

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-55-31>

УДК 519.854.2+004.021

Шенгелія Володимир Олександрович, магістр

<https://orcid.org/0009-0001-6356-962X>

Жданова Олена Григорівна, к.т.н., доцент

<https://orcid.org/0000-0002-8787-846X>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

МЕТОДИ СКЛАДАННЯ МАРШРУТІВ ПОЛЬОТУ ДЛЯ БПЛА З ПЕРЕСУВНИМ ЦЕНТРОМ ОБСЛУГОВУВАННЯ

Шенгелія В.О., Жданова О.Г. Методи складання маршрутів польоту для БПЛА з пересувним центром обслуговування. У статті розглядаються методи складання маршрутів польоту для безпілотних літальних апаратів (БПЛА) з пересувним центром обслуговування (ПЦО). Досліджувана задача передбачає наявність одного БПЛА та одного ПЦО, який має множину місць дислокації. Метою роботи є підвищення якості маршрутів польоту. Для розв'язання задачі розроблено жадібний алгоритм, алгоритм штучної бджолоїної колонії та табу-пошук. Запропонований бджолоїний алгоритм налічує три способи організації роботи бджіл-розвідників і п'ять способів організації роботи бджіл-фуражирів, комбінації яких утворюють 15 різноманітних алгоритмів. Для алгоритму табу-пошуку розроблено чотири способи роботи локальної оптимізації. Експерименти показали, що найкращий із алгоритмів табу-пошуку працював значно швидше за найкращий з бджолоїних алгоритмів, але в середньому трохи поступився в якості побудованих маршрутів.

Ключові слова: безпілотні літальні апарати (БПЛА), пересувний центр обслуговування (ПЦО), теорія маршрутизації, комбінаторна оптимізація, жадібний алгоритм, алгоритм штучної бджолоїної колонії, алгоритм табу-пошуку.

Shenheliia V., Zhdanova O. Path planning for UAVs with a moving maintenance center. The article examines methods for path planning of unmanned aerial vehicles (UAVs) with a mobile maintenance center (MMC). The task under study involves a single UAV and a single MMC, which has multiple deployment locations. The aim of the work is to improve the quality of flight paths. To solve the problem, a greedy algorithm, an artificial bee colony algorithm, and a tabu search algorithm were developed. The proposed bee algorithm includes three methods for organizing the work of scout bees and five methods for organizing the work of forager bees, with combinations forming 15 different algorithms. For the tabu search algorithm, four methods of local optimization were developed. Experiments showed that the best of the tabu search algorithms operated significantly faster than the best of the bee algorithms, but on average, it slightly lagged in the quality of the constructed flight paths.

Keywords: unmanned aerial vehicles (UAV), mobile maintenance center (MMC), combinatorial optimization, routing theory, greedy algorithm, artificial bee colony algorithm, tabu search algorithm.

Постановка проблеми. Безпілотні літальні апарати (БПЛА) стрімко починають займати важливе місце в різних сферах діяльності людини, від військових до цивільних застосувань. Вже активно використовуються рятувальними службами для доставки медикаментів та їжі у віддалені та важкодоступні місця. Військові застосовують БПЛА під час ведення розвідки для виявлення та атаки ворожих цілей, що допомагає проводити операції без прямої участі пілотів. Протягом останніх років виникло багато інтересу до їх використання у комерційних потребах, це трапилось завдяки розвитку технологій і зменшенню витрат на виробництво даних апаратів. Великі логістичні компанії, такі як Amazon, United Parcel Service і DHL, шукають альтернативні підходи зменшення витрат на останній кілометр доставки замовлень клієнтам. Amazon в 2021 році оголосив, що вже почав розробку власних БПЛА й всієї необхідної під них інфраструктури для вирішення цієї проблеми. Однак існують численні виклики, пов'язані з використанням БПЛА. Основним обмеженням є тривалість польоту, цю проблему можна вирішити розробивши різні типи акумуляторів, використовуючи двигуни внутрішнього згоряння або гібридні системи. Проте, більш перспективним рішенням проблеми визнали пересувні центри обслуговування (ПЦО), які здатні перезаряджати чи замінювати акумулятори, зберігати БПЛА та взаємодіяти з ними [1]. ПЦО є інноваційним концептом, який передбачає використання мобільної платформи (наприклад, автомобіля, корабля або наземної станції), яка служить як точка запуску та посадки для БПЛА. Застосування ПЦО дозволяє вирішити проблему тривалості польоту та рухатися далі у напрямку автоматизації систем, через це основною областю досліджень стало створення енергоефективних маршрутів. Таким чином, розробка ефективних методів складання маршрутів польоту БПЛА з ПЦО може стати ключовим елементом у розширенні можливостей використання цих апаратів в різних галузях.

