

Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system

Mai Đức Thịnh, Lê Bá Minh Phúc

Tháng 10, 2025

Động cơ nghiên cứu

- ▶ Sự phát triển mạnh mẽ của UAV và trí tuệ nhân tạo đã mở rộng ứng dụng trong nhiều lĩnh vực như giám sát, cứu hộ, nông nghiệp, quân sự,...
- ▶ Một UAV đơn lẻ bị hạn chế về năng lượng, khả năng tính toán và phạm vi hoạt động \Rightarrow cần phối hợp nhiều UAV.
- ▶ Hệ thống UAV **không đồng nhất** (về tốc độ, tầm bay, cảm biến, tải trọng,...) mang lại hiệu quả cao hơn nhưng làm bài toán lập kế hoạch đường bay trở nên phức tạp.
- ▶ Các nghiên cứu trước chủ yếu tập trung vào UAV đồng nhất hoặc một UAV đơn lẻ, chưa xét đến bài toán các UAV không đồng nhất

Đầu vào bài toán

Các tham số đầu vào của mô hình, mô tả đặc tính của đội UAV, môi trường hoạt động và nhiệm vụ:

- ▶ UAV $U = \{U_1, U_2, \dots, U_n\}$: mỗi UAV U_i có các đặc tính $\langle V_i^{max}, H_i, W_i \rangle$
- ▶ Khu vực quét $R = \{R_1, R_2, \dots, R_m\}$ với diện tích cần quét A_i
- ▶ Khoảng cách giữa các khu vực: ma trận $D = \{D_{i,j}\}$
- ▶ Vận tốc quét cho phép: ma trận $V = \{V_{i,j}\}$

Tham số phụ

- ▶ Thời gian quét:

$$TS_{i,j} = \frac{A_j}{V_{i,j} \times W_i}$$

nếu $V_{i,j} \neq 0$, ngược lại là vô cùng.

- ▶ Thời gian bay:

$$TF_{i,j,k} = \frac{D_{j,k}}{V_i^{max}}$$

Đầu ra bài toán

Đầu ra là ma trận nhị phân $X = \{x_{i,j,k}\}$:

- ▶ $x_{i,j,k} = 1$ nếu UAV U_i bay từ khu vực R_j đến R_k
- ▶ $x_{i,j,k} = 0$ nếu ngược lại

với $1 \leq i \leq n$, $0 \leq j \leq m$, $1 \leq k \leq m$

Ràng buộc bài toán

- ▶ (C1): Mỗi UAV chỉ cất/hạ cánh một lần.

$$\forall i, \sum_{j=1}^m x_{i,0,j} \leq 1, \quad \sum_{j=0}^m x_{i,j,0} \leq 1$$

- ▶ (C2): Tổng số UAV hoạt động không vượt quá số UAV sẵn có.

$$\sum_{i=1}^n \sum_{j=1}^m x_{i,0,j} \leq n$$

Ràng buộc bài toán

- ▶ (C3): UAV được chọn phải quay trở lại căn cứ.

$$\text{Nếu } \sum_{j=1}^m x_{i,0,j} = 1 \Rightarrow \sum_{j=0}^m x_{i,j,0} = 1$$

- ▶ (C4): Mỗi khu vực chỉ được ghé thăm đúng một lần.

$$\forall j, \sum_{i=1}^n \sum_{k=1}^m x_{i,j,k} = 1, \sum_{i=1}^n \sum_{k=0}^m x_{i,k,j} = 1$$

Hàm mục tiêu của bài toán

Tổng chi phí thời gian cho một UAV riêng lẻ $F(U_i)$. Đại lượng này tính tổng thời gian bay và thời gian quét cho tất cả các chặng bay được gán cho UAV U_i

$$F(U_i) = \sum_{j=0}^m \sum_{k=1}^m x_{i,j,k} (TS_{i,k} + TF_{i,j,k})$$

Mục tiêu tổng thể của bài toán là tối thiểu hóa thời gian hoàn thành của UAV cuối cùng theo công thức

$$\min \max_{1 \leq i \leq n} F(U_i)$$

Thuật toán: Phân vùng cho các UAV

Algorithm 1: The pseudo-code of the region allocation phase.

