(x_i) = 0,15(0,2 \left\lceil \frac{x_i}{0,2} \right\rceil - 0,05)^2 I_{\left\{ \left\lceil \frac{x_i}{0,2} \right\rceil > 0 \right\}},$$

$$h(x_i) = 0,15(0,2 \left\lceil \frac{x_i}{0,2} \right\rceil + 0,05)^2 I_{\left\{ \left\lceil \frac{x_i}{0,2} \right\rceil < 0 \right\}}, \quad j(x_i) = x_i^2 I_{\left\{ \left\lceil \frac{x_i}{0,2} \right\rceil = 0 \right\}},$$

$$f_{T5}(x) = \sum_{t=1}^{20} (y_t - x_1 - x_2 o_{t2} - x_2^2 o_{t3})^2,$$

де  $y_t, o_{t1}, o_{t2}, o_{t3}$  — фінансові показники, розглянуті в [37],

$$f_{T6}(x) = \sum_{t=1}^T \{y_t \log(F(x_t, b)) + (1 - y_t) \log(1 - F(x_t, b))\},$$

де  $F(x_t, b)$  визначає емпіричну функцію розподілу, що залежить від невідомого параметра  $b \in \{0, 1\}$ ,

$$f_{T7}(x) = \sum_{t=1}^n (2y_i - 1) \operatorname{sgn}(x_i b).$$

Основним показником швидкодії алгоритму у даному дослідженні буде кількість звернень до функції, для якої шукаємо максимум. Надалі СМА1, СМА2, СМА5, СМА10 — результати для розширеного СМА-ES-алгоритму з 1, 2, 5 та 10 піками відповідно.

Проведемо спочатку аналіз функції (4), перевагою дослідження якої є залежність від розмірності простору пошуку оптимального значення  $x \in R^n$ . Згадані вище генетичні алгоритми програмно реалізовані за допомогою мови Python з використанням модуля СМА. Результати моделювання на основі методу Монте-Карло наведено на рис. 2. Як видно з рис. 2, з ростом кількості піків для розширеного СМА-ES-алгоритму кількість звернень до цільової функції  $f_{T1}$  і мінімальна кількість звернень при всіх значеннях розмірності простору  $n$  для 5 та 10 піків у розширеному СМА-ES-алгоритмі спадають. Крім того, для малої розмірності  $n$  вибір великої кількості піків недоцільний.

Середні значення кількості звернень до функції  $f_{T1}$  із середньоквадратичними відхиленнями (в дужках) при  $I = 10^4$  симуляцій наведені в табл. 1, з якої видно, що середньоквадратичне відхилення значно нижче для СМА5 та СМА10 для великих значень розмірності  $n$ , це підтверджує ефективність розширеного алгоритму.

Проаналізуємо також відповідні значення функцій  $f_{T2} - f_{T7}$ , представлених у табл. 2 (середні значення кількості звернень до функцій  $f_{T2} - f_{T7}$  з середньоквадратичними відхиленнями (в дужках) при  $I = 10^4$  симуляцій). Як бачимо, зберігається тенденція вибору кількості піків для розширеного СМА-ES-алгоритму; для задач більшої розмірності кількість піків повинна бути більшою. Дані власності свідчать про існування взаємозалежності між розмірністю початкової задачі та кількістю вибраних піків.

