СИСТЕМНИЙ
АНАЛІЗ

В.П. ГОРБУЛІН, Л.Ф. ГУЛЯНИЦЬКИЙ, І.В. СЕРГІЄНКО

УДК 519.8

# ОПТИМІЗАЦІЯ МАРШРУТІВ КОМАНДИ БПЛА ЗА НАЯВНОСТІ АЛЬТЕРНАТИВНИХ ТА ДИНАМІЧНИХ ДЕПО

**Анотація.** Запропоновано змістовну постановку та математичні моделі проблем оптимізації маршрутів команди безпілотних літальних апаратів (БПЛА) під час обстеження чи обслуговування заданої множини об'єктів за наявності альтернативних та динамічних депо (місць старту чи/або фінішу) і ресурсних обмежень. До таких проблем належать, зокрема, планування польотів БПЛА, що використовують рухомі платформи як депо. Критеріями оптимізації є як сумарна довжина маршрутів, так і кількість задіяних БПЛА. Розроблено та реалізовано алгоритми розв'язування сформульованих задач комбінаторної оптимізації, які базуються на оптимізації мурашиними колоніями, табу пошуку та повному переборі. Наведено результати обчислювального експерименту.

**Ключові слова:** оптимізація маршрутів, БПЛА, мурашині алгоритми, динамічні депо, табу пошук, ґремліни.

## ВСТУП

Останнім часом поширилася тенденція до збільшення використання безпілотних апаратів, зокрема безпілотних літальних апаратів (БПЛА). Світовий досвід розвитку безпілотної авіації, насамперед, у таких розвинених країнах, як США, Німеччина, Ізраїль, показує, що БПЛА в найближчі 10–15 років зможуть виконувати більшість завдань, які сьогодні пов'язані з пілотованими засобами. Уже зараз БПЛА використовують для контролю технічного стану, безпеки та процесу функціонування різних об'єктів і систем, зокрема в екології, сільському чи лісовому господарстві, на залізничному транспорті, під час організації морських пошуково-рятувальних операцій, гасінні пожеж, доставлянні кореспонденції та товарів, а також істотне значення БПЛА мають для військової галузі [1–3].

Задачі оптимізації маршрутів як одиночних БПЛА, так і їхніх команд [4] належать до NP-повних задач, через те точні алгоритми не можна використовувати на практиці. Це стосується навіть найбільш «простих» і досліджених задач, коли потрібно знайти оптимальний маршрут руху БПЛА між двома точками: старту і фінішу (див., наприклад, [2, 5]). Дослідники запропонували низку метаевристичних методів, серед яких алгоритми оптимізації мурашиними колоніями [6–8], генетичні алгоритми [9, 10], а також алгоритми штучних бджолиних колоній [11, 12] та оптимізації потоком частинок [13].

У роботі розглянуто спеціальні проблеми оптимізації маршрутів БПЛА, що діють як команда, яка має завдання обстежити задану множину об'єктів. Особливістю є те, що БПЛА у процесі виконання завдання можуть стартувати та приземлюватися в альтернативних пунктах (депо), які в свою чергу можуть рухатися заданою траєкторією, а закріплення за ними відбувається під час планування завдання. Необхідно знайти маршрути БПЛА мінімальної сумарної довжини з можливою мінімізацією їхньої кількості, що забезпечує скорочення часу виконання завдання і економію енергоресурсів, та з урахуванням певних обмежень.

### 1. ЗАГАЛЬНА ПРОБЛЕМАТИКА ОПТИМІЗАЦІЇ МАРШРУТІВ БПЛА З ДЕКІЛЬКОМА ДЕПО

Розглянемо проблеми пошуку оптимальних маршрутів, які полягають у тому, що задана команда БПЛА, яка може стартувати із заданих точок пуску та закінчувати маршрут у різних місцях (депо), має облетіти низку заданих об'єктів (точок на місцевості) з мінімізацією сумарної довжини маршрутів чи тривалості польотів за умов, що кожен об'єкт обстежується одним і лише одним БПЛА і всі об'єкти мають бути обстеженими. До того ж часто слід враховувати ще й додаткові обмежувальні умови (дальність польоту без підзаряджання чи дозаправляння, вантажопідйомність, мінімальний радіус розвороту, погодні умови тощо) [1, 14–16].

У деяких випадках постановка таких проблем є подібною до відомого класу задач маршрутизації транспортних засобів [17, 18], зокрема, з кількома депо [19, 20]. Зважаючи на це, в подальшому вживатимемо і відповідні терміни, що мають конкретний сенс у контексті досліджуваних проблем [6, 21]. Так, зокрема, вважатимемо, що депо — це місце старту та/або приземлення чи дозаправляння БПЛА.

