

УДК 004.896:519.863

DOI: 10.18372/2073-4751.84.20904

Тюшкевич Д.А.,
orcid.org/0009-0001-5111-1056,
gpost.paris@gmail.com

Гузій М.М., к.т.н.
orcid.org/0000-0003-4807-8862
mykola.huzii@npp.kai.edu.ua

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТАЕВРИСТИЧНИХ АЛГОРИТМІВ РОСВОГО ІНТЕЛЕКТУ ГРУПОВОГО УПРАВЛІННЯ БПЛА

Державний університет «Київський авіаційний інститут»

Вступ

З стрімким розвитком сучасних технологій для безпілотних літальних апаратів, а також активним впровадженням БПЛА у різноманітні сфери діяльності людини можна визначити тенденцію, в ході якої увага розробників та науковців переходить від управління поодинокими апаратами до координації групи БПЛА об'єднаними в одну мережу для з можливістю спільної роботи для вирішення різноманітних задач. Функціонування такої мережі базується на принципі розподілу навантаження між окремими агентами, що дозволяє реалізувати завдання, виконання яких є неможливим або енергетично недоцільним для автономних апаратів через обмежений радіус дії та дефіцит обчислювальних потужностей.

Однак введення такої концепції рою дронів одночасно і формулює певні виклики у побудові системи управління. Таким викликом може бути, наприклад, при виборі підходу у централізованому керуванні ризик критичної затримки передачі даних, що в певних динамічних умовах може загрожувати неможливістю оперативного реагування на загрози.

При централізованій архітектурі мережі вихід з ладу хоча би одного керуючого агента або ж придушення каналу зв'язку спричиняє повну втрату контролю над усією групою дронів [7]. Саме тому, найкращим рішенням є використання децентралізованої,

локальної моделі системи управління, де кожен агент мережі приймає рішення на основі тих даних, якими він володіє [7].

На цьому принципі будуються “роеві” алгоритми функціонування, такі як наприклад вже класичний PSO (Particle swarm optimization) [1] і сучасні метаевристичні: алгоритм яструбів Харріса (ННО) або алгоритм беркута (АО) тощо. Зокрема останні набувають нових обертів у розвитку і пропонують нові принципи адаптивної поведінки. Так, наприклад, відносно новий, представлений в 2021 році African Vultures Optimization Algorithm (AVOA) демонструє високу ефективність у вирішенні складних оптимізаційних задач у реальному часі [2], алгоритм Snake Optimizer (SO) розроблений у 2022 році, має здатність до швидкої адаптації траєкторій у середовищах з екстремальною щільністю перешкод [3] або ж алгоритм ННО завдяки механізму кооперативного полювання дозволяє групі БПЛА гнучко перерозподіляти ролі при виявленні динамічних цілей або зміні польотного завдання [4]. Відносно новий алгоритм Dung Beetle Optimizer (DBO) представлений у 2023 році пропонує високу швидкість збіжності та точність пошуку рішень, і демонструє значну ефективність у задачах глобальної оптимізації [5].

Проте питання найбільш ефективного алгоритму для специфічних умов польоту, наприклад, при необхідності швидкого та

ефективного оминання перешкод, роботі в умовах щільної протидії з боку засобів приглушення зв'язку, або в умовах обмежених обчислювальних здатностей бортових комп'ютерів залишається недостатньо вивченим, що і зумовлює необхідність проведення порівняльного аналізу існуючих алгоритмів для визначення їхніх сильних та слабких сторін у задачах роєвого управління БПЛА.

Аналіз останніх досліджень та публікацій

Сучасний стан наукових досліджень у сфері управління БПЛА характеризується переходом від класичних підходів до складних метаевристичних методів роєвого інтелекту. Такі базові алгоритми як Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) та Firefly algorithm (FA) дозволяють агентам обмінюватися даними про навколишнє середовище, що компенсує обмеженість бортових сенсорів та ресурсів окремих апаратів [8]. Основним аспектом в управлінні роєм є вибір архітектури. Таким чином централізована система управління має суттєві обмеження в масштабуванні та стійкості до збоїв [7]. Проблемою може бути, наприклад, експоненціальне зростання затримки у прийнятті рішень при додаванні нового агента до рою, при цьому в децентралізованих системах це обмеження зняте, при будь-якій кількості дронів у рої затримка залишається стабільно низькою [7]. Це підкреслює необхідність впровадження децентралізованих метаевристичних методів для великих груп БПЛА. Зокрема такі алгоритми як Particle swarm optimization (PSO) та Grey Wolf Optimizer (GWO) показують значні успіхи в задачах динамічного розподілу навантаження, але ці алгоритми працюють на основі випадкового пошуку, їх результати можуть відрізнятися при кожному запуску.

Досвід показує: що точніше алгоритм намагається знайти ціль, то

менш передбачуваними можуть бути його окремі результати, що потребує додаткової перевірки їх надійності [9]. Найсвіжіші дослідження 2023-2025 років зосереджені на плануванні траєкторій у складних 3D-просторах, пропонують, наприклад, інтегровані методи планування місій на основі кластеризації та оптимізаційних алгоритмів для роботи в умовах мінливого навантаження [10] або ж успішно застосовують Self-Organizing Migrating Algorithm (SOMA) для 3D-навігації підкреслюючи важливість ефективного балансу між фазами дослідження (Exploration) та експлуатації (Exploitation) для оминання динамічних перешкод [11].