Аналіз останніх досліджень і публікацій. Наявна велика кількість типів задач маршрутизації БПЛА, їх класифікують в залежності від цільової функції, обмежень та змінних

факторів. Більшість задач складання маршрутів польоту для БПЛА ставить за мету оптимізувати пройдену відстань та витрачений час. Обмеження ресурсу заряду акумулятору, наявність заборонених для польоту зон або перешкод, вантажопідйомність є найбільш поширеними. Змінні фактори представляють собою кількість БПЛА та їх характеристики, наявність динамічного середовища або ПЦО, а також багато інших. Ці особливості грають значну роль під час створення математичної моделі та вибору методів розв'язання поставленої задачі.

Методи складання маршрутів польоту для БПЛА поділяються на традиційні та сучасні інтелектуальні. Традиційні включають методи найшвидшого спуску, оптимального керування та кореляції похідних. Вони мають деякі суттєві недоліки, такі як погана адаптивність та низька стабільність у відтворенні отриманих результатів, з легкістю можуть зациклюватись в локальній оптимізації, що не допустимо для маршрутизації БПЛА [2]. Серед сучасних інтелектуальних алгоритмів можна зустріти еволюційні, метаевристики основані на ройовому інтелекті, методи локального пошуку та багато інших [3]. Дана група методів вже довела свою ефективність та використовується в більшості робіт, пов'язаних зі складанням маршрутів польоту для БПЛА. Далі окремо розглянуто переваги та недоліки найпопулярніших з них.

Генетичні алгоритми. Незважаючи на переваги, такі як глобальна оптимізація, можливість паралельних обчислень та гнучкість (застосування до широкого діапазону задач), вони мають деякі недоліки, а саме низьку швидкість збіжності, проблеми з масштабованістю (збільшення розмірності може значно знизити якість), потребують великої кількості обчислювальних ресурсів [4]. Алгоритм все ж таки є ефективним в розв'язанні задач маршрутизації БПЛА. Проте, складність визначення принципів роботи операторів, часто призводять до того, що розроблений алгоритм є неефективним [5] та потребує модифікацій чи повної переробки [6].

Метод рою частинок. Завдяки своїй ефективності з часом став набирати популярність для розв'язання задач маршрутизації. Не зважаючи на просту структуру і малу кількість параметрів, даний метод не дуже підходить для розв'язання дискретних задач, адже через наявну схему роботи виникає потреба у введенні форм «рух» і «швидкість», доволі часто це відбувається неефективно [7]. Проте, особливої уваги вартує основний недолік, від якого страждає якість його побудови, а саме часте впадання в локальні оптимуми [8].

Мурашиний алгоритм. Даний метод оптимізації є більш популярним та ефективним за якістю складання маршрутів польоту БПЛА ніж попередньо розглянуті. Має низьку вагомих відомих переваг, а саме наявний глобальний пошук, децентралізовану обробку та підвищення якості знайдених розв'язків з часом [9]. Проте, алгоритм страждає від повільної збіжності, що призводить до його низької швидкості роботи [10].

Бджолиний алгоритм. Ефективний для задач складання маршрутів польоту БПЛА, адже передбачає локальне вдосконалення знайдених розв'язків, простий у реалізації, з гнучкою параметризацією як кількості бджіл-розвідників та бджіл-фуражирів, так і умови завершення роботи. Під час пошуку може вимагати великої кількості обчислень через наявну постійну локальну оптимізацію [11]. В порівнянні з мурашиним алгоритмом показує значно кращі результати як в якості отриманих маршрутів, так і за часом виконання [12].

Методи локального пошуку. Група методів може бути майже такою ж ефективною як ройові, проте кращою за часом роботи, перевірка цієї гіпотези заслуговує на увагу. Серед наявних алгоритмів найбільшої уваги заслуговує алгоритм табу-пошуку, адже надає можливість уникнення локальних мінімумів, здатний працювати зі складними обмеженнями, простий в реалізації та доволі ефективний на практиці.

Для подальшої розробки та аналізу ефективності серед ройових обрано алгоритм штучної бджолиної колонії, адже попередньо він виявився кращим за якістю отриманих маршрутів. Для оцінки ефективності побудови методів локального пошуку обрано алгоритм табу-пошуку.

Постановка досліджуваної задачі. Нехай маємо один БПЛА, множину об'єктів розвідки та ПЦО, що також має множину (більше трьох) місць дислокації. Ресурс польоту БПЛА є обмеженим, приземлившись на місці дислокації ПЦО, він проходить обслуговування (заміну акумулятору або дозаправку паливом), після чого повністю відновлює ресурс польоту. Об'єкти розвідки мають різну вагу (пріоритет для відвідання), при цьому кожен з них можна відвідати лише один раз протягом виконання усього маршруту польоту.

БПЛА починає рух з першого місця дислокації і облітає певні об'єкти розвідки, до закінчення ресурсу польоту має здійснити посадку на друге місце дислокації, де проходить обслуговування. Послідовність дій, описана попередньо, аналогічна для наступних підмаршрутів. Завдання полягає

у визначенні підмаршрутів, що дозволять обстежити максимальну зважену кількість об'єктів розвідки під час даного вильоту.

