

УДК 519.8

DOI [https://doi.org/10.24144/2616-7700.2019.1\(34\).69-78](https://doi.org/10.24144/2616-7700.2019.1(34).69-78)**Л. Ф. Гуляницький<sup>1</sup>, В. В. Сторчевий<sup>2</sup>**

<sup>1</sup> Інститут кібернетики ім. В.М. Глушкова НАН України, Київ,  
завідувач відділу,

доктор технічних наук, старший науковий співробітник

[leonhul icyb@gmail.com](mailto:leonhul icyb@gmail.com)

ORCID: <https://orcid.org/0000-0002-1379-4132>

<sup>2</sup> НТУУ «Київський політехнічний інститут імені Ігоря Сікорського», Київ,  
магістрант

[v.storchevyi@gmail.com](mailto:v.storchevyi@gmail.com)

ORCID: <https://orcid.org/0000-0002-8570-575X>

## ОДНА СПЕЦІАЛЬНА ЗАДАЧА МАРШРУТИЗАЦІЇ БПЛА

Метою статті є огляд задач маршрутизації транспортних засобів, їх класифікація, а також розгляд проблеми маршрутизації групи безпілотних літальних апаратів (БПЛА) при обстеженні заданих об'єктів. Запропонована математична модель проблем, які полягають в тому, що перед заданою групою літальних апаратів, які можуть стартувати з різних точок пуску і повинні закінчувати маршрут у спеціальних зонах приймання, що мають певну ємкість, стоїть завдання облетіти ряд заданих об'єктів (точок на місцевості) з мінімізацією сумарної довжини маршрутів або тривалості польотів за умови, що кожен об'єкт відвідується одним і тільки одним БПЛА і всі об'єкти повинні бути відвіданими. Показано, що модель руху транспортних засобів з декількома депо можна адаптувати для планування польоту групи БПЛА в ряді задач обстеження. Пропонуються алгоритми розв'язування: повний перебір розв'язків, гібридний локальний пошук – табуйований пошук із використанням жадібного алгоритму для побудови початкового маршруту, макс-мін алгоритм мурашиних систем MMAS. Оскільки у відкритому доступі немає бібліотек даних для сформульованих постановок задач, то був згенерований набір задач за допомогою системи Google Maps API, що дало змогу отримати задачі, які наближені до реальності (використані координати сіл, селищ, міст). Задачі мають різну кількість обстежуваних об'єктів, літальних засобів, початкових та кінцевих депо. Для кожного з алгоритмів був здійснений підбір параметрів і потім проведений обчислюваний експеримент, який дозволив порівняти три розроблені алгоритми розв'язування задачі маршрутизації групи БПЛА. У результаті визначено, що метод прямого перебору можна застосувати лише до задач дуже малої розмірності, при цьому потрібно багато часу на виконання алгоритму. Локальний пошук знаходив покращені розв'язки на задачах середньої розмірності, а макс-мін алгоритм мурашиних систем MMAS показав себе найкраще на задачах великої розмірності, тому він і визначений як перспективний для дослідження. Визначено подальший напрямок покращення макс-мінного алгоритму мурашиних систем шляхом використання принципів диверсифікації пошуку в них.

**Ключові слова:** безпілотні літальні апарати, оптимізація маршруту, мультидепо, математична модель, комбінаторна оптимізація, мурашина система.

**1. Вступ.** Доставка товарів – одна з ключових функцій у логістичних системах. Успішність доставки багато в чому залежить від правильно складеного маршруту руху транспортних засобів з урахуванням особливостей кожного клієнта. Ця проблема має назву задача маршрутизації транспортних засобів (VRP – vehicle routing problem). Вперше вона сформульована у 1959 р. Данцігом і Рамсером [1]. Задачі VRP лежать на перетині двох добре відомих задач. Це задача комівояжера (якщо вантажопідйомність кожного транспортного засобу

вважається достатньою, то VRP зводиться до задачі багатьох комівояжерів) та задача про ранець (розв'язок даної задачі, по суті, еквівалентний розв'язанню задачі VRP за умови, що всі відстані приймаються рівними нулю).

