Теорія оптимальних рішень

УДК 519.8

Л.Ф. ГУЛЯНИЦЬКИЙ, О.В. РИБАЛЬЧЕНКО

ФОРМАЛІЗАЦІЯ ТА РОЗВ'ЯЗУВАННЯ ОДНОГО ТИПУ ЗАДАЧ МАРШРУТИЗАЦІЇ БПЛА

Подано формальну постановку задачі маршрутизації безпілотних літальних апаратів, в якій розглядаються одночасно етапи планування операції та відвідування літальними апаратами встановлених цілей. Розроблено метаевристичний алгоритм розв'язування та проведено його аналіз на основі результатів аналізу обчислювального експерименту.

Вступ. Розглядається формальна постановка задачі маршрутизації безпілотних літальних апаратів (БПЛА), яка описує етап планування операції з відвідування літальними апаратами встановлених цілей. Припускається, що з n пускових точок може бути випущено m літальних апаратів, кожен з яких характеризується можливою дальністю польоту. Існує k цілей, що мають бути відвідані хоча б одним з БПЛА. Кінцева мета операції – це відвідування одним з БПЛА кожної з цілей. Але, оскільки дальність одного польоту обмежена, при проведенні операції може бути задіяно більше, ніж один апарат. До того ж, з метою оптимізації витрат, пускова точка вильоту може відрізнятися від точки посадки. Отже, окрім побудови маршрутів обльоту, потрібно призначити пускові точки. Також для БПЛА існує можливість заміни джерел живлення у пускових точках, тож після приземлення він може отримати нове завдання та продовжити виконання плану. Водночас, важливими аспектами є мінімізація загального часу проведення операції та витрат енергії, що витрачається під час виконання операції. Таким чином, задача полягає у складанні такого плану проведення операції, за якого, з одного боку, завдання буде виконано якомога швидше, а з іншого – витрати енергії для переміщення БПЛА будуть якомога меншими.

Це також вказує на актуальність проблеми оптимізації плану операції з використанням БПЛА, яку буде розглянуто.

Математична постановка задачі. Планувальник операції отримує наступну інформацію:

n – кількість пускових точок (баз БПЛА), з яких можуть здійснюватися вильоти та які можуть приймати апарати;

m – кількість БПЛА;

m_{j0} – максимальна дальність польоту j-го БПЛА, км;

m_{j1} – швидкість польоту j-го БПЛА, км/год;

k – кількість цілей, що необхідно відвідати;

C_{ij} – відстань між точками i та j.

Окремо зазначимо, що застосування формули обчислення відстані між точками на двовимірній площині призведе до похибки через викривлення траєкторії при переміщенні над поверхнею. Для розрахунку відстані по поверхні Землі використаємо формулу гаверсинуса – важливе рівняння у навігації, яке дозволяє обчислити відстань між точками на сфері, за їх довготою та широтою. Є окремим випадком більш загальної формули сферичної тригонометрії, закону гаверсинусів, відносно сторін та кутів сферичних трикутників [1].

Висотою польоту при обчислюванні відстані буде знехтувано.

Виконання перельоту з точки i у точку j апаратом l описуватиметься змінними x_{ijl}:

$$x_{ijl} = \begin{cases} 0, \text{ переліт не виконується} \\ 1, \text{ переліт виконується} \end{cases}, \quad i, j = \overline{1, n+k}, \ l = \overline{1, m}. \tag{1}$$

Загальна відстань D, яку мають подолати всі БПЛА під час виконання завдання згідно з отриманим планом операції обчислюється наступним чином:

$$D = \sum_{i=1}^{n+k} \sum_{j=1}^{n+k} \sum_{l=1}^{m} x_{ijl} C_{ij} \ . \tag{2}$$

Загальний час виконання завдання T згідно з отриманим планом операції обчислюється згідно з формулою:

$$T = \max_{l=\overline{1,m}} \frac{1}{m_{l1}} \sum_{i=1}^{n+k} \sum_{j=1}^{n+k} \sum_{l=1}^{m} x_{ijl} C_{ij} \ , \tag{3}$$

де m_{l1} – згадана вище швидкість l-го БПЛА.

Обмеження. Довжина шляху кожного БПЛА не має перевищувати його максимально допустимої відстані перельоту:

$$\sum_{i=1}^{n+k} \sum_{j=1}^{n+k} x_{ijl} C_{ij} \le m_{l0}, \ l = \overline{1, m}. \tag{4}$$

Кожен БПЛА відвідує певну кількість цілей, отже, прибуття та відправлення від кожної цілі має здійснитися лише один раз. Також, кожна ціль має бути відвідана строго одним БПЛА. Отже, для перевірки коректності розв'язку з точки зору обходу цілей, введемо обмеження.