Дано:

$J = \{ 1, 2, \dots, n \}$ — множина об'єктів розвідки, кожен має вагу $1 \leq w_j \leq 10, j = 1, \dots, n$;

$\{ A_1, A_2, \dots, A_m \}$ — множина місць дислокації ПЦО, де $3 < m \leq 100$;

t — ресурс польоту БПЛА в часовому вимірі (без обслуговування);

τ — час відновлення ресурсу польоту (обслуговування);

v — швидкість польоту БПЛА.

Проміжні дані:

c_{pq} ($p, q \in \{ 1, 2, \dots, n, n+1, n+2, \dots, n+m \}$) — матриця часів перельоту.

Значення під індексами $n+1, n+2, \dots, n+m$ визначають час перельоту до місця дислокації ПЦО.

Необхідно знайти такі підмаршрути обльоту об'єктів розвідки:

$$J_1 = \{ A_1, j_1^1, j_2^1, \dots, j_{n_1}^1, A_2 \},$$

$$\dots$$

$$J_{m-1} = \{ A_{m-1}, j_1^{m-1}, j_2^{m-1}, \dots, j_{n_{m-1}}^{m-1}, A_m \},$$

де n_1, \dots, n_{m-1} вказують на кількість обстежених об'єктів розвідки кожного з підмаршрутів, ці величини також відносяться до шуканих.

До знайдених підмаршрутів накладаються наступні обмеження:

$$\{ j_1^1, j_2^1, \dots, j_{n_1}^1 \} \cup \dots \cup \{ j_1^{m-1}, j_2^{m-1}, \dots, j_{n_{m-1}}^{m-1} \} \subseteq J, \quad (1.1)$$

$$\{ j_1^p, j_2^p, \dots, j_{n_p}^p \} \cap \{ j_1^q, j_2^q, \dots, j_{n_q}^q \} = \emptyset, \quad p, q = \overline{1, m-1}, p \neq q. \quad (1.2)$$

Формула 1.1 вказує, що усі відвідані об'єкти розвідки включені або є еквівалентними заданій множині об'єктів розвідки. Формула 1.2 вводить обмеження, що кожен об'єкт розвідки може бути відвіданий лише раз. Сумарний час польоту кожного підмаршруту не має перевищувати ресурс польоту БПЛА у часовому вимірі:

$$C_1 = c_{A_1 j_1^1} + \sum_{q=1}^{n_1-1} c_{j_q^1 j_{q+1}^1} + c_{j_{n_1}^1 A_2} \leq t,$$

$$\dots$$

$$C_{m-1} = c_{A_{m-1} j_1^{m-1}} + \sum_{q=1}^{n_{m-1}-1} c_{j_q^{m-1} j_{q+1}^{m-1}} + c_{j_{n_{m-1}}^{m-1} A_m} \leq t.$$

Цільовою функцією є максимізація сукупної ваги обстежених об'єктів розвідки:

$$\sum_{i=1}^{m-1} \sum_{l=j_1^i}^{j_{n_i}^i} w_l \rightarrow \max.$$

Розроблені алгоритми. Вводиться умова використання двовимірного простору для спрощення процесу розробки алгоритмів. Пошук розв'язків кожним з них відбувається послідовно за підмаршрутами. Очікується, що під час ініціалізації задач наявна матриця, що містить відомості про час переміщення та співвідношення ваги об'єкту розвідки до часу перельоту для всіх наявних

точок. Жадібний алгоритм введено як референтну точку відліку для оцінки якості результатів, які були отримані за допомогою розроблених алгоритмів.

Жадібний алгоритм (Greedy). Натхнений класичним алгоритмом найближчого сусіда, що зазвичай найчастіше обирається людиною для пошуку розв'язків подібних задач. У базовій постановці на кожному кроці вибору підлягає найближча за відстанню точка, проте ця модифікація передбачає пошук найбільшого співвідношення ваги об'єкта розвідки до відстані. Пошук відбувається без урахування відвіданих об'єктів розвідки, проте враховує наявність ресурсу польоту для приземлення на кінцевий ПЦО.

Алгоритм має описаний нижче вигляд.

ЦИКЛ ЗА ПІДМАРШРУТАМИ:

1. ВКЛЮЧИТИ початковий ПЦО в підмаршрут.
2. ВИКОНУВАТИ (формування підмаршруту):
 - 2.1. ВИЗНАЧЕННЯ найкращого об'єкта розвідки для візиту з поточної точки.
 - 2.2. ЯКЩО наступний для відвідання об'єкт розвідки не знайдено, ТО:
 - 2.2.1. ВКЛЮЧИТИ кінцевий ПЦО в підмаршрут.
 - 2.2.2. ВИЙТИ З ЦИКЛУ.

ІНАКШЕ:

- 2.2.1. ВКЛЮЧИТИ знайдений об'єкт розвідки в підмаршрут.
- 2.2.2. ПОЗНАЧИТИ знайдений об'єкт розвідки як відвіданий.
- 2.2.3. ПРИЗНАЧИТИ знайдений об'єкт розвідки поточним.
- 2.2.4. ПЕРЕРАХУВАТИ залишок ресурсу польоту.

