УДК 004.8:65.05:681.5 DOI: 10.31673/2786-8362.2024.029215

Бриль В.В., Задонцев Ю.В., д.т.н.

АНАЛІЗ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ ГЛОБАЛЬНОЇ ПОШУКОВОЇ ОПТИМІЗАЦІЇ

Bryl V.V., Zadontsev Y.V. Analysis of evolutionary algorithms of global search optimization. The article provides a detailed analysis of evolutionary algorithms of global search engine optimization, in particular the genetic algorithm (GA) and the modified algorithm for searching a school of fish (FSS). The strengths and weaknesses of the selected algorithms are investigated, their efficiency and adaptability are evaluated. The main scientific result is the development and testing of FSS modifications, which showed increased efficiency for tasks with a large number of variables. It is recommended to use modified algorithms in optimization tasks in various industries where complex target functions are present, in particular in engineering, finance and logistics. The proposed system uses an evolutionary approach to adapt and improve search algorithms, including the use of genetic algorithms to evaluate and select the most relevant results. This allows you to take into account the user's previous experience and generate results that better meet his needs. The introduction of evolutionary algorithms into search engines can improve the quality of results and provide a more personalized user experience.

Keywords: abstract, evolutionary algorithm, search engines

Бриль В.В., Задонцев Ю.В. Аналіз еволюційних алгоритмів глобальної пошукової оптимізації. У статті виконано детальний аналіз еволюційних алгоритмів глобальної пошукової оптимізації, зокрема генетичного алгоритму (GA) та модифікованого алгоритму пошуку косяка риб (FSS). Досліджено сильні та слабкі сторони обраних алгоритмів, оцінено їхню ефективність та адаптивність. Основним науковим результатом є розробка та тестування модифікацій FSS, які показали підвищену ефективність для задач із великою кількістю змінних. Рекомендується використовувати модифіковані алгоритми в задачах оптимізації в різних галузях.

Ключові слова: анотація, еволюційний алгоритм, пошукові системи

Вступ

Проблеми оптимізації поширені в різних областях, таких як інженерія, інформатика, економіка та дослідження операцій. Багато реальних проблем оптимізації включають складні цільові функції з декількома локальними оптимістами, що робить їх складними для вирішення за допомогою традиційних методів оптимізації. Глобальні методи оптимізації пошуку спрямовані на пошук глобального оптимального рішення, ефективно досліджуючи весь простір пошуку, не потрапляючи в пастку локальної оптимізації.

Еволюційні алгоритми (ЕА) виникли як потужний клас методів глобальної пошукової оптимізації, натхненний принципами природної еволюції. Ці алгоритми імітують процеси природного відбору, розмноження і мутації, що спостерігаються в біологічних системах. ЕА працюють на популяції кандидатських рішень, ітеративно застосовуючи такі оператори, як відбір, кросовер та мутація, щоб розвивати кращі рішення протягом наступних поколінь [1].

Сила еволюційних алгоритмів полягає в їх здатності ефективно досліджувати складні пошукові простори, адаптивно направляючи пошук до перспективних регіонів. На відміну від методів на основі градієнтів, які покладаються на похідну інформацію і можуть легко застрягти в локальній optima, ЕА використовують підхід на основі населення і стохастичні оператори, що дозволяє їм уникнути локальної оптимізації і в кінцевому підсумку сходитися до глобального оптимуму або майже оптимального рішення [2].

Аналіз останніх досліджень. Було проведено детальний аналіз еволюційних алгоритмів глобальної пошукової оптимізації, зокрема генетичного алгоритму (GA) та алгоритму пошуку косяка риб (FSS). Автори використаних джерел акцентують увагу на використанні цих алгоритмів для вирішення задач із багатьма локальними оптимумами. Розглянуто методи їхньої адаптації до складних задач, зокрема через оптимізацію параметрів та структури алгоритмів.