Input: region set R and UAV set U

Output: regions assigned to each UAV

```
1  $LeftRegion \leftarrow R$ ;  
2  $\forall i \in [1, n], Region[i] \leftarrow \emptyset, FinishTime[i] \leftarrow 0$ ;  
3 repeat  
4    $i \leftarrow$  index of the UAV which would complete the coverage task  
   at the earliest time;  
5    $j \leftarrow$  index of the last region that would be visited by  $U_i$ ;  
6   Calculate effective time ratios of regions in  $LeftRegion$ ;  
7    $k \leftarrow$  index of the region that has the largest effective time  
   ratio;  
8   Remove  $R_k$  from  $LeftRegion$ ;  
9   Add  $R_k$  into  $Region[i]$ ;  
10   $FinishTime[i] += TS_{i,k} + TF_{i,j,k}$ ;  
11 until  $LeftRegion \neq \emptyset$ ;
```

Thuật toán: Tối ưu lộ trình cho từng UAV

Algorithm 2: Pseudo-code of the order optimization phase.

Input: number of ants N , evolution algebra of populations G

Output: optimized coverage sequence for each UAV

```
1 for  $r = 1$  to  $n$  do
2    $Region[r] \leftarrow$  the set of regions assigned to  $U_r$ ;
3   Place  $M$  ants in the regions belonging to  $Region[r]$  randomly;
4    $L \leftarrow$  tour length obtained from the nearest neighbour
    approach;
5    $\forall i, j \in [0, m], \tau(i, j) \leftarrow \frac{1}{m \times L}, \eta(i, j) \leftarrow \frac{1}{D_{i,j}}$ ;
6   for  $t = 1$  to  $G$  do
7     for  $a = 1$  to  $N$  do
8        $R_i \leftarrow$  the current region in which the ant  $a$  is placed;
9        $\Pi_a \leftarrow Region[r] \setminus \{R_i\}, Order[a] \leftarrow \{R_i\}$ ;
10      repeat
11         $R_i \leftarrow$  the current region in which  $a$  is placed;
12         $q \leftarrow$  a randomly generated number ranging in  $[0, 1]$ ;
13         $R_k \leftarrow$  the next region selected according to Eq. (6);
14        Move the ant  $a$  from  $R_i$  to  $R_k$ ;
15        Put  $R_k$  into  $Order[a]$ ;
16        Remove  $R_k$  from  $\Pi_a$ ;
17        Update the local pheromone according to Eq. (10);
18      until  $\Pi_a = \emptyset$ ;
19    end
20    Select the best ant that finds the shortest path in current
    solutions;
21    Update the global pheromone according to Eq. (11);
22  end
23 end
```

Thuật toán: Tối ưu lộ trình cho từng UAV

$$ETR_{i,j,k} = \frac{TS_{i,k}}{TF_{i,j,k} + TS_{i,k}} = \frac{A_k V_i^{max}}{A_k V_i^{max} + v_{i,k} W_i D_{j,k}} \quad (1)$$

$$R_k = \begin{cases} \arg \max_{R_j \in \Pi_a} \tau(i,j)^\alpha \eta(i,j)^\beta, & \text{if } q \leq q^0, \\ \text{Roulette Wheel Selection,} & \text{otherwise.} \end{cases} \quad (2)$$

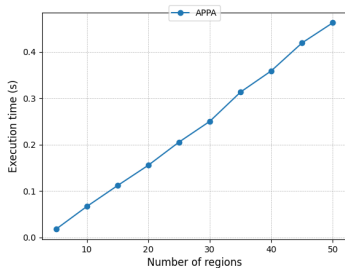
$$p(i,j) = \begin{cases} \frac{\tau(i,j)^\alpha \eta(i,j)^\beta}{\sum_{v_k \in \Pi_a} \tau(i,k)^\alpha \eta(i,k)^\beta}, & \text{if } R_j \in \Pi_a, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$$\tau(i,j) = (1 - \rho)\tau(i,j) + \rho\tau_0 \quad (4)$$

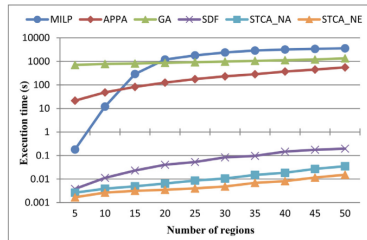
$$\tau(i,j) = (1 - \varepsilon)\tau(i,j) + \varepsilon\Delta\tau(i,j) \quad (5)$$

$$\text{where } \Delta\tau(i,j) = \begin{cases} \frac{1}{L_{gb}}, & \text{if } (R_i, R_j) \in P_{gb}, \\ 0, & \text{otherwise.} \end{cases}$$