Бджолиний алгоритм (ABC). Побудований на основі алгоритму штучної бджолоїної колонії, який є натхненим поведінкою рою бджіл в процесі пошуку нектару. Бджоли-розвідники вилітають з вулика у різних випадкових напрямках та шукають місця знаходження нектару, формуючи початкові розв'язки. З часом повертаються у вулик, передаючи інформацію про розташування нектару. Бджоли-фуражири летять у місця, де його очікується найбільше, та проводять процедуру пошуку в околах знаходження нектару. Даний процес повторюється доти, доки не виконається певна кількість ітерацій без покращення отриманого розв'язку.

Алгоритм має схему, подану нижче.

ЦИКЛ за підмаршрутами:

ВИКОНУВАТИ:

1. ЦИКЛ за кількістю бджіл-розвідників:
 - 1.1. ВИПУСТИТИ бджолу-розвідника.
2. ОБРАТИ найкращих бджіл-розвідників.
3. ЦИКЛ за найкращими бджолами-розвідниками:
 - 3.1. ПОКИ результат покращується:
 - 3.1.1. ЦИКЛ за бджолами-фуражирами:
 - 3.1.1.1. ВИПУСТИТИ бджолу-фуражира.
 4. ЯКЩО результат покращився, ТО СКИНУТИ В НУЛЬ кількість ітерацій без покращення.

ІНАКШЕ ІНКРЕМЕНТУВАТИ кількість ітерацій без покращення.

ПОКИ кількість ітерацій без покращення < максимальної кількості ітерацій без покращення.

ЗБЕРЕГТИ отриманий підмаршрут.

ВИКЛЮЧИТИ відвідані об'єкти розвідки з огляду.

Запропоновано три способи організації роботи бджіл-розвідників. Розглянемо їх детально.

Випадковий (Random — R). Під час побудови підмаршруту визначається список об'єктів розвідки, які можна відвідати з поточного місця дислокації ПЦО. Випадковим чином генерується кількість тих, які має відвідати БПЛА в процесі побудови, об'єкти розвідки довільно обираються зі створеного раніше списку. Після чого підмаршрут перевіряється на допустимість, в разі успішності перевірки додається до вже знайдених.

Ймовірнісний (Probable — P). На кожному кроці пошуку об'єктів розвідки створюється список тих, що можна відвідати з поточної точки. Будується одинична пряма з нанесеними відрізками співвідношень ваги до часу перельоту для обраних об'єктів розвідки. Генерується раціональне число від нуля до одиниці, що вказує на точку, яка належить відріzkу певного об'єкта розвідки. Таким чином, ймовірність вибору об'єкта розвідки прямо пропорційна вазі та обернено пропорційна часу перельоту для кожного з об'єктів розвідки.

Ймовірнісний в околі (Probable in the neighborhood — PN). Додатково враховується, що шуканий об'єкт розвідки має знаходитись в околі радіусом одна четверта від максимального ресурсу польоту БПЛА.

Локальна оптимізація початкових підмаршрутів є ключовою у знаходженні розв'язків. Запропоновано п'ять способів організації її роботи, кожен з яких розглянуто нижче.

Додати найближчий об'єкт розвідки (Add nearest — AN). Випадково обирається один з об'єктів розвідки підмаршруту, в залежності від ефективності перебудованого підмаршруту найближчий об'єкт розвідки до даного додається для відвідання перед або після нього.

Перебудувати підмаршрут ймовірно (Rebuild probable — RP). З підмаршруту випадковим чином обирається об'єкт розвідки, після чого наступні за ним об'єкти розвідки виключаються, а сам підмаршрут добудовується за способом організації роботи ймовірнісних бджіл-розвідників.

Перебудувати підмаршрут ймовірно в околі (Rebuild probable in the neighborhood — RPN). Подібний до попереднього, але при перебудові підмаршруту точки обираються з околу.

Перебудувати підмаршрут ймовірно та додати найближчий об'єкт розвідки (Rebuild probable and add nearest — RPAAN). Представляє собою комбінацію способів організації роботи RP та AN, спочатку підмаршрут підлягає перебудові, а потім покращується додаванням найближчого об'єкта розвідки.

Перебудувати підмаршрут ймовірно в околі та додати найближчий об'єкт розвідки (Rebuild probable in the neighborhood and add nearest — RPNAAN). Аналогічний попередньому, але перебудова підмаршрутів відбувається з урахування пошуку об'єктів розвідки в околі.

Для подальшого дослідження розроблено три способи організації роботи бджіл-розвідників і п'ять способів організації роботи бджіл-фуражирів, комбінації яких утворюють 15 різноманітних алгоритмів.

Алгоритм табу-пошуку (Tabu). Перш за все, відбувається пошук допустимого розв'язку. За допомогою локальних модифікацій поточний підмаршрут ітеративно змінюється, намагаючись перейти до кращого розв'язку. При цьому отриманий розв'язок може погіршуватися, що дозволяє вийти з «пастки» локального мінімуму. Усі розв'язки, отримані в процесі пошуку, додаються до списку заборонених (табу), отже вони не будуть оцінюватися як потенційні кандидати для подальшого відвідування.

