

**ДВОРІВНЕВИЙ АЛГОРИТМ
ДЛЯ ЗАДАЧІ МАРШРУТИЗАЦІЇ БПЛА
З РУХОМИМИ ЦІЛЯМИ**

Вступ. Безпілотні літальні апарати (БПЛА, Unmanned aerial vehicles – UAV) знаходять застосування у багатьох галузях цивільної авіації. Успіх БПЛА це наслідок їхньої доступності та гнучкості в експлуатації. Здатність пересуватися довільною місцевістю та досягати віддалених локацій робить БПЛА невід'ємним інструментом підвищення операційної ефективності ланцюгів постачання.

Водночас БПЛА відіграють ключову роль у сфері оборони. Вони надають можливості як для прихованого стеження за супротивником, так і для безпосередньої інтервенції. Також БПЛА зменшують ризики для особового складу порівняно з пілотованими літальними апаратами завдяки можливості віддаленого керування.

Залежно від особливостей предметної області виникає конкретна задача маршрутизації БПЛА. У роботі [1] проведено огляд таких задач, включно з динамічними задачами маршрутизації. Втім, динамічність присутніх у літературі результатів полягає або у зміні маршруту внаслідок реакції на зміни у доступній інформації [2, 3], або у присутності рухомих депо зі сталими цілями [4].

Натомість предметна область оборони часто має справу з рухомими цілями. Зазвичай такими задачами займається теорія диференційних ігор. Проте, отримані у її рамках класичні результати [5, 6] часто є занадто громіздкими для ефективного застосування у практичних задачах. Особливо ситуація ускладнюється присутністю великої кількості незалежних переслідувачів та переслідуваних.

У літературі [7] досягнуто консенсусу, що прикладні задачі маршрутизації потребують цілої команди БПЛА для ефективної роботи з групою цілей. Таким чином виникає потреба у нових методах маршрутизації групи БПЛА з групою рухомих цілей. Ми запропонуємо один такий алгоритм заснований на аналогічному алгоритмі для задачі із сталими цілями [8].

Запропоновано дворівневий алгоритм побудови маршрутів групи БПЛА, що переслідують групу рухомих цілей. Проведено порівняльний аналіз ефективності методу з класичною евристикою найближчого сусіда. Розглянуто випадки кластеризації цілей за позицією та напрямком. Отримані результати свідчать про ефективність дворівневого алгоритму за умови доречної кластеризації.

Ключові слова: безпілотний літальний апарат, дрон, оптимізація маршрутів, методи кластеризації.

1. Загальна постановка задачі маршрутизації БПЛА з рухомими цілями

Вхідними даними задачі переслідування групи рухомих цілей групою БПЛА на декартовій площині у загальному випадку є:

- кількість цілей n ;
- їхні початкові позиції $z_i(0) = (x_i(0), y_i(0)), 1 \leq i \leq n$, де z_i – функція від часу, значення якої це вектори з двома дійсними координатами x_i, y_i ;
- кількість БПЛА m ;
- їхні початкові позиції $z'_j(0) = (x'_j(0), y'_j(0)), 1 \leq j \leq m$;
- закони руху цілей, задані у вигляді $\dot{z}_i = f_i(z, z', c_i), 1 \leq i \leq n$, де z, z' – вектори позицій усіх цілей та БПЛА відповідно, а c_i – певні константи, що описують технічні характеристики цілі, наприклад швидкість руху;
- допустимі множини руху БПЛА задані в аналогічному вигляді $\dot{z}'_j \in F_j(z, z', c'_j), 1 \leq j \leq m$, де F_j набуває множинних значень, а c'_j – певні константи, що описують технічні характеристики БПЛА, наприклад швидкість руху.

Задачу маршрутизації БПЛА визначимо як задачу розбиття множини цілей на (можливо порожні) впорядковані послідовності $I_j = (i_{j,1}, i_{j,2}, \dots, i_{j,k_j}), 1 \leq j \leq m$. Невпорядковане об'єднання послідовностей має дорівнювати множині усіх цілей: $I_1 \cup I_2 \cup \dots \cup I_m = \{1, 2, \dots, n\}$. Для заданого розбиття, БПЛА з індексом j буде переслідувати цілі з індексами з послідовності I_j у вказаному порядку відповідно до їхніх законів руху та з дотриманням обмежень, що накладаються на рух самого БПЛА. Час перехоплення цілі з індексом i позначимо t_i . З контексту буде зрозуміло, який саме БПЛА здійснює перехоплення.