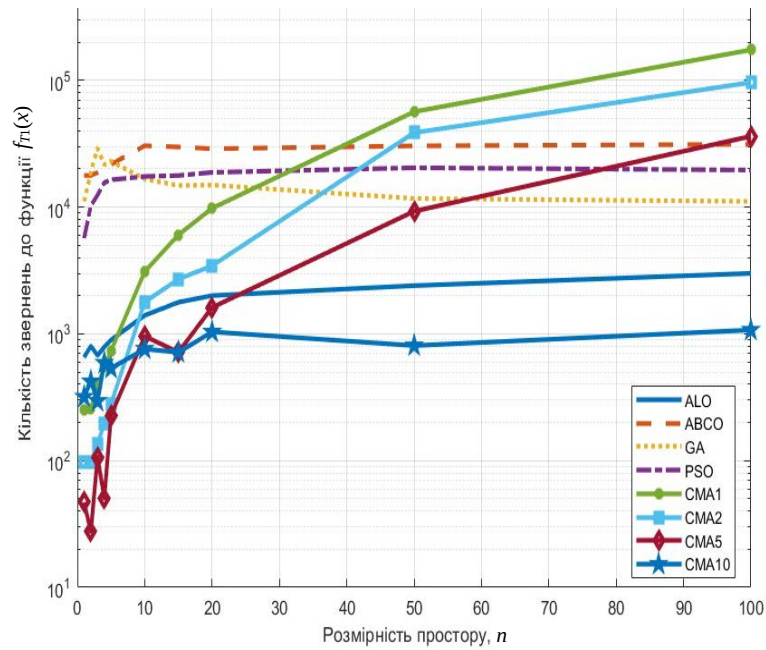


Рис. 2

Таблиця 1

| $n$ | ALO                | ABCO                  | GA                     | PSO                  | CMA1                    | CMA2                 | CMA5                 | CMA10              |
|-----|--------------------|-----------------------|------------------------|----------------------|-------------------------|----------------------|----------------------|--------------------|
| 1   | 653,29<br>(40,29)  | 17760,57<br>(4,86)    | 11053,35<br>(407,69)   | 5670<br>(289,13)     | 253,434<br>(23)         | 97,85<br>(28,4)      | 46,9<br>(31,9)       | 317,69<br>(54,1)   |
| 2   | 801,12<br>(43,55)  | 17770,57<br>(174,57)  | 18812,8<br>(3558,68)   | 10260<br>(2740,64)   | 255,714<br>(23,05)      | 98,28<br>(35,1)      | 27,5<br>(33,7)       | 420,36<br>(61,9)   |
| 3   | 676,17<br>(44,08)  | 18711,64<br>(425,82)  | 29092,2<br>(10125,89)  | 12445,0<br>(4250,23) | 390,201<br>(31,48)      | 135,15<br>(38,4)     | 106,77<br>(40,9)     | 294,2<br>(72,3)    |
| 4   | 788,62<br>(48,28)  | 19875,02<br>(548,51)  | 21582,25<br>(6503,63)  | 15577,5<br>(5007,7)  | 545,424<br>(64,09)      | 196,57<br>(52,4)     | 50,47<br>(51,8)      | 588,93<br>(73,4)   |
| 5   | 895,02<br>(54,92)  | 21724,72<br>(1064,06) | 23019,37<br>(7686,432) | 16417,5<br>(5594,08) | 735,248<br>(218,27)     | 280,97<br>(73,6)     | 227,32<br>(84,3)     | 535,98<br>(93,6)   |
| 10  | 1398,5<br>(152,4)  | 30484,31<br>(7233,62) | 16607,22<br>(4432,16)  | 17452,5<br>(8794,08) | 3099,11<br>(575,27)     | 1783,78<br>(129,9)   | 956,94<br>(145,7)    | 759,67<br>(98,4)   |
| 15  | 1775,5<br>(403,29) | 29811,17<br>(9955,86) | 14805,83<br>(4016,91)  | 17740,0<br>(8116,33) | 6035,376<br>(878,96)    | 2697,28<br>(154,5)   | 716,63<br>(184,1)    | 714,67<br>(129,3)  |
| 20  | 2004<br>(611,56)   | 28858,22<br>(5285,7)  | 14930,58<br>(4204,75)  | 18782,5<br>(6876,74) | 9791,484<br>(1542,77)   | 3449,49<br>(430,87)  | 1614,28<br>(287,5)   | 1036,53<br>(256,8) |
| 50  | 2400,5<br>(705,3)  | 30397,71<br>(8156,89) | 11687,08<br>(1231,79)  | 20420,0<br>(8866)    | 56514,825<br>(12220,98) | 38626,37<br>(1002,4) | 9267,20<br>(503,4)   | 807,64<br>(383,7)  |
| 100 | 3000,8<br>(806,23) | 31289,91<br>(9665,79) | 11083,29<br>(608,27)   | 19605<br>(9676,14)   | 173977,32<br>(76227,7)  | 96187,69<br>(1985,4) | 36216,10<br>(1498,1) | 1072,23<br>(789,3) |