Задачі маршрутизації БПЛА за типом депо можна розділити на такі класи:

1) старт і приземлення відбуваються в одному депо, інакше кажучи, воно є водночас і стартовим, і фінішнім;

2) старт і приземлення можуть відбуватися в декількох депо, які також водночас є і стартовими, і фінішними, а кожний БПЛА закріплено за конкретним депо;

3) існує дві групи депо: стартові, за якими вже закріплено конкретні БПЛА, та фінішні, в яких можуть завершувати маршрут БПЛА, без апріорного припису до них задіяних БПЛА;

4) як і в п. 3, існують стартові та фінішні депо, але конкретний БПЛА може використовувати довільне стартове депо, а також оптимізувати вибір фінішного депо в процесі виконання завдання.

Задачі класів 2–4 назвемо задачами з альтернативними депо. Клас задач 4 виникає тоді, коли використовують рухомі депо, наприклад, літаки-носії чи пересувні або плавучі платформи. Клас 4 назвемо задачами із динамічними депо.

Загалом розв'язування задач маршрутизації команди БПЛА можна поділити на три етапи:

— розбиття множини об'єктів на диз'юнктні підмножини, за кожною з яких закріплюють один БПЛА;

— побудова маршруту кожного БПЛА з оптимізацією його довжини;

— уточнення маршрутів за необхідності (у разі несуттєвих змін обставин, урахування погодних умов у конкретний період тощо).

Зауважимо, що в деяких випадках (наприклад, із застосуванням оптимізації мурашиними колоніями) перший та другий етапи об'єднують в одну оптимізаційну задачу [5, 6].

### 2. КОМБІНАТОРНА МОДЕЛЬ ПРОБЛЕМ МАРШРУТИЗАЦІЇ З АЛЬТЕРНАТИВНИМИ ДЕПО

Розглянемо спочатку задачу маршрутизації, коли кожний БПЛА апріорі закріплений за конкретним стартовим депо, а його приземлення може відбуватися в одному із альтернативних фінішних депо.

Позначимо $n$ кількість досліджуваних об'єктів, $m$ — кількість БПЛА, $b^i$ — стартове депо для $i$-го БПЛА, $E^i$ — множину альтернативних фінішних депо для $i$-го БПЛА, $R^i$ — польотний ресурс $i$-го БПЛА, $i=1, \ldots, m$. Нехай $Y=\{y_1, \ldots, y_n\}$ — множина досліджуваних об'єктів. Зауважимо, що можливий випадок, коли $b^i \in E^i$.

Розглянемо розбиття множини $Y$ на спеціальні блоки розбиття $Y^i$: вони є або порожніми, або впорядкованими послідовностями елементів $Y$ (тобто розміщеннями) і задовольняють умовам $\bigcup_{i=1}^m Y^i = Y$, $Y^i \cap Y^j = \varnothing$.

Таким чином, такі спеціальні блоки розбиття можуть відрізнятися один від одного як елементами, так і їхнім упорядкуванням. Вони визначатимуть польотні завдання для кожного БПЛА: як множину об'єктів, які варто обстежити, так і порядок їхнього обходу.

Маршрут $i$-го БПЛА подамо вектором $X^i = (x_1^i, \ldots, x_{l(i)}^i)$, де $x_1^i = b^i$, $x_{l(i)}^i = e$, $e \in E^i$, а $(x_2^i, \ldots, x_{l(i)-1}^i)$ — один із зазначених блоків розбиття $Y^i$ з упорядкуванням його елементів.

Довжину маршруту $i$-го БПЛА в цих позначеннях запишемо так:

$$f(X^i) = \begin{cases} \sum\limits_{j=1}^{l(i)-1} d(x_j^i, x_{j+1}^i), & \text{якщо } Y^i \neq \varnothing, \\ 0 & \text{в іншому випадку,} \end{cases}$$

де $d(x_j^i, x_{j+1}^i)$ — відстань між відповідними пунктами (депо, об'єктами).

Тоді цільову функцію досліджуваної задачі оптимізації маршрутів БПЛА можемо записати

$$F(X) = \sum_{i=1}^m f(X^i), \tag{1}$$

де $X = (X^1, \ldots, X^m)$ — елемент простору розв'язків задачі.

Ресурсні обмеження визначають умову допустимості всього розбиття $Y$:

$$f(X^i) \leq R^i, \ i=1, \ldots, m. \tag{2}$$

Зазначимо, що запропонована математична модель маршрутизації (1), (2) в окремих випадках включає також формалізацію задач, що мають прикладне значення:

— усі БПЛА стартують лише з одного депо і повертаються назад;

— для кожного БПЛА вказано одне конкретне депо із декількох наявних місць старту і завершення маршруту [14];

— для кожного БПЛА зафіксовано одне із декількох наявних депо для старту, а повернення може відбуватися до будь-якого з цих чи окремих депо [6].