Незважаючи на досягнуті результати, питання вибору найбільш робастного алгоритму для специфічних умов обмежених обчислювальних потужностей [8] та активної протидії засобам зв'язку [7] залишається недостатньо вивченим, що визначає актуальність подальших порівняльних досліджень.

Мета дослідження

Мета даної статті полягає у проведенні системного порівняльного аналізу та кількісній оцінці щодо ефективності сучасних метаевристичних алгоритмів роєвого інтелекту для задач децентралізованого траєкторного управління групою БПЛА. Зокрема будуть розглянуті такі "роєві" алгоритми: African Vultures Optimization Algorithm (AVOA), алгоритм яструбів Харрика (ННО), Dung Beetle Optimizer (DBO) та алгоритм Snake Optimizer (SO). Дослідження спрямоване на визначення найбільш робастних методів, здатних забезпечувати стабільне функціонування рою в умовах динамічних перешкод, обмежених обчислювальних ресурсів та активної радіоелектронної протидії.

Основні завдання дослідження

Для досягнення поставленої мети слід виконати наступні завдання:

1. Опис математичних моделей. Описати математичну модель кожного з чотирьох порівнювальних алгоритмів (AVOA, ННО, DBO, SO) та визначити їхні ключові переваги в управлінні групами автономних об'єктів у порівнянні з традиційними методами оптимізації.

2. Порівняння критеріїв. На основі стандартних тестів, наведених у описі алгоритмів порівняти швидкості збіжності та точність знайдених рішень.

3. Аналіз ітераційної стабільності. Оцінити показники стандартного відхилення кожного алгоритму для порівняння надійності та підтвердження відтворюваності результатів у динамічних умовах середовища.

4. Аналіз поведінки алгоритмів в умовах перешкод. Порівняти на основі даних про уникнення локальних оптимумів поведінку алгоритмів з високою щільністю перешкод.

5. Висновки. Обґрунтувати доцільність застосування конкретного алгоритму для різних типів місій, таких як: швидкий переліт, детальний пошук чи то супровід цілі.

Методологія дослідження

Методологія дослідження ґрунтується на проведенні серії чисельних експериментів із використанням методів імітаційного моделювання для порівняльної оцінки ефективності порівнювальних в даній статті алгоритмів (AVOA, ННО, DBO, SO). Щоб забезпечити ідентичність умов, для кожного алгоритму було обрано єдине середовище реалізації - мова програмування Python версії 3.12.9 із використанням спеціалізованих бібліотек для глобальної оптимізації, таких як `mealpy` та `numpy`, а також програмний та адаптований код за оригінальними публікаціями [12, 13].

Всі реалізації алгоритмів будуть проведені на єдиній апаратній платформі. Для оцінки масштабованості алгоритмів експерименти проводилися для стандартному рівні складності $D =$

30 розмірностей. Кількість агентів $N = 30$, максимальна кількість ітерацій $T_{max} = 500$. Враховуючи вимоги до реального часу в децентралізованих мережах БПЛА, де час відгуку має становити близько 70 мс [7], такий обсяг ітерацій дозволяє оцінити здатність алгоритму знаходити прийнятне рішення в умовах обмеженого польотного вікна. Оцінка пошукової ефективності алгоритмів проводилася на наборі з 23 класичних математичних тестових функцій (F1–F23) [6], серед яких є унімодальні (F1–F7), мультимодальні (F8–F13) та фіксованої розмірності (F14–F23). Порівняльний аналіз здійснювався за трьома основними метриками, а саме: середнє значення, стандартне відхилення та аналіз кривих збіжностей. Такий методологічний підхід дозволяє перевірити результати експерименту з еталонними даними першоджерел та об'єктивно визначити найбільш робастний алгоритм для подальшої адаптації в задачах роєвого управління БПЛА [9].

Основна частина. Математичні моделі порівнювальних алгоритмів

Алгоритм AVOA моделює стратегію пошуку та локального уточнення [2] та має основний принцип в механізмі “голоду” (динамічний коефіцієнт переходу), значенні, яке безпосередньо визначає перехід від пошуку до уточнення рішення. Сама модель включає в себе декілька етапів що балансують між пошуком дослідженням простору та локальним уточненням рішень. На початку роботи алгоритму агенти розподіляються випадковим чином, і після оцінки фітнес-функції кожного агента визначаються два найкращих агента-лідера: Best Vulture 1 та Best Vulture 2 [2]. Ці рішення стають орієнтиром для всіх агентів, і кожен з них окремо випадковим чином обирає, за яким з лідерів слідувати.

Ключовим параметром що регулює перехід між двома фазами є показник "голоду

$$F = (2 \cdot rand + 1) \cdot z \cdot \left(1 - \frac{t}{T_{max}}\right) \quad (1)$$

де: F - визначення голоду; t - поточна ітерація; z - випадкові числа з діапазоном $[-1,1]$; $rand$ - випадкові числа з діапазоном $[0,1]$. При умові $|F| > 1$, агенти здійснюють глобальну розвідку (фаза Exploration, тобто дослідження). На цьому етапі подальший вибір стратегії залежить також від параметра P_1 (зазвичай вказується як 0.6) [2]. Якщо $rand \leq P_1$, агент продовжує рух випадковим чином задля уникання застрягання в локальних оптимумах. Якщо $rand \geq P_1$ агент переміщується далі від лідерів, намагаючись знайти нові перспективні зони поблизу вже відомих джерел. При $|F| < 1$, вони здійснюють локальну оптимізацію (фаза Exploitation, тобто експлуатації).