Таблиця 2

| $f$      | ALO                | ABCO                  | GA                     | PSO                   | CMA1                | CMA2               | CMA5               | CMA10             |
|----------|--------------------|-----------------------|------------------------|-----------------------|---------------------|--------------------|--------------------|-------------------|
| $f_{T2}$ | 1053,2<br>(321,24) | 26868,59<br>(9304,66) | 16740,95<br>(5551,51)  | 11557,5<br>(4412,91)  | 587,4<br>(87,9)     | 478,1<br>(83,1)    | 543,1<br>(93,4)    | 567,4<br>(132,1)  |
| $f_{T3}$ | 2014,3<br>(615,24) | 22473,09<br>(4393,99) | 27562,27<br>(13830,33) | 14448,25<br>(6771,35) | 11933,4<br>(148,07) | 8560,23<br>(200,2) | 7934,23<br>(200,2) | 9678<br>(298,45)  |
| $f_{T4}$ | 749,875<br>(57)    | 17802,31<br>(5,74)    | 12287,37<br>(740,11)   | 7875,25<br>(1620,49)  | 135,73<br>(32,44)   | 111,34<br>(43,12)  | 189,86<br>(86,31)  | 231,81<br>(92,84) |
| $f_{T5}$ | 1290,7<br>(462,86) | 38657,57<br>(7813,46) | 25345,70<br>(8129,27)  | 11658,5<br>(4513,03)  | 519,49<br>(42,84)   | 456,11<br>(44,29)  | 387,83<br>(51,7)   | 322,4<br>(53,11)  |
| $f_{T6}$ | 575<br>(3,78)      | 17850,4<br>(3,52)     | 10729<br>(4,56)        | 5500<br>(7,95)        | 20,46<br>(12,63)    | 24,19<br>(22,84)   | 20,32<br>(13,65)   | 19,28<br>(14,56)  |
| $f_{T7}$ | 575<br>(2,1)       | 17850,6<br>(4,22)     | 24631,14<br>(7196,34)  | 5500<br>(9,67)        | 20,79<br>(12,83)    | 21,75<br>(18,12)   | 19,4<br>(22,31)    | 18,32<br>(25,63)  |

## Висновок

Проведено порівняльний аналіз метаевристичних алгоритмів оптимізації оцінки параметрів складних систем. Запропоновано розширений алгоритм стратегії еволюційної адаптації на основі коваріаційної матриці для оцінки параметрів складних систем. Розширений CMA-ES базується на основі використання сумішної розподілів з невизначеною величиною розмірності суміші та з відомим базовим розподілом. Дослідження показують, що з ростом кількості піків для розширеного CMA-ES-алгоритму кількість звернень до цільової функції спадає, що і підтверджує ефективність запропонованого розширеного алгоритму CMA-ES. Однак для малої розмірності  $n$  вибір великої кількості піків недоцільний. Дані властивості свідчать про існування взаємозалежності між розмірністю початкової задачі та кількістю вибраних піків.