Хоча задача маршрутизації транспортних засобів була сформульована досить давно, вона не втрачає своєї актуальності та інтересу дослідників, оскільки на практиці виникають все більш складні задачі як за кількістю параметрів, що оптимізуються, так і за кількістю обмежень. Зазвичай задача маршрутизації фігурує при використанні автомобільного транспорту, оскільки існує розвинена мережа автомобільних доріг, за рахунок цього існують різні варіанти досягнення пункту призначення. Для повітряного або морського транспорту ситуація була зовсім інша, оскільки існує сформована система торгових шляхів, транспортних коридорів, якими зазвичай користуються. Ситуація почала суттєво змінюватися з набуванням популярності безпілотних літальних апаратів (БПЛА). Із розширенням функціоналу БПЛА і збільшенням доступності дедалі більше розширюється сфера їх використання: ЖКГ – для енергоаудиту будинків, лісництва – для боротьби з незаконною вирубкою, охоронні підприємства – для виявлення порушників, у сільському господарстві – для зрошення, захисту від замерзання, моніторингу структури площ для посівів і контролі за використанням угідь, урожайністю тощо [5]. Одним із напрямів застосування БПЛА є військова та інші спеціальні сфери, де БПЛА використовуються для ведення повітряної розвідки – як тактичної, так і стратегічної [6,7].

**2. Постановка задачі обстеження об'єктів.** Задачі маршрутизації БПЛА у спеціальних сферах часто відносяться до класу multi depot vehicle routing problem (MDVRP). MDVRP можуть бути подані наступним чином [8]: маємо граф  $G(V, E)$ , де  $V = \{0, \dots, n\}$  – множина вершин (депо, споживачі), а  $E$  – множина ребер. Кожне ребро відображає собою маршрут від вузла  $i$  до вузла  $j$ . Множина  $V$  розбита на дві підмножини:  $V_c = \{v_1, \dots, v_N\}$ , яка відображає споживачів, та підмножину  $V_d = \{v_{N+1}, v_{N+2}, \dots, v_M\}$  яка відповідає депо. Кожен споживач  $v_i \in V_c$  має невід'ємний попит, а кожне ребро з  $E$  характеризується вартістю, відстанню або часом в дорозі. Маємо декілька транспортних засобів, кожен з яких має характеристику вантажопідйомності чи місткості. Задача полягає в побудові маршруту таким чином, щоб: кожен маршрут починався і закінчувався в одному і тому ж депо; кожен клієнт обслуговувався лише один раз; загальний попит на маршруті не перевищував ємкість транспортного засобу; загальна довжина маршруту була мінімальною. На рисунку 1 зображено відмінності MDVRP від VRP [4].

**3. Змістовна постановка проблеми.** Розглядається проблема пошуку оптимальних маршрутів для групи спеціальних літальних апаратів, в першу чергу, БПЛА. Вона полягає у тому, що перед заданою групою БПЛА, які можуть стартувати з різних точок пуску та мати можливість закінчувати маршрут в різних місцях (депо), стоїть завдання облетіти низку заданих об'єктів (точок на місцевості) з мінімізацією сумарної довжини маршрутів чи тривалості польотів за умов, що кожен об'єкт відвідується одним і лише одним БПЛА і всі об'єкти повинні бути відвіданими. При цьому часто слід враховувати ще й ряд додаткових обмежуючих умов (дальність польоту без підзаряджання чи дозаправлення, вантажопідйомність, погодні умови, тощо). Приземлення БПЛА може здійснюватися в декількох заданих територіально зонах. Припускається

