IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Hierarchical Planning for Vehicle Routing and Scheduling in Marsupial Robotic Systems

Donghyun Kim[1] and Jinwhan Kim[1]

***Abstract*—This letter presents a hierarchical planning approach to the vehicle routing and scheduling problem (VRSP) for marsupial robotic systems, a specialized class of heterogeneous robotic systems in which one type of mobile robot is capable of carrying another. While traditional VRSPs have been widely studied, the marsupial variant (MVRSP) has received relatively little attention. To address the NP-hard nature of MVRSP, this work introduces a hierarchical planning structure that decomposes the problem into two subproblems with reduced complexity: a high-level routing problem, formulated as a mixed-integer linear program (MILP), and a low-level scheduling problem, modeled in the Planning Domain Definition Language (PDDL). These subproblem solutions are integrated to generate complete mission plans. The proposed approach is validated through qualitative plan visualizations and quantitative Monte Carlo simulations in an autonomous subsea mapping scenario, where an unmanned surface vehicle carries multiple underwater vehicles. Results show that the hierarchical planner significantly improves both planning efficiency and solution quality compared to baseline methods.**

***Index Terms*—Planning, Scheduling and Coordination, Multi-Robot Systems, Cooperating Robots.**

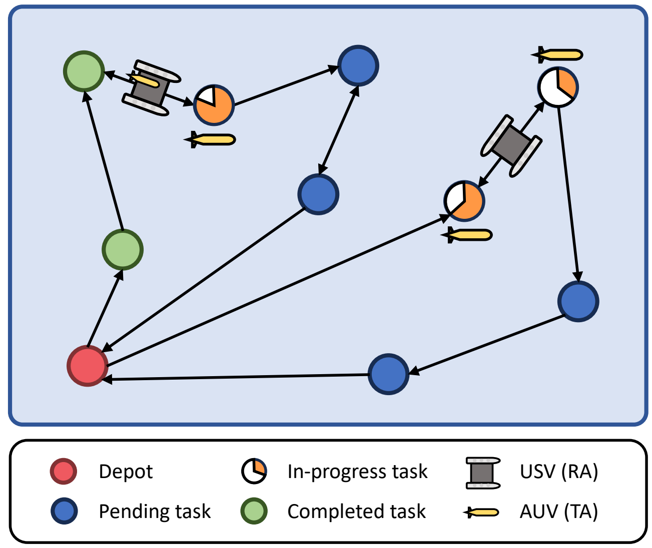


Fig. 1. Sample scenario in which two fleets of marsupial systems, each consisting of a routing agent (RA) and task agents (TAs), conduct a mission involving spatially distributed tasks.

I. Introduction

INTEREST in heterogeneous robotic systems is growing, as complementary capabilities enable more complex missions and more efficient task execution. Among these, marsupial systems, inspired by mammals such as kangaroos that carry their young in a pouch, pair multiple small robots with a larger carrier from which they can be launched and later recovered, thereby enhancing the system's overall capabilities. First demonstrated with a UGV and a small ground explorer [1], the paradigm has since expanded to UGV–UAV [2], USV–UAV [3], and USV–AUV [4] configurations. Marsupial systems have enabled efficient mission execution across diverse domains, including environmental monitoring [5], search and rescue [6], and reconnaissance [7].

This letter addresses a mission, shown in Fig. 1, involving spatially distributed tasks over a large area and focuses on the time-efficient operation of a specialized marsupial system. The system is composed of a routing agent (RA) and task agents (TAs), each specialized for distinct aspects of the mission. The RA is responsible for transporting and retrieving the task agents, while the TAs perform assigned tasks at designated locations. We refer to the resulting mission planning problem as the Marsupial Vehicle Routing and Scheduling Problem (MVRSP), whose objective is to minimize the overall completion time. According to [8], MVRSP belongs to the class of ST-SR-TA-XD (single-task robots, single-robot tasks, time-extended assignment, cross-schedule dependencies). Problems in this class are NP-hard, making it infeasible to solve them entirely except for trivial cases.