З іншого боку, запропонований підхід до формалізації став кроком до розроблення математичних моделей задач маршрутизації у випадку, коли літальні апарати (БПЛА чи спеціальні крилаті ракети багаторазового використання) запускають зі спеціального літака-носія чи іншої рухомої платформи, про що йдеться далі.

## 3. МАТЕМАТИЧНА МОДЕЛЬ ПРОБЛЕМ МАРШРУТИЗАЦІЇ З ДИНАМІЧНИМИ ДЕПО

Припустимо, що старт БПЛА і завершення їхніх маршрутів можуть здійснюватися за використання певної рухомої платформи, яку назвемо динамічними депо [21]. Наприклад, літак-носій з проекту DARPA [22], де як безпілотні авіаційні системи, названі ґремлінами, можуть використовуватися не лише дрони, а і спеціальні керовані крилаті ракети.

Отже, розглянемо проблему пошуку оптимальних маршрутів для команди БПЛА, коли вони можуть стартувати з різних точок та закінчувати маршрут у різних місцях (зонах відомої траєкторії рухомого депо). До того ж приймання БПЛА рухомим депо може здійснюватися в декількох територіально заданих зонах. Вважатимемо також, що характеристики кожної зони вибирають таким чином, що у конкретній зоні допускається приймання заданої кількості БПЛА, тобто у одній зоні можуть завершувати свій маршрут всі чи частина БПЛА. Кожну зону будемо ідентифікувати її центроїдом. Отже, маршрут кожного з БПЛА повинен закінчуватися в одній з таких зон, але конкретна зона для нього не вказується. Кількість місць старту БПЛА може перевищувати їхню кількість, але це створює додаткові можливості щодо вибору місць (чи моментів) старту. Також зазначимо, що в процесі розв'язування задачі маршрутизації може оптимізуватися і кількість задіяних БПЛА.

Отже, за умови існування декількох можливих зон приймання БПЛА і місць їхнього старту, а також завершення маршрутів кожного БПЛА лише в одній з виділених зон варто визначити маршрути БПЛА з оптимізацією сумарної довжини (тривалості виконання завдання) та їхньої кількості і місця (зони) їхнього приймання.

Змістовна постановка задачі маршрутизації команд БПЛА для пошуку об'єктів може бути такою [21]. Задано координати можливих місць старту, кількість БПЛА, координати об'єктів, які необхідно обстежити, координати зон приймання, польотний ресурс кожного БПЛА, ємність зон приймання.

Критерієм є мінімум сумарної довжини маршрутів БПЛА, що і визначатиме час виконання завдання, а також мінімальну кількість задіяних БПЛА.

Позначимо $m$ загальну кількість БПЛА, $b$ — кількість місць старту, $e$ — кількість зон приймання БПЛА ($e \geq 1$), $n$ — кількість об'єктів, які варто обстежити, $R^k$ — польотний ресурс $k$-го БПЛА, $k = 1, \ldots, m$.

Розглянемо зважений граф $G = (Y, U)$, де вершини $Y$ відповідають об'єктам, місцям старту та зонам приймання, тобто $\| Y \| = n + b + e$, задаючи таку їхню нумерацію: від 1 до $n$ — об'єкти, від $n+1$ до $n+b$ — місця старту, а від $n+b+1$ до $n+b+e$ — зони приймання (їхні центроїди).

Позначимо $V = \{1, \ldots, n\}$ множину вершин, які відповідають об'єктам, $B = \{n+1, \ldots, n+b\}$ — множину вершин, які відповідають місцям старту, $E = \{n+b+1, \ldots, n+b+e\}$ — вершини, що відповідають зонам приймання. Зрозуміло, що для вершин із множини $B$ матимемо лише вихідні дуги, а для вершин із $E$ — лише вхідні.

Введемо матрицю $X = (x_{ij})$, $x_{ij} \in \{0, 1\}$, розміру $m \times (n+b+e)$, яка складається із трьох блоків. Перший блок сформовано із стовпчиків від 1 до $n$, другий — від $n+1$ до $n+b$, третій — від $n+b+1$ до $n+b+e$, а рядки відповідають номерам БПЛА. Ця матриця буде описувати фактично етап розбиття, на якому кожному БПЛА приписують ті об'єкти, які він має обстежити, а також одне місце старту та місце завершення маршруту.