У роботі "State of the Art in Global Optimization" [1] автори представили фундаментальні концепції та методи глобальної оптимізації, зокрема детерміновані і евристичні підходи. Вони детально описали класичні методи глобальної оптимізації, як-от метод гілок та меж (B&B). Основною перевагою роботи є огляд основних підходів та встановлення базових понять у глобальній оптимізації. Однак автори не розглянули новіші методи, такі як генетичні алгоритми (GA) або інші стохастичні підходи, що є важливим для подальших досліджень [1].

У "Firefly Algorithm, Stochastic Test Functions and Design Optimization" [3] автор Yang Xin-She детально описує алгоритм світлячків (Firefly Algorithm), який базується на привабливості світлячків і дозволяє уникати локальних мінімумів. Основною перевагою алгоритму є його здатність працювати зі складними та мультимодальними функціями. Однак автор визнає, що алгоритм може не гарантувати збіжності до глобального оптимуму [3].

У роботі "A novel numerical optimization algorithm inspired from weed colonization" [4] запропоновано алгоритм оптимізації на основі колонізації бур'янами (Weed Optimization Algorithm), який ефективно застосовується до задач із розривними або негладкими функціями. Основним досягненням авторів є створення алгоритму, що уникає передчасної збіжності. Однак, питання налаштування параметрів залишається невирішеним, що може призвести до непередбачуваних результатів у деяких задачах [4].

У роботі "An algorithm for the traveling salesman problem" [5] автори запропонували алгоритм для розв'язання задачі комівояжера, який став основою для багатьох сучасних методів оптимізації. Хоча алгоритм ефективно вирішує дискретні задачі, він обмежений у своїй здатності знаходити оптимуми для задач із неперервними і складними пошуковими просторами [5].

Результати огляду наукових робіт свідчать про високу ефективність існуючих еволюційних алгоритмів у задачах оптимізації. Водночас, залишаються недостатньо вирішеними питання адаптації цих алгоритмів до задач із великою кількістю змінних та їх оптимізації для умов мультимодальних цільових функцій. Зокрема, проблемою залишається передчасна збіжність до локальних мінімумів та нестабільність результатів через налаштування параметрів.

Тематика дослідження спрямована на всебічний аналіз еволюційних алгоритмів, зокрема GA та FSS, із метою підвищення їхньої ефективності для задач глобальної пошукової оптимізації. У статті запропоновано модифікації алгоритму FSS, які показали кращі результати для задач із великою кількістю змінних, що відкриває перспективи для подальших досліджень у напрямку оптимізації еволюційних алгоритмів.

Проаналізовані роботи пропонують ефективні методи оптимізації для різних класів задач, проте існують певні обмеження та невирішені питання, такі як: передчасна збіжність, нестабільність результатів через параметри алгоритмів та відсутність гарантій збіжності до глобального оптимуму. Ці питання залишають простір для подальших досліджень і вдосконалень.

Метою роботи є всебічний аналіз та оптимізація алгоритмів для ефективного розв'язання різноманітних задач при цьому основними цілями є визначення часової та просторової складності алгоритмів їх масштабованості та можливостей для паралельного виконання оцінка точних значень часу виконання та використання пам'яті на конкретних наборах даних розробка нових ефективних алгоритмів та вдосконалення існуючих для вирішення складних NP-повних задач.

Методи дослідження – аналіз часової складності, який передбачає оцінку часу виконання алгоритму в залежності від розміру вхідних даних.

Також метою є детальний аналіз різного роду еволюційних алгоритмів, плани їх роботи і базові принципи. Були детально розглянуті слабкі і сильні сторони. На базі аналізу обрано два найкращих алгоритми, котрі були обрані задля вирішення завдання глобальної пошукової оптимізації.

Виклад основного матеріалу дослідження.

Глобальна оптимізація особливо важлива при роботі зі складними, неопуклими, мультимодальними або розривними цільовими функціями, де може існувати кілька локальних оптимумів. У таких випадках традиційні методи оптимізації, які покладаються на локальний пошук або градієнтні методи, можуть потрапити в пастку локальної оптимізації, не знайшовши справжнього глобального оптимуму.

