

УДК 629.7.01

Я. А. Кулик, к. т. н., доц.; Б. П. Книш, к. т. н., доц.; В. М. Папінов, к. т. н., проф.

МОДЕЛЮВАННЯ ОГЛЯДУ ТЕРИТОРІЇ ЗА ДОПОМОГОЮ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ НА ОСНОВІ АЛГОРИТМУ ОПТИМІЗАЦІЇ МУРАШИНОЇ КОЛОНІЇ

В роботі розглянуто проблему визначення довжини оптимального маршруту, який дозволить виконати огляд території за найкоротший час, що важливо у разі виконання моніторингу лісів, річок, транспорту, будівель, сільськогосподарських угідь, підрахунку об'єктів тощо. Для вирішення цієї проблеми запропоновано використання безпілотних літальних апаратів та різні методи оптимізації маршруту, серед яких можна виділити ймовірнісні методи пошуку рішення з мінімальним часом (MTS). А саме евристику, крос-ентропійну оптимізацію, байєсівський алгоритм оптимізації та генетичні алгоритми, методи оптимізації ройового інтелекту на основі спостережень за живою природою (оптимізація мурашиних колоній (ACO), штучні колонії бджіл, зграї криланів тощо).

Запропоновано використання алгоритму оптимізації мурашиної колонії, оскільки це дозволяє дотримуватись балансу між різними параметрами, а саме обчислювальними ресурсами безпілотних літальних апаратів та оптимальною довжиною маршруту його руху. Проведено експериментальні дослідження огляду території за допомогою безпілотних літальних апаратів за різної кількості ітерацій на основі алгоритму оптимізації мурашиної колонії шляхом моделювання в середовищах WeBots та tsp-problem-ga-aco-comparisson, які є симуляторами різноманітних пристроїв, зокрема безпілотних літальних апаратів, що компенсує вплив зовнішнього середовища на керування польотом безпілотних літальних апаратів вбудованими програмними засобами для збереження поточного курсу руху безпілотних літальних апаратів.

Встановлено, що використання алгоритму оптимізації мурашиної колонії дозволяє виконувати огляд території за менший час, ніж при використанні генетичного алгоритму, який являє собою стандартний алгоритм багатьох систем керування за замовчуванням, знаходячи при цьому баланс між оптимальністю маршруту та обчислювальними ресурсами.

Ключові слова: *безпілотні літальні апарати, мурашині алгоритми, оптимізація мурашиної колонії, огляд території, алгоритм.*

Вступ

Останнім часом розвиток дистанційних пристроїв та засобів набув швидкого темпу. Особливо популярним стало застосування безпілотних літальних апаратів (БПЛА), які виконують задачі, пов'язані з оглядом території. Прикладом може бути моніторинг лісів, річок, транспорту, будівель, сільськогосподарських угідь, підрахунок об'єктів тощо [1]. Зазвичай задача огляду території сформульована приблизно. Оператор БПЛА вирішує її шляхом введення послідовності координат території для обстеження з допомогою БПЛА. Для зменшення навантаження на оператора існують системи автоматичного розрахунку траєкторії польоту з вибором оптимального маршруту [2, 3]. Проте ці системи часто працюють недостатньо ефективно і використовують стандартні методи руху, не враховуючи досягнення в галузі оптимізації [4]. Тому виникає необхідність у підвищенні ефективності руху БПЛА під час огляду території.

Час польоту БПЛА обмежений ємністю акумулятора або запасом пального і огляд території не повинен перевищувати цей час. Тому виникає необхідність оптимально спланувати проходження маршруту польоту БПЛА.

Для визначення довжини оптимального маршруту, яка дозволить виконати огляд території за найкоротший час, існують різні методи оптимізації. Серед них можна виділити ймовірнісні методи пошуку рішення з мінімальним часом (MTS) – евристика, крос-ентропійна оптимізація, байєсівський алгоритм оптимізації та генетичні алгоритми (GA) [5], група методів оптимізації ройового інтелекту на основі спостережень за живою природою: оптимізації мурашиних колоній (ACO) [6], штучної колонії бджіл, зграї криланів тощо [7].

Метод оптимізації мурашиних колоній підходить для задач різної складності, де потрібно дотримуватись балансу між різними параметрами. В нашому випадку – це обчислювальні ресурси БПЛА та оптимальна довжина маршруту його руху. Задача огляду території, заданої групою координат, близька до задачі комівояжера [8], з якою також досить добре справляється метод оптимізації мурашиних колоній, який реалізується за допомогою відповідного алгоритму.