Позначимо $\delta_j$ максимальну кількість БПЛА, які можуть бути прийняті в зоні $j$, $j = 1, \ldots, e$.

Задачу розбиття (як окремий етап чи складову загального процесу розв’язування) можна записати так.

Обмежувальні умови:

$$\sum_{j=n+1}^{n+b} x_{ij} \le 1, \ \ i=1,\ldots,m; \tag{3}$$

$$\sum_{j=n+b+1}^{n+b+e} x_{ij} \le 1, \ \ i=1,\ldots,m; \tag{4}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \ \ i=1,\ldots,n; \tag{5}$$

$$\sum_{i=1}^{m} x_{ij} \le 1, \ \ j=n+1,\ldots,n+b; \tag{6}$$

$$\sum_{j=n+1}^{n+b} \sum_{i=1}^{m} x_{ij} = m; \tag{7}$$

$$\sum_{j=n+b+1}^{n+b+e} \sum_{i=1}^{m} x_{ij} = m; \tag{8}$$

$$\sum_{i=1}^{m} x_{ij} \le \delta_{j-n-b}, \ \ j=n+b+1,\ldots,n+b+e; \tag{9}$$

$$L^i(\overline{x_i}) \le R^i, \ \ i=1,\ldots,m. \tag{10}$$

Задача полягає в пошуку такого розв’язку $X$ за умов (3)–(10), для якого

$$F(X) \equiv \sum_{i=1}^{m} L^i(\overline{x_i}) \to \min, \tag{11}$$

де $\overline{x_i}$ — це $i$-й вектор-рядок матриці $X$, а $L^i(\overline{x_i})$ — довжина маршруту $i$-го задіяного БПЛА як розв’язок задачі комівояжера, яка визначається вершинами, що відповідають ненульовим компонентам вектора $\bar{x}_i$, починаючи з певного стартового місця $s \in B : x_{is} = 1$ і закінчуючи фінішною зоною $f \in E : x_{if} = 1$.

Для незадіяних БПЛА відповідний вектор-рядок матиме всі нульові координати.

Зрозуміло, що час виконання завдання можна отримати як максимум серед тривалостей польотів БПЛА, а ці тривалості визначаються довжиною маршруту та середньою швидкістю відповідного БПЛА.

Матрицю $X$ як елемент простору розв’язків задачі (3)–(11) у компактному вигляді можна записати: $X = (\overline{x_1}, \ldots, \overline{x_m})^{\mathrm{T}}$, де $\overline{x_i} = (x_{i1}, x_{i2}, \ldots, x_{i,n+b+e})$, Т — знак транспонування.

За виконання умови (3) кожен $i$-й БПЛА стартує лише з одного із можливих місць старту (чи в момент часу, який і визначає таке місце); якщо деякий БПЛА незадіяний, то відповідний рядок стає нульовим. Аналогічно формула (4) описує ситуацію, коли до конкретного центроїда від об’єктів або веде одне і лише одне ребро графу $G$, або жодного. За виконання умови (5) кожен об’єкт $j \in V$ обстежується лише одним БПЛА; за виконання умови (6) з кожного місця або стартує один БПЛА, або це місце не використовують; за виконання умов (6), (7) кількості використаних місць і задіяних БПЛА однакові. В умові (8) описано ситуацію, коли всі БПЛА повертаються у фінішні зони. Можливість приймати в кожній зоні $j \in E$ (за іншим трактуванням — у відповідний момент часу) не більше, ніж $\delta_j$ БПЛА, відображено у формулі (9). Обмеження на польотний ресурс кожного літального апарату визначено умовою (10).

Зауважимо, що не виключаються випадки, коли в одній зоні (чи в момент часу) може прийматися лише один БПЛА, тоді $\delta_j = 1$, чи всі БПЛА, тоді $e = 1$, а $\delta_1 = m$.

Таким чином, цю задачу можна вважати розвитком проблеми маршрутизації транспортних засобів із декількома депо (Multiple Depot Vehicle Routing Problem, MDVRP) з додатковими обмеженнями [19, 23].

Задача MDVRP складніша, ніж класична задача Vehicle Routing Problem. До того ж MDVRP є NP-складною задачею, це означає, що не існує ефективного алгоритму для отримання оптимального розв'язку. Такі точні методи, як метод гілок і меж, гілок і відтинання є неефективними для розв'язування MDVRP через надмірну трудомісткість, тому використовують наближені алгоритми.