При цьому, якщо $F \in [0,5;1]$ подальша стратегія залежить від P_2 (зазвичай вказується як 0.4) [2]. Якщо $rand \geq P_2$, агенти здійснюють інтенсивне локальне обстеження. Якщо $rand < P_2$, агенти рухаються безпосередньо до обраного агента-лідера (швидке скорочення дистанції до

поточної найкращої точки). Якщо ж $F < 0,5$ стратегія базується вже на параметрі P_3 (зазвичай вказується як 0.6) [2]. Якщо $rand \geq P_3$, агенти починають конкурувати між собою, що математично виражається у швидкій зміні позицій навколо джерела. Якщо $rand < P_3$, агенти збираються в одній точці для уточнення фінального результату.

Алгоритм ДВО заснований у 2023 році і містить у собі імітацію життя гнойних жуків: кочення кулі, танець, розмноження, пошук їжі та крадіжку [5]. Фази в даному алгоритмі розділена на чотири специфічні групи, які відповідають за певні етапи. Кочення кулі і танець відповідають за фазу Exploration, під час якої оновлення координатів агентів рою здійснюється шляхом лінійного переміщення у просторі пошуку, а математична модель базується на передбаченні зміни позиції на основі вектора попереднього руху з урахуванням випадкового коефіцієнта відхилення [5], забезпечує спрямований пошук оптимального рішення вздовж прямої траєкторії:

$$x_i(t+1) = x_i(t) + \alpha \cdot k \cdot x_i(t-1) + b \cdot \Delta x \quad (2)$$

де: α - ваговий коефіцієнт інтенсивності пошуку; $k \in (0, 0.2]$ - коефіцієнт дефектності (відхилення від прямолінійної траєкторії для уникнення локальних оптимумів); b - випадкові числа з діапазоном $[0, 1]$.

Якщо на шляху з'являється перешкода, алгоритм змінює напрямок руху за допомогою спеціального оберту. Це дозволяє агенту вчасно змінити

маршрут і не "застрягнути" на місці, продовжуючи пошук у новому напрямку. Після дослідження перспективних зон рій переходить до наступної фази для їх детального вивчення. Ця фаза забезпечує посилений пошук в перспективних зонах.

Алгоритм формує межі для локальної зони R , звертаючи увагу на поточне найкраще рішення:

$$R = L \cdot \left(1 - \frac{t}{T_{max}}\right) \quad (3)$$

де: L - початковий радіус пошуку.

Такий підхід забезпечує поступове звуження області пошуку, що дозволяє

$$x_i(t+1) = x_{best} + b_1 \cdot (x_i(t) - L) + b_2 \cdot (x_i(t) - R) \quad (4)$$

Це дозволяє прибрати похибки та отримати максимально точно координати цілі, завдяки чому алгоритм забезпечує високу точність фінального результату. При цьому, навіть коли більшість агентів зайняті уточненням

$$x_i(t+1) = x_{Gbest} + S \cdot g \cdot (|x_i(t) - x_{Gbest}| + |x_i(t) - x_{best}|) \quad (5)$$

де: x_{best} - найкраще рішення знайдене за весь час пошуку; S - коефіцієнт гнучкості; g - число за нормальним розподілом для уникнення шаблонних рухів.

Ця особливість дозволяє окремим дронам здійснювати переліт між різними ділянками місцевості якщо в них було виявлено кращі показники, алгоритм при цьому не застрягає в одній точці і дає можливість швидко перегрупуватись при зміні польотного завдання або ж виявленні нових цілей.

Алгоритм ННО запропонований у 2019 році та базується на стратегії спільного пошуку та захоплення цілі групою агентів рою. Особливістю даного алгоритму є можливість адаптивно змінювати тактику в залежності від динаміки цілі та рівня "енергії" системи [4]. Під час фази Exploration коли координати цілі невідомі, агенти розсереджуються територією для пошуку останньої. Спосіб реалізації даної фази випадковим чином обирається серед двох стратегій: пошук на основі позицій інших учасників та суто випадковий пошук.

агентам зосередитися на детальному вивченні найбільш вигідних варіантів і не витратити час на перевірку завідомо неефективних ділянок[5]. Наступним кроком, коли зона пошуку є максимально звуженою, є точне визначення найкращої точки, під час якого група агентів здійснює пошуки у дуже малому радіусі навколо лідерів:

знайдених координат, рій зберігає гнучкість і не втрачає можливість знайти більш привабливі варіанти в інших зонах завдяки обміну інформацією між усіма агентами

Фактор, завдяки якому обирається стратегія залежить від показника енергії E , який поступово зменшується в процесі роботи:

$$E = 2E_0 \cdot \left(1 - \frac{t}{T_{max}}\right) \quad (6)$$

де: E_0 - початковий рівень енергії що визначається випадковим числом в діапазоні $[-1, 1]$. При умові $|E| \geq 1$ триває етап розвідки, при $|E| < 1$ алгоритм переходить до фази Exploitation. Ця фаза також має стратегії, які також залежать від показника E . При $|E| \geq 0.5$ відбувається "м'яке оточення", при якому агенти виконують плавне коригування траєкторії. Якщо $|E| < 0.5$, відбувається "жорстке оточення", при якому дрони швидко оточують ціль з фокусом усіх ресурсів в одній точці.