Метою роботи є підвищення ефективності огляду території за допомогою БПЛА на основі алгоритму оптимізації мурашиної колонії.

Алгоритм оптимізації мурашиної колонії для огляду території за допомогою БПЛА

Оптимізація мурашиної колонії (ACO) — метаевристичний алгоритм на основі рою, який моделює поведінку мурашиних колоній у природі під час пошуку їжі [9].

Штучні мурахи, на відміну від біологічних, рухаються через дискретне середовище, визначене вузлами (вершинами), і мають пам'ять, а феромони відкладаються на віртуальній феромонній карті. Переходячи від одного вузла до іншого, мурахи залишають сліди феромонів на шляху між вузлами.

Сліди феромонів приваблюють інших мурах, які виділяють більше феромонів, що призводить до накопичення слідів феромонів. Негативний зворотний зв'язок використовується через випаровування феромонів і стримує мурах від того самого шляху та дозволяє постійно шукати кращі рішення.

Використання алгоритму оптимізації мурашиної колонії для БПЛА призведе до того, що найбільш інтенсивні феромонні доріжки будуть найбільш імовірним курсом для БПЛА. Причому чим оптимальніший шлях, тим більш імовірно його використовуватимуть інші мурахи, а отже інші БПЛА.

БПЛА здійснює огляд території, заданої групою координат, з мінімальними витратами часу, оскільки час польоту обмежений ємністю акумулятора. Ця задача близька до математичної задачі комівояжера, а саме для заданого набору міст із відомою відстанню між ними необхідно знайти найкоротший шлях, який дозволить відвідати кожне місто лише один раз [8]. Її можна представити у вигляді графа з N вершин, що представляють міста, і набір E ребер, які повністю з'єднують вершини N , з довжиною d_{ij} ($(i; j) \in E$), яка є відстанню між містами i та j . По суті для вирішення цієї задачі необхідно знайти гамільтоновий цикл графа $G = (N; E)$ для усіх $n = |N|$ вершин G , і його довжина визначається сумою довжин усіх ребер, з яких він складається. Оптимальним розв'язком є перестановка π вершин $1, 2, \dots, N$ так, щоб довжина $f(\pi)$ була мінімальною і визначалася як

$$f(\pi) = \sum_{i=1}^N d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)}, \quad (1)$$

де $d_{\pi(i)\pi(i+1)}$ – довжина ребра між містом i та містом $i+1$ для всіх комбінацій перестановок π від 1 до N ; $d_{\pi(n)\pi(1)}$ – довжина ребра між містом n та містом 1 для всіх комбінацій перестановок π від 1 до N .

Видимість вершини i відносно вершини j обернено пропорційна довжині:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (2)$$

БПЛА виконує огляд території на основі алгоритму оптимізації мурашиної колонії, що генерує колонію штучних мурах, які пересуваються між вершинами у пошуках оптимального маршруту. Алгоритм складається з двох основних етапів, які повторюються: проходження маршруту та оновлення феромонів (рис. 1).