Thực nghiệm



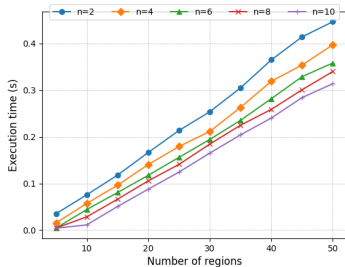
Hình: Chạy thử nghiệm



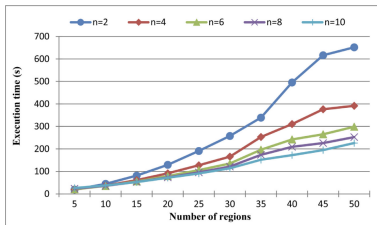
Hình: Bài báo

Hình: Thời gian chạy thuật toán với $u = 0.05$, $d = 0.9$

Thực nghiệm



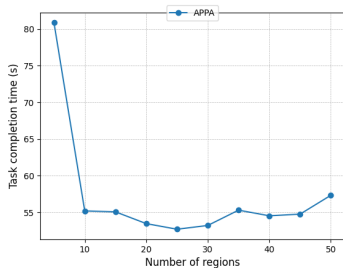
Hình: Chạy thử nghiệm



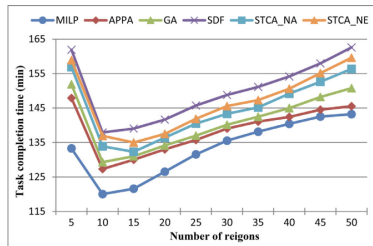
Hình: Bài báo

Hình: Thời gian chạy thuật toán với $u = 0.05$, $d = 0.9$, số UAV từ 2 -> 10

Thực nghiệm



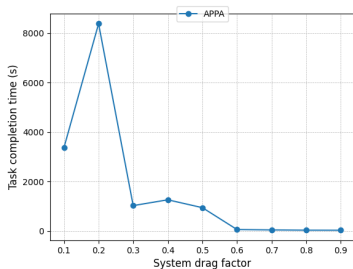
Hình: Chạy thử nghiệm



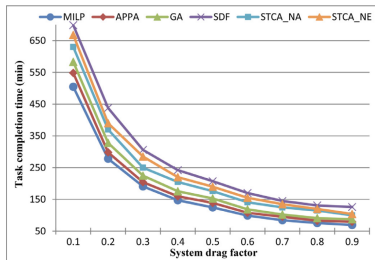
Hình: Bài báo

Hình: Thời gian hoàn thành nhiệm vụ của UAV với $u = 0.02$, $d = 0.9$

Thực nghiệm



Hình: Chạy thử nghiệm



Hình: Bài báo

Hình: Thời gian hoàn thành nhiệm vụ của UAV với số lượng khu vực là 15 và $u=0.01$

Thuật toán Alternating Refinement (1/3): Khởi tạo

Algorithm 1: ALTERNATING_REFINEMENT
($uavs, regions, V, base, max_iters, eps_rel, local_gain$)

Input : Danh sách UAVs, vùng quét, ma trận vận tốc V , căn cứ $base$, số lặp tối đa max_iters , ngưỡng hội tụ eps_rel , hệ số cải thiện cục bộ $local_gain$.

Output: Bộ phân công tốt nhất, thời gian hoàn thành và makespan nhỏ nhất.

```
1  $assignment \leftarrow \text{REGION\_ALLOCATION\_PHASE1}(uavs, regions, V, base)$ 
2 foreach  $u \in uavs$  do
3    $assignment[u] \leftarrow$   
    $\text{OPTIMIZE\_ORDER\_FOR\_UAV}(u, assignment[u], base, V)$ 
4  $ft \leftarrow \text{COMPUTE\_FINISH\_TIMES}(uavs, assignment, V, base)$ 
5  $best\_ms \leftarrow \text{MAX\_VALUE}(ft)$ 
6  $best\_assignment \leftarrow \text{COPY}(assignment)$ 
7  $best\_ft \leftarrow \text{COPY}(ft)$ 
```