To effectively address this computational complexity, this letter presents a hierarchical mission planning approach. The mission, consisting of a set of tasks, is first partitioned into multiple task groups. To partition effectively, two clustering-based heuristic methods are proposed. These methods generate task groupings in a computationally efficient manner, enabling the original problem to be decomposed into two subproblems of relatively lower complexity. The high-level problem is a routing subproblem that determines which fleet is assigned to each task group and in what order, while the low-level problem is a scheduling subproblem that decides which task agent performs each task and when. To solve these, the routing subproblem is formulated as a Mixed-Integer Linear Program (MILP), and the scheduling subproblem is modeled using the

Manuscript received: June 18, 2025; Revised September 19, 2025; Accepted October 26, 2025.

This paper was recommended for publication by Editor Chao-Bo Yan upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by National R&D Program through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT (2020M3C1C1A02086308). *(Corresponding author: Jinwhan Kim.)*

[1]Donghyun Kim and Jinwhan Kim are with the Department of Mechanical Engineering, KAIST (Korea Advanced Institute of Science and Technology), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea {rlaehdgusqwe, jinwhan}@kaist.ac.kr

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Planning Domain Definition Language (PDDL). Finally, these solutions are integrated within a hierarchical framework to generate the mission plan. The proposed hierarchical planner is evaluated through both qualitative analysis using plan visualizations and quantitative analysis based on Monte Carlo simulations.

II. Related work

The MVRSP is closely related to the Vehicle Routing Problem (VRP), a generalization of the Traveling Salesman Problem that incorporates practical constraints such as capacities, time windows, and heterogeneous fleets. VRP variants are typically solved by exact methods (e.g., mixed integer programming, column generation, and Dantzig–Wolfe decomposition) for small-scale instances, while metaheuristics (e.g., tabu search, large-neighborhood search, genetic algorithms) are used for larger problems, with numerous variants and hybrid extensions proposed [9].

Unlike most VRP variants, MVRSP must capture concurrency, multiple visiting, and launch–recovery constraints that tightly couple RA and TAs. Similar challenges have been surveyed in the multi-robot planning and scheduling literature [10], where decomposition and coordination are emphasized for handling task allocation, temporal constraints, and inter-agent dependencies. To model such couplings, an action-based formulation is required, and the Planning Domain Definition Language (PDDL) provides such a model. PDDL 2.1 [11] introduced temporal and numeric features for time and resource constraints, and recent work has extended PDDL to multi-robot missions [12], [13]. Relevant applications of the MVRSP include an undersea survey mission with a USV and multiple AUVs [14], [15]. In particular, [14] modeled the mission in PDDL and incorporated a TSP-based constraint to guide the planner, which improved planning performance.

However, PDDL-based planning also suffers from scalability limitations. To address this, decomposition strategies have been proposed, such as bi-level planning for UAV–UGV coordination [16] and hierarchical decomposition for heterogeneous multi-robot routing [17]. These approaches improve scalability by decoupling allocation/routing or inter-agent interactions, but are difficult to generalize to MVRSP, where RAs and TAs are tightly coupled. Meanwhile, clustering-based decomposition planning has been widely used as a representative strategy for addressing large-scale problems [18]–[20], in which tasks are first grouped into clusters and then detailed plans are derived within each cluster to effectively reduce computational complexity. Building on this idea, our approach integrates clustering-based decomposition with a PDDL symbolic planner, thereby improving scalability while preserving coupling constraints.

As an alternative paradigm, multi-agent reinforcement learning (MARL) has recently gained attention for its ability to learn adaptive and decentralized coordination policies through interaction [21]–[24]. In contrast, our symbolic, decomposition-based planner achieves coordination by explicitly reasoning over coupling and scheduling constraints in MVRSP, offering interpretability and formal reliability that data-driven policies often lack.