*Y. Litvinchuk*

## ON A GENERALIZATION OF EVOLUTIONARY ALGORITHMS

**Yuliia Litvinchuk**

Yuriy Fedkovych Chernivtsi National University

orcid id: 0000-0002-5277-0109

*julilitvintchuk@gmail.com*

In the real world, there are many optimization scenarios with a large number of constraints and objective functions that are discontinuous, nonlinear, nonconvex, and multimodal in nature. In addition, optimization problems are multidimensional with mixed variable types such as integer, real, discrete, binary, and have a different range of values, which requires normalization. Therefore, the problem search space may not always be smooth. Thus, evolutionary algorithms began to be used and developed in computational processes and solving complex engineering tasks, as well as in artificial intelligence methods. The proposed work considers a set of methods for estimating the optimal parameters of training functions using evolutionary and genetic search algorithms in comparison with the extended CMA-ES algorithm. The metaheuristic optimization algorithms ALO, ABCO, GA, PSO and classical CMA-ES, which are based on the behavior of living organisms in real natural environments, are analyzed. The Monte Carlo method was used for the analysis, which makes it possible to draw conclusions about the distribution of the number of calculations of the objective function. The work also proposes an extension of the CMA-ES algorithm using mixtures of normal distributions with an undefined dimension of the mixture and with a known base distribution for estimating optimal values of known test functions. The developed algorithm is built according to the assumption of multip peaked distribution of parameters of complex systems, including hyperparameters of neural networks and parameters of stochastic differential equations. Research shows that with the increase in the number of peaks for the extended CMA-ES algorithm, the number of calls to the objective function decreases and thus confirms the effectiveness of the proposed extended CMA-ES algorithm. However, for low dimensionality,  $n$  choosing a large number of peaks is impractical. These properties suggest the existence of interdependence between the dimension of the initial problem and the number of selected peaks.

**Keywords:** mixture distribution, hyperparameters optimization, genetic algorithms, CMA-ES algorithm, Monte–Carlo method, stochastic differential equations.