Нижче наведено схему алгоритму.

ЦИКЛ за підмаршрутами:

1. ФОРМУВАННЯ допустимого розв'язку.
2. ІНІЦІАЛІЗУВАТИ табу-списку.
3. ПРИЗНАЧИТИ поточний розв'язок рекордним.
4. ВИКОНУВАТИ (локальний пошук):
 - 4.1. ПОШУК допустимих розв'язків в околі.
 - 4.2. ЯКЩО один із знайдених розв'язків краще рекордного, ТО:
 - 4.2.1. ПРИЗНАЧИТИ даний розв'язок рекордним.
 - 4.2.2. ОБНУЛИТИ кількість ітерацій без покращення.
ІНАКШЕ ІНКРЕМЕНТУВАТИ кількість ітерацій без покращення.
 - 4.3. ВКЛЮЧИТИ знайдені розв'язки в табу-список.
ПОКИ кількість ітерацій без покращення < максимальної кількості ітерацій без покращення.
5. ЗБЕРЕГТИ отриманий підмаршрут.
6. ВИКЛЮЧИТИ відвідані об'єкти розвідки з огляду.

У розробленій системі алгоритмів для ініціалізації початкового допустимого розв'язку використано жадібний алгоритм. Для локальної оптимізації застосовано всі способи організації роботи бджіл-фуражирів, окрім AN, оскільки постійні спроби додавання найближчого об'єкта розвідки до єдиного підмаршруту не дозволяють отримати бажану якість оптимізації початкового розв'язку.

Експериментальне дослідження. Експеримент спрямовано на визначення найкращого з розроблених алгоритмів для досліджуваної задачі. Кожним із запропонованих алгоритмів розв'язується 100 індивідуальних задач у 10 прогонах, що дозволяє оперувати даними 1000 розв'язаних під час порівняння результатів роботи. Даний процес має місце для кожної з розглянутих розмірностей.

Кількість місць дислокації ПЦО (розмірність) варіюється від 4 до 10. Кількість об'єктів розвідки в задачі визначається як розмірність помножена на коефіцієнт в діапазоні від 6 до 15. Отже,

для кожної з розглянутих розмірностей буде наявно 10 груп задач, які мають по 10 індивідуальних та відрізняються за кількістю об'єктів розвідки. Наприклад, для розмірності 10 кількість об'єктів розвідки індивідуальних задач буде варіюватись від 60 до 150 одиниць.

Експеримент проводився за умови, що жодна інша програма не запущена на комп'ютері з процесором Intel Core i7-1185G7 з частотою 3.00 – 4.80 ГГц, а також наявною оперативною пам'яттю DDR4 на 32 ГБ з постійною частотою 3200 МГц.

Таблиця 1 відображає результати роботи алгоритмів для розмірності 10. Зображені там характеристики є ключовими для визначення ефективності розроблених алгоритмів.

Таблиця 1. Результати експерименту для розмірності 10

Алгоритм	Відхилення значення ЦФ від ЦФ жадібного (%)			Час роботи (мс.)		
	Найкр.	Серед.	Найгір.	Макс.	Серед.	Мін.
Greedy	0	0	0	8.75	2.91	1.75
ABC-R-AN	77.71	40.83	-33.67	747.40	536.01	378.09
ABC-P-AN	79.18	48.86	-31.31	779.53	570.45	471.25
ABC-PN-AN	81.72	49.64	-27.88	731.15	509.68	488.90
ABC-R-RP	75.47	44.45	-25.59	1331.06	1082.33	731.00
ABC-P-RP	77.57	50.75	-24.40	1539.47	1119.88	940.62
ABC-PN-RP	78.63	51.67	-23.41	2206.13	1065.68	971.76
ABC-R-RPN	72.20	47.12	-26.37	1506.19	1032.25	624.71
ABC-P-RPN	75.17	51.35	-24.17	1497.63	1066.04	845.61
ABC-PN-RPN	82.33	51.84	-23.69	1670.43	1122.77	876.43
ABC-R-RPAAAN	86.14	51.56	-14.97	1684.92	1416.36	952.03
ABC-P-RPAAAN	89.10	52.34	-13.42	2014.57	1408.14	1182.65
ABC-PN-RPAAAN	91.47	52.99	-12.63	2051.16	1575.52	1224.42
ABC-R-RPNAAAN	87.07	52.84	-13.14	1440.79	1255.88	892.45
ABC-P-RPNAAAN	92.09	53.83	-12.63	1595.07	1385.08	1092.57
ABC-PN-RPNAAAN	95.27	54.36	-12.04	1910.80	1308.39	1106.50
Tabu-RP	71.78	40.77	-16.38	551.98	250.38	113.76
Tabu-RPN	77.72	41.16	-15.96	514.23	233.64	104.44
Tabu-RPAAAN	84.54	47.22	-14.25	648.15	272.92	113.45
Tabu-RPNAAAN	80.88	46.88	-14.85	647.62	253.93	104.24

Метрикою якості маршрутів польоту, що використовується в даному експерименті, є середнє відхилення значення цільової функції (ЦФ) алгоритмів від ЦФ жадібного. Результати експерименту для всіх розмірностей проілюстровано на одному графіку, який зображено на рисунку 1.

