

*Яценко Олександр,
асистент кафедри комп'ютерних наук та інформаційних технологій,
Житомирський державний університет імені Івана Франка,
м. Житомир, Україна*

ЕВРИСТИЧНІ АЛГОРИТМИ КЕРУВАННЯ РОЄМ БПЛА

Алгоритми керування роєм безпілотних літальних апаратів (БПЛА) використовуються для керування групою або роєм БПЛА з метою досягнення спільних завдань, таких як виконання розвідувальних місій, пошуку та рятування, патрулювання, та інших. Важливо, щоб рої БПЛА працювали разом, уникнули зіткнень і виконали завдання ефективно. Зазвичай алгоритми керування роєм БПЛА базуються на технологіях штучного інтелекту, машинного навчання та передових системах навігації. Комбінування цих алгоритмів допомагає створити ефективні та безпечні рої БПЛА для виконання різних завдань.

Значну роль у розробці систем керування роєм БПЛА відіграють евристичні алгоритми: вони забезпечують ефективне та адаптивне вирішення складних задач, що виникають під час організації колективної поведінки безпілотників, особливо в умовах невизначеності та динамічних змін середовища [1]. Евристика – це метод розв'язання задачі, який базується на досвіді, спостереженнях та інтуїції, і використовується там, де аналітичні методи можуть бути складними або неможливими для застосування.

Секція 4. Технології розробки інформаційних систем

Прикладами евристичних алгоритмів є:

1) Генетичні алгоритми. Генетичні алгоритми є підвидом еволюційних алгоритмів та використовуються для розв'язання задач, у яких класичні методи оптимізації є неефективними або занадто складними через великий простір пошуку чи обмеження. Загалом ці алгоритми імітують процес еволюції популяції рішень і включають такі етапи: створення початкової популяції потенційних рішень (індивідів), які представлені у вигляді хромосом (зазвичай у бінарному, числовому або іншому форматі) (ініціалізація); відбір найбільш пристосованих індивідів для створення наступного покоління (селекція); поєднання частин «хромосом батьків» для створення нових індивідів (нащадків); внесення випадкових змін у деякі гени хромосом для забезпечення різноманітності в популяції та уникнення локальних екстремумів (мутація); відбір індивідів, що перейдуть у наступне покоління, наприклад, через повну або часткову заміну; умова зупинки [2].

2) Ройові алгоритми. Ройові алгоритми – це алгоритми оптимізації та штучного інтелекту, що базуються на спостереженнях за поведінкою груп організмів у природі (таких як мурахи, бджоли, риби чи птахи). Ці алгоритми використовують колективну поведінку агентів («частинок»), що взаємодіють один з одним та з навколишнім середовищем, для вирішення складних задач. Найбільш відомими ройовими алгоритмами є: алгоритм оптимізації рою частинок (Particle Swarm Optimization, PSOP), мурашиний алгоритм (Ant Colony Optimization, ACO), алгоритм рою бджіл (Bee Algorithm) та алгоритм імітації зграї птахів (Bird Flocking).

Особливості ройових алгоритмів: децентралізація (немає єдиного керівного центру і кожен агент працює автономно, спираючись на локальні правила і взаємодії); колективний інтелект (група агентів досягає «розумної» поведінки завдяки взаємодії між собою); адаптивність (система здатна адаптуватися до змін середовища або умов задачі); емерджентність (загальна поведінка рою виникає в результаті простих локальних взаємодій) [3]. Основною перевагою ройових алгоритмів є їх здатність ефективно вирішувати складні задачі в умовах невизначеності та великого пошукового простору.

3) Алгоритм табу-пошуку (*Tabu Search*). Алгоритм табу-пошуку – метод оптимізації, що використовується для вирішення складних комбінованих задач. Основна ідея алгоритму полягає у використанні так званого табу-списку (списку заборонених рішень) для керування процесом пошуку та уникнення циклів.

Основні етапи алгоритму табу-пошуку: ініціалізація (генерується початкове рішення (може бути випадковим або заданим) та встановлюється порожній табу-список та інші параметри алгоритму (наприклад, максимальна довжина списку або критерії зупинки)); генерація сусідів (визначається набір «сусідів» для поточного рішення (нових рішень, що утворюються шляхом незначної зміни поточного)); вибір найкращого сусіда (серед сусідів обирається найкращий за певною цільовою функцією, навіть якщо це рішення гірше за поточне); оновлення табу-списку (поточне рішення додається до табу-списку, якщо табу-список перевищує максимальну довжину, видаляється найстаріший запис);

Секція 4. Технології розробки інформаційних систем

оновлення найкращого рішення; перевірка критерію зупинки і, якщо критерій зупинки виконано – завершення алгоритму.