Мета глобальної оптимізації полягає в тому, щоб ретельно вивчити весь пошуковий простір, не застрягаючи в локальній оптимізації, і в кінцевому підсумку визначити глобальне оптимальне рішення або набір майже оптимальних рішень [5].

Оптимізація є процесом знаходження точки, яка мінімізує функцію. У контексті задач оптимізації терміни «глобальний локальний мінімум» і «глобальний мінімум» відносяться до різних понять:

Глобальний локальний мінімум - це точка в просторі пошуку, де цільова функція досягає локального мінімального значення, але це не обов'язково найменше значення по всьому простору пошуку. Іншими словами, глобальний локальний мінімум є найнижчою точкою в певному районі або області простору пошуку, але можуть існувати й інші регіони з ще нижчими значеннями функцій.

Глобальний мінімум, з іншого боку, є найменшим можливим значенням цільової функції над усім пошуковим простором. Він являє собою найкраще рішення, яке оптимізує цільову функцію в усьому світі, не обмежуючись будь-яким місцевим регіоном або районом.

Система у області глобальної оптимізації розрізняє наступні класи:

- детерміновані;
- евристичні.
- стохастичні;

Глобальна оптимізація знаходить своє "покликання" у моделюванні хімічних процесів, фінансовому прогнозуванні, Big Data аналізі, створенні лазерного обладнання, оцінці ефективності роботи підприємства, в обчислювальній фізиці та у вирішенні систем нелінійних рівнянь.

У задачах оптимізації кінцевою метою часто є пошук глобального мінімуму, оскільки він являє собою справжнє оптимальне рішення. Однак через складність пошукового простору або наявність декількох локальних мінімумів може бути складно знайти глобальний мінімум за допомогою методів локального пошуку або методів на основі градієнта [6].

Глобальні алгоритми оптимізації, такі як генетичні алгоритми, імітація відпалу або методи, пов'язані з гілками, призначені для більш ретельного вивчення простору пошуку та уникнення потрапляння в глобальні локальні мінімуми. Ці алгоритми використовують стратегії, щоб уникнути локальної оптимізації та продовжувати пошук, поки не буде знайдено глобального мінімуму або задовільного наближення. Варто зазначити, що в деяких випадках знаходження точного глобального мінімуму може бути обчислювально нездійсненним або непотрібним, а врегулювання високоякісного глобального локального мінімуму може бути практичним компромісом, особливо у великомасштабних задачах [7].

Результати дослідження. Результати цього дослідження мають значний потенціал для практичного застосування в різноманітних сферах, де виникають складні оптимізаційні задачі. Розроблена система може бути адаптована для вирішення реальних проблем в інженерії, фінансах, логістиці та інших галузях, де потрібно знаходити оптимальні рішення в умовах багатовимірності та нелінійності. Особливо цінним є порівняльний аналіз алгоритмів GA та FSS, який надає користувачам можливість вибору найбільш підходящого методу залежно від специфіки конкретної задачі.

Основна задача пошуку max функції $f(x)$ для допустимої множини $X \subseteq R^n$ зводиться до пошуку точки $x \in X$, що $f(x) \leq f(x)\left(f\left(x_{.}\right) \geq f(x)\right)$ для всіх $x \in X$. Моделюємо ситуацію, де стоїть задача мінімізації функції f . Лімітування, котрі базуються на обчислювальній похибці,

Таблиця 1

Порівняльна таблиця алгоритмів оптимізації [8]