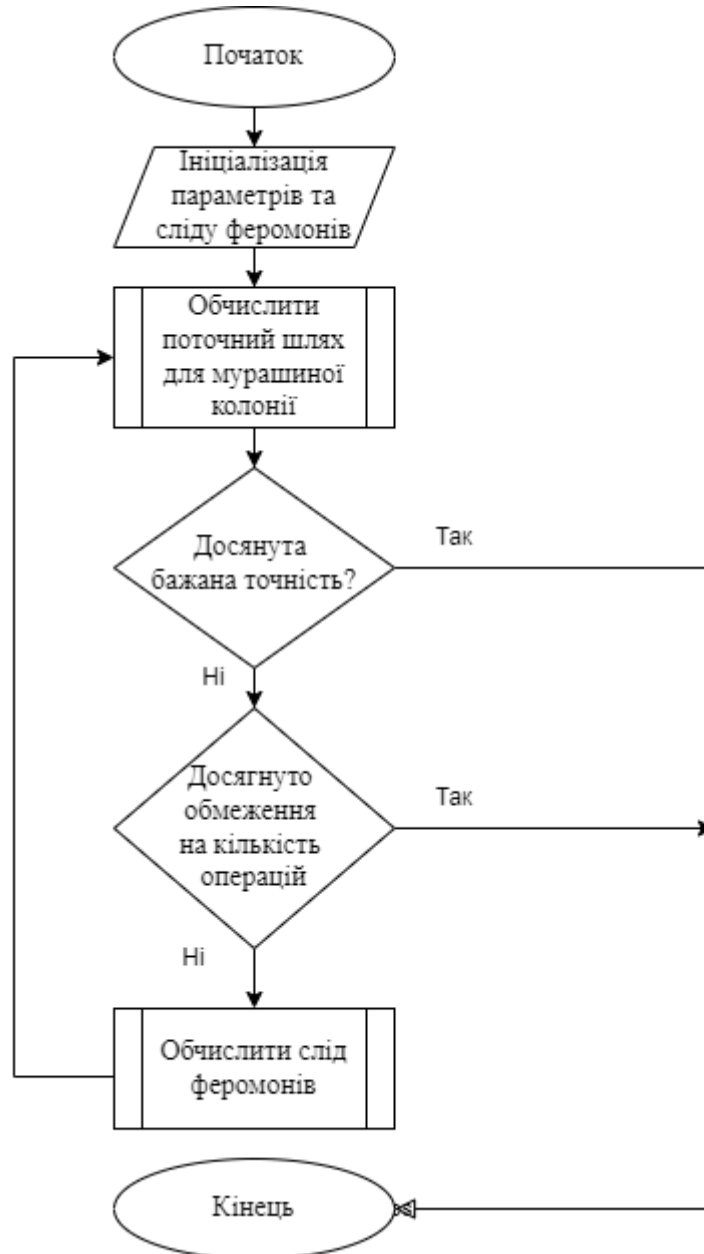


Рис. 1. Схема алгоритму оптимізації мурашиної колонії

На кожному кроці побудови графа штучна мураха k застосовує ймовірнісне правило вибору дії, яке називається «правилом рулетки». Кожна вершина пов'язана з певною ймовірністю бути випадково обраною із набору вершин, що раніше не відвідувались.

Алгоритм зупиняється, коли знайдено задовільне рішення або коли досягнуто максимальну кількість ітерацій [8].

Моделювання огляду території з допомогою БПЛА в середовищі WeBots

Моделювання огляду території за допомогою БПЛА на основі алгоритму оптимізації мурашиної колонії проводилося в середовищі WeBots. Це середовище дозволяє виконувати моделювання програмного керування пристроїв, зокрема БПЛА. Керування є спрощеним, тобто вплив зовнішнього середовища вважається прогнозованим, погодні умови – нормальні та зручні для польоту. WeBots створює умови для моделювання, коли вплив зовнішнього середовища на керування польотом БПЛА компенсується вбудованими програмними засобами для збереження поточного курсу руху БПЛА.

Моделювання дозволяє суттєво покращити ефективність реальної системи, знайти недоліки, передбачити структурні зміни на етапі проектування та виявити стратегічні недоліки, які несуть довгостроковий характер.

У середовищі для моделювання було створено тестовий віртуальний світ, де можлива імітація виконання завдання огляду території для БПЛА. Перевагою використання віртуального середовища є відсутність ризиків матеріальних втрат у випадку помилок оператора БПЛА або програми керування, можливість проведення значно більшої кількості досліджень, спрощений збір даних та можливості керування часом у середовищі.

Для експериментальної перевірки ефективності огляду території за допомогою БПЛА на основі алгоритму оптимізації мурашиної колонії створено середовище у вигляді віртуального світу (рис. 2), вплив якого на характеристики БПЛА незначний.



Рис. 2. Віртуальний світ, створений в середовищі WeBots для експериментальної перевірки ефективності огляду

території за допомогою БПЛА на основі алгоритму оптимізації мурашиної колонії

Віртуальний світ містить територію з різним ландшафтом, деревами та техногенними об'єктами. В якості моделі БПЛА обрано Mavic 2 Pro, який має хорошу систему відеоспостереження, вбудовану систему уникнення зіткнень та максимальний час польоту 30 хвилин. Ділянка території для огляду задавалась у вигляді набору координат, які були вхідними даними для БПЛА, що виконував огляд цієї території на основі програми керування, яка будує маршрут після оптимізації набору координат алгоритмом оптимізації мурашиної колонії (рис. 1) на основі формул (3 – 6).