Найважливішими метриками ефективності розбиття це сумарна довжина маршрутів усіх БПЛА та максимальна довжина маршруту одного БПЛА (англ. *makespan*). Наведемо явні формули сумарної та максимальної довжин відповідно:

$$\sum_{j=1}^m \sum_{l=1}^{k_j} d(z'_j(t_{i_{j,l-1}}), z'_j(t_{i_{j,l}})), \tag{1}$$

$$\max_{j=1..m} \sum_{l=1}^{k_j} d(z'_j(t_{i_{j,l-1}}), z'_j(t_{i_{j,l}})). \tag{2}$$

Тут функція d позначає довжину частини траєкторії між відповідними точками площини, а $t_{i_{j,0}} = 0$ – момент початку руху БПЛА з індексом j .

Мета задачі маршрутизації це мінімізація однієї з цих метрик. Взагалі кажучи, один алгоритм маршрутизації може переважати інший за сумарною довжиною, але програвати за максимальною [9]. У такому разі вибір алгоритму для застосування на практиці це задача відповідного фахівця, який має враховувати особливості предметної області, що знаходяться за межами нашого дослідження. З огляду на це, ми не проводили порівняльний аналіз значень двох метрик.

2. Конкретна постановка задачі маршрутизації БПЛА з рухомими цілями

В цій статті розглядається випадок спрощених законів руху як цілей, так і БПЛА, а саме:

- вважається, що кожна ціль рухається зі сталою швидкістю за наперед вказаною лінійною траєкторією (виконує певне завдання і не має можливості відхилитися від курсу), тобто $\dot{z}_i = c_i w_i, 1 \leq i \leq n$, де $w_i = (u_i, v_i)$ – одиничний вектор з дійсними координатами u_i, v_i ;
- БПЛА перехоплюють цілі миттєво (часом перехоплення можна знехтувати у масштабі часу переслідування);
- допустимі рухи БПЛА обмежені лише їхньою максимальною швидкістю та не залежать від їхньої минулої траєкторії (зміна курсу відбувається миттєво), тобто $\dot{z}'_j \in B(c'_j)$, де $B(c'_j)$ позначає кулю з центром в нулі та радіусом c'_j .

Останні спрощення суперечать фізичній природі процесу переслідування, проте не впливають на розв'язок суттєво, адже задача – стійка за фізично змістовних законів руху. З міркувань стійкості, ми також не розглядаємо питання точності обчислень у цій статті. Водночас програмний продукт, розроблений нами для проведення усіх експериментів, належно дбає про стійкість обчислень, зокрема, про точність визначення часу та позиції перехоплення одним БПЛА однієї цілі. Зі стійкості цього кроку алгоритму за теоремою про арифметичні дії впливатиме й стійкість усього алгоритму.

3. Класична евристика найближчого сусіда для задачі маршрутизації БПЛА

Класична евристика маршрутизації це пошук найближчого сусіда. Опишемо алгоритм маршрутизації заснований на ній:

- створюємо чергу з пріоритетом та наповнюємо її всіма БПЛА. Вважаємо, що вона підтримує ефективні операції пошуку елемента з найменшим пріоритетом (першого елемента) та оновлення пріоритету цього елемента. Пріоритет БПЛА з індексом j – час перехоплення попередньої цілі, або ж нуль якщо БПЛА лише починає свій шлях;
- позначаємо усі цілі, які ще не перехоплені та наповнюємо ними структуру даних, яка підтримує ефективні ітерацію та видалення елемента за ідентифікатором, наприклад хеш-таблицю;
- повторюємо наступні дії поки є хоча б одна не перехоплена ціль:
 1. Розглядаємо перший БПЛА у черзі, нехай його індекс j .
 2. Для кожної ще не перехопленої цілі з індексом i обчислюємо час перехоплення t_i нашим БПЛА.
 3. Обираємо для переслідування ціль i^* з найменшим часом t_{i^*} .
 4. Оновлюємо пріоритет БПЛА з індексом j у черзі значенням t_{i^*} .
 5. Позначаємо ціль з індексом i^* як вже перехоплену.