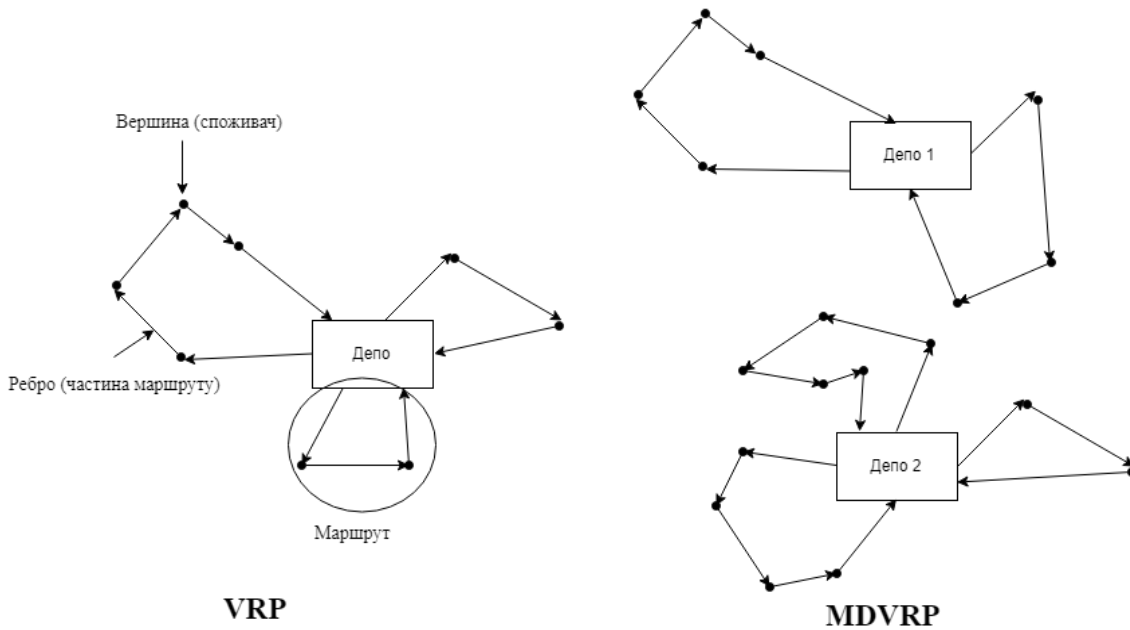


Рис. 1. Порівняння VRP та MDVRP

також, що характеристики кожної зони вибираються так, що у конкретній зоні можливе приймання заданої кількості БПЛА: у такій зоні можуть завершувати свій маршрут всі чи частина БПЛА. Кожну зону будемо ідентифікувати її центроїдом. Отже, маршрут кожного із БПЛА повинен закінчуватися в одній із таких зон, але конкретна зона для БПЛА не вказується. Цю проблему назвемо задачею маршрутизації з альтернативними депо. Кількість місць старту БПЛА може перевищувати кількість самих БПЛА – це створює додаткові можливості для вибору місць (чи моментів) старту. Також відмітимо, що в процесі розв’язування задачі маршрутизації може оптимізуватися також і кількість задіяних БПЛА.

Задача формулюється так. За умови наявності декількох можливих зон приймання БПЛА і місць старту БПЛА, а також завершення маршрутів БПЛА лише в одній із виділених зон слід визначити як маршрути БПЛА з оптимізацією сумарної довжини (тривалості польотів) та їх задіяної кількості, так і місця (зони) їх приймання. Зауважимо, що в одній зоні може завершуватися декілька маршрутів БПЛА. Задано координати місць старту, кількість БПЛА, координати точок, які необхідно відвідати, координати зон приймання, польотний ресурс кожного БПЛА, ємність зон приймання.

Критерієм виступатиме мінімум сумарної довжини (часу) польоту для виконання завдання; мінімум БПЛА.

**4. Математична модель.** Розглянемо комбінаторну математичну модель проблеми оптимізації маршрутів БПЛА за наявності декількох депо [9,10].

Позначимо  $u$  кількість депо,  $n$  – кількість об’єктів,  $m$  – кількість БПЛА. Нехай  $B = \{b_1, \dots, b_u\}$  – це множина депо, а  $Y = \{y_1, \dots, y_n\}$  – множина об’єктів.

Розглянемо розбиття множини  $Y$  на спеціальні блоки розбиття  $Y^i$ , які є

або порожніми, або впорядкованими послідовностями елементів  $Y$  (тобто, розміщенням) і задовольняють умовам:

$$\bigcup_{i=1}^m Y^i = Y,$$

$$Y^i \cap Y^j = \emptyset, \quad i \neq j, \quad i, j = 1, \dots, m.$$

Таким чином, такі спеціальні блоки розбиття можуть відрізнитися між собою як елементами, так і їх порядком. Вони і будуть визначати польотні завдання для кожного БПЛА: як множину об'єктів, які слід відвідати, так і порядок їх обходу.

Якщо початкове розміщення БПЛА в депо не задано, то варіанти призначення можуть породжуватися експертним шляхом на основі аналізу ситуації чи випадковим чином.