Оцінювання ефективності огляду території за допомогою БПЛА на основі алгоритму оптимізації мурашиної колонії

Отримати характеристику, яка б слугувала оцінювальним критерієм ефективності систем огляду територій є нетривіальною задачею і єдиного та універсального рішення для цього не існує [10]. Тому для оцінки ефективності цієї системи будемо враховувати, що час роботи БПЛА залежить від ємності акумулятора (або бензобака) та є однією з основних характеристик, яку складно змінити без значного збільшення витрат або погіршення інших характеристик. Таким чином, можливість виконати огляд визначається часом розрядження акумулятора, тому час є одним з головних критеріїв ефективності.

Також ефективний огляд території за допомогою БПЛА вимагає виконання задачі по плануванню шляху БПЛА, що передбачає використання функції витрат:

$$J_G = K \cdot \sum_{i=1}^{N+1} d_{ij} + C = K \cdot \sum_{i=1}^{N+1} d_{ij} + K1 \cdot \sum_{l=1}^M h_l + K2 \quad (3)$$

де J_G – витрати на політ; d_{ij} – довжина сегментів шляху; C – величина, яка об'єднує вартість ризику та вартість стабілізації польоту; N – кількість координат для огляду території; K – коефіцієнт затрат на одиницю довжини при польоті; $K1$ – одиниця зміни висоти польоту БПЛА (складається з витрат пального або електроенергії акумулятора БПЛА); M – кількість змін висоти для БПЛА; h – поточна висота БПЛА під час польоту; $K2$ – величина, яка об'єднує вартість ризику під час польоту і розраховується на основі статистичних даних вартості ремонту БПЛА при пошкодженнях [11].

Як видно з формули (7) найменш затратним буде БПЛА, який виконує огляд території за найкоротшим шляхом і, відповідно, за мінімальний час.

Виконання оптимізації пройденого шляху БПЛА для огляду території за найкоротший час здійснювалося за допомогою програмного забезпечення tsp-problem-ga-aco-comparisson [12], яке дозволяє генерувати набір вершин та шукати шляхи обходу цих вершин за допомогою різних методів, зокрема за допомогою алгоритму оптимізації мурашиної колонії.

На рис. 3 показано оптимальні маршрути БПЛА, що виконує огляд території за допомогою алгоритму оптимізації мурашиної колонії за згенерованими випадковими 24-ма координатами.