## 4. АЛГОРИТМИ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ

**Повний перебір.** Найпростіший точний метод, який можна застосувати, — це прямий перебір усіх можливих варіантів. Алгоритм повного перебору всіх можливих розв'язків задачі складається з чотирьох етапів. Перший етап — формування всіх можливих перестановок номерів стартових депо. Для визначення кількості перестановок із $b$ елементів маємо формулу $P_b = b!$. Другий етап — розбиття множини з $n$ об'єктів, які необхідно обстежити. Загальна кількість варіантів розбиття довільної $n$-елементної множини дорівнює числу Белла, яке можна обрахувати як суму чисел Стірлінга другого роду $B_n = \sum_{m=0}^{n} S(n, m)$ [24]. Третій етап — формування всіх можливих розміщень для $e$ зон приймання. Загальна кількість розміщень визначається за формулою $A_e^b = \frac{b!}{(b-e)!}$. Четвертий етап — композиція всіх маршрутів, отриманих на перших трьох кроках, загальна кількість варіантів $k = b! \sum_{m=0}^{n} S(n, m) \frac{b!}{(b-e)!}$. Така надзвичайно велика кількість варіантів дозволяє зробити висновок, що методом повного перебору неможливо знайти оптимальний розв'язок реальних задач за прийнятний час. Його застосування виправдане лише для задач невеликої розмірності, щоб отримати точний розв'язок і дослідити точність прикладних алгоритмів.

**Локальний пошук.** Алгоритми локального пошуку використовують поняття околу в просторі розв'язків $X$ задачі. Варіант розв'язку складається з множини маршрутів — послідовності номерів об'єктів, які обстежуються кожним БПЛА. Усі маршрути починаються з стартового депо та закінчуються зоною приймання.

Для генерації околів $O(x)$ довільної точки $x \in X$ застосовують оператор переміщення, який є одним з найбільш ефективних способів модифікації маршруту [20].

Для розв'язування сформульованої задачі далі пропонуються два наближені алгоритми, розроблені на основі табу пошуку та мурашиних алгоритмів.

**Табуйований пошук.** Цей алгоритм, який довів свою ефективність для розв'язування низки задач класу MDVRP [17, 20], пропонується гібридизувати з конструктивним алгоритмом, що реалізує принцип жадібного вибору.

Наведемо псевдокод алгоритму.

**procedure TABU_SEARCH_WITH_GREEDY** $(x)$

ініціалізація_алгоритму;
  $x :=$ допустимий варіант розв'язку, сформований жадібним алгоритмом;
  $T := \varnothing$;
  $x_{rec} := x$;
  **while** не_виконується_критерій_завершення **do**
    пошук_прийнятного_варіанта $y \in O(x) \setminus T$;
    $x := y$;
    $T := T \cup x$;
    **if** $f(x) < f(x_{rec})$ **then** $x_{rec} := x$;
    **if** $|T| >$ максимальний_розмір_табу_списку **then**
      видалення_з_множини_$T$_найстарішого_елемента;
  **endwhile**
**end**

**Макс-мін алгоритм мурашиних систем.** Інтенсивніше використання кращих розв'язків у процесі пошуку призводить до підвищення продуктивності алгоритмів оптимізації мурашиною колонією [25], але у такому випадку збільшується ризик передчасної збіжності. Тому доречно комбінувати використання кращих розв'язків та процедури запобігання передчасній збіжності. Для задоволення цих двох потреб розроблено спеціальний макс-мін алгоритм мурашиних систем як покращення алгоритму мурашиних систем [26]. Як і в класичному алгоритмі, для уникнення стагнації розв'язків введено інтервал значень для феромона, тобто введено поняття нижньої та верхньої межі.

Обчислювальну схему алгоритма можна записати таким псевдокодом:

**procedure MMAS** $(x)$

ініціалізація_алоритму;
**while** кількість_ітерацій_без_покращення_менша_заданої **do**
формування_популяції_мурах;
  **while** не_всі_вершини_відвідані **do**
    **for** чергова_мураха_з_популяції **do**
      ініціалізація_поточного_кроку_мурахи;
      $M =$ оновлення_пам'яті_мурахи;
      $A =$ локальна_матриця_мурашиних_маршрутів;
      сформувати_множину_припустимих_вершин_з_урахуванням_
        наявного_ресурсу;
      $p =$ обчислити_ймовірність_переходів $(A, M, \Pi)$;
      наступний_стан = правило_прийняття_рішення $(p, \Pi)$;
      вибравши вершину, перейти_в_наступний_стан;
      $M =$ оновити_внутрішній_стан;
      вилучити вибрану вершину із списку допустимих;
    **endfor**
  **endwhile**
завершити_діяльність;
відкласти_феромон_на_найкращому_маршруті_на_групі_останніх_ітерацій;
випаровування_феромона;
оновлення_рекорду $(x)$;
**endwhile**
**end**

У наведеному псевдокоді П — предикат, що описує обмеження задачі (детальніший опис обчислювальної схеми алгоритмів оптимізації мурашиними колоніями див., наприклад, в [25–27]).