Один і тільки один БПЛА може покинути ціль:

$$\sum_{l=1}^{m}\sum_{i=1}^{n+k} x_{ijl}C_{ij} = 1,\ j = \overline{n+1, n+k}. \quad (5)$$

Один і тільки один БПЛА може прибути до цілі:

$$\sum_{l=1}^{m}\sum_{j=1}^{n+k} x_{ijl}C_{ij} = 1,\ i = \overline{n+1, n+k}. \quad (6)$$

Слід зазначити, що у формулах (5) та (6) при перевірці кількості перельотів, до цілі та від цілі відповідно, переглядаються також перельоти між депо та цілями. Проте перевірка за цими формулами для самих депо не відбувається.

Таким чином, виконується вимога відвідування кожної цілі. Наступна група обмежень пов’язана з вильотами з депо. Необхідно забезпечити, щоб кожен з БПЛА, що має відвідати набір цілей, виконав переліт від одного з депо до цілі, яка не має вхідних перельотів від інших цілей.

Означимо M як множину індексів апаратів, що мають призначені цілі в отриманому плані.

Умова вильоту БПЛА з депо:

$$\sum_{i=1}^{n}\sum_{j=n+1}^{n+k} x_{ijl} = 1,\ l \in M. \quad (7)$$

Умова повернення у депо:

$$\sum_{i=1}^{n}\sum_{j=n+1}^{n+k} x_{jil} = 1,\ l \in M. \quad (8)$$

Виконання даних умов забезпечує перевірку коректності маршруту кожного БПЛА. Якщо в отриманому плані виконання завдання наявні апарати без призначених цілей, обмеження (7) та (8) не мають сенсу, тому перевірка відбувається для індексів з множини M.

Для розв’язування сформульованої задачі (1) – (8) розроблено наближений алгоритм оптимізації мурашиною колонією (ОМК) з використанням детермінованого локального пошуку (ДЛП) як вбудовану процедуру.

Загальна схема алгоритму ОМК. Серед відомих алгоритмів ройового інтелекту (swarm intelligence) в комбінаторній оптимізації значного поширення набули алгоритми ОМК (ant colony optimization – ACO) [2]. Вони успішно застосовуються для розв’язування багатьох типів задач комбінаторної оптимізації, починаючи з класичної задачі комівояжера.

Основні компоненти обчислювальної схеми мурашиних алгоритмів такі: модель задачі, що подається спеціальним графом; феромонні значення; евристична інформація; пам'ять (локальна та глобальна). В алгоритмах ОМК формується спеціальна модель задачі, що розв’язується, тому вони належать до класу моделе-орієнтованих методів. Модель задачі маршрутизації БПЛА подається у вигляді зваженого графа $G(V,E)$, де $v_i \in V,\ i = 1,...,n+k$ – вершини,

що відповідають компонентам розв'язку, а $e_{ij} \in E$, $e_{ij} = (v_i, v_j)$, $v_i, v_j \in V$ – ребра, які відповідають можливим з'єднанням (переходам) між відповідними вершинами. Для кожного ребра визначена функція вартості з'єднання, що відповідає відстані по поверхні між вершинами, з'єднаними даним ребром, якщо одна з вершин – ціль, інша – ціль чи депо або ж нескінченності, якщо обидві вершини є депо.

На кожному кроці алгоритму для будь-якої вершини $i \in V$ може бути побудована множина сусідніх вершин N_i.

Умови задачі, визначають набір обмежень $\Pi = \Pi(V, E, t)$ для елементів V та E, описані формулами (4) – (8), які визначають припустимість зав'язків між компонентами та з'єднаннями, а в підсумку – і побудованого з них розв'язку. Розв'язки задачі подаються як припустимі шляхи на графі G.

Евристична інформація η_{ij} – це числове значення, що не залежить від знайдених на попередніх кроках розв'язків і відображає ступінь бажаності включення в побудований фрагмент розв'язку того чи іншого нового ребра графа моделі $e_{ij} \in E$. Евристичні значення η_{ij} базуються на апріорній інформації, що відображає умови конкретної задачі та надається джерелом, відмінним від мурах.

Рівень феромону (феромонний слід) – τ_{ij}, що відповідає ребру $e_{ij} \in E$, – це додатне число, яке показує, наскільки часто мурахами використовувалося це ребро на попередніх кроках чи при формуванні повного розв'язку. Феромонні сліди виконують роль довготривалої пам'яті для мурах [2].