Для пункту 2 у нашій моделі достатньо розв'язати квадратне рівняння $d(z_i(t), z'_j(t_i)) = c'_j(t - t_i)$. За умови, що швидкість руху цілі c_i строго менша граничної швидкості c'_j руху БПЛА, це рівняння завжди має дійсний розв'язок $t \geq t_i$.

Зауважимо, що задачу з нерухомими цілями можна розглядати як частковий випадок задачі з рухомими цілями. Для випадку нерухомих цілей відомо [10], що евристика пошуку найближчого сусіда не є алгоритмом апроксимації. Іншими словами, не існує певної константи C такої, що для довільних вхідних даних отриманих цим алгоритмом розв'язок буде не більш ніж у C разів гіршим за оптимальний розв'язок. Оскільки евристика не є алгоритмом апроксимації у частковому випадку, то й у загальному випадку теж.

4. Дворівневий алгоритм з кроком кластеризації

Водночас, у критичних предметних областях ця властивість – вкрай бажана. Як наслідок, виникає потреба в алгоритмах апроксимації з поліноміальним часом роботи. Для задачі з нерухомими цілями одним з них є алгоритм Арори [11]. Цей алгоритм дозволяє за поліноміальний час досягнути довільної наперед заданої константи апроксимації $C > 1$, тобто це схема апроксимації (з поліноміальним часом роботи). В його основі лежить розбиття простору на серію сіток із залежними від бажаної точності розмірами.

Такий процес можна описати як ієрархічну кластеризацію [12]. Саме тому ми розглянули дворівневий алгоритм маршрутизації де задачі розбиття цілей між БПЛА та впорядкування цілей кожного окремого БПЛА відбуваються окремо. Для доречності безпосереднього порівняння, задачу впорядкування цілей кожного окремого БПЛА ми також будемо розв'язувати евристикою найближчого сусіда. Натомість для розбиття цілей між різними БПЛА буде використано кластеризацію методом k середніх.

Пропонований нами алгоритм має наступні кроки:

1) за початковими положеннями $z_i(0), 1 \leq i \leq n$ усіх цілей провести кластеризацію методом k середніх з $k = \min(n, m)$ кластерів. Оскільки така задача кластеризації – NP-складна, цей крок виконується наближеним ітеративним методом;

2) застосувати угорський алгоритм [13] для розподілу кластерів між БПЛА. Для алгоритму використати матрицю попарних відстаней між початковими положеннями $z'_j(0), 1 \leq j \leq m$ усіх БПЛА та центроїдами усіх кластерів;

3) всередині кожного кластера використати попередній алгоритм маршрутизації заснований на евристиці пошуку найближчого сусіда.

5. Обчислювальні експерименти

Опишемо методологію проведення обчислювальних експериментів. Всього було проведено чотири експерименти. Кожен з них складався з 30 сценаріїв (наборів вхідних даних). Отримані результати у кожному експерименті групували та візуалізували кількома способами. Нарешті, результати обох алгоритмів порівнювалися між собою якісно та кількісно.

У кожному з сценаріїв частина даних була сталою, а решта обиралися випадково з певного ймовірнісного розподілу. В усіх експериментах: $n = 30$, $m = 4$, $c_i \in [5, 25]$, $c'_j = 30$. Для відтворюваності експериментів, ми опублікували використані генератори випадкових чисел та отримані набори даних на GitHub [14].

Перший експеримент: випадкові початкові позиції та випадкові напрямки руху цілей. Усі розподіли у цьому експерименті рівномірні.

– початкові позиції цілей обираються з квадрату $[-1000, 1000] \times [-1000, 1000]$;

– кут руху вибирається з напівінтервалу $[0, 2\pi)$.

Другий експеримент: скупчені початкові позиції та випадкові напрямки руху.

– обираються m випадкових центрів скупчення зі згаданого квадрата;

– для кожного центра обирається декілька точок з кулі з цим центром та радіусом 150.

Третій експеримент: випадкові початкові позиції та скупчені напрямки руху.

– обираються m випадкових кутів зі згаданого напівінтервалу;

– для кожного куті обирається декілька напрямків руху розподілу фон Мізеса з цим кутом як центром та параметром $\kappa = 8$ (ідейний аналог нормального розподілу на колі) [15].