З графіку, зображеного на рисунку 1, добре видно, що зі збільшенням кількості місць дислокації ПЦО значення середнього відхилення ЦФ розроблених алгоритмів від ЦФ жадібного зростає. Отже, зі збільшенням розмірності розроблені алгоритми стають більш ефективними в порівнянні з жадібним. Значення зростають без суттєвих коливань, що свідчить про сталість запропонованих алгоритмів.

Результати роботи бджолиних алгоритмів вказують на те, що використання способу організації роботи бджіл-розвідників R призводить до найгірших результатів, спосіб P в середньому поступається PN на 1%. Розглядаючи локальну оптимізацію, можна прийти до висновку, що спосіб організації роботи AN виявився найгіршим. Майже однакові значення показали RP та RPN, відхилення варіюється в діапазоні до 1%. RPAAAN та RPNAAAN є кращими за попередні в середньому більш ніж на 2%, а другий ефективніше за перший майже на 1.5%. Жорсткіші обмеження за околону способам організації роботи призводять до кращих результатів.

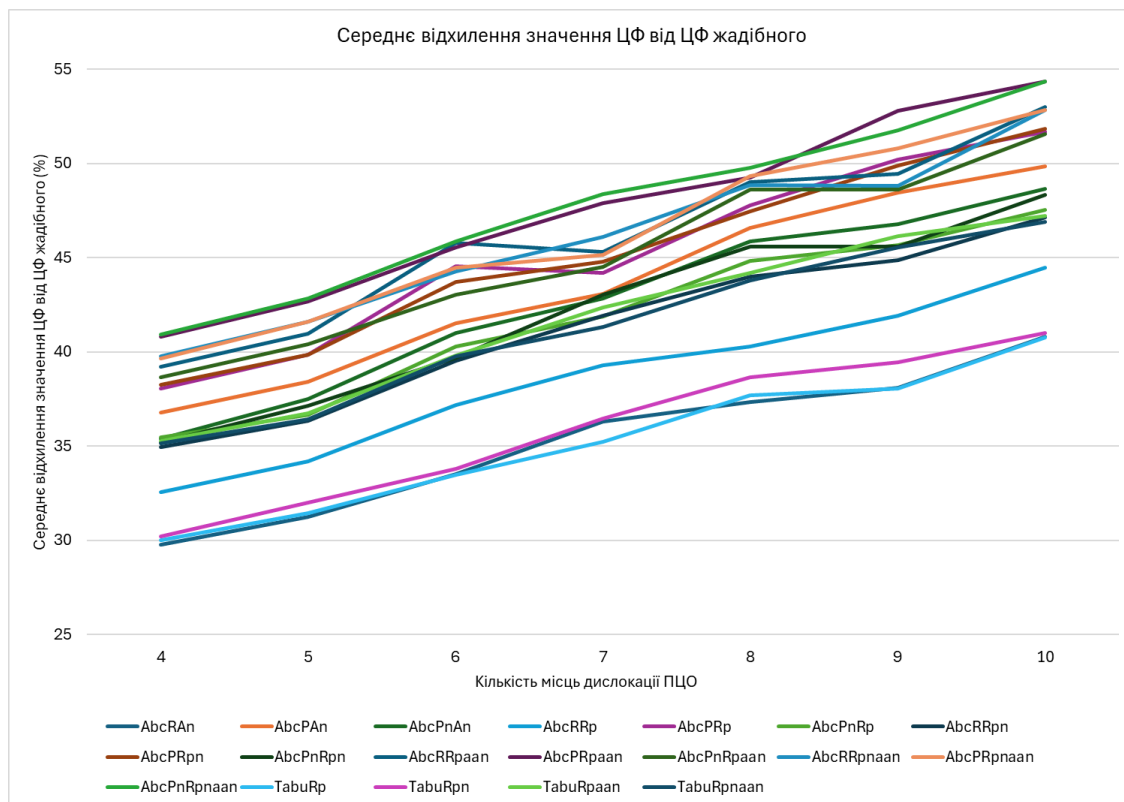


Рис.1. Середнє відхилення значення ЦФ алгоритмів від ЦФ жадібного

Для алгоритмів табу-пошуку попередні твердження, відносно локальної оптимізації, є схожими, проте RPAAN є навпаки кращим за RPNAAN приблизно на 0.5%. Відхилення в значеннях між кращим бджолиним і табу-пошуку складає приблизно 7%, а для розмірності 4 є лише біля 2.5%. Ці результати були очікуваними під час висування гіпотези для даного дослідження, проте для повного її підтвердження треба проаналізувати швидкодію. Графік середнього часу роботи алгоритмів проілюстровано на рисунку 2.