4) Імітація відпалу. Евристичний алгоритм імітації відпалу (Simulated Annealing, SA) – це метод оптимізації, що використовується для пошуку наближених рішень складних задач, зокрема тих, що мають велику кількість можливих рішень або є NP-повними. Змодельований алгоритм відпалу може дати обґрунтоване наближення для функції з великим простором пошуку.

Опис алгоритму: ініціалізація (початкове рішення вибирається випадково, і йому призначається певна «температура»); ітерація (випадковим чином генерується сусіднє рішення) і якщо це рішення краще, воно приймається, якщо гірше – приймається з певною ймовірністю, що залежить від температури і різниці в «енергії» (яка, в контексті оптимізації, може бути функцією вартості або якості рішення); охолодження (температура зменшується впродовж деякого часу (зазвичай експоненційно або лінійно)); завершення (завершується, коли температура досягає мінімального значення або коли число ітерацій перевищує заданий ліміт).

5) Штучні нейронні мережі (ШНМ). ШНМ – евристичний метод заснований на імітації роботи нейронів у мозку тварин та використовує адаптивні алгоритми для пошуку рішень, наближених до оптимальних. Відповідно до цього підходу, ШНМ намагаються навчитися з даних, виконуючи ітеративний процес пошуку, який не завжди гарантує знайдення ідеального рішення, але дозволяє досягти ефективних та корисних результатів в реальних умовах.

ШНМ працюють як евристичний метод, для якого характерними є: навчання через зворотний зв'язок («навчаються» на основі помилок) і після кожної ітерації мережа коригує свої параметри (ваги нейронів), щоб наблизитись до правильного результату, що дозволяє мережам шукати оптимальні рішення в умовах неповних або шумних даних; невизначеність та локальні оптимуми (як і в багатьох евристичних методах, процес навчання ШНМ може призводити до локальних мінімумів у просторі можливих рішень, а це означає, що мережа може не знайти глобально оптимальне рішення, але зазвичай вона знаходить рішення, достатньо хороші для практичного застосування); пошук за допомогою проб і помилок; застосування в складних завданнях (ШНМ використовуються для вирішення складних задач, таких як класифікація, регресія, розпізнавання образів, прогнозування тощо та здатні працювати з великими об'ємами даних, де традиційні методи не завжди ефективні).

Переваги евристичного підходу в нейронних мережах:

- гнучкість: нейронні мережі можуть адаптуватися до нових умов, змінюючи свої параметри в процесі навчання;
- адаптивність: можуть ефективно працювати з великими і складними наборами даних, знаходячи рішення без чітких аналітичних або математичних моделей;

Секція 4. Технології розробки інформаційних систем

- швидкість та ефективність: для деяких задач нейронні мережі можуть значно скоротити час, необхідний для отримання результату, порівняно з іншими методами.

Недоліки:

- локальні мінімуми: ШНМ можуть «застрягти» в локальних мінімумах або субоптимальних рішеннях;
- для досягнення хороших результатів ШНМ потрібні значні обсяги даних для навчання.
- такі параметри, як кількість шарів, розмір кожного шару та швидкість навчання, часто потребують ретельного налаштування і можуть сильно вплинути на результат.

Таким чином, хоча ШНМ не є суто евристичним методом, їх процес навчання можна розглядати як евристичний, оскільки вони використовують ітеративний, адаптивний підхід для вирішення складних завдань з неповними або неточними даними.

Як видно з наведеного вище та аналізу наукових праць, евристичні алгоритми можуть стати потужними інструментами для вирішення складних задач керування роєм БПЛА. Їхні використання надає такі переваги в порівнянні з класичними:

1) Адаптивність до динамічних умов. Евристичні алгоритми, такі як алгоритм рою частинок або генетичні алгоритми, дозволяють швидко адаптувати маршрут і поведінку рою до змін у середовищі, таких як нові перешкоди, зміна погодних умов або втрати зв'язку між БПЛА [3, 4].