Загальну обчислювальну схему алгоритмів ОМК можна подати так [3, 4].

procedure *ACO* (*x*)
ініціалізація_алоритму;
while критерій_завершення_не_задоволений **do**
формування_популяції_мурах; {поточне покоління}
foreach мураха_з_популяції **do** {життєвий цикл мурахи}
ініціалізація_мурахи;
M = *оновлення_пам'яті_мурахи*;
while поточний_стан ≠ повний_розв'язок **do**
A = *локальна_ матриця_мурашиних_маршрутів*;
сформувати_множину припустимих_вершин; (1*)
p = *обчислити_ймовірність_переходів*(*A*, *M*, П);
наступний_стан = *правило_прийняття_рішення*(*p*, П);
перейти_в_наступний_стан(наступний_стан);
if онлайнове_покрокове_оновлення_феромону **then**
відкласти_феромон_на_відвіданій_дузі;
поновити_ матрицю_мурашиних_маршрутів_A;

endif
M = оновити_внутрішній_стан;
endwhile
if онлайнове_відстрочене_оновлення_феромону **then**
foreach відвіданої_дуги_побудованого_розв'язку **do**
відкласти_феромон_на_відвіданій_дузі;
оновити_матрицю_мурашиних_маршрутів_A;
endforeach
endif
завершити діяльність;
endforeach
випаровування_феромону;
оновлення_рекорду (*x*);
дії_Демона; {необов'язково} (2*)
endwhile
end

Особливості реалізації алгоритму. У розробленому алгоритмі як множину припустимих вершин (1*) обрано множину невідвіданих вершин, до яких одна з мурах зі своєї поточної позиції може дістатися не більше, ніж за два кроки [3]. Як дії Демона (2*) виступає оптимізація отриманого розв'язку за допомогою алгоритму ДЛП.

Спочатку для кожної мурахи k із популяції здійснюється формування підмножини припустимих вершин $N_i^k \subseteq N_i$ із множини вершин, сусідніх для тієї вершини $i \in V$, яка є останньою у поточному фрагменті маршруту. Отже, N_i^k – множина можливих переходів за один крок для мурахи k з вершини i. Потім додатково будується D_i^k – множина точок, до яких можливий перехід у два кроки, тобто, всі невідвідані точки з можливістю прямого переходу від однієї з точок попередньої множини N_i^k: $D_i^k = \bigcup_{s \in N_i^k} N_s{}^k$. Іншими словами, множина D_i^k – це об'єднання множин допустимих переходів для кожної з точок s з множини допустимих переходів для обраної вершини i та мурахи k.

Перехід k-ї мурахи з вершини i в j через вершину s на поточній ітерації t здійснюється з ймовірністю, що розраховується за наступною формулою:

$$p_{isj}{}^k = \frac{[\tau_{is}(t) \cdot \tau_{sj}(t)]^{\alpha} \cdot [\eta_{is}(t) \cdot \eta_{sj}(t)]^{\beta}}{\sum_{r \in N_i^k} [\tau_{ir}(t)]^{\alpha} \cdot [\eta_{ir}(t)]^{\beta} \cdot \sum_{r \in N_s^k} [\tau_{rj}(t)]^{\alpha} \cdot [\eta_{rj}(t)]^{\beta}}, \quad (9)$$

де $N_{ilj}{}^k = \{l : \ l \in N_i{}^k, \ \ j \in N_l{}^k\}$.

Формула (9) описує ймовірності для переходів через проміжну вершину, тобто, двокрокового алгоритму ОМК. Для прямих переходів до вершин з множини N_i^k застосовується наступна формула:

$$q_{ij}{}^k = \frac{[\tau_{ij}^{\alpha}(t) \cdot \eta_{ij}^{b}(t)]}{\sum_{r \in N_i^k} [\tau_{ir}^{\alpha}(t) \cdot \eta_{ir}^{b}(t)]}. \qquad (10)$$

Вибір точки для наступного переходу здійснюється серед елементів множини $D_i^k \cup N_i^k$, тобто, може бути обрано як прямий, так і двокроковий перехід. Ймовірність кожного переходу для мурахи k з вершини i обчислюється як $p_{isj}{}^k$, $s \in N_i^k$, $j \in D_i^k$ для двокрокових переходів та $q_{ij}{}^k$, $j \in N_i^k$ для прямих переходів.

Детермінований локальний пошук. Алгоритми ДЛП – це сім'я ітераційних методів, заснована на частковому перебиранні варіантів на кожній ітерації серед точок околу поточної точки, тобто серед сусідніх до неї. В алгоритмах цього типу замість повного перебору застосовується спрямований локальний перебір у підмножинах варіантів, які називаються околами [4]. Загальна схема локального пошуку в задачах мінімізації така: починаючи з деякого припустимого розв'язку задачі, новий розв'язок із кращим значенням цільової функції шукають у його околів і якщо такий розв'язок знайдено, то він приймається і пошук поліпшення розв'язку далі здійснюється вже в його околі. Процес закінчується, коли досягнуто локального оптимуму, тобто коли в околі поточного розв'язку немає ніякого іншого варіанта з меншим значенням цільової функції.