При зміні траєкторії цілі дрони використовують функцію Lévy flights [4] для того, щоб робити різкі зигзагоподібні рухи і не втратити орієнтир в складних умовах.

Алгоритм SO представлений у 2022 році для вирішення складних задач

оптимізації в умовах обмеженого простору [3]. Робота цього алгоритму регулюється завдяки двом основним показникам - стабільності $Temp$:

$$Temp = \exp\left(\frac{-t}{T_{max}}\right) \quad (7)$$

Та доступності цілі Q :

$$Q = c_1 \cdot \exp\left(\frac{t - T_{max}}{T_{max}}\right) \quad (8)$$

де: c_1 - константа масштабування.

При умові, що показник доступності цілі низький ($Q < 0.25$) агенти рою виконують пошук по всій території, при якому агенти обирають випадкові напрямки польоту для того, щоб максимально швидко охопити невідомий простір та знайти перспективні райони або цілі

$$x_i(t + 1) = x_{max}(t) \pm c_1 \cdot A \cdot (x_{max} - x_{min}) \cdot rand \quad (9)$$

де: x_{rand} - випадкова позиція агента;
 A - коефіцієнт здатності до маневру.

При $Q > 0.25$ ціль є ідентифікованою, і агенти починають рух до цілі, при цьому характер руху залежить від показника $Temp$. Якщо $Temp > 0.6$ рій рухається з значним збереженням дистанції між агентами, що дозволяє оминати перешкоди у щільному середовищі. Агенти не скупчуються в одній точці, а підходять до цілі з різних напрямків[3]. При $Temp < 0.6$ система переходить до етапу наведення, дрони починають обмін даними та вибудовують найкоротшу траєкторію до найкращої знайденої точки мінімізуючи будь-які відхилення.

Результати досліджень.

За результатами проведення серії імітаційних експериментів та чисельного моделювання було виконано комплексний порівняльний аналіз чотирьох метаевристичних алгоритмів: AVOA, DBO, ННО та SO. Для забезпечення об'єктивності оцінки кожен алгоритм було протестовано на наборі з 23 класичних бенчмарк-функцій з фіксованою кількістю ітерацій та розмірністю популяції. Нижче наведено зведені результати моделювання. Основними показниками ефективності алгоритмів було обрано середнє значення цільової функції для оцінки точності та стандартне відхилення для демонстрації стабільності та робастності методу. Метрика, яка була використана для кожного алгоритму - Mean(Std).

Таблиця 1. Результати порівняльного аналізу ефективності алгоритмів

Функція	f_{min}	AVOA	DBO	ННО	SO
F1	0	6.61e-316(~0)	2.94e-145(8.81e-145)	1.40e-237(~0)	7.45e-143(2.23e-142)
F2	0	1.2100e-158 (3.6200e-158)	4.6083e-07 (1.3199e-06)	3.9700e-122 (7.2700e-122)	2.9089e-46 (3.7459e-46)
F3	0	9.1000e-236 (0.0000e+00)	4.1707e-01 (1.2512e+00)	1.5700e-158 (4.7000e-158)	4.4387e-59 (1.1444e-58)
F4	0	4.4300e-168 (0.0000e+00)	9.0366e-07 (2.7100e-06)	1.3300e-115 (3.2500e-115)	3.1375e-41 (5.3482e-41)
F5	0	2.2700e-01 (5.4100e-01)	2.7986e+01 (4.7003e-01)	1.9900e-04 (1.8300e-04)	2.8879e+01 (6.2273e-02)
F6	0	0.0000e+00 (0.0000e+00)	1.0737e+00 (2.9033e-01)	0.0000e+00 (0.0000e+00)	1.4344e+00 (9.8462e-01)