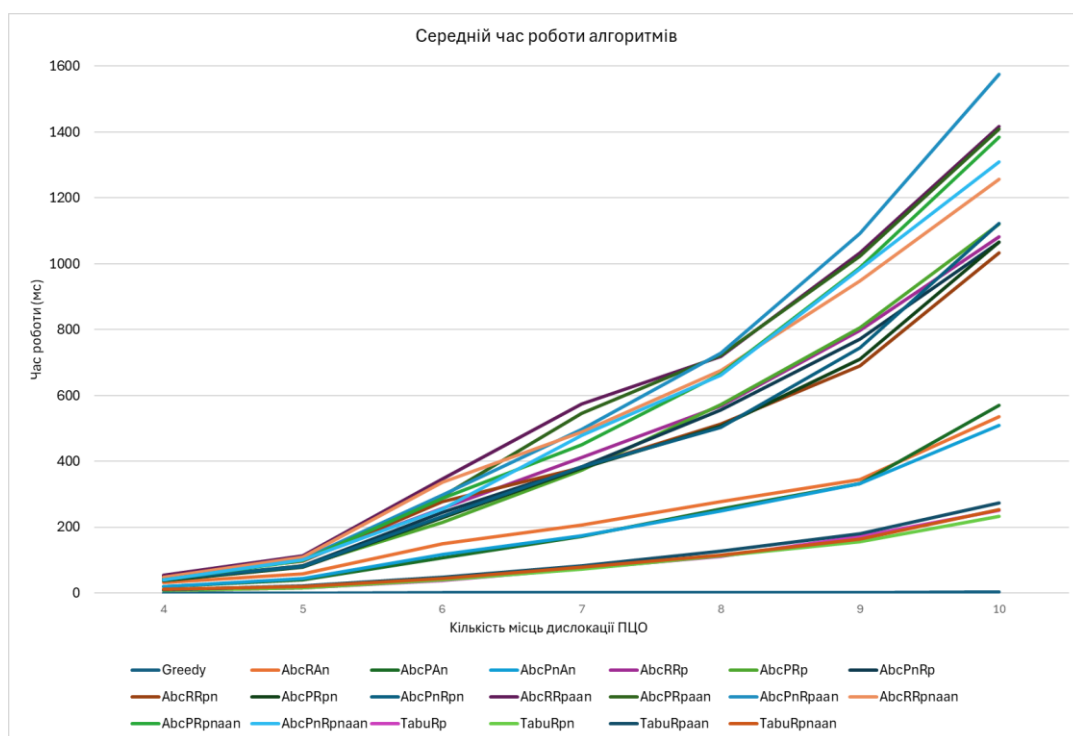


Рис. 2. Середній час роботи алгоритмів

Графік середнього часу роботи алгоритмів, зображений на рисунку 2, показує, що збільшення кількості місць дислокації ПЦО веде до стрімкого, але без суттєвих коливань, зростання часу необхідного бджолиним та алгоритмам табу-пошуку для знаходження маршрутів польоту. Алгоритми табу-пошуку показали себе як найшвидші, Tabu-RPAAAN виявився у майже 5 разів швидше за ABC-PN-RPNAAN, отже поставлену в даному дослідженні гіпотезу було доведено.

Можна зазначити, що використання алгоритму ABC-PN-RPNAAN призводить до найкращий розв'язків у порівнянні з розробленими. Tabu-RPAAAN є значно швидшим за попередній, але відрізняється трохи гіршим середнім відхиленням значення ЦФ від ЦФ жадібного.

Висновки. Розроблено та експериментально досліджено ефективність алгоритмів штучної бджолиної колонії та табу-пошуку. Встановлено, що алгоритм ABC-PN-RPNAAN гарантує найкращий розв'язок серед запропонованих, а Tabu-RPAAAN виявився значно, майже в 5 разів, швидшою альтернативою попередньому, проте відрізняється трохи гіршою, в середньому менш ніж на 4%, якістю побудови маршрутів.

Напрямами покращення якості роботи алгоритмів визначено наступні способи побудови маршрутів польоту: реверсивний (від кінцевого підмаршруту), випадковий (кожну ітерацію підмаршрут обирається випадково), у напрямку зустрічі (почергово з початку та кінця), повний (одразу весь маршрут польоту). Для побудови початкових допустимих розв'язків, а також локальної оптимізації, можна задіяти численну кількість існуючих евристичних правил або розробити їх самостійно.

Наукова новизна отриманих результатів полягає у вдосконаленні методів розв'язання задач складання маршрутів польоту для БПЛА з ПЦО. Практичне значення одержаних результатів зводиться до можливого використання розроблених алгоритмів у маршрутизації БПЛА з ПЦО в умовах реального світу.