2) Зменшення обчислювальної складності. Замість пошуку точного рішення в усьому просторі можливих варіантів, евристики зосереджуються на знаходженні наближених рішень за прийнятний час. Це особливо важливо для великого рою, де класичні методи оптимізації можуть бути надто ресурсомісткими [5].

3) Енергоефективність. Евристики допомагають зменшити витрати енергії, забезпечуючи оптимальні маршрути для кожного апарата. Наприклад, алгоритм мурашиної колонії ефективно розподіляє завдання між БПЛА, зменшуючи зайві переміщення та витрати енергії [4].

4) Масштабованість. Евристичні підходи добре працюють із великими групами БПЛА, оскільки обчислення зазвичай базуються на локальних правилах. Це дозволяє підтримувати ефективність роботи навіть за значного збільшення розміру рою [1].

5) Координація та уникнення конфліктів. Евристичні алгоритми, такі як бджолиний алгоритм або модифіковані алгоритми рою частинок, забезпечують уникнення зіткнень і синхронізацію між БПЛА, що є ключовим для безпеки та ефективності рою [3].

6) Гнучкість у розподілі завдань. Евристичні алгоритми дозволяють оптимально розподіляти завдання між БПЛА відповідно до їхніх позицій, ресурсів і поточного стану. Це підвищує ефективність виконання місій, таких як розвідка, доставка вантажів або рятувальні операції [1].

Секція 4. Технології розробки інформаційних систем

Евристичні алгоритми є важливим інструментом для ефективного керування роєм БПЛА, оскільки вони дозволяють вирішувати складні задачі оптимізації, координації та адаптації в умовах невизначеності [6, 7]. Так для оптимізації маршруту БПЛА можна використати мурашиний, генетичний алгоритми або алгоритм A*; для координації та взаємодії між БПЛА – алгоритм рою часток та бджолиний алгоритм; для уникнення зіткнень та подолання перешкод – колективної поведінки чи силового поля; для розподілу завдань між БПЛА – генетичний алгоритм або алгоритм рою бджіл; для адаптації до змін середовища – алгоритм мурашиної колонії.

Загалом евристичні алгоритми забезпечують високий рівень автономності та надають багато можливостей для покращення ефективності та адаптивності рою БПЛА дозволяючи вирішувати складні завдання оптимізації, координації та уникнення зіткнень у реальному часі. Ці переваги роблять евристичні ключовою технологією для сучасних і майбутніх систем безпілотного управління.

Список використаних джерел та літератури

1. Computational Intelligence Algorithms for UAV Swarm Networking and Collaboration: A Comprehensive Survey and Future Directions / P. Cao et al. *IEEE Communications Surveys & Tutorials*. 2024. P. 1. URL: <https://doi.org/10.1109/comst.2024.3395358> (date of access: 14.11.2024).
2. Schwefel H.-P. Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Basel : Birkhäuser Basel, 1977. URL: <https://doi.org/10.1007/978-3-0348-5927-1> (date of access: 14.11.2024).
3. Yan K., Xiang L., Yang K. Cooperative Target Search Algorithm for UAV Swarms with Limited Communication and Energy Capacity. *IEEE Communications Letters*. 2024. P. 1. URL: <https://doi.org/10.1109/lcomm.2024.3374797> (date of access: 14.11.2024).
4. State-of-the-Art and Future Research Challenges in UAV Swarms / S. Javed et al. *IEEE Internet of Things Journal*. 2024. P. 1. URL: <https://doi.org/10.1109/jiot.2024.3364230> (date of access: 14.11.2024).
5. Advances in Swarm Intelligence / ed. by Y. Tan, Y. Shi, M. Tuba. Cham : Springer International Publishing, 2020. URL: <https://doi.org/10.1007/978-3-030-53956-6> (date of access: 14.11.2024).
6. Tsourdos A., White B., Shanmugavel M. Cooperative Path Planning of Unmanned Aerial Vehicles. Chichester, UK : John Wiley & Sons, Ltd, 2010. URL: <https://doi.org/10.1002/9780470974636> (date of access: 14.11.2024).
7. Parsons S. Principles of Robot Motion: Theory, Algorithms and Implementations by Howie Choset, Kevin M. Lynch. *The Knowledge Engineering Review*. 2007. Vol. 22, no. 2. P. 209–211. URL: <https://doi.org/10.1017/s0269888907218016> (date of access: 14.11.2024).