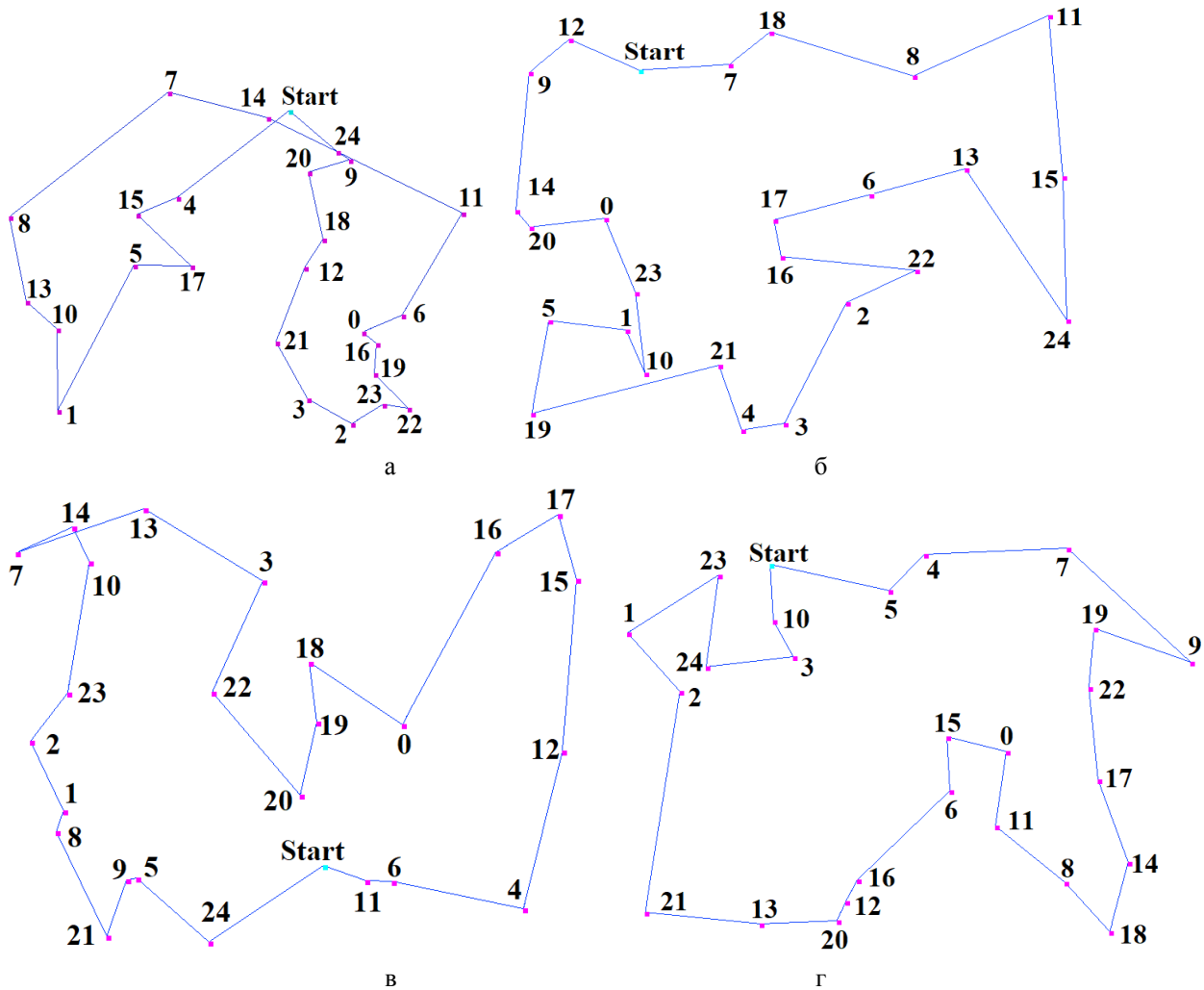


Рис. 3. Оптимальні маршрути БПЛА відносно початку руху Start за згенерованими випадковими 24-ма координатами: а – перший маршрут; б – другий маршрут; в – третій маршрут; г – четвертий маршрут

На рис. 4 – 7 показано графіки довжин оптимального маршруту залежно від кількості ітерацій на основі алгоритму оптимізації мурашиної колонії, відповідно до рис. 3, та генетичного алгоритму. Порівняння проводиться саме з генетичним алгоритмом, скільки він досить широко використовується для знаходження оптимального шляху.