Після першого етапу, коли відоме стартове розміщення БПЛА, фрагмент маршруту для  $i$ -го БПЛА, може бути описаний так. Спочатку задається вектор  $\tilde{X}^i = (b_i^0, \tilde{Y}^i)$ , де  $b_i^0 \in B$ , а  $\tilde{Y}^i$  – це один із зазначених блоків розбиття з упорядкуванням. Зрозуміло, що якщо  $\tilde{Y}^i = \emptyset$ , то відповідний БПЛА в такому варіанті планування маршрутів не задіяний.

Нехай  $\tilde{Y}^i = (y_1^i, \dots, y_s^i)$ , де  $y_s^i \in Y$ ,  $i = 1, \dots, s$ , а  $1 \leq s \leq n$ . Для задоволення умов на обмеження льотного ресурсу (що вимагає підзарядки чи дозаправлення в депо), виконаємо таку процедуру: починаючи з  $y_1^i$ , визначаємо такий максимально можливий підфрагмент із розташованих підряд  $t$  компонент  $\tilde{X}^i$ ,  $t \leq s$ , що маршрут  $(b_i^0, y_1^i, \dots, y_t^i, b_1^i)$ , де  $b_1^i \in B$  – найближче до об'єкта  $y_t^i$  депо, задовольняє умовам обмежень на ресурс  $i$ -го БПЛА, а при  $t < s$  підфрагмент  $(b_i^0, y_1^i, \dots, y_t^i, y_{t+1}^i, b_1^i)$  уже не задовольняє. Продовживши цю процедуру з наступними підфрагментами, вставляємо в  $\tilde{X}^i$  відповідні депо, останнє із яких визначатиме кінець маршруту. У разі, якщо відвідання вже першого компонента  $y_1^i$  не може бути реалізоване, тобто додавання будь-якого депо для повернення БПЛА неможливе із-за перевищення ресурсу, то таке розбиття множини  $Y$  вважається неприпустимим; аналогічно, воно є неприпустимим, якщо при вставленні депо залишаються компоненти  $\tilde{X}^i$ , включення яких в маршрут призводить до перевищення польотного ресурсу.

Після успішного завершення такого опрацювання отримуємо послідовність із  $l(i)$  елементів, яка починається з  $b_i^0$  і закінчується  $b_i^{k(i)}$ ,  $b_i^0, b_i^{k(i)} \in B$ , де  $l(i) = k(i) + s + 1$ , а  $k(i)$  – кількість депо, вставлених шляхом виконання описаної процедури. Цю послідовність, для зручності, позначимо  $X^i$ , так що надалі вважатимемо  $X^i = (x_1^i, \dots, x_{l(i)}^i)$ .

Довжина маршруту  $i$ -го БПЛА в цих позначеннях визначатиметься так:

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

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

Цільову функцію досліджуваної задачі оптимізації маршрутів БПЛА тоді можемо подати так:

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

де  $X = (X^1, \dots, X^m)$  – елемент простору розв'язків задачі, який визначається способом розбиття множини об'єктів  $Y$  та розташуванням депо  $B$ .

Важливо відзначити, що запропонована математична модель маршрутизації в якості частинних випадків включає також такі постановки задач, що мають важливе прикладне значення:

1) Всі БПЛА стартують лише з одного депо і повертаються в нього. В цьому разі задача може зводитися до відомої задачі маршрутизації транспортних засобів [11].

2) Для кожного БПЛА вказане одне із конкретних депо із декількох наявних в якості місця старту і завершення маршруту [12].

3) Для кожного БПЛА однозначно зафіксоване одне із декількох наявних депо для старту, а повернення можливе в будь-яке із цих депо [13].

З іншого боку, запропонований підхід до формалізації може служити кроком до розробки математичних моделей задач маршрутизації літальних апаратів у випадку, коли літальні апарати (БПЛА чи спеціальні крилаті ракети багаторазового використання) запускаються зі спеціального літака-носія [10].

**5. Алгоритми розв'язування задачі.** MDVRP є більш складною задачею, ніж класична задача VRP. Крім того, MDVRP є NP-складною задачею, це означає, що не існує ефективного алгоритму для отримання оптимального її розв'язку. Точні методи, такі як метод гілок і меж, метод гілок і відтинання, є неефективними для розв'язку MDVRP через надмірну трудомісткість, тому на практиці використовуються наближені методи [14].