Алгоритм	Переваги	Недоліки
Алгоритм косяка риб (FSS)	• Не потрібна градієнтна інформація: алгоритм FSS не вимагає градієнтної інформації або похідних обчислень, що робить його придатним для задач оптимізації з недиференційованими або розривними цільовими функціями.	• Передчасна збіжність: Незважаючи на заходи, вжиті для підтримки різноманітності, алгоритм FSS все ще може страждати від передчасної збіжності, особливо в складних задачах оптимізації з багатьма локальними оптимістами.
Алгоритм кажанів (Bat-inspired algorithm)	• Потенціал розпаралелювання: Оцінка кандидатських рішень в алгоритмі BA може бути розпаралелена, що потенційно призводить до більш швидкої збіжності на паралельних обчислювальних системах.	• Передчасна збіжність: Як і багато метаевристичних алгоритмів, алгоритм BA може страждати від передчасної збіжності, особливо в задачах з багатьма локальними оптимальних рішень.
Алгоритм Firefly (світлячків)	• Автоматичний поділ: алгоритм автоматично поділяє населення на підгрупи, що дозволяє ефективно досліджувати простір пошуку та зменшує ймовірність потрапляння в локальну оптимізаці..	• Відсутність теоретичних гарантій: Алгоритм FA не забезпечує теоретичних гарантій збіжності до глобального оптимального рішення, особливо для непуклих.
Генетичний алгоритм (GA)	• Потенціал глобальної оптимізації: ГА має можливість уникнути локальної оптимізації, підтримуючи різноманітне населення рішень, збільшуючи шанси на пошук глобальної оптимізації.	• Ландшафтна чутливість: Продуктивність залежить від характеристик ландшафту придатності, таких як міцність, оманливість або нейтралітет.
Weed Optimization Algorithm	• Стійкість: IWO рідше застряє в локальних мінімумах порівняно з деякими іншими алгоритмами. Це тому, що це дозволяє отримати деяку випадковість і дозволяє уникнути занадто швидкого зближення в одному рішенні.	• Налаштування параметрів: Хоча кількість параметрів є керованою, пошук оптимальних налаштувань для цих параметрів може бути складним завданням і може вимагати певних рішень.

або нестачею технічних ресурсів, як правило, забороняють віднайти математично точне вирішення даної планової задачі. В даній ситуації потрібно прямувати до пошуку приблизного результату, іншими словами, точки з незліченної кількості ε-оптимальних рішень $X^{\varepsilon}_{*} = \left\{ x \in X: f(x) \le f\left(x_{*}\right) + \varepsilon \right\}$. Пошук математично точного результату потрібно вважати як окрему подію пошуку наближеного рішення $\varepsilon = 0$ [7].

Засіб задання множини X надає можливість виокремити досить жорстку класифікацію. Якщо$X = R^n$, то завдання можна класифікувати до класу задач безсумнівної оптимізації. Коли прийнятна множина задана завдяки обмеженням – це умовна оптимізація. Водночас, якщо X задача класифікується до сфери дискретної оптимізації.

Зазвичай виділяють 2 основні класи методів вирішення задач кінцевомірної оптимізації – орієнтовані та математично точні. Математично точні методи надають можливість забезпечувати оптимальність встановленого вирішення. До цієї класифікації співставляють різноманітні варіації методу меж та гілок. Задля математично точних методів властива висока трудомісткість, котра досить часто забороняє застосовувати їх задля подолання реальних задач. Евристичні методи побудовані на гіпотезах про властивості оптимального вирішення. На противагу математично точним рішенням, евристичні методи не забезпечують оптимальність віднайденого вирішення. Проте, в обставинах обмеженості обчислювальних можливостей евристики, як правило, є єдиним варіантом пошуком вирішення задачі. Гібридні та поширені методи, за якими, евристичні методи використовуються задля пошуку вирішення задачі, точні - задля аргументу оптимальності. Результативність гібридних методів спричинена тим, що евристичні алгоритми досить часто мають високу швидкість збіжності до оптимуму у порівняно з точними методами [7].