Четвертий експеримент: скупчені позиції та напрямки руху. Поєднує особливості другого та третього експериментів.

Покажемо зображення одного сценарію з кожного експерименту на рисунку.

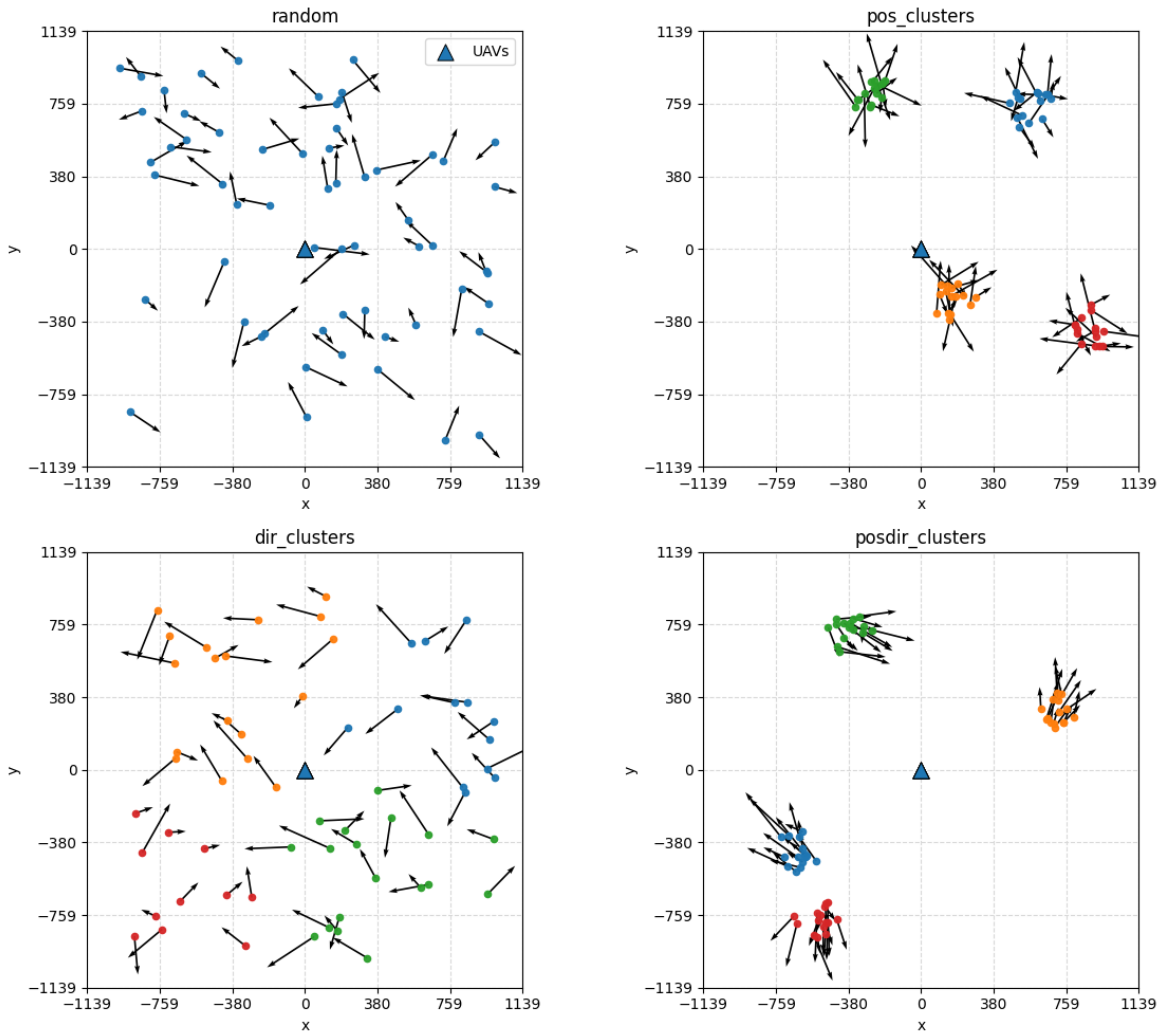


РИСУНОК. Початкові розташування та напрямки руху в одному з тестів кожного експерименту

З точки зору кластеризації k середніх, маємо такі якісні властивості:

- у першому експерименті жодних змістовних кластерів виділити неможливо у жоден момент часу;
- у другому – початково існують кластери за позицією які з часом зникають;
- у третьому – початково змістовних кластерів не існує проте з плином часу вони з'являються;
- у четвертому – змістовні кластери існують одразу і зберігаються з плином часу.