Thuật toán Alternating Refinement (2/3): Vòng lặp chính

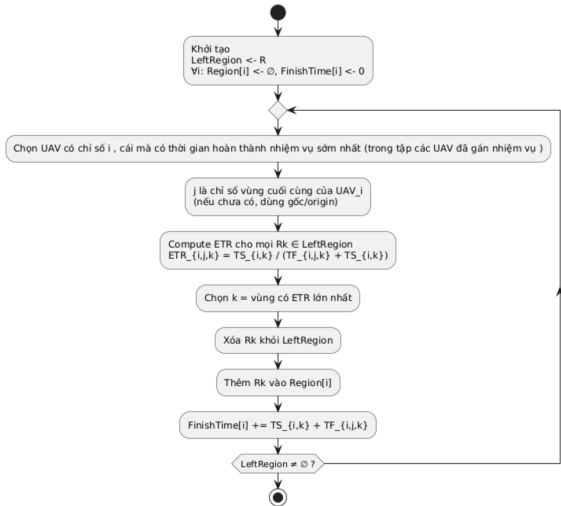
```
1 repeat
2   TRY_LOCAL_TRANSFER(uavs, assignment, V, base, local_gain)
   // có thể chuyển một vùng từ UAV chậm nhất nếu giảm makespan
3   foreach u ∈ uavs do
4     new_assignment[u] ←
      OPTIMIZE_ORDER_FOR_UAV(u, new_assignment[u], base, V)
5   new_ft ←
      COMPUTE_FINISH_TIMES(uavs, new_assignment, V, base)
6   ms ← MAX_VALUE(new_ft)
7   improvement ← best_ms − ms
8 until max_iters
```

Thuật toán Alternating Refinement (3/3): Cập nhật và dừng

```
1 if improvement < 0 then
2   | break                                // tái phân vùng làm xấu hơn, giữ kết quả cũ
3 assignment  $\leftarrow$  new_assignment
4 ft  $\leftarrow$  new_ft
5 if improvement > threshold then
6   | best_assignment  $\leftarrow$  COPY(assignment)
7   | best_ft  $\leftarrow$  COPY(ft)
8   | best_ms  $\leftarrow$  ms
9 else
10  | break                                // cải thiện quá nhỏ để tiếp tục
11 return (best_assignment, best_ft, best_ms)
```

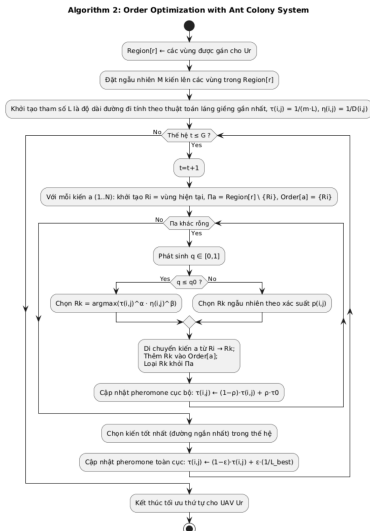
Sơ đồ thuật toán

Algorithm 1: Region Allocation Phase



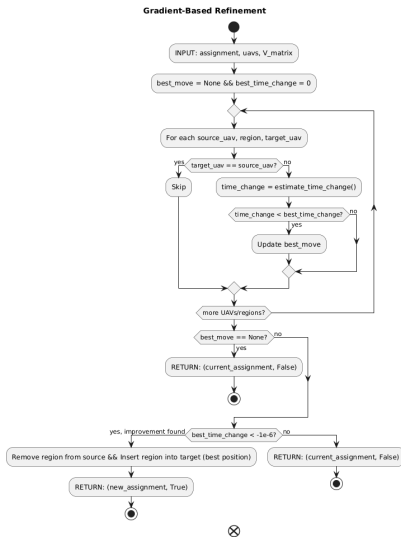
Hình: Thuật toán region allocation (phase 1)

Sơ đồ thuật toán



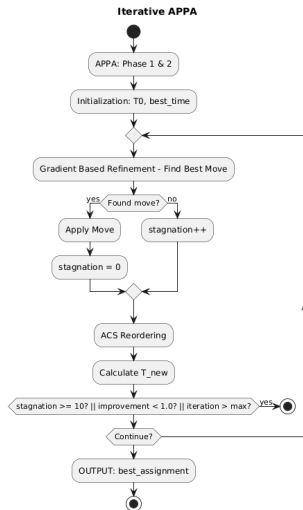
Hình: Thuật toán order optimization (phase 2)