Найпоширеніший план організації точних методів – метод гілок та меж (МГМ), побудований на розбитті допустимої множини. Для комбінаторної, дискретної та загальної математичної оптимізації використовуються галузеві та зв'язані алгоритми, щоб знайти найкраще рішення. Величезна різноманітність задач оптимізації, таких як планування виробництва та планування екіпажу, класифікуються як NP-Hard, оскільки вони не можуть бути вирішені за поліноміальний час. Алгоритм, пов'язаний з гілкою (B&B) є часто використовуваним методом для отримання точних рішень задач оптимізації NP-hard. Метод, який був вперше розроблений Лендом і Дойгом, часто називають алгоритмом; однак, більш точно сказати, що B&B містить сімейство алгоритмів, які всі мають спільну процедуру первинного рішення [7].

Схема роботи методу: Під час всього періоду роботи підтримується список $\{X_i\}$ підмножин допустимої множини. Він містить один елемент - допустимої множини X. Наступним кроком обирається один з елементів – підмножина X_i. Якщо X_i необхідне характеристикам відсіву, даний об'єкт видаляється зі списку. В іншій ситуації алгоритм підлягає дробленню на менші дрібні підмножини за правилом декомпозиції. Результати підмножини заміщають в переліку обраний елемент. Алгоритм доводить цикл роботи до кінця, коли в послідовності $\{X_i\}$ відсутні елементи [7].

Правилом відсіву – це будь-яка функція $\xi : S \to \{0, 1\}$, котра визначена на деякій множині S підмножин R^n і відповідає кожній підмножині з S число 0 або 1. Якщо $V \in S$ виконується $\xi(V) = 1$, тоді можна стверджувати, що підмножина V відповідає правилам відсіву ξ . У такому разі можна стверджувати, що V не відповідає правилам відсіву ξ.

Нехай задані параметри відсіву $Y = \left\{\xi 1 \ldots \xi_m\right\}$. Кінцевий список підмножин $C = \left\{X_1, \ldots, X_t\right\}$ для будь-якого $j = \{1, \ldots, t\}$ знайдеться хоча б одне $i = \{1, \ldots m\}$ таке, що $\xi_i\left(X_j\right) = 1$, називаємо покриттям безлічі X задля набору правил відсіву Y.

Приведу доказ формулювання часто використовуваних правил відсіву. Характеристика - це призначення $g : S \to R$, встановлене множиною S підмножиною R^n, а для $V \in S$застосовується $g(V) \leq \min\limits_{x \in V \cap X} f(x)$. Рекорд – це найменше значення головної функції, отримане під час активного розв'язання системи. Коли $x_{1, \ldots,} x_k$ - логіка точок, в котрих шукається корінь головної функції, $f_k = f\left(x_r\right) = \min\limits_{i = 1, \ldots k} f\left(x_i\right)$ - рекорд; x_r - рекордне рішення [6]. Правило відсіву ξ_{rbc} представляється наступним чином:

$$\xi_{rbc} = \begin{cases} 1, \ g(V) \geq f_k - \varepsilon \\ 0, \ g(V) < f_k - \varepsilon \end{cases}$$

Правила відсіву мають визначатися з урахуванням наявності покриття і у підсумку повинно виходити так, що рекорд, який ми шукаємо – ε-оптимальне рішення [8].

Рекорд черговості точок $x_{1, \ldots,} x_k$ базується на базі розбиваючих множин. Як приклад, коли множини $\{X_i\}$ є паралелепіпедами - у ролі компонентів послідовності $x_{1, \ldots,} x_k$ беруть їх центри. Є варіант, при якому послідовність точок $x_{1, \ldots,} x_k$ формується самостійним від побудови покриття методом за допомогою варіативних евристичних способів. У ролі евристик використовують різноманітні локальні алгоритми (Ньютона, метод градієнтного спуску), крім того є складніші алгоритми, як приклад, стохастичні та генетичні методи [8].

Найважливіший компонент методу - правило декомпозиції $\Delta : S \to 2^S$, воно є функцією, обраною на множині S підмножин R_n та порівнює підмножини $V \in S$ кінцевий список його підмножин $V_1, \ldots, V_m$, при цьому, $V = U_{i=1}^{m} V_i$, $V_i \cap V_j = \varnothing$ для всіх $1 \le i < j \le m$ [8].