Ефективність алгоритмів локального пошуку істотно залежить від вибору відповідного типу околу $O(x)$. Чим більше окіл, тим імовірніше отримати кращий результат, але розширення околів швидко стає непрактичним. Утім для багатьох задач комбінаторної оптимізації алгоритми з околами розміром більше, ніж $O(n^2)$, де n позначає розмірність задачі, стають неефективними й тому досить рідко використовуються на практиці.

У реалізованому алгоритмі використовується алгоритм 2-opt Ліна [4].

Дослідження ефективності розробленого алгоритму. Дослідження проводилося на основі аналізу результатів обчислювального експерименту із розв'язування задачі маршрутизації БПЛА з наступними параметрами:

- $n = 4$ – кількість депо;
- $m = 3$ – кількість БПЛА;
- $k = 23$ – кількість цілей.

Кожній цілі та депо відповідає точка на карті, програма отримує на вхід широту та довготу точки.

Використовувалося таке апаратне забезпечення: процесор Intel Core i7, 2.2 GHz, з OSX та оперативною пам’яттю 16 GB.

В таблиці наведені результати розв'язування задачі з 23 цілями, 4 депо та 3 БПЛА з використанням алгоритму ОМК без двокрокових переходів (класичний мурашиний алгоритм); алгоритму ОМК з двокроковими переходами (ОМК з далекоглядними мурахами); алгоритму ДЛП, що використовується Демоном (2*).

Для алгоритмів ОМК було здійснено 10 запусків з датчиками псевдовипадкових чисел з рівномірним розподілом, ініціалізованими різними початковими значеннями. Алгоритм ДЛП запущено окремо 10 разів, як початкові значення використано результат, отриманий на першій ітерації кожного запуску ОМК з двокроковими переходами. У таблиці наведено усереднені результати 10 незалежних між собою запусків для кожного розглянутого алгоритму.

ТАБЛИЦЯ. Усереднені результати обчислень

Алгоритм	Класичний ОМК	ОМК з далекогляд-ними мурахами	ДЛП
Цільова функція *f*, км	269,8836	255,4535	278,6548
Відносна похибка ε, %	6,63	0,93	10,09
Час, с	40,22	43,06	1,03

Для обчислення відносної похибки ε використовується f^* – найменше відоме значення цільової функції для даної задачі ($f^* = 253,097$), а відносна похибка розв’язку обчислюється за формулою $\varepsilon = 100 \cdot (f - f^*) / f^*$.

З таблиці можна зробити висновок, що розроблений алгоритм з двокроковими переходами дозволив отримати істотно кращий середній результат.

Висновки. Сформульовано та формалізовано один з типів задачі маршрутизації БПЛА, запропоновано і реалізовано алгоритм її розв’язування. Виконано дослідження розробленого спеціального алгоритму ОМК, отримані результати розв’язування задачі з реальними об’єктами на місцевості. Завдяки порівнянню отриманих розв’язків продемонстровано переваги розробленого алгоритму. Отримано відносну похибку результатів роботи кожного алгоритму, що дозволяє порівняти їх ефективність.

Л.Ф. Гуляницкий, О.В. Рыбальченко

ФОРМАЛИЗАЦИЯ И РЕШЕНИЕ ОДНОГО ТИПА ЗАДАЧ МАРШРУТИЗАЦИИ БПЛА

Рассмотрена формальная постановка задачи маршрутизации БПЛА, в которой одновременно рассматриваются этапы планирования операции и посещение аппаратами установленных целей. Разработан метаэвристический алгоритм решения и проведен его анализ на основании результатов анализа вычислительного эксперимента.

L.F. Hulianytskyi, O.V. Rybalchenko

FORMALIZATION AND SOLUTION OF THE PARTICULAR UAV ROUTING PROBLEM

The formal statement of the particular UAV routing problem is covered. The problem under examination unites the stages of pre-mission planning and visiting targets. The methaeuristic algorithm for solving the problem have been developed and analyzed based on the results of computational experiment.

Список літератури

1. Van Brummelen G. Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry. Princeton, N.J., Oxford: Princeton University Press, 2013. P. 160 – 162.
2. Dorigo M., Stützle T. Ant colony optimization. Cambridge (MA): MIT Press, 2004. 348 p.
3. Гуляницький Л.Ф. Новий алгоритм оптимізації мурашиними колоніями. Пр. Міжнар. конф., присвяченої 60-річчю заснування Інституту кібернетики імені В.М. Глушкова НАН України "*Сучасна інформатика: проблеми, досягнення та перспективи розвитку*" (Київ, 13 – 15 грудня 2017). К.: ІК імені В.М. Глушкова НАН України, 2017. С. 41 – 43.
4. Гуляницький Л.Ф., Мулеса О.Ю. Прикладні методи комбінаторної оптимізації. К.: Видавничо-поліграфічний центр "Київський університет", 2016. 142 с.

Одержано 16.04.2018