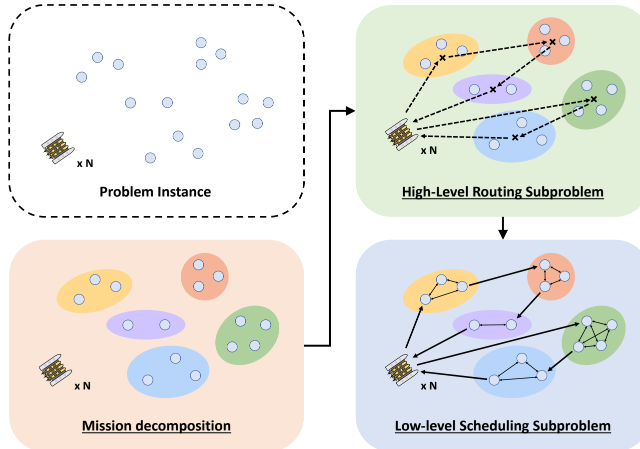


Fig. 2. The hierarchical planning process. Mission partitioning is employed to divide original problem into two hierarchical subproblems: a high-level routing subproblem and a low-level scheduling subproblem.

III. Problem description

We consider mission scenarios such as the undersea survey mission illustrated in Fig. 1, where spatially distributed tasks must be completed using multiple marsupial robotic fleets. Each fleet consists of a single resource agent (RA) and several task agents (TAs). The RA transports, deploys, and retrieves its TAs until all assigned tasks are completed, with multiple TAs operating concurrently within a fleet. At the end of the mission, each RA must recover all TAs and return to the starting location. Fleets operate independently, and all interactions are restricted to agents within the same fleet. The MVRSP is defined as the problem of generating routing and scheduling plans for multiple marsupial fleets to complete all tasks while minimizing the mission makespan.

To formalize the MVRSP, we adopt several modeling assumptions. All task agents are assumed to be homogeneous in capability and performance, and coordination is managed by a centralized system without explicitly modeling communication delays. In addition, energy constraints such as battery capacity are not explicitly modeled, under the assumption that agents have sufficient resources to complete their assigned tasks. These assumptions allow us to concentrate on the core scheduling aspects of the problem. While the undersea survey mission serves as an illustrative case, the formulation generalizes to other marsupial systems provided that: (i) tasks are spatially distributed, (ii) RAs specialize in routing and transportation while TAs are specialized for tasks that RAs cannot perform, (iii) TAs require retrieval due to endurance or routing limitations, and (iv) coordination is centralized with sufficiently reliable communication. These conditions are satisfied in domains such as UAV–UGV inspection, agricultural rovers deploying drones for crop monitoring and spraying, and disaster response with ground–aerial robot teams, illustrating that the proposed formulation extends beyond the marine domain to diverse mission contexts.

IV. Hierarchical planning

In this chapter, we present a hierarchical mission planning method based on a divide-and-conquer approach to over-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

come the optimization challenges of MVRSP. The first step is mission partitioning, and here we propose two efficient mission partitioning methods. Once the mission is partitioned, the original problem can be divided into two subproblems with relatively lower complexity. The high-level problem is a routing subproblem, while the low-level problem is a scheduling problem. Ultimately, partial plans generated from the scheduling problem are sequentially connected according to the routing order obtained from the routing problem, forming a comprehensive plan. The overall flowchart of the hierarchical mission planning is shown in Fig.2.

A. Mission Partioning

Mission $\mathcal{M}$ is defined as a set of spatially distributed tasks. A mission partitioning $\mathcal{P}$ can be defined as a partition of $\mathcal{M}$, which is a collection of task groups—subsets of $\mathcal{M}$—with the following three properties [25]:

$$A \neq \emptyset \qquad \forall A \in \mathcal{P} \tag{1}$$

$$A \cap B = \emptyset \qquad \forall A, B \in \mathcal{P},\ A \neq B \tag{2}$$

$$\bigcup_{A \in \mathcal{P}} A = \mathcal{M}. \tag{3}$$

In simple terms, a mission partitioning of M is a collection of nonempty, mutually exclusive, collectively exhaustive subsets of M. However, mission partitioning is not simply a partition problem; it is intricately linked with both the routing and scheduling subproblems. If we were able to know the optimal solution of each subproblem for all possible partition combinations in advance, we could first identify the optimal partition combination and then generate the actual plans for each subproblem to determine the optimal plan. Unfortunately, both routing and scheduling subproblems are NP-hard, which means we cannot know the optimal solution of the subproblems beforehand. Therefore, it is necessary to develop an appropriate mission partitioning algorithm that considers the implicit potential utility of partial plans, and we propose the following two methods.