Найпростіший точний метод, який можна застосувати – це прямий перебір всіх можливих варіантів. Алгоритм повного перебору всіх можливих розв'язків задачі складається з чотирьох етапів. Перший етап – формування всіх можливих перестановок початкових депо. Для визначення кількості перестановок з  $b$  елементів маємо формулу  $P_b = b!$ . Другий етап – розбиття множини з  $n$  об'єктів, які необхідно відвідати. Загальна кількість поділів довільної  $n$ -елементної множини дорівнює числу Белла, яке можна обчислити як суму чисел Стірлінга другого роду,  $B_n = \sum_{m=0}^n S(n, m)$ . Третій етап – формування всіх можливих розміщень для  $e$  зон приймання. Загальна кількість розміщень визначається за наступною формулою  $A_e^b = \frac{b!}{(b-e)!}$ . Останній етап – композиція всіх маршрутів отриманих на перших трьох кроках, загальна кількість варіантів  $k = b! \sum_{m=0}^n S(n, m) \frac{b!}{(b-e)!}$ . Через таку надзвичайно велику кількість варіантів можна зробити висновок, що методом повного перебору за прийнятний час неможливо знайти оптимальний розв'язок. Його застосування виправдане лише для задач невеликої розмірності, де є шанс отримувати точний розв'язок.

Алгоритми локального пошуку використовують поняття околу в просторі розв'язків задачі  $X$ . Варіант розв'язку складається із множини маршрутів – послідовності номерів об'єктів, які відвідуються кожним БПЛА. Кожен маршрут починається з стартового депо та закінчується зоною приймання. Для

генерації околів  $O(x)$  довільної точки  $x \in X$  застосовується оператор переміщення, який є одним з найбільш ефективних способів модифікації маршруту [15].

Для розв'язування задачі пропонується один із алгоритмів локального пошуку – табуйований пошук, який довів свою ефективність при розв'язанні задач класу MDVRP [16]. Для формування початкового розв'язку використовується конструктивний алгоритм, який реалізує принцип жадібного вибору.

---

**procedure** TABU\_SEARCH\_WITH\_GREEDY ( $x$ )

ініціалізація\_алгоритму;

$x :=$ припустимий\_варіант\_розв'язку\_сформований\_жадібним\_алгоритмом;

$T := \emptyset$  ;

$x_{rec} := x$ ;

**while** не\_виконується\_критерій\_завершення **do**

пошук\_прийняттого\_варіанта  $y \in O(x) \setminus T$ ;

$x := y$ ;

$T := T \cup x$ ;

**if**  $f(x) < f(x_{rec})$  **then**

$x_{rec} := x$ ;

**end if**

**if**  $|T| >$  максимальний\_розмір\_табу\_списку **then**

видалення\_з\_множини\_T\_найстарішого\_елемента;

**end if**

**end while**

**end procedure**

---

Більш інтенсивне використання кращих розв'язків у ході пошуку призводить до підвищення продуктивності алгоритмів оптимізації мурашиною колонією [17], але при цьому підвищується ризик передчасної збіжності. Тому доречно комбінувати використання кращих розв'язків та методи запобігання передчасній збіжності. Для задоволення цих двох потреб розроблений спеціальний максі-мінний алгоритм мурашиних систем (Max-Min Ant System, MMAS) як покращеного алгоритму мурашиних систем [18]. Для уникнення стагнації розв'язку вводиться інтервал значень для феромону, тобто вводиться поняття нижньої та верхньої межі. На початку роботи алгоритму матриця феромонів ініціалізується значеннями нижньої межі феромону. Після кожної ітерації тільки один агент залишає слід феромону, зазвичай це агент, що знайшов кращий маршрут на поточній ітерації. Для розв'язування сформульованої задачі використовується модифікований максі-мінний алгоритм мурашиних систем. Розв'язок будується покроково: поточна мураха вибирає наступну вершину графа задачі, після переходу в яку ця вершина стає недоступною для відвідання мурахами, які роблять свій крок пізніше.