Sơ đồ thuật toán



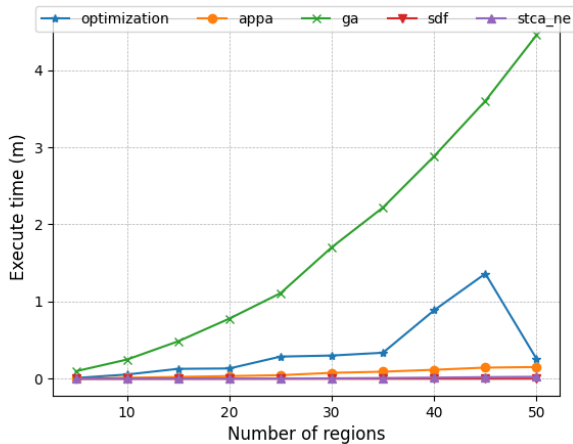
Hình: Thuật toán tối ưu lại phân vùng

Sơ đồ thuật toán



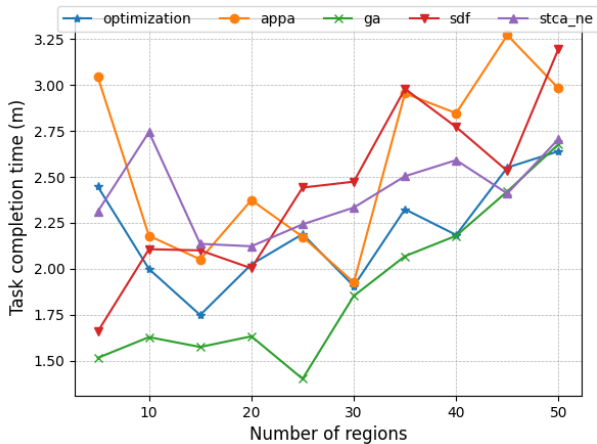
Hình: Tổng quan thuật toán tối ưu lại APPA

So sánh trên nhiều thuật toán



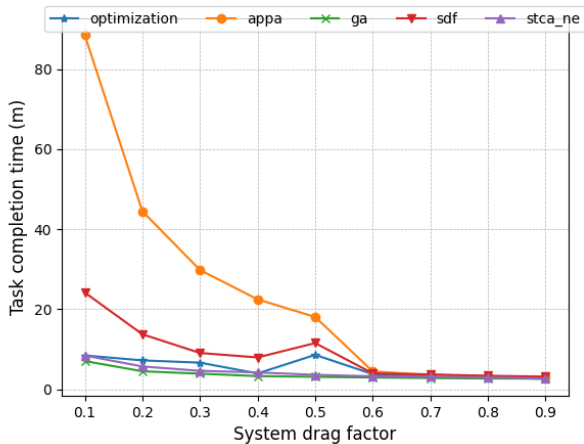
Hình: So sánh thời gian thực thi giữa các thuật toán

So sánh trên nhiều thuật toán



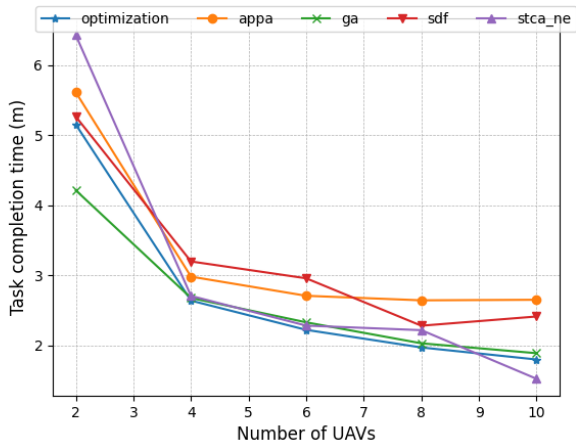
Hình: So sánh thời gian hoàn thành nhiệm vụ giữa các thuật toán