На початку роботи алгоритму матриця феромонів ініціалізується значеннями нижньої межі феромона. Після кожної ітерації тільки одна мураха, яка знайшла найкращий маршрут на поточній ітерації, модифікує феромонний слід. Для розв'язування сформульованої задачі використовують модифікований макс-мін алгоритм мурашиних систем. Розв'язок будується покроково: поточна мураха вибирає наступну вершину графа задачі і переходить до неї, потому ця вершина стає недосяжною для відвідування мурахами, які роблять свій крок пізніше.

## 5. ОБЧИСЛЮВАЛЬНИЙ ЕКСПЕРИМЕНТ

Оскільки у відкритому доступі немає даних для наведеної задачі, було згенеровано набір задач з використанням системи Google Maps API. Ці задачі мають розмірність від 10 до 500 об'єктів, які необхідно обстежити, у середньому 4–7 місць старту та 2–5 зон приймання.

Отримані в обчислювальному експерименті результати наведено в табл. 1, де MMAS — модифікований макс-мін алгоритм мурашиних систем; Tabu Search — гібридний табу пошук з використанням оператора переміщення; Bruteforce — алгоритм повного перебору; $f_0$ — початковий розв'язок; $f_*$ — розв'язок задачі, отриманий алгоритмом; $q = \frac{f_0 - f_*}{f_0} \cdot 100\,\%$ — відносне покращення знайденого розв'язку щодо початкового наближення; $t$ — час виконання алгоритму (у мілісекундах); як і раніше, $n$ — кількість об'єктів, які необхідно обстежити; жирним шрифтом виділено найкращі розв'язки серед усіх варіантів, знайдених алгоритмами.

**Таблиця 1**

| Задача | $n$ | MMAS | | | | Tabu Search | | | | Bruteforce | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $f_0$ | $f_*$ | $q$ | $t$ | $f_0$ | $f_*$ | $q$ | $t$ | $f_*$ | $t$ |
| 1 | 10 | 2757.1 | 2193.9 | 20.4 | 61 | 2465.5 | 2157.3 | 12.5 | 6 | **1690.1** | 62 |
| 2 | 16 | 3987.4 | 3299.2 | 17.2 | 69 | 4510.5 | 3669.4 | 18.6 | 7 | **3151.2** | 23621 |
| 3 | 30 | 7696.4 | **4369.5** | 43.2 | 158 | 6941.6 | 4478.9 | 35.4 | 30 | — | — |
| 4 | 50 | 7947.9 | **5738.7** | 27.7 | 294 | 7784 | 6443 | 17.2 | 149 | — | — |
| 5 | 100 | 13619.2 | 9693.4 | 28.8 | 978 | 16430 | **9665.8** | 41.1 | 212 | — | — |
| 6 | 500 | 28725.4 | **19698.2** | 31.4 | 44712 | 29906 | 24869.6 | 16.8 | 35840 | — | — |

Усі алгоритми реалізовано на однаковій програмній базі на мові Kotlin. Пошук розв'язків здійснювався на персональному комп'ютері з 16 ГБ оперативної пам'яті та восьмиядерним процесором з тактовою частотою 3.6 ГГц. Це дозволило порівнювати і час роботи алгоритмів.

**Параметри.** Для алгоритму локального пошуку застосовано такі параметри: розмір списку заборон — 5; кількість ітерацій без покращення розв'язку — 500. Для макс-мін алгоритму мурашиних систем використано такі параметри: однакова кількість мурах і БПЛА — $m$; кількість ітерацій без покращення розв'язку — 1000; ступінь значущості феромонного сліду $\alpha = 1$; ступінь значущості евристичної інформації $\beta = 1$; коефіцієнт випаровування $\rho = 0.1$ [25–27].

Параметри підібрано шляхом вибору найкращих із серії попередніх випробувань.

Як випливає з результатів експерименту, розроблений модифікований макс-мін алгоритм мурашиних систем продемонстрував кращі показники точності в порівнянні з алгоритмами локального пошуку та прямого перебору.

**ВИСНОВКИ**

Запропоновано змістовну постановку сформульованих задач маршрутизації для виконання обстеження та/або обслуговування заданої множини об'єктів командою БПЛА з умовою завершення маршруту в певних зонах приймання (депо) та обмеженнями на ресурси БПЛА як спеціальної задачі комбінаторної оптимізації.