---

**procedure** MMAS( $x$ )

ініціалізація\_алгоритму;

**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;
    сформувані_множину_припустимих_вершин;
     $p :=$  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_припустимих;
  end for
end while
завершити_діяльність;
відкласти_феромон_на_найкращому_маршруті_на_поточній_ітерації;
випаровування_феромону;
оновлення_рекорду ( $x$ );
end while
end procedure

```

Тут  $\Pi$  – предикат, що описує обмеження задачі.

**6. Обчислювальний експеримент.** Оскільки у відкритому доступі немає даних для цієї задачі, то був згенерований набір задач за допомогою використання Google Maps API. Згенеровані задачі мають розмірність від 10 до 500 об'єктів, які необхідно відвідати, у середньому 4–7 місць старту та 2–5 зон приймання.

Додаткові позначення:  $n$  – кількість об'єктів, що необхідно відвідати; MMAS – модифікований мах-мінний алгоритм мурашиних систем; TabuSearch – локальний пошук (табу пошук) з використанням оператора переміщення; BruteForce – алгоритм прямого перебору;  $f_{\text{поч}}$  – початковий розв'язок;  $f_{\text{роз}}$  – розв'язок задачі, отриманий алгоритмом;  $q = \frac{f_{\text{поч}} - f_{\text{роз}}}{f_{\text{поч}}} \cdot 100\%$  – відносне покращення початкового розв'язку (у відсотках);  $t$  – час виконання алгоритму (у мілісекундах).

Всі алгоритми реалізовані на одній і тій же 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 дозволяє порівнювати час роботи алгоритмів. Основні результати наведено у табл. Жирним шрифтом виділені найкращі розв'язки між всіма алгоритмами.

Для локального пошуку застосовані наступні параметри:  $l = 5$  – розмір списку заборон,  $iter = 500$  – кількість можливих ітерацій, на яких не відбувається покращення розв'язку. Для алгоритму мурашиних систем MMAS:  $noa = m$  (кількість БПЛА) – кількість мурах,  $noi = 1000$  – кількість можливих ітерацій, на яких не відбувається покращення розв'язку,  $\alpha = 1$  – ступінь значущості феромонного сліду,  $\beta = 1$  – ступінь значущості евристичної інформації,  $e = 0.1$  – стала випаровування.

Параметри підібрані на основі аналізу результатів проведеної серії випробувань. Основні результати експерименту подані у таблиці 1.

Таблиця 1. Результати обчислювального експерименту

Задача	n	MMAS				TabuSearch				Bruteforce	
		$f_{поч}$	$f_{роз}$	q	t	$f_{поч}$	$f_{роз}$	q	t	$f_{роз}$	t
1	10	2757	2193	20.4	61	2465	2157	12.5	6	<b>1690</b>	62
2	16	3987	3299	17.2	69	4510	3669	18.6	7	<b>3151</b>	23621
3	30	7696	<b>4369</b>	43.2	158	6941	4478	35.4	30	-	-
4	50	7947	<b>5738</b>	27.7	294	7784	6443	17.2	149	-	-
5	100	13619	9693	28.8	978	16430	<b>9665</b>	41.1	212	-	-
6	500	28725	<b>19698</b>	31.4	44712	29906	24869	16.8	35840	-	-

Як впливає з цих результатів експерименту, макс-мінний алгоритм мурашиних систем MMAS продемонстрував підвищені показники точності в порівнянні з алгоритмами локального пошуку та прямого перебору.

**7. Висновки та перспективи подальших досліджень.** Запропоновано змістовну постановку виділеної задачі маршрутизації для виконання обльоту об'єктів групою БПЛА з умовою завершення маршруту в зонах приймання та обмеженнях на ресурси БПЛА як спеціальної задачі комбінаторної оптимізації.

Показано, що математична модель руху групи БПЛА може бути зведена до моделі маршрутизації транспортних засобів із декількома депо. Для розв'язування сформульованої задачі маршрутизації групи БПЛА запропоновано і реалізовано декілька алгоритмів, найкраще з яких себе показав макс-мін алгоритм мурашиних систем.

Напрямок подальших досліджень бути покращення мурашиного алгоритму за рахунок використання мурашиних алгоритмів із диверсифікацією пошуку [19] та застосування острівної моделі з природнім паралелізмом [20].

### Список використаної літератури

1. Dantzig G. B., Ramser J. H. The truck dispatching problem. *Management Science*. 1959. Vol. 6, No. 1. P. 80–91.
2. Baldacci R., Toth P., Vigo D. Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*. 2010. Vol. 175, No. 1. P. 213–245.
3. Гуляницький Л. Ф. Проблема оптимізації маршрутов транспортних средств с временными окнами. *Компьютерная математика*. 2007. № 1. С. 122–132.