## ПОСИЛАННЯ

1. Hamster Ch., van Heijster P. Waves in a stochastic cell motility model. *Bulletin of Mathematical Biology*. 2023. Vol. 85. DOI: <https://doi.org/10.1007/s11538-023-01164-1>
2. Vastola J., Holmes W. Chemical langevin equation: A path-integral view of Gillespies derivation. *Physical Review E*. 2020. Vol. 101. DOI: <https://doi.org/10.1103/PhysRevE.101.032417>
3. Hansen N., Müller S., Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*. 2003. Vol. 11. P. 1–18. DOI: <https://doi.org/10.1162/106365603321828970>
4. Loshchilov I., Hutter F. CMA-ES for hyperparameter optimization of deep neuralnetworks. 2016. arXiv:1604.07269v1 [cs.NE]. Vol. 25. 9 p. DOI: <https://doi.org/10.48550/arXiv.1604.07269>
5. Hansen N., Ros R., Mauny N., Schoenauer M., Auger A. *Impacts of Invariance in Search: When CMA-ES and PSO Face Ill-conditioned and non-separable problems*. *Applied Soft Computing*. 2011. Vol. 11. <https://doi.org/10.1016/j.asoc.2011.03.001>
6. Bharadiya J. Convolutional neural networks for image classification. *International Journal of Innovative Research in Science Engineering and Technology*. 2023. Vol. 8. 673 p. DOI: <https://doi.org/10.5281/zenodo.7952031>
7. Voß Th., Hansen N., Igel Ch. Improved step size adaptation for the MO-CMA-ES. *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. 2010. DOI: <https://doi.org/10.1145/1830483.1830573>
8. Loshchilov I. A computationally efficient limited memory CMA-ES for large scale optimization. *Proceeding of the sixteenth annual conference on Genetic and Evolutionary Computation Conference (GECCO2014)*. 2014. DOI: <https://doi.org/10.1145/2576768.2598294>
9. Arabas J., Jagodzinski D. Towards a matrix-free covariance matrix adaptation evolution strategy. *IEEE Transactions on Evolutionary Computation*. 2019. DOI: <https://doi.org/10.1109/tevc.2019.2907266>
10. Liang Y., Wang X., Zhao H., Han T., Wei Z., Li Y. A covariance matrix adaptation evolution strategy variant and its engineering application. *Applied Softcomputing*. 2019. Vol. 83. 105680 p. DOI: <https://doi.org/10.1016/j.asoc.2019.105680>
11. Sakamoto N., Akimoto Y. Modified box constraint handling for the covariance matrix adaptation evolution strategy. *Proceedings of the genetic and evolutionary computation conference companion on — GECCO 2017*. DOI: <https://doi.org/10.1145/3067695.3075986>
12. Dharmesh Dabhi, Kartik Pandya, Joao Soares, Fernando Lezama, Zita Vale. Cross entropy covariance matrix adaptation evolution strategy for Solving the Bi-Level Bidding Optimization Problem in Local Energy Markets. *Energies*. 2022. Vol. 15, N 13. 4838 p. DOI: <https://doi.org/10.3390/en15134838>
13. Ahrari A., Deb K., Preuss M. Multimodal Optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations. *Evolutionary Computation*. 2017. Vol. 25, N 3. P. 439–471. DOI: [https://doi.org/10.1162/evco\\_a\\_00182](https://doi.org/10.1162/evco_a_00182)
14. Dang V.-H., Vien N.A., Chung T. A covariance matrix adaptation evolution strategy in reproducing kernel Hilbert space. *Genetic Programming and Evolvable Machines*. 2019. DOI: <https://doi.org/10.1007/s10710-019-09357-1>
15. Gillespie D.T. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 2001. Vol. 115, N 4. P. 1716–1733. DOI: <https://doi.org/10.1063/1.1378322>
16. Peterson A.C., Kelley W.G. The theory of differential equations. 2010. DOI: <https://doi.org/10.1007/978-1-4419-5783-2>
17. Petrovskii S., Sekerci Y. Mathematical modelling of plankton-oxygen dynamics under the climate change. *Bull Math Biol*. 2015. Vol. 77, N 11. P. 2325–2353. DOI: <https://doi.org/10.1007/s11538-015-0126-0>
18. Thiruthummal A., Anand & Kim Eun-Jin. Monte–Carlo simulation of stochastic differential equation to study information geometry. *Entropy*. 2022. Vol. 24. 1113 p. DOI: <https://doi.org/10.3390/e24081113>
19. Abdulle Assyr, Blumenthal Adrian. Stabilized multilevel Monte–Carlo method for stiff stochastic differential equations. *Journal of Computational Physics*. 2013. Vol. 251. P. 445–460. DOI: <https://doi.org/10.1016/j.jcp.2013.05.039>
20. Brand J., Yang M., Pahl E. Stochastic differential equation approach to understanding the population control bias in full configuration interaction quantum Monte–Carlo. *Physical Review B*. 2022. Vol. 105. DOI: <https://doi.org/10.1103/PhysRevB.105.235144>