Аналіз цільової ефективності модифікованого алгоритму FSS. Значення розміру формації під час роботи алгоритму FSS продемонстрований у табл. 2.

Таблиця 2

Значення масштабів формації на час роботи алгоритму

Масштаби формації	Термін виконання алгоритму, (с)
5	0,013
10	0,008
25	0,007
35	0,007
50	0,007
75	0,008
100	0,005
200	0,01
500	0,012
1000	0,02

Аналіз ефекту масштабів формації під час виведення результату алгоритму будувалося на базі пошуку екстремуму функції $y = 0,1\left(x^2 + z^2\right) + 10$, найбільш ефективні показники продемонстровані у таблиці 2. З цієї таблиці зрозуміло, що оптимальні масштаби формації – 1 сотня одиниць. Максимальний час отримання результату алгоритму за умови малої формації – 5 одиниць, при великій формації – 1000 і 500 одиниць.

Задля дослідження швидкості пошуку від значення max ваги оператора, визначена функція $y = 3x^2 + xz + 2z^2 - x - 4z$. Результати аналізу продемонстровані у таблиці 3.

Таблиця 3

Значення максимальної ваги оператора на час роботи алгоритму.

Вага оператора	Час виконання алгоритму, (с)
5	0,012
10	0,012
25	0,012
35	0,012
50	0,011
75	0,011
100	0,013
200	0,015
500	0,016
1000	0,014

Таблиця 3 демонструє оптимальну вагу риби – 75.

У таблиці 4 продемонстровано вплив чисельності ітеративних процесів на точність та час завершення алгоритму, задля функції $y = 0,1\left(x^2 + z^2\right) + 10$.

У таблиці 4 продемонстрована точність виконання алгоритму збільшується з чисельністю ітеративних обробок. Час виконання алгоритму також збільшується з чисельністю ітеративних процесів.

Таблиця 4

Значення чисельності ітеративних обробок на точність та час виконання алгоритму

Чисельність ітеративних обробок	Точність виконання алгоритму, (%)	Час виконання алгоритму, (с)
15	99,9412	0,007
25	99,97482	0,045
35	99,987	0,055
50	100	0,056
75	100	0,033
100	100	0,038
200	100	0,045
350	100	0,11
500	100	0,115

Аналіз роботи алгоритму Fish School Search продемонстрували, що результативність алгоритму залежна від визначених параметрів, чим краще параметри підібрані, тим точнішим буде результат та меншим час виконання алгоритму.

Проведення аналізу результативності виконання генетичного алгоритму. Значення масштабу формації під час виконання генетичного алгоритму. Аналіз значення масштабів формації під час роботи алгоритму проводилася на базі пошуку екстремуму функції $y = 0,1(x^2 + z^2) + 10$, найбільш результативні значення продемонстровані у таблиці 5. З цієї таблиці можна побачити, що оптимальний масштаб формації становить 100 особин.

Задля аналізу залежності швидкості пошуку рішень від значення вірогідності симбіозу хромосом була визначена функція $y = 3x^2 + xz + 2z^2 - x - 4z$. Отримані результати продемонстровані у таблиці 5.

Таблиця 5

Значення масштабів формації під час виконання алгоритму

Масштаби формації	Час виконання алгоритму, (с)
5	0,17
10	0,091
25	0,032
35	0,032
50	0,026
75	0,025
100	0,023
200	0,026
500	0,037
1000	0,077

Таблиця 6

Значення вірогідності симбіозу хромосом під час виконання алгоритму

Вірогідність симбіозу, (%)	Час виконання алгоритму, (с)
95	0,051
90	0,061
85	0,046
80	0,046
75	0,053
70	0,09
65	0,098
60	0,165

Таблиця 5 демонструє, що оптимальна вірогідність симбіозу хромосом для визначеної функції - складає 80-85%.