4. Montoya-Torres J. R., López Franco J., Nieto Isaza S. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*. 2015. Vol. 79. P. 115–129.
5. Golden B., Raghavan S., Wasil E. The vehicle routing problem: latest advances and new challenges. Springer Science+Business Media, 2008.
6. Ponda S. S., Johnson L. B., Geramifard A. Cooperative mission planning for multi-UAV teams. *Handbook of Unmanned Aerial Vehicles*. Dordrecht: Springer. 2015. P. 1447–1490.
7. Liu Y., Liu Z., Shi J. Optimization of base location and patrol routes for unmanned aerial vehicles in border intelligence, surveillance, and reconnaissance. *Journal of Advanced Transportation*. 2019. Vol. 2019. С. 1–13.
8. Renaud J., Laporte G., Voctor F. F. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*. 1996. Vol. 23, No. 3. P. 229–235.
9. Гуляницький Л. Ф. Розробка математичної моделі проблеми оптимізації маршрутів групи БПЛА за наявності декількох депо. Математичне та імітаційне моделювання систем. *МОДС 2019: тези доповідей Чотирнадцятої міжнародної науково-практичної конференції (Чернігів, 24 - 26 червня 2019 р.)* / М-во освіти і науки України, Нац. академія наук України, Академія технологічних наук України, Інженерна академія України та ін. Чернігів: ЧНТУ, 2019. С. 324–327.
10. Горбулін В. П., Гуляницький Л. Ф., Сергієнко І. В. Постановки та математичні моделі проблем оптимізації маршрутів літальних апаратів із динамічними депо. *Управляющие*



- системи и машины*. 2019. №1. С. 2–15.
11. Toth P., Vigo D. Vehicle routing: problems, methods, and applications. *Society for Industrial and Applied Mathematics*. 2014. P. 462.
  12. Austin R. Unmanned aircraft systems: UAVS design, development and deployment. Chichester, UK: John Wiley & Sons, Ltd, 2010. P. 365.
  13. Гуляницький Л. Ф., Рибальченко О. В. Формалізація та розв'язування одного типу задач маршрутизації БПЛА. *Теорія оптимальних рішень*. 2018. Вип. 17. С. 107–114.
  14. Pereira F. B., Tavares J. Bio-inspired algorithms for the vehicle routing problem. *Studies in Computational Intelligence 161*, Springer. 2009. P. 79.
  15. Kohl N. Exact methods for time constrained routing and related scheduling problems. *PhD thesis, Department of Mathematical Modelling, Technical University of Denmark*. 1995.
  16. Soto M., Sevaux M., Rossi 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.
  17. Dorigo M., Stützle T. Ant colony optimization: overview and recent advances. *Springer International Publishing AG*. 2019. P. 311–352.
  18. Stützle T., Hoos H. H. Ant system. *Future Generation Computer Systems*. 2000. Vol. 16, No. 8. P. 889–914.
  19. 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.)*. К.: ІК ім. В.М. Глушкова НАН України. 2017. С. 41–43.
  20. Mora A. M., García-Sánchez P., Merelo J. J. Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal. *Soft Computing*. 2013. Vol. 17, No. 7. P. 1175–1207.

### Hulianytskyi L. F., Storchevyy V. V. One special problem of UAV routing.

The purpose of the article is to review the tasks of routing vehicles, their classification, as well as consideration of the problem of routing the group of unmanned aerial vehicles (UAVs) during the survey of specified objects. The proposed mathematical model of the problems is that before a given group of aircraft, which can start from different points of launch and must complete the route in special receiving zones having a certain capacity, the task is to fly a number of specified objects (points on the ground) minimizing the total length of the routes or the duration of the flights, provided that each object is visited by one and only one UAV and all objects must be visited. It is shown that the model of traffic of vehicles with multiple depots can be adapted for flight planning of the UAV group in a number of survey tasks. The next solution algorithms are offered: brute force, hybrid local search - tabulated search using greedy algorithm for constructing an initial route, max-min algorithm of ant system (MMAS). Since there are no data libraries for formulated task statements in open access, a set of tasks was generated using the Google Maps API, which allowed you to get tasks that are close to reality (using the