*1) **Hierarchical clustering**:* The first partitioning method is a distance-based approach that employs hierarchical clustering [26]. This method iteratively merges nearby clusters to form a hierarchy, which can be visualized using a dendrogram, as illustrated in Fig. 3. Whereas algorithms such as k-means need the number of clusters to be specified upfront, hierarchical clustering builds a nested cluster structure that can be interpreted afterward using a dendrogram.

In hierarchical clustering, a horizontal cut, referred to as the cut-off distance, is applied to the dendrogram to define clusters. Nodes located below intersected branches are grouped together. Consequently, the clustering outcome depends on how this cut-off distance is chosen. A common method for determining the cut-off distance is to use a clustering index, which quantitatively evaluates the quality of clustering. The cut-off distance is selected where the index value is maximized. Among various clustering indices, the Dunn index and the Silhouette index are most commonly used. These indices measure intra-cluster cohesion and inter-cluster separation

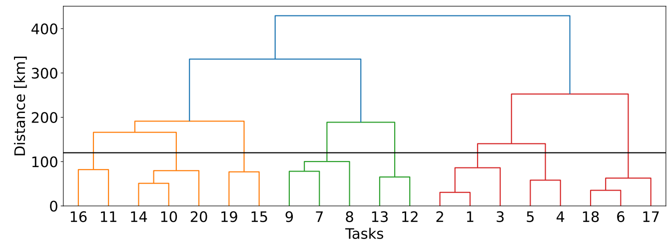
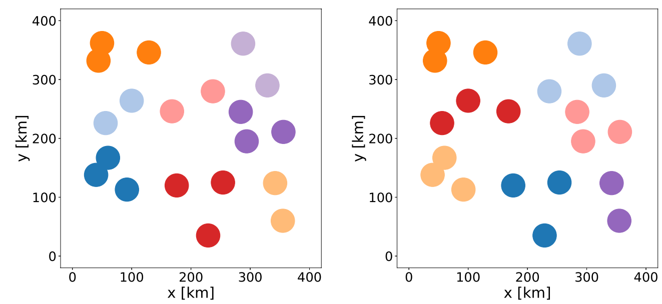


Fig. 3. Dendrogram representing the result of hierarchical clustering. The cut-off distance is shown as a black horizontal line. Nodes below each intersected branch are grouped into the same cluster.



(a) (b)

Fig. 4. Clustering results: (a) Hierarchical, (b) Balanced.

based on distance. However, in the context of our scheduling subproblem, clustering quality is also affected by the number of tasks within each task group. To reflect this, we propose a new clustering index, called the homogeneity index, which quantifies workload balance by comparing the number of tasks in each group to the number of TAs.

$$HI = exp\left(-\frac{1}{m}\sum_{k=1}^{m}\left(|C_k| - N_A\right)^2\right) \tag{4}$$

TABLE I
THE PARAMETERS OF HOMOGENEITY INDEX.

Parameter	Description
m	Number of clusters
C_k	Set of kth cluster
N_A	Number of task agents

The homogeneity index, formally defined in Equation 4, applies an exponential decay function to the mean square error between the number of tasks in each group and the number of TAs. The parameters used in this formulation are summarized in Table I. A lower MSE implies a better match and yields a higher homogeneity index, effectively capturing potential utility. By combining this index with hierarchical clustering, the proposed method enables mission partitioning that considers both spatial distribution and workload balance. An example of the resulting partition is shown in Fig. 4a.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