Якісні результати експериментів наведено у табл. 1 – 4.

З табл. 1–3 видно, що в перших трьох експериментах крок кластеризації за початковою позицією шкодить ефективності за обома метриками.

З табл. 4 видно, що в останньому експерименті перевага кластеризації колосальна і становить 2–3 рази залежно від метрики.

З табл. 1–4 видно, що кількісні результати алгоритму найближчого сусіда майже не залежать від експерименту, тобто він явно використовує не всі властивості вхідних даних.

ТАБЛИЦЯ 1. Порівняння класичного та дворівневого алгоритмів у першому експерименті

Алгоритм	Середня сумарна довжина (1.1)	Середня максимальна довжина (1.2)
Найближчого сусіда	32774.07	13412.12
З кластеризацією	55654.21	30034.14

ТАБЛИЦЯ 2. Порівняння класичного та дворівневого алгоритмів у другому експерименті

Алгоритм	Середня сумарна довжина	Середня максимальна довжина
Найближчого сусіда	30226.45	10936.28
З кластеризацією	35062.40	15973.80

ТАБЛИЦЯ 3. Порівняння класичного та дворівневого алгоритмів у третьому експерименті

Алгоритм	Середня сумарна довжина	Середня максимальна довжина
Найближчого сусіда	35701.33	14462.95
З кластеризацією	48818.49	28109.28

ТАБЛИЦЯ 4. Порівняння класичного та дворівневого алгоритмів у четвертому експерименті

Алгоритм	Середня сумарна довжина	Середня максимальна довжина
Найближчого сусіда	35719.66	13412.12
З кластеризацією	13232.20	5761.62

Висновки. За умови існування змістовних кластерів впродовж усього часу задачі крок кластеризації варто застосовувати одразу. Якщо ж заданий сценарій не володіє такою властивістю, то безпосередньо застосовувати крок кластеризації не варто.

Водночас є підстави вважати, що зміна способу кластеризації може надати дворівневному підходу якісну перевагу в усіх експериментах, де змістовні кластери існують хоча б певний час. Зокрема, ми плануємо проведення наступних досліджень у майбутньому:

1) порівняльний аналіз методів кластеризації: лише за позицією, лише за напрямком руху, комбінований (наприклад, за зваженим інтегралом відстані між парою цілей як мірою їхньої близькості);

2) порівняльний аналіз класичної та ієрархічної кластеризації. Частковий випадок з нерухомими цілями дає підстави вважати, що ієрархічна кластеризація дозволить покращити впорядкування цілей призначених кожному окремому БПЛА;

3) розробка власне алгоритму апроксимації заснованого на динамічній ієрархічній кластеризації.

Фінансування. Автори не отримували фінансування для проведення досліджень та написання статті.

Подяки. Автор висловлює подяку В.В. Семенову та рецензентам за конструктивні поради, щодо написання статті.