Функція	f_{\min}	AVOA	DBO	ННО	SO
F7	0	1.3400e-04 (1.0400e-04)	3.9860e-03 (2.8290e-03)	2.7100e-04 (3.4000e-04)	3.3812e-04 (2.6183e-04)
F8	-418.9·D	1.52e+02 (3.21e+02)	-1.13e+04 (3.51e+03)	1.50e+03 (1.96e+03)	8.20e+03(4.30e+02)
F9	0	0.0000e+00 (0.0000e+00)	0.0000e+00 (0.0000e+00)	0.0000e+00 (0.0000e+00)	6.8968e+01 (1.0067e+01)
F10	0	4.4400e-16 (0.0000e+00)	1.4567e-09 (4.3525e-09)	4.4400e-16 (0.0000e+00)	9.7460e-01 (1.1996e+00)
F11	0	0.0000e+00 (0.0000e+00)	1.1102e-17 (3.3307e-17)	0.0000e+00 (0.0000e+00)	9.7518e-02 (2.0278e-01)
F12	0	2.3000e-07 (2.1600e-07)	4.3453e-02 (1.2202e-02)	7.1000e-07 (6.9100e-07)	4.7685e+00 (2.2375e+00)
F13	0	5.9600e-08 (5.1900e-08)	2.2277e+00 (3.6465e-01)	1.2600e-03 (3.7700e-03)	2.8457e+00 (3.9823e-01)
F14	0.998	1.2958 (0.6354)	3.3001 (1.5589)	1.6926 (0.8929)	2.4817 (1.0595)
F15	0.000307	3.3200e-04 (5.9700e-05)	8.4800e-04 (4.2700e-04)	4.5100e-04 (4.2600e-04)	2.5557e-03 (5.9378e-03)
F16	-1.0316	-1.0300e+00 (2.6200e-14)	-1.0316e+00 (0.0000e+00)	-1.0300e+00 (4.8500e-12)	3,49 (2,12)
F17	0.39788	3.9800e-01 (9.6700e-15)	3.9789e-01 (0.0000e+00)	3.9800e-01 (4.9900e-06)	3.9789e-01 (0.0000e+00)
F18	3	3.0000e+00 (1.4300e-05)	3.0000e+00 (0.0000e+00)	8.7900e+00 (1.0700e+01)	3.0000e+00 (1.9610e-15)
F19	-3.8627	-3.8700e+00 (3.2900e-08)	-3.8620e+00 (2.3640e-03)	-3.8700e+00 (2.0100e-05)	-3.8628e+00 (0.0000e+00)
F20	-3.3223	-3.2600e+00 (5.9800e-02)	-3.2347e+00 (7.4690e-02)	-3.2400e+00 (8.6700e-02)	-3.3224e+00 (3.8694e-12)
F21	-10.1532	-1.0200e+01 (1.1400e-11)	-5.5650e+00 (1.5294e+00)	-6.0700e+00 (2.0400e+00)	-8.3858e+00 (2.5298e+00)
F22	-10.4029	-1.0400e+01 (3.5700e-11)	-8.1501e+00 (2.5087e+00)	-5.6200e+00 (1.5900e+00)	-8.2975e+00 (2.4608e+00)
F23	-10.5364	-10.5364 (0.0000)	-5.1663 (2.0482)	-1.8882 (1.4471)	-2.0263 (0.8358)

Варто зазначити, що в наведеній таблиці 1 значення $0.0000e+00$ і подібні не є абсолютним нулем, але є таким, що до нього наближено. При аналізі даних можна виділити чітку спеціалізацію кожного алгоритму залежно від цільової функції. Дані демонструють, що універсального алгоритму не існує, однак AVOA та ННО демонструють найбільш збалансовані показники майже в усіх бенчмарках.

В унімодальних функціях безумовним лідером є AVOA і підтверджує себе як найкращий алгоритм для локального пошуку. ННО

майже зрівнявся за показниками точності, та навіть перевершив її за AVOA у функції F5. DBO та SO виглядають значно слабшими через високу похибку, маючи схильність до передчасної зупинки на гладких ландшафтах.

В мультимодальних функціях AVOA та ННО знову показати себе стабільно. DBO в функції F8 показав найкращим, з суттєвим відривом результат є близьким до еталонного значення. Це підтверджує, що стратегія пошуку в даному алгоритмі чудово підходить до глобальної розвідки. SO

показав на функції F9 значно гіршу стабільність, вказуючи на те, що у алгоритму виникають труднощі при роботі з великою кількістю ідентичних локальних пасток.

В функціях фіксованої розмірності AVOA також демонструє результати, близькі до еталонних, що підтверджує його стабільність на складних ландшафтах. SO показує низьке відхилення на низьковимірних функціях F19 та F20, що визначає його достатньо

надійним для специфічних малорозмірних задач.

За результатами аналізу ітераційних кривих збіжності було оцінено та порівняно часову ефективність та обчислювальну складність досліджуваних алгоритмів. Криві збіжності відображають зміну найкращого значення фітнес-функції залежно від номера поточної ітерації t , що дозволяє наочно порівняти швидкість переходу від фази Exploration до фази Exploitation

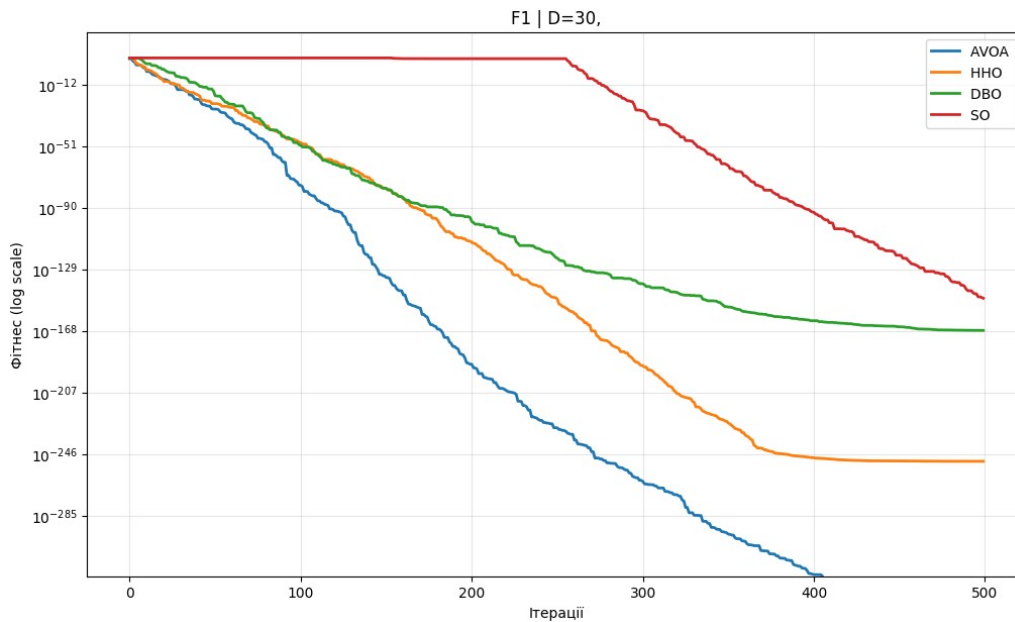


Рис.1. Динаміка мінімізації похибки позиціонування F1

За графіком на рис. 1. AVOA показав найкращу ефективність досягнувши майже еталонного показника точності, DBO та HHO на початкових ітераціях демонстрували однаковий