Показано, що запропонована математична модель руху команди БПЛА є розвитком задачі маршрутизації транспортних засобів із декількома депо. Важливою умовою є те, що депо можуть переміщуватися заданою траєкторією в процесі виконання завдання, це дозволяє охопити клас задач з рухомими депо, наприклад, проблеми планування маршрутів ґремлінів, які стартують і фінішують з використанням літака-носія. Подібні оптимізаційні задачі можна розв'язувати методом повного перебору лише для малої їхньої розмірності. Для розв'язування сформульованої задачі маршрутизації команди БПЛА розроблено макс-мін алгоритм мурашиних систем, особливістю якого є покрокова взаємодія мурах для формування розв'язків, та гібридний алгоритм табу пошуку.

Напрямом подальших досліджень може бути покращення мурашиного алгоритму за рахунок використання процедур диверсифікації пошуку [28], а також паралельної реалізації острівної моделі [29].

СПИСОК ЛІТЕРАТУРИ


1. Ponda S.S., Johnson L.B., Geramifard A., How J.P. Cooperative mission planning for multi-UAV teams. In: Valavanis K., Vachtsevanos G. (Eds). *Handbook of Unmanned Aerial Vehicles*. Dordrecht: Springer, 2015. P. 1447–1490.
2. Otto A., Agatz N., Campbell J., Golden B., Pesch E. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*. 2018. Vol. 72, Iss. 4. P. 411–458.
3. Горбулін В.П. Забезпечення оборони та безпеки України: актуальні проблеми і шляхи їх вирішення. *Вісн. НАН України.* 2019. № 9. С. 3–18.
4. Coutinho W.P., Battarra M., Fliege J. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering*. 2018. Vol. 120. P. 116–128.
5. Zhao Y., Zheng Z., Liu Y. Survey on computational-intelligence-based UAV path planning. *Knowledge-Based Systems.* 2018. Vol. 158. P. 54–64.
6. Гуляницький Л.Ф., Рибальченко О.В. Формалізація та розв'язування одного типу задач маршрутизації БПЛА. *Теорія оптимальних рішень*. 2018. № 17. С. 107–114.
7. Perez-Carabaza S., Besada-Portas E., Lopez-Orozco J.A., Jesus M. Ant colony optimization for multi-UAV minimum time search in uncertain domains. *Applied Soft Computing*. 2018. Vol. 62. P. 789–806.
8. Cekmez U., Ozsiginan M., Sahingoz O.K. Multi-UAV path planning with multi colony ant optimization. In: *International Conference on Intelligent Systems Design and Applications* (2017, December). Cham: Springer, 2017. P. 407–417.
9. Chiang W.C., Li Y., Shang J., Urban T.L. Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization. *Applied Energy*. 2019. Vol. 242. P. 1164–1175.
10. Binol H., Bulut E., Akkaya K., Guvenc I. Time optimal multi-UAV path planning for gathering ITS data from roadside units. In: *88th Vehicular Technology Conference (VTC-Fall)* (2018, August). IEEE, 2018. P. 1–5.
11. Xu C., Duan H., Liu F. Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning. *Aerospace Science and Technology*. 2010. Vol. 14, Iss. 8. P. 535–541.