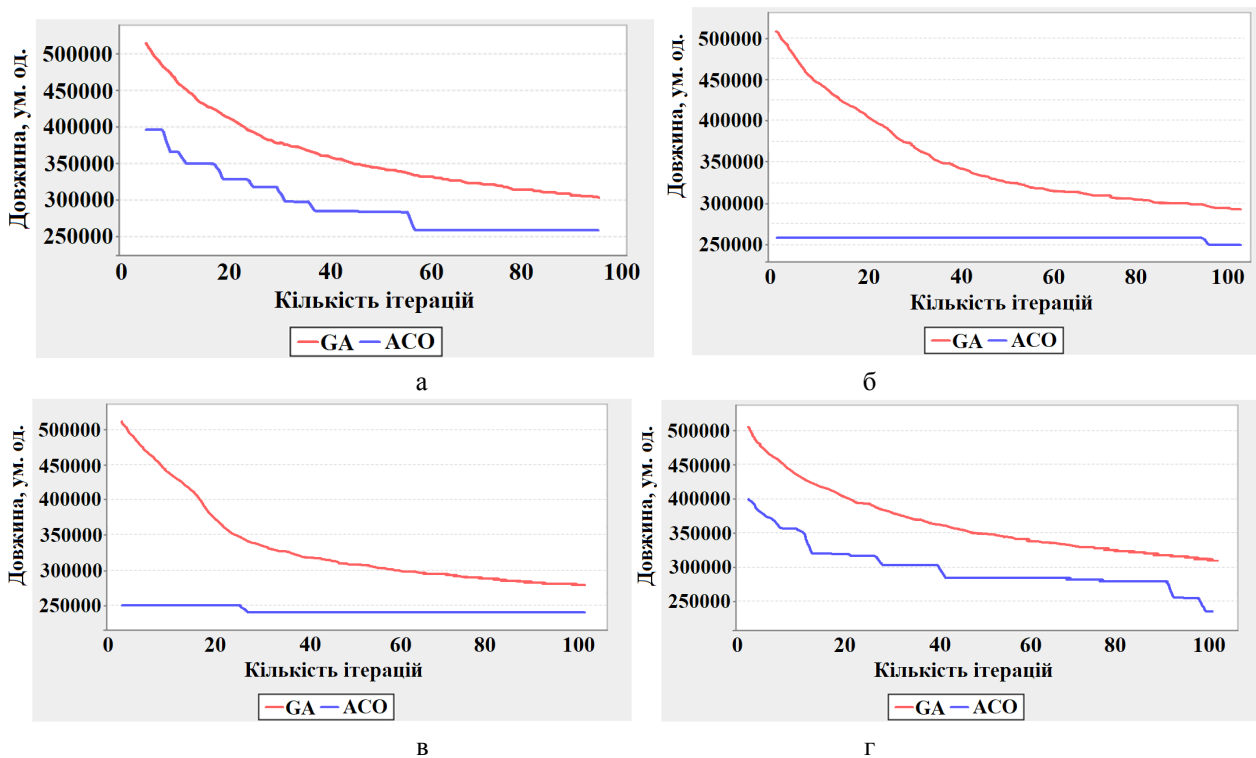


Рис. 4. Графіки довжини першого оптимального маршруту залежно від кількості ітерацій на основі алгоритму оптимізації мурашиної колонії (ACO) та генетичного алгоритму (GA): а – перший маршрут; б – другий маршрут; в – третій маршрут; г – четвертий маршрут

Графіки на рис. 4 отримані за таких параметрів:

- кількість мурах у колонії $m=50$,
- швидкість випаровування феромону $\rho=0,1$,
- коефіцієнт інтенсивності феромону $\alpha=0,1$,
- коефіцієнт видимості феромону $\beta=0,8$,
- дистанція видимості феромону – 2,
- кількість точок маршруту (координат для відвідування) – $n=25$,
- кількість ітерацій (критерій зупинки обчислень) – 100.

З рис. 4 видно, що алгоритм оптимізації мурашиної колонії зі збільшенням кількості ітерацій зменшує довжину оптимального маршруту. Таким чином, для першого маршруту зменшення відбувається з 400 тис. ум. од. за ітерації 5 до 260 тис. ум. од. за ітерації 55; для другого з 260 тис. ум. од. за ітерації 3 до 250 тис. ум. од. за ітерації 95; для третього з 250 тис. ум. од. за ітерації 3 до 240 тис. ум. од. за ітерації 25; для четвертого з 400 тис. ум. од. за ітерації 3 до 240 тис. ум. од. за ітерації 100. Як видно, найбільше скорочення довжини відбувається на початкових ітераціях, тоді як на останніх ітераціях скорочення оптимальної довжини майже не відбувається. Тому для зменшення обчислювальної складності методу є сенс обмежити кількість ітерацій.

Також, для порівняння, на рис. 4 додано графіки зменшення оптимального маршруту на основі генетичного алгоритму. Таким чином, для першого маршруту зменшення відбувається з 510 тис. ум. од. за ітерації 5 до 300 тис. ум. од. за ітерації 95; для другого з 510 тис. ум. од. за ітерації 3 до 290 тис. ум. од. за ітерації 100; для третього з 510 тис. ум. од. за ітерації 3 до 275 тис. ум. од. за ітерації 100; для четвертого з 510 тис. ум. од. за ітерації 3 до 310 тис. ум. од. за ітерації 100. Видно, що алгоритм оптимізації мурашиної колонії дозволяє побудувати коротший маршрут, а саме для першого маршруту – на 40 тис. ум. од., для другого – на 40 тис. ум. од., для третього – на 35 тис. ум. од., для четвертого – на 50 тис. ум.

од. Таким чином, алгоритм оптимізації мурашиної колонії дозволяє виконати огляд території за менший час та є більш ефективним.

Алгоритм оптимізації мурашиної колонії знаходить рішення за прийнятний час, проте не гарантує, що знайдений розв'язок не є субоптимальним. Зменшити ймовірність знаходження субоптимального розв'язку можна збільшенням кількості ітерацій, проте навіть багатократне збільшення кількості ітерацій в більшості випадків призведе до незначного скорочення довжини маршруту або не призведе взагалі.