So sánh trên nhiều thuật toán



Hình: So sánh thời gian hoàn thành nhiệm vụ giữa các thuật toán

So sánh trên nhiều thuật toán

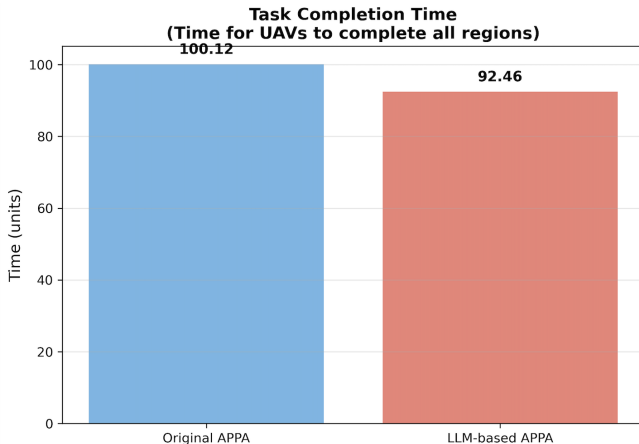


Hình: So sánh thời gian hoàn thành nhiệm vụ giữa các thuật toán

Thử nghiệm với LLM trong phase 1

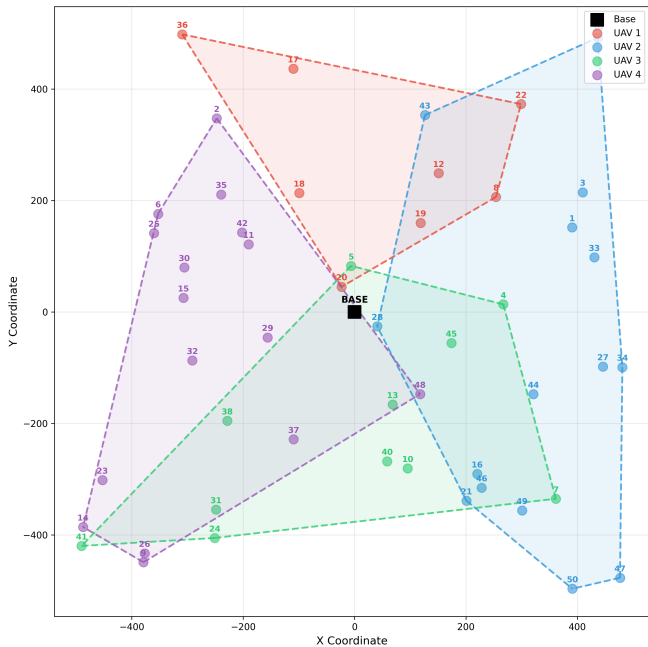
- ▶ Pha 1 sử dụng LLM để phân vùng
- ▶ Pha 2 giữ nguyên thuật toán ACO gốc

Thử nghiệm với LLM trong phase 1

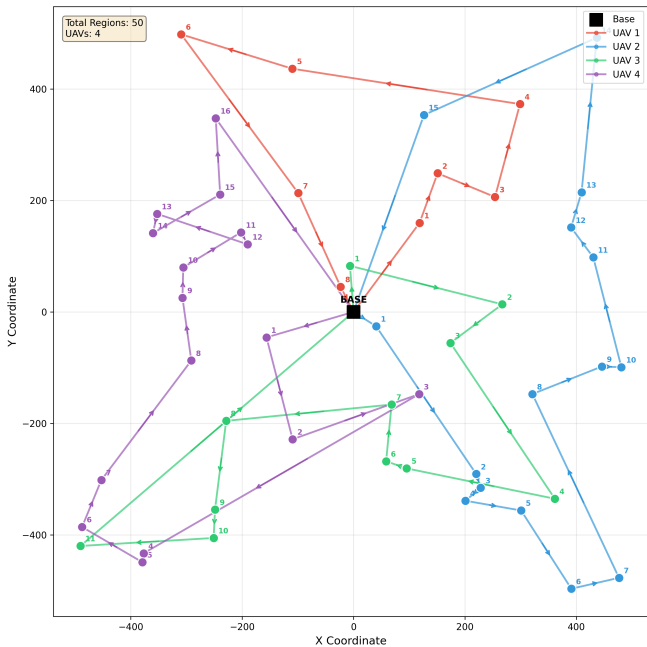


Hình: So sánh thời gian hoàn thành nhiệm vụ giữa các thuật toán

Visualize APPA



Visualize APPA



Metrics

Comprehensive Metrics Comparison Dashboard