12. Tian G., Zhang L., Bai X., Wang B. Real-time dynamic track planning of multi-UAV formation based on improved artificial bee colony algorithm. In: *37th Chinese Control Conference (CCC)* (2018, July). IEEE, 2018. P. 10055–10060.
13. Shakhatreh H., Khreishah A., Alsarhan A., Khalil I., Sawalmeh A., Othman, N.S. Efficient 3D placement of a UAV using particle swarm optimization. In: *8th International Conference on Information and Communication Systems (ICICS)* (2017, April). IEEE, 2017. P. 258–263.
14. Austin R. Unmanned aircraft systems. UAVs design, development and deployment. West Sussex: John Wiley and Sons, 2010. 365 p.
15. Tsourdos A., White B., Shanmugavel M. Cooperative path planning of unmanned aerial vehicles. West Sussex: John Wiley and Sons, 2011. 212 p.
16. Shima T., Rasmussen S. UAV cooperative decision and control. Challenges and practical approaches. Philadelphia: SIAM, 2009. 186 p.
17. Toth P., Vigo D. (Eds.). Vehicle routing: problems, methods, and applications. Philadelpia: SIAM, 2014. 462 p.
18. Braekers K., Ramaekers K., Nieuwenhuyse I.V. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*. 2016. Vol. 99. P. 300–313.
19. Karakatic S., Podgorelec V. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*. 2015. Vol. 27. P. 519–532.
20. Soto M., Sevaux M., Rossi A., Reinholz A. Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem. *Computers & Industrial Engineering*. 2017. Vol. 107. P. 211–222.
21. Горбулін В.П., Гуляницький Л.Ф., Сергієнко І.В. Постановки та математичні моделі проблем оптимізації маршрутів літальних апаратів із динамічними депо. *Управляющие системы и машины*. 2019. № 1. С. 3–10.
22. Husseini T. Gremlins are coming: DARPA enters Phase III of its UAV programme. Army Technology, 3 July 2018. URL: https://www.army-technology.com/features/gremlins-darpa-uav-programme/.
23. Гуляницкий Л.Ф. Проблема оптимизации маршрутов транспортных средств с временными окнами. *Компьютерная математика.* 2007. № 1. С. 122–132.
24. Андерсон Дж. Дискретная математика и комбинаторика. Москва; СПб; Киев: ИД «Вильямс», 2003. 957 с.
25. Dorigo M., Stützle T. Ant colony optimization: Overview and recent advances. In: *Handbook of Metaheuristics*. Cham: Springer, 2019. P. 311–352.
26. Stützle T., Hoos H.H MAX-MIN ant system. *Future Generation Computer Systems*. 2000. Vol. 16, Iss. 8. P. 889–914.
27. Гуляницький Л.Ф., Мулеса О.Ю. Прикладні методи комбінаторної оптимізації. Київ: ВПЦ «Київський університет», 2016. 142 с.
28. Гуляницький Л.Ф. Новий алгоритм оптимізації мурашиними колоніями. *Сучасна інформатика: проблеми, досягнення та перспективи розвитку: Пр. Міжн. конф., присвяченої 60-річчю заснування ІК ім. В.М. Глушкова НАН України (*Київ, 13–15 грудня 2017 р.). Київ: Ін-т кібернетики ім. В.М. Глушкова НАН України, 2017. С. 41–43.
29. Mora A.M., García-Sánchez P., Merelo J.J., Castillo P.A. Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal. *Soft Computing*. 2013. Vol. 17, Iss. 7. P. 1175.–1207.

*Надійшла до редакції 16.09.2019*

## В.П. Горбулин, Л.Ф. Гуляницкий, И.В. Сергиенко

**ОПТИМИЗАЦИЯ МАРШРУТОВ КОМАНДЫ БПЛА ПРИ НАЛИЧИИ АЛЬТЕРНАТИВНЫХ И ДИНАМИЧЕСКИХ ДЕПО**

**Аннотация.** Предложены содержательная постановка и математические модели проблем оптимизации маршрутов команды беспилотных летательных аппаратов (БПЛА) при обследовании или обслуживании заданного множества объектов при наличии альтернативных и динамических депо (мест старта и/или приземления) и ресурсных ограничений. К таким задачам, в частности, относятся проблемы планирования полетов БПЛА, использующих движущие платформы в качестве депо. Критериями оптимизации являются как суммарная длина маршрутов, так и количество задействованных БПЛА. Разработаны и реализованы алгоритмы решения сформулированных задач комбинаторной оптимизации, основанные на оптимизации муравьиными колониями, табу поиске и полном переборе. Представлены результаты вычислительного эксперимента.

**Ключевые слова:** оптимизация маршрутов, БПЛА, муравьиные алгоритмы, динамические депо, табу поиск, гремлины.

## V.P. Horbulin, L.F. Hulianytskyi, I.V. Sergienko

**OPTIMIZATION OF UAV TEAM ROUTES AT THE AVAILABILITY OF ALTERNATIVE AND DYNAMIC DEPOTS**

**Abstract.** The paper considers the problems of optimization of unmanned aerial vehicle (UAV) routes which act as a team when inspecting or supporting a given set of objects in the presence of alternative and dynamic depots (starting and/or landing sites) and resource constraints. Problem definition and mathematical models are proposed. Such problems, in particular, include UAV flight planning problems, which use moving platforms as a depot. The optimization criteria are both the total length of the routes and the number of UAVs involved. Algorithms for solving formulated combinatorial optimization problems based on ant colony optimization, tabu search, and exhaustive search have been developed and implemented. The results of the computational experiment are presented.

**Keywords:** route optimization, UAV, ant algorithms, dynamic depo, tabu search, gremlins.

**Горбулін Володимир Павлович,**
академік НАН України, професор, перший віце-президент НАН України, Київ,
e-mail: horbulin@nas.gov.ua.

**Гуляницький Леонід Федорович,**
доктор техн. наук, завідувач відділу Інституту кібернетики ім. В.М. Глушкова НАН України, Київ,
e-mail: leonhul.icyb@gmail.com.

**Сергієнко Іван Васильович,**
академік НАН України, професор, директор Інституту кібернетики ім. В.М. Глушкова НАН України,
Київ, e-mail: incyb@incyb.kiev.ua.