результат, але при цьому HHO все ж отримав кращі показники. Алгоритм SO потребував найбільше часу для початку етапу підвищення точності та мінімізації похибки

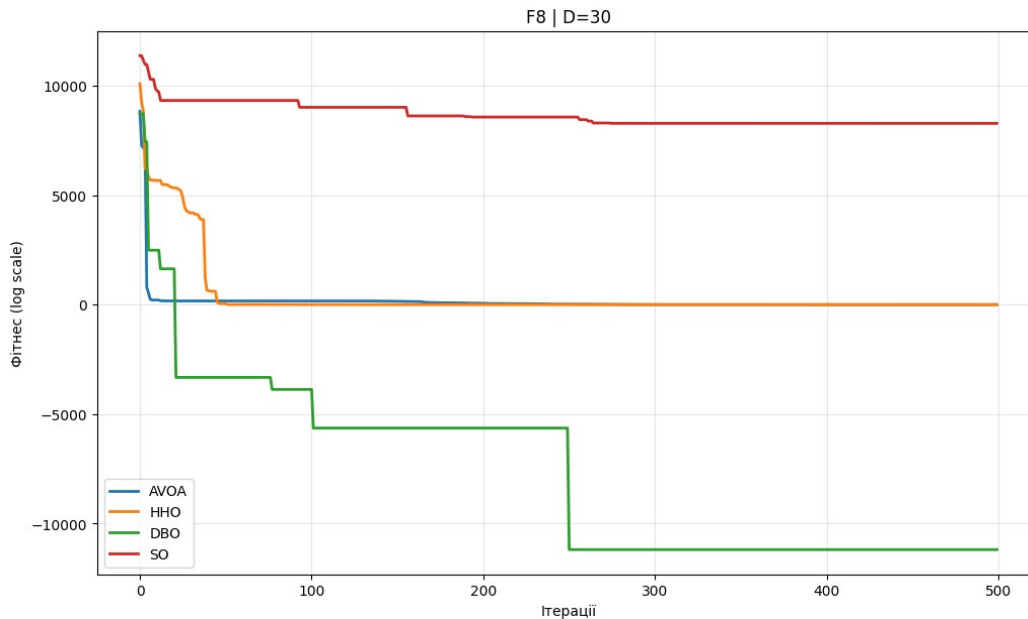


Рис.2. Ефективність подолання перешкод та виходу з локальних пасток F8

За графіком на рис.2. алгоритм DBO показав найкращу ефективність та точність пошуку глобального мінімуму. AVOA та HHO показали задовільну точність, стабілізувавшись у зоні

локальних мінімумів поблизу нульових значень. SO ж виявився найменш ефективним, застрягши в локальному екстремумі з критично низькою точністю відносно еталону.

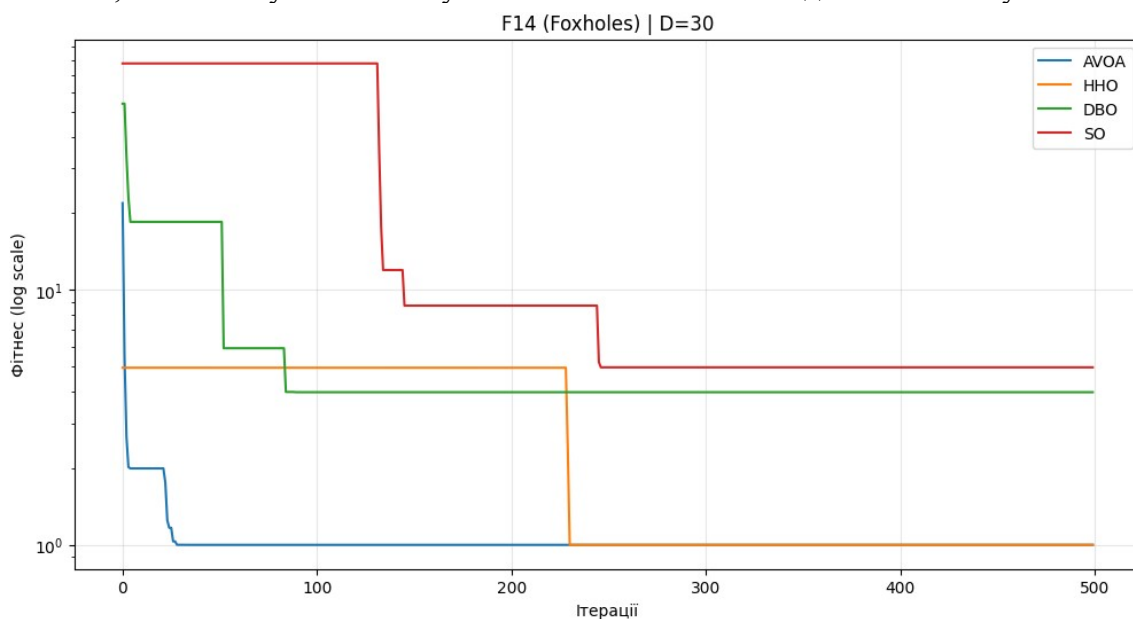


Рис. 3. Динаміка подолання складної мережі перешкод F14

За графіком на рис.3. можна визначити що алгоритми SO та DBO виявилися менш ефективними і застрягли в локальних «норах» ландшафту з нижчою точністю, в той же час коли AVOA та HHO продемонстрували