Список літератури

1. Hulianytskyi L., Byshovets N., Zhdanova O. About the Problem of Drone Routing. *Cybernetics and Computer Technologies*, 2024. 3. P. 34–47. <http://dx.doi.org/10.34229/2707-451X.24.3.4>
2. Long Y. et al. Dynamic Truck–UAV Collaboration and Integrated Route Planning for Resilient Urban Emergency Response. *IEEE Transactions on Engineering Management*. 2024. Vol. 71. P. 9826–9838. <http://dx.doi.org/10.1109/TEM.2023.3299693>
3. Arsie A., Frazzoli E. Efficient routing of multiple vehicles with no explicit communications. *International Journal of Robust and Nonlinear Control*. 2007. No. 2. P. 154–164. <http://dx.doi.org/10.1002/rnc.1258>
4. Horbulin V.P., Hulianytsky L.F., Sergienko I.V. Optimization of UAV Team Routes in the Presence of Alternative and Dynamic Depots. *Cybernetics and Systems Analysis*, 2020. No. 2. P. 195–203. <http://dx.doi.org/10.1007/s10559-020-00235-8>
5. Friedman A. *Differential games*. Dover Publications, 2013. 368 p.
6. Isaacs R. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Dover Publications, 2012. 741 p.
7. Kucherenko Yu., Naumenko M., Kuznetsova M. Analysis Experience Use Unbeatural Vehicle Apparatus And Determination Their Further Development During Conduct Netsset Central Operations. *Systems of Arms and Military Equipment*. 2018. No. 1. P. 25–30. <http://dx.doi.org/10.30748/soivt.2018.53.03>
8. Ding C., Cheng Y., He M. Two-level genetic algorithm for clustered traveling salesman problem with application in large-scale TSPs. *Tsinghua Science and Technology*. 2007. No. 4. P. 459–465. [https://doi.org/10.1016/S1007-0214\(07\)70068-8](https://doi.org/10.1016/S1007-0214(07)70068-8)
9. Kuo R. J. et al. Applying NSGA-II to vehicle routing problem with drones considering makespan and carbon emission. *Expert Systems with Applications*. 2023. No. 221. P. 119777. <https://doi.org/10.1016/j.eswa.2023.119777>
10. Frieze A., Pegden W. The bright side of simple heuristics for the TSP. *arXiv preprint*, 2023. <https://arxiv.org/abs/2310.03222>
11. Arora S. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*. 1998. No. 5. P. 753–782. <http://dx.doi.org/10.1145/290179.290180>
12. Ran X. et al. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artificial Intelligence Review*. 2023. No. 8. P. 8219–8264. <http://doi.org/10.1007/s10462-022-10366-3>
13. Vásconez J.P. et al. Smart Delivery Assignment through Machine Learning and the Hungarian Algorithm. *Smart Cities*. 2024. No. 3. P. 1109–1125. <https://doi.org/10.3390/smartcities7030047>
14. Clustered pursuit generated test suite. *GitHub*. https://github.com/nika-skybytska/clustered-pursuit/blob/f64acb39a5b23a902e2cae7f6f8d69ae316fd0fb/tests/generated_suite.json (звернення: 30.08.2025)
15. Benlakhdar S., Rziza M., Thami R.O. In-depth Analysis of von Mises Distribution Models: Understanding Theory, Applications, and Future Directions. *Statistics, Optimization & Information Computing*. 2024. No. 4. P. 1210–1230. <https://doi.org/10.19139/soic-2310-5070-1919>

Одержано 04.09.2025

Скибицький Нікіта Максимович,

аспірант факультету комп'ютерних наук та кібернетики

Київського національного університету імені Тараса Шевченка, Київ.

<https://orcid.org/0009-0002-9507-3589>

n.skybytskyi@knu.ua

MSC 90C27, 68Q25

Nikita Skybytskyi

Two-Level Algorithm for the UAV Routing Problem with Moving Targets

Taras Shevchenko National University of Kyiv, Kyiv

Correspondence: n.skybytskyi@knu.ua

Introduction. UAV routing problems gain relevance considering expanding uses in the defense industry. Contrary to the civil aviation with static depots and end users, the problems arising in defense require drone routing with moving targets.

The theory of differential games has extensively studied the problems related to pursuit of dynamic targets in the general case, but the resulting solutions are difficult to apply in practical scenarios. Hence combinatorial methods are needed for simplified practical models. In particular, we consider all targets to be moving with a constant velocity vector.

This article proposes a two-level algorithm for the problem of routing a group of UAVs with moving targets. We compare different clustering methods, including clustering by position, velocity, and a combined approach.

The purpose of the article is to show the superiority of a two-level algorithm over the classical algorithm in solution quality under certain conditions. Additionally, we establish the necessity of the combined clustering methods in the general case and predict a correspondence between optimal clustering methods and practical routing scenarios.

Results. We established that a two-level algorithm is better than the original algorithm under clustering conditions. We developed software that can be used to solve instances of the UAVRP with moving targets.

Conclusions. A two-level algorithm improves the process of routing a group of UAVs in pursuit of a group of moving targets. A combined clustering strategy proved superior for total route length minimization.

Keywords: unmanned aerial vehicle, drone, route optimization, clustering methods.