Висновки

В роботі запропоновано використовувати алгоритм оптимізації мурашиної колонії для розрахунку оптимального маршруту огляду території за допомогою БПЛА.

Проведено експериментальні дослідження оптимального маршруту огляду території шляхом моделювання з використанням середовища WeBotста tsp-problem-ga-aso-comparisson. Дослідження показали, що використання алгоритму оптимізації мурашиної колонії дозволяє виконувати огляд території за менший час, ніж при використанні генетичного алгоритму, знаходячи при цьому баланс між оптимальністю маршруту та обчислювальними ресурсами. Варто також враховувати, що знайдене рішення може бути субоптимальним.

СПИСОК ЛІТЕРАТУРИ

1. Книш Б. П. Оцінювання стану об'єктів безпілотними літальними апаратами / Б. П. Книш, М. А. Алексєєв // Вісник Вінницького політехнічного інституту. – 2019. – № 2. – С. 58 – 65.
2. Classification of multi-UAV architectures. In Handbook of unmanned aerial vehicles / I. Maza, A. Ollero, E. Casado [et al.] // Springer. – 2015. – P. 953 – 975.
3. Modelling the one channel systems of a delivery of goods provided by unmanned aerial vehicles / R. N. Kvetny, Y. A. Kulyk, B. P. Knysh [et al.] // INTL Journal of electronics and telecommunications. – 2020. – Volume 2020, № 3. – P. 487 – 492. – DOI: <https://doi.org/10.24425/ijet.2020.134003>.
4. Senanayake M. Search and tracking algorithms for swarms of robots: A survey / M. Senanayake, I. Senthoran, J. C. Barca // Robotics and Autonomous Systems. – 2016. – Volume 75. – P. 422 – 434.
5. Jones K. O. Comparison of ant colony optimisation and differential evolution / K. O. Jones, A. Bouet // International conference on Computer systems and technologies (ACM). – 2007. – P. 25.
6. Carabaza S. P. Ant Colony Optimization for Multi-UAV Minimum Time Search in Uncertain Domains / S. P. Carabaza, E. Besada, J. A. Lopez-Orozco // Applied Soft Computing. – 2018. – Volume 62. – P. 789 – 806. – DOI: [10.1016/j.asoc.2017.09.009](https://doi.org/10.1016/j.asoc.2017.09.009).
7. Cimino M. G. Combining stigmergic and flocking behaviors to coordinates swarms of drones performing target search / M. G. Cimino, A. Lazzeri, G. Vaglini // In Information, Intelligence, Systems and Applications (IISA). – 2015. – P. 1 – 6.
8. Shtovba S. D. Ant Algorithms : Theory and Applications / S. D. Shtovba // Program Comput Soft. – 2005. – Volume 31. – P. 167 – 178. – DOI: <https://doi.org/10.1007/s11086-005-0029-1>.
9. Dorigo M. Ant Colony Optimization / M. Dorigo, T. Stützle // The MIT Press. – 2004. – P. 305. – DOI: <http://dx.doi.org/10.1007/b99492>.
10. Birdsey L. Identifying self-organization and adaptability in complex adaptive systems / L. Birdsey, C. Szabo, K. Falkner // In Self-Adaptive and Self-Organizing Systems (SASO). – 2017. – P. 131 – 140.
11. Lin N. A novel improved bat algorithm in UAV path planning / N. Lin, J. Tang, X. Li, L. Zhao // Computers, Materials & Continua. – 2019. – Volume 61, № 1. – P. 323 – 344. – <https://doi.org/10.32604/cmc.2019.05674>.
12. Yaseen M. Experimental Comparison between Genetic Algorithm and Ant Colony Optimization on Traveling Salesman Problem / M. Yaseen, J. Razia, Md. T. Rahman // International Journal of Scientific Research in Science, Engineering and Technology. – February 2021. – Volume 8, Issue 1. – P. 155 – 162. – DOI: <https://doi.org/10.32628/IJSRSET218135>.

Стаття надійшла до редакції 18.09.2022.

Стаття пройшла рецензування 25.09.2022.

Кулик Ярослав Анатолійович – к. т. н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій.

Книш Богдан Петрович – к. т. н., доцент кафедри загальної фізики, tutmos-3@i.ua.

Папінов Володимир Миколайович – к. т. н., професор кафедри автоматизації та інтелектуальних інформаційних технологій.

Вінницький національний технічний університет.