найкращі показники точності. Також графік відображає ступінчасте підвищення точності обчислень. Зокрема, алгоритму HHO знадобилося близько 230 ітерацій для успішного виходу з локального мінімуму та

переходу до фази фінальної мінімізації похибки

Таблиця 2. Зведена характеристика ефективності алгоритмів

Параметр порівняння	AVOA	DVO	ННО	SO
Точність локалізації (Exploitation)	Висока	Низька	Висока	Низька
Глобальна розвідка (Exploration)	Висока	Найвища	Середня	Помірна
Ітераційна стабільність (Std)	Найвища	Помірна	Середня	Низька
Швидкість збіжності	Дуже висока	Середня	Висока	Повільна

Висновки

В ході проведення дослідження було здійснено порівняльний аналіз чотирьох сучасних (2021-2023 роки представлення) метаевристичних алгоритмів за допомогою 23-ох класичних бенчмарк функцій. Аналіз показав, що алгоритм AVOA має найвищу точність та швидкість збіжності, особливо на унімодальних функціях, і також виявився єдиним, хто зміг знайти точний глобальний мінімум на складних ландшафтах фіксованої розмірності. Отримані показники роблять цей алгоритм ідеальним для, наприклад, прецизійного наведення групи БПЛА на статичну ціль або утримання стабільної геометрії рою в просторі.

Водночас, алгоритм DVO виявився найбільш ефективним у роботі з мультимодальними ландшафтами високої складності, зокрема на функції F8, де має найкращий результат серед усіх алгоритмів, що робить його ефективним у задачах автономного планування маршрутів у невідомому середовищі з високою щільністю перешкод і підтверджує його перевагу в задачах з пошуку в глобальному просторі.

Алгоритм ННО показав збалансовані результати, продемонструвавши високу точність і низьку похибку на вигнутих ландшафтах, що вказує на перспективність застосування для динамічного супроводження рухомих об'єктів. Проте виявив схильність до нестабільності на складних мультимодальних задачах.

SO виявився найменш ефективним у більшості тестів, продемонструвавши повільну адаптацію до ландшафту та схильність до передчасної збіжності на початкових етапах пошуку. Найкращі показники стандартного відхилення має алгоритм AVOA, що робить його більш привабливим для використання в задачах оптимізації, водночас високі значення Std алгоритмів DVO та SO на унімодальних функціях вказують на їхню чутливість до початкових параметрів ініціалізації популяції.

Тож варто зазначити, що для задач, що вимагають максимальної точності та стабільності обчислень, доцільно використовувати алгоритми AVOA або ННО. У випадках пошуку в надскладних мультимодальних середовищах з великою кількістю локальних пасток перевагу варто віддавати алгоритму DVO.

Список використаних джерел

1. Hassanien A. E., Emary E. Swarm Intelligence: Principles, Advances, and Applications. *1st ed. CRC Press*, 2016. DOI: 10.1201/9781315222455.
2. Abdollahzadeh B., Gharehchopogh F., Mirjalili S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* 2021. Vol. 158. 107408. DOI: 10.1016/j.cie.2021.107408.
3. Hashim F., Hussien A. Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowl. Based Syst.* 2022. Vol. 242. 108320.
4. Opposition-based learning Harris hawks optimization with advanced transition rules: Principles and analysis / A. A. Heidari et al. *Expert Systems with Applications.* 2020. Vol. 158. 113528. DOI: 10.1016/j.eswa.2020.113528.
5. Xue J.-K., Shen B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing.* 2022. Vol. 79. DOI: 10.1007/s11227-022-04959-6.
6. A Novel Meta-Heuristic Algorithm for Numerical and Engineering Optimization Problems: Piranha Foraging Optimization Algorithm (PFOA) / S. Cao et al. *IEEE Access.* 2023. PP. 1-1. DOI: 10.1109/ACCESS.2023.3267110.
7. Kalashnikova V. Methods of managing an automated mobile system. *INNOVATIVE TECHNOLOGIES AND SCIENTIFIC SOLUTIONS FOR INDUSTRIES.* 2024. No. 4(30). P. 67–84. DOI: 10.30837/2522-9818.2024.4.067.
8. Використання алгоритмів ройового інтелекту при проектуванні систем управління груп безпілотних літальних апаратів / О. В. Каратанов та ін. *Молодий вчений.* 2021. № 10 (98). С. 98–103.
9. Бабич С. О., Гуляницький Л. Ф. Алгоритми ройового інтелекту для задачі динамічного розподілу навантаження. *Журнал науковий огляд.* 2018. № 3(46).
10. UAV Swarm Mission Planning and Load Sensitivity Analysis Based on Clustering and Optimization Algorithms / Y. Yan et al. *Appl. Sci.* 2023. Vol. 13. 12438. DOI: 10.3390/app132212438.
11. Diep Q. B., Truong T. C., Zelinka I. Planning trajectory for UAVs using the self-organizing migrating algorithm. *PLOS ONE.* 2025. Vol. 20(7). e0327016. DOI: 10.1371/journal.pone.0327016.
12. Lancephil. Dung-Beetle-Optimizer [Source code]. GitHub. 2023. URL: <https://github.com/Lancephil/Dung-Beetle-Optimizer>.
13. Hashim F. Snake Optimizer (SO) [Source code]. MathWorks File Exchange. 2022. URL: <https://www.mathworks.com/matlabcentral/fileexchange/106465-snake-optimizer>.