Таблиця 6 демонструє значення чисельності ітеративних обробок на точність та час виконання алгоритму для функції $y = 0,1\left(x^2 + z^2\right) + 10$.

Таблиця 6

Значення чисельності ітеративних обробок на точність та час виконання алгоритму

Чисельність ітеративних обробок	Точність виконання алгоритму, (%)	Час виконання алгоритму, (с)
15	99,9998	0,025
25	100	0,03
35	100	0,054
50	100	0,073
75	100	0,094
100	100	0,121
200	100	0,267
350	100	0,399
500	100	0,584

Таблиця 6 демонструє точність виконання алгоритму і час збільшується з чисельністю ітеративних обробок.

Аналіз роботи генетичного алгоритму демонструють, що результативність алгоритму залежить від заданих параметрів, чим точніше параметри визначені, тим точнішим є результат та меншим час виконання.

Зіставлення параметрів результативності роботи алгоритмів FSS і GA. Задля зіставлення параметрів результативності роботи алгоритму пошуку FSS у рамках виконання задачі глобальної оптимізації було визначено генетичний алгоритм.

Таблиця 7 демонструє результати обчислень значень оптимуму функції Де Джонга $f(x) = \sum_{i=1}^{n} x_i^2,\ i = 1{:}n$, з різною кількістю змінних.

Таблиця 7

Точність отриманого результату визначеними алгоритмами для функції Де Джонга

Чисельність змінних / Точність, (%)	3	4	5
FSS	99,988	99,99	99,993
GA	99,999	100	100

Беручи в увагу дані таблиці 7 не важко дійти до висновку: алгоритми – досить точні у своїх результатах, тому що точність FSS алгоритму складає майже 100%, а у GA - 100%.

Таблиця 8 та 9 демонструє відсутність автономії кількості ітеративних обробок, потрібних задля пошуку оптимуму функцій Розенброка $f(x) = \sum_{i=1}^{n-1} 100\left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2,\ i = 1{:}n-1$ та Растрігіна $f(x) = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos\left(2\pi x_i\right)\right),\ i = 1{:}n$.

Таблиця 8

Чисельність ітеративних обробок задля функції Розенброка з іншою чисельністю змінних

Чисельність змінних / Чисельність ітерацій	3	4	5
FSS	170	250	330
GA	130	370	650

Таблиця 9

Чисельність ітеративних обробок задля пошуку оптимуму функції Растрігіна з різноманітною чисельністю змінних

Чисельність змінних / Чисельність ітерацій	3	4	5
FSS	150	170	200
GA	210	390	735

Згідно таблиць 8 і 9 можемо помітити, що алгоритму FSS потрібно менша кількість ітеративних обробок при пошуку екстремуму багатомірної функції, розмірність більше 3-х змінних.

Таблиця 10 демонструє відсутність автономії часу виконання, потрібного задля пошуку оптимуму функції Швефеля $f(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{x_i}\right), i=1:n$.

Таблиця 10

Час потрібний задля пошуку оптимуму функції Растрігіна з різною чисельністю змінних

Чисельність змінних / Час роботи, (с)	3	4	5
FSS	0,18	0,29	0,404
GA	0,155	0,37	0,64

Таблиця 10 демонструє те, що алгоритм Fish School Search – швидший, в порівняні з GA. При досліджені ЕА задля пошуку оптимуму функцій можна побачити, що ці два алгоритми - універсальні методи задля пошуку екстремуму, незважаючи на складності функцій. Задля знаходження оптимуму простих функцій найкращим чином буде застосовувати класичні методи, вони, у цій ситуації, виконують свою роботу швидше ніж ЕА. Під час вирішення питання глобальної оптимізації для функції зі складним ландшафтом доцільніше використовувати алгоритм FSS, через те, що звичайні алгоритми з високою частотою спричинює некоректний результат, а GA працює дещо повільніше. За допомогою модифікації алгоритму Fish School Search, алгоритм сходиться на функціях з гладкою поверхнею. Алгоритму Fish School Search потрібна менша чисельність ітеративних обробок задля функцій багатьох змінних, на функціях з невеликою чисельністю змінних генетичний алгоритм має перевагу у чисельності ітеративних обробок. Коректність розрахунків алгоритмів - майже 100%. Слід уточнити, під час роботи алгоритму і точність пошуку рішення напряму впливають задані дані.