Список бібліографічного опису

1. Zear A., Ranga V. (2019). Path planning of unmanned aerial vehicles: Current State and future challenges. *First International Conference on Sustainable Technologies for Computational Intelligence*, 409–419. DOI: 10.1007/978-981-15-0029-9_32.
2. Aggarwal S., Kumar N. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, 149, 270–299. DOI: 10.1016/j.comcom.2019.10.014.
3. Hooshyar M., Huang, Y.-M. (2023). Meta-heuristic algorithms in UAV path planning optimization: A systematic review (2018–2022). *Drones*, 7 (12), 687. DOI: 10.3390/drones7120687.
4. Cakir M. (2015). 2D path planning of UAVs with genetic algorithm in a constrained environment. 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO). DOI: 10.1109/icmsao.2015.7152235.
5. Roberge V., Tarbouchi M., Labonte G. (2013). Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, 9 (1), 132–141. DOI: 10.1109/tii.2012.2198665.
6. Liu J. (2022). An improved genetic algorithm for Rapid UAV Path Planning. *Journal of Physics: Conference Series*, 2216 (1), 012035. DOI: 10.1088/1742-6596/2216/1/012035.
7. Shen L., Wang Y., Liu K., Yang Z., Shi X., Yang X., Jing K. (2020). Synergistic path planning of multi-UAVs for air pollution detection of ships in ports. *Transportation Research Part E: Logistics and Transportation Review*, 144, 102128. DOI: 10.1016/j.tre.2020.102128.
8. Asma A., Sadok B. (2017). Dynamic distributed PSO joints elites in multiple robot path planning systems: Theoretical and practical review of new ideas. *Procedia Computer Science*, 112, 1082–1091. DOI: 10.1016/j.procs.2017.08.128.
9. Li B., Qi X., Yu B., Liu L. (2020). Trajectory planning for UAV based on improved ACO algorithm. *IEEE Access*, 8, 2995–3006. DOI: 10.1109/access.2019.2962340.
10. Perez-Carabaza S., Besada-Portas E., Lopez-Orozco J. A., de la Cruz J. M. (2018). Ant colony optimization for multi-UAV minimum time search in uncertain domains. *Applied Soft Computing*, 62, 789–806. DOI: 10.1016/j.asoc.2017.09.009.
11. Yu S., Li Y., Ma C. (2021). Research on UAV trajectory planning based on artificial bee colony algorithm. 2021 9th International Conference on Intelligent Computing and Wireless Optical Communications (ICWOC). DOI: 10.1109/icwoc52624.2021.9530209.
12. Tian G., Zhang L., Bai X., Wang B. (2018). Real-time dynamic track planning of multi-UAV formation based on improved artificial bee colony algorithm. 2018 37th Chinese Control Conference (CCC). DOI: 10.23919/chicc.2018.8482622.

References

1. Zear A., Ranga V. (2019). Path planning of unmanned aerial vehicles: Current State and future challenges. *First International Conference on Sustainable Technologies for Computational Intelligence*, 409–419. DOI: 10.1007/978-981-15-0029-9_32.

2. Aggarwal S., Kumar N. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, 149, 270–299. DOI: 10.1016/j.comcom.2019.10.014.
3. Hooshyar M., Huang, Y.-M. (2023). Meta-heuristic algorithms in UAV path planning optimization: A systematic review (2018–2022). *Drones*, 7 (12), 687. DOI: 10.3390/drones7120687.
4. Cakir M. (2015). 2D path planning of UAVs with genetic algorithm in a constrained environment. *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*. DOI: 10.1109/icmsao.2015.7152235.
5. Roberge V., Tarbouchi M., Labonte G. (2013). Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, 9 (1), 132–141. DOI: 10.1109/tii.2012.2198665.
6. Liu J. (2022). An improved genetic algorithm for Rapid UAV Path Planning. *Journal of Physics: Conference Series*, 2216 (1), 012035. DOI: 10.1088/1742-6596/2216/1/012035.
7. Shen L., Wang Y., Liu K., Yang Z., Shi X., Yang X., Jing K. (2020). Synergistic path planning of multi-UAVs for air pollution detection of ships in ports. *Transportation Research Part E: Logistics and Transportation Review*, 144, 102128. DOI: 10.1016/j.tre.2020.102128.
8. Asma A., Sadok B. (2017). Dynamic distributed PSO joints elites in multiple robot path planning systems: Theoretical and practical review of new ideas. *Procedia Computer Science*, 112, 1082–1091. DOI: 10.1016/j.procs.2017.08.128.
9. Li B., Qi X., Yu B., Liu L. (2020). Trajectory planning for UAV based on improved ACO algorithm. *IEEE Access*, 8, 2995–3006. DOI: 10.1109/access.2019.2962340.
10. Perez-Carabaza S., Besada-Portas E., Lopez-Orozco J. A., de la Cruz J. M. (2018). Ant colony optimization for multi-UAV minimum time search in uncertain domains. *Applied Soft Computing*, 62, 789–806. DOI: 10.1016/j.asoc.2017.09.009.
11. Yu S., Li Y., Ma C. (2021). Research on UAV trajectory planning based on artificial bee colony algorithm. *2021 9th International Conference on Intelligent Computing and Wireless Optical Communications (ICWOC)*. DOI: 10.1109/icwoc52624.2021.9530209.
12. Tian G., Zhang L., Bai X., Wang B. (2018). Real-time dynamic track planning of multi-UAV formation based on improved artificial bee colony algorithm. *2018 37th Chinese Control Conference (CCC)*. DOI: 10.23919/chicc.2018.8482622.