Тюшкевич Д.А., Гузій М.М.

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТАЕВРИСТИЧНИХ АЛГОРИТМІВ РОЄВОГО ІНТЕЛЕКТУ ГРУПОВОГО УПРАВЛІННЯ БПЛА

В даній статті робота присвячена проведенню порівняльного аналізу та оцінку ефективності сучасних, представлених не пізніше 2021 року, метаевристичних алгоритмів *African Vultures Optimization Algorithm (AVOA)*, *Harris Hawks Optimization (HHO)*, *Dung Beetle Optimizer (DBO)* та *Snake Optimizer (SO)* що можуть бути використані для задачі децентралізованого управління роєм безпілотних літаючих апаратів (БПЛА). Дослідження є актуальним та зумовлено тим, що вектор в управлінні БПЛА переходить від поодинокого керування до координації автономних роїв, які можуть виконувати більше трудоміких задач, але при цьому мають потребу в мінімізації ризиків, притаманні централізованим систем, а саме зменшити критичну затримку передачі даних та підвищити стійкість до втрати керуючих вузлів.

Методологія дослідження проводилась за допомогою імітаційного моделювання в середовищі Python версії 3.12.9. Набір для тестування складався з 23-ох класичних бенчмарк-функцій, на яких було перевірено чотири вище згаданих алгоритми. Обраними метриками для порівняння стали точність Mean, ітераційна стабільність Std та швидкість збіжності за фіксованих для кожного алгоритму параметрів розмірності $D = 30$.

Результатом чисельних експериментів є демонстрація чіткої спеціалізації кожного з алгоритмів: встановлено, що AVOA має найбільш високу, наближену до еталону, точність локалізації та виняткову стабільність в більшості бенчмарків і в порівнянні з конкурентами. Це робить алгоритм найкращим вибором для задач наведення рою або утримання у просторі. Найкращим алгоритмом для глобальної розвідки в умовах складного рельєфу є DBO, який продемонстрував найкращу ефективність в плануванні маршрутів невідомими ділянками з великою кількістю перешкод. HHO як алгоритм показав достатньо збалансовані показники точності та стабільності, в особливості на вигнутих ландшафтах. В загальному не поступаючись AVOA, алгоритм HHO також чудово підходить для задач супроводу цілей. Алгоритм SO в середньому виявився найменш ефективним через повільну збіжність, проте має достатню ефективність у використанні в задач з малою розмірністю.

Ключові слова: БПЛА, ройовий інтелект, метаевристичні алгоритми, децентралізоване управління, AVOA, DBO, HHO, SO, глобальна оптимізація.

Tiushkevych D.A., Huzii M.M.

COMPARATIVE ANALYSIS OF SWARM INTELLIGENCE METAHEURISTIC ALGORITHMS FOR GROUP UAV CONTROL

This article is devoted to conducting a comparative analysis and evaluating the efficiency of modern metaheuristic algorithms introduced no later than 2021—namely, *African Vultures Optimization Algorithm (AVOA)*, *Harris Hawks Optimization (HHO)*, *Dung Beetle Optimizer (DBO)*, and *Snake Optimizer (SO)* which can be applied to decentralized unmanned aerial vehicle (UAV) swarm control tasks. The research is timely and driven by the shift in UAV management from single-unit control to the coordination of autonomous swarms capable of executing labor-intensive tasks while minimizing risks inherent to centralized

systems, specifically by reducing critical communication latency and enhancing resilience against control node failures.

The research methodology was implemented using simulation modeling in a Python version 3.12.9 environment. The testing suite consisted of 23 classic benchmark functions, on which the four aforementioned algorithms were evaluated. Accuracy (Mean), iterative stability (Std), and convergence speed at a fixed dimensionality of $D = 30$ for each algorithm were selected as the primary comparison metrics.

The results of the numerical experiments demonstrated a clear specialization for each algorithm: it was established that AVOA possesses the highest, near-benchmark localization accuracy and exceptional stability across most benchmarks compared to its competitors. This makes the algorithm the optimal choice for swarm targeting or spatial station-keeping tasks. DBO proved to be the superior algorithm for global exploration in complex terrains, demonstrating the highest efficiency in path planning through unknown areas with a high density of obstacles. The HHO algorithm showed sufficiently balanced indicators of accuracy and stability, particularly on curved landscapes. Generally competitive with AVOA, HHO is also excellently suited for target tracking tasks. The SO algorithm proved to be the least effective on average due to slow convergence, yet it maintains sufficient effectiveness for use in low-dimensional tasks.

Keywords: UAV, swarm intelligence, metaheuristic algorithms, decentralized control, AVOA, DBO, HHO, SO, global optimization.

Стаття подана до редакції: 10/12/2025

Стаття прийнята до опублікування: 21/12/2025

Стаття опублікована: 30/12/2025

Стаття поширюється на умовах ліцензії CC BY 4.0