Висновки

У рамках даної роботи було проведено комплексне дослідження еволюційних алгоритмів глобальної пошукової оптимізації, зокрема генетичного алгоритму (GA) та модифікованого алгоритму пошуку косяка риб (FSS). Робота охоплює теоретичні основи оптимізації, детальний аналіз різних еволюційних алгоритмів, їх практичну реалізацію та експериментальну оцінку ефективності. Експериментальні результати на різних тестових функціях показали високу точність обох алгоритмів, при цьому FSS продемонстрував кращу ефективність для функцій з більшою кількістю змінних. Важливим аспектом роботи стало дослідження впливу параметрів алгоритмів на їх ефективність, що дозволило визначити оптимальні налаштування для різних типів задач. Таким чином, дана робота не лише розширює теоретичне розуміння еволюційних алгоритмів, але й надає практичний інструментарій для їх ефективного використання у вирішенні складних оптимізаційних задач реального світу. Важливо відзначити, що робота не обмежується лише реалізацією відомих алгоритмів, але й включає їх модифікацію та удосконалення. Зокрема, модифікований алгоритм FSS показав підвищену ефективність у порівнянні з класичним варіантом, що відкриває перспективи для подальших досліджень у напрямку оптимізації еволюційних алгоритмів.

Список використаної літератури:

1. Floudas C. A., Pardalos P. M. State of the Art in Global Optimization: Computational Methods and Applications. New York City : Springer; 1996th edition, 2011. 664 p.
2. An Algorithm for the Traveling Salesman Problem / J. D. C. Little et al. 11th ed. FB &c Ltd, Dalton House, 60 Windsor Avenue, London, SW19 2RR : Forgotten Books, 2018. 83 p.
3. Yang X.-S. Firefly algorithm, stochastic test functions and design optimisation. International journal of bio-inspired computation. 2010. Vol. 2, no. 2. P. 78–84. URL: https://doi.org/10.48550/arXiv.1003.1409.
4. Mehrabian A. R., Lucas C. A novel numerical optimization algorithm inspired from weed colonization. Ecological informatics. 2006. Vol. 1, no. 4. P. 355–366. URL: https://doi.org/10.1016/j.ecoinf.2006.07.003.
5. The Expanded Invasive Weed Optimization Metaheuristic for Solving Continuous and Discrete Optimization Problems / H. Josiński et al. The Scientific World Journal. 2014. Vol. 2014. P. 1–14. URL: https://doi.org/10.1155/2014/831691.
6. Yang X.-S. Cuckoo search via Levy flights. in: Proc. of World Congress on Nature & Biologically Inspired Computing. 2009. Vol. 1, no. 1. P. 210-214. URL: https://doi.org/10.48550/arXiv.1003.1594.
7. Ibe O. C. Markov Processes for Stochastic Modeling. 2nd ed. Amsterdam, Netherlands : Elsevier, 2013. 424 p.
8. Robbins H., Monro S. A Stochastic Approximation Method. 22nd ed. Virginia, USA. : Institute of Mathematical Statistics, 1951. 407 p.

Автори статті

Бриль Владислав – студент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0002-9144-8187

Задонцев Юрій – кандидат технічних наук, доцент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0007-2192-4746

Authors of the article

Bryl Vladyslav – student, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0002-9144-8187

Zadontsev Yuriy – Candidate of Science (technic), Associate Professor, State University of Information and CommunicationTechnologies, Kyiv, Ukraine.

ORCID: 0009-0007-2192-4746