coordinates of villages, towns, cities). Tasks have a different number of surveyed objects, aircraft, start and finish depots. For each algorithm, a selection of parameters was made and then a computational experiment was conducted, which allowed comparing the three developed algorithms for solving the routing problem of the UAV group. As a result, it is determined that the method of brute force can be applied only to problems of very small dimension, while it takes a lot of time to execute the algorithm. The local search found improved solutions to problems of medium dimension, and the max-min algorithm of ant system proved to be the best on problems of large dimension, therefore it is defined as perspective for research. The further direction of improvement of the max-min algorithm of ant system is determined by using the principles of search diversification in them.

**Keywords:** unmanned aerial vehicles, route optimization, multiple depots, mathematical model, combinatorial optimization, ant system.

## References

1. Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80–91.
2. Baldacci, R., Toth, P., & Vigo, D. (2010). Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175 (1), 213–245.
3. Huliannytskyi, L.F. (2007). Problema optimizatsii marshrutov transportnykh sredstv s vremennymi oknami. *Computer Mathematics*, 1, 122–132 [in Russian].
4. Montoya-Torres, J. R., López Franco, J., & Nieto Isaza, S. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79, 115–129.
5. Golden, B. L., Raghavan, S., & Wasil, E. A. (Eds.). (2008). The vehicle routing problem: latest advances and new challenges. *Springer Science & Business Media*, 43.
6. Ponda, S. S., Johnson, L. B., & Geramifard, A. (2015). Cooperative mission planning for multi-UAV teams. In: *Handbook of Unmanned Aerial Vehicles*. Dordrecht: Springer, 1447–1490.
7. Liu, Y., Liu, Z., & Shi, J. (2019). Optimization of base location and patrol routes for unmanned aerial vehicles in border intelligence, surveillance, and reconnaissance. *Journal of Advanced Transportation*, 2019, 1–13.
8. Renaud, J., Laporte, G., & Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23(3), 229–235.
9. Huliannytskyi, L. F. (2019). Rozrobka matematychnoi modeli problemy optymizatsii marshrutiv hrupy BPLA za naiavnosti dekilkokh depo. In *Mathematical and simulation modeling of systems*, 324–327 [in Ukrainian].
10. Horbulin, V. P., Huliannytskyi, L. F., & Serhiienko, I. V. (2019). Postanovky ta matematychni modeli problem optymizatsii marshrutiv litalnykh aparativ iz dynamichnymy depo. *Control systems and machines*, 1, 2–15 [in Ukrainian].
11. Toth, P., & Vigo, D. (2014). Vehicle routing: problems, methods, and applications. *Society for Industrial and Applied Mathematics*.
12. Austin, R. (2010). Unmanned aircraft systems: UAVS design, development and deployment. Chichester, UK: John Wiley & Sons, Ltd.
13. Huliannytskyi, L. F., & Rybalchenko, O. V. (2018). Formalizatsiia ta rozviazuvannia odnogo typu zadach marshrutyzatsii BPLA. *The theory of optimal solutions*, 17, 107–114 [in Ukrainian].
14. Pereira, F. B., & Tavares, J. (2009). Bio-inspired algorithms for the vehicle routing problem. *Studies in Computational Intelligence* 161, Springer.
15. Kohl, N. (1995). Exact methods for time constrained routing and related scheduling problems. *PhD thesis, Department of Mathematical Modelling, Technical University of Denmark*.
16. Soto, M., Sevaux, M., & Rossi, A. (2017). Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem. *Computers & Industrial Engineering*, 107, 211–222.
17. Dorigo, M., & Stützle, T. (2019). Ant colony optimization: overview and recent advances. *Springer International Publishing AG*, 311–352.
18. Stützle, T., Hoos, H.H. (2000). Ant system. *Future Generation Computer Systems*, 16(8), 889–914.
19. Huliannytskyi, L. F. (2017). Novyi alhorytm optymizatsii murashynomy koloniiamy. *Modern informatics: problems, achievements and prospects of development*, 41–43 [in Ukrainian].
20. Mora, A. M., García-Sánchez, P., & Merelo, J. J. (2013). Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal. *Soft Computing*, 7(7), 1175–1207.

Одержано 10.05.2019