21. Doumbia M., Oudjane N., Warin X. Unbiased Monte–Carlo estimate of stochastic differential equations expectations. *ESAIM: Probability and Statistics*. 2017. DOI: <https://doi.org/10.1051/ps/2017001>
22. Spigler R. Monte–Carlo-type simulation for solving stochastic ordinary differential equations. *Mathematics and Computers in Simulation*. 1987. Vol. 29. P. 243–251. DOI: [https://doi.org/10.1016/0378-4754\(87\)90134-0](https://doi.org/10.1016/0378-4754(87)90134-0)
23. Appleby J., Mao X. Stochastic stabilisation of functional differential equations. *Systems & Control Letters*. 2005. Vol. 54. P. 1069–1081. DOI: <https://doi.org/10.1016/j.sysconle.2005.03.003>
24. Browning A., Warne D. Burrage K., Baker R., Simpson M. Identifiability analysis for stochastic differential equation models in systems biology. *Journal of The Royal Society Interface*. 2020. Vol. 17. 20200652 p. DOI: <https://doi.org/10.1098/rsif.2020.0652>
25. Kelly D. Melbourne I. Smooth approximation of stochastic differential equations. *The Annals of Probability*. 2014. DOI: <https://doi.org/10.1214/14-AOP979>
26. Shen Jun, Lu Kening. Wong–Zakai approximations and center manifolds of stochastic differential equations. *Journal of Differential Equations*. 2017. Vol. 263. DOI: <https://doi.org/10.1016/j.jde.2017.06.005>
27. Tsarkov Y.F., Yasinsky V.K., Malyk I.V. Stability in impulsive systems with Markov perturbations in averaging scheme. 2. Averaging principle for impulsive Markov systems and stability analysis based on averaged equations. *Cybernetics and Systems Analysis*. 2011. Vol. 47, N 1. P. 44–54. DOI: [10.1007/s10559-011-9288-4](https://doi.org/10.1007/s10559-011-9288-4)
28. Priyadarshini Ishaani, Cotton Chase. A novel LSTM-CNN-grid search-based deep neural network for sentiment analysis. *The Journal of Supercomputing*. 2021. Vol. 77. DOI: <https://doi.org/10.1007/s11227-021-03838-w>
29. Wang Linnan, Zhao Yiyang, Jinnai Yuu, Tian Yuandong, Fonseca Rodrigo. Neural architecture search using deep neural networks and Monte–Carlo tree search. *Proceedings of the AAAI conference on artificial intelligence*. 2020. Vol. 34. P. 9983–9991. DOI: <https://doi.org/10.1609/aaai.v34i06.6554>
30. Sun Yuting, Ding Shifei, Zhang Zichen, Jia Weikuan. An improved grid search algorithm to optimize SVR for prediction. *Softcomputing*. 2021. Vol. 25. DOI: <https://doi.org/10.1007/s00500-020-05560-w>
31. Xue Yu, Wang Yankang, Liang Jiayu. A self-adaptive gradient descent search algorithm for fully-connected neural networks. *Neurocomputing*. 2022. Vol. 478. DOI: <https://doi.org/10.1016/j.neucom.2022.01.001>
32. Літвінчук Ю.А., Малик І.В. Розширений алгоритм стратегії еволюції адаптації ко-варіаційної матриці. *Буковинський математичний журнал*. 2023. Vol. 10, N 2. P. 137–143. DOI: <https://doi.org/10.31861/bmj2022.02.09>
33. Roeva O., Zoteva D., Roeva G., Lyubenova V. An efficient hybrid of an ant lion optimizer and genetic algorithm for a model parameter identification problem. *Mathematics*, 2023. Vol. 11. 1292 p. DOI: <https://doi.org/10.3390/math11061292>
34. Yuce B., Packianather M., Mastrocinque E., Pham D., Lambiase A. Honey bees inspired optimization method: the bees algorithm. *Insects*. 2013. Vol. 4, N 4. P. 646–662. DOI: <https://doi.org/10.3390/insects4040646>
35. Albadr M.A., Tiun S., Ayob M., AL-Dhief F. Genetic algorithm based on natural selection theory for optimization problems. *Symmetry*. 2020. Vol. 12, N 11. 1758 p. DOI: <https://doi.org/10.3390/sym12111758>
36. Xuefeng W., Chen M. Application of mathematical model based on optimization theory and particle swarm algorithm in radar station layout optimization. *Journal of physics: conference series*. 2021. Vol. 1848, N 1. 12087 p. DOI: <https://doi.org/10.1088/1742-6596/1848/1/012087>
37. Hansen N., Auger A. CMA-ES. *Proceedings of the 13th annual conference companion on genetic and evolutionary computation — GECCO 2011*. DOI: <https://doi.org/10.1145/2001858.2002123>
38. Dorsey Robert, Mayer Walter. Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. *Journal of Business & Economic Statistics*. 1995. N 13. P. 53–66. DOI: <https://doi.org/10.1080/07350015.1995.10524579>

Отримано 09.10.2023