*2) **Balanced clustering**:* The second method, referred to as balanced clustering, formulates a clustering problem as an Integer Linear Programming (ILP) model that simultaneously considers the distance between tasks and the number of tasks within each task group. This formulation is motivated by the observation that the utility of a subproblem tends to increase when the number of tasks in a cluster closely matches the number of available task agents. The objective is to minimize the total intra-cluster distance. The corresponding ILP formulation is defined below, with the notation summarized in Table II:

$$\min \quad \sum_{k\in M}\sum_{i\in N}\sum_{j\in N} d_{ij}x_{ik}x_{jk} \tag{5}$$

$$\text{s.t.} \quad \sum_{i\in N} x_{ik} = c_k, \qquad \forall k \in M \tag{6}$$

$$\sum_{k\in M} x_{ik} = 1, \qquad \forall i \in N \tag{7}$$

$$x_{ik} \in \{0,1\}, \qquad \forall i \in N, \forall k \in M \tag{8}$$

$$c_k = \begin{cases} a & \text{if } 1 \le k \le \lfloor \frac{n}{a} \rfloor \\ n - a \cdot \lfloor \frac{n}{a} \rfloor & \text{otherwise.} \end{cases} \tag{9}$$

TABLE II
NOTATION USED IN THE ILP FORMULATION.

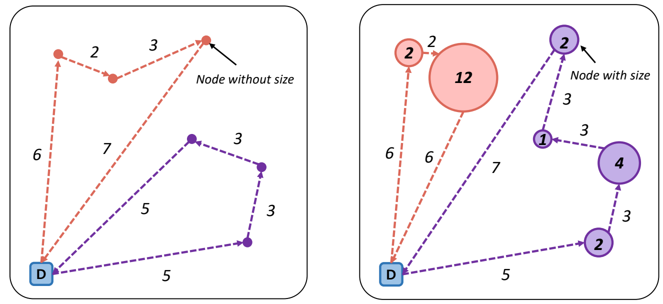
Decision variable	Description
$x_{ik} \in \{0,1\}$	$x_{ik} = 1$ if the task $i \in N$ is assigned to cluster $k \in M$
Parameter	**Description**
n	The number of tasks
a	The desired number of tasks in each cluster
N	Set of tasks; $\{1, 2, \cdots, n\}$
M	Set of clusters; $\{1, 2, \cdots, \lceil \frac{n}{a} \rceil\}$
c_k	The number of tasks in cluster $k \in M$
d_{ij}	Distance from task $i \in N$ to task $j \in N$

The clustering result obtained from this ILP model is illustrated in Fig. 4b. The result shows that the number of tasks in each cluster closely matches the number of task agents, indicating that the proposed method effectively balances task distribution.

B. Routing subproblem

Once the mission is partitioned into task groups, the next step is to determine which fleet will visit which task group and in what order. This problem can be viewed as a VRP. VRP is commonly defined using either a min-sum or a min-max objective. In this study, we aim to minimize the time required for each fleet to complete its mission in parallel. Thus, we adopt the min-max approach [27].

To formulate this as a VRP, we define the nodes as the centroids of the tasks within each task group. Next, we must establish the cost between these nodes. In most VRP formulations, the cost matrix is based on the Euclidean distance between nodes to minimize travel distance or time. However, since our goal is to minimize the overall mission completion time, both travel time and task execution time must be considered.



(a) (b)

Fig. 5. Comparison of cost formulations. (a) Euclidean cost results in similar travel costs (18 vs. 16) but large imbalances when task execution is considered (31 vs. 21). (b) Node-weighted cost achieves a balanced plan (28 vs. 30) despite longer routes, effectively mitigating bias.

To address this, we propose a node-weighted cost that incorporates not only the Euclidean distance but also the size of each node, i.e., the total workload in each task group. The proposed cost is defined as follows:

$$c_{ij} = d_{ij} + \frac{v_r}{v_t} \sum_{k\in C_j} W_k, \tag{10}$$

where d_{ij} is the Euclidean distance, v_r and v_t denote RA travel speed and TA task speed, respectively, and $\sum_{k\in C_j} W_k$ represents the workload of task group C_j.

Fig. 5 compares different cost formulations. Using only Euclidean distance yields nearly identical travel costs but ignores execution time, leading to unbalanced fleet workloads. In contrast, the proposed node-weighted cost balances both travel and execution, producing more equitable plans and mitigating bias.

C. Scheduling subproblem

The last step of the hierarchical planning process is to generate detailed execution schedules for the task agents within each task group. This scheduling subproblem focuses on deciding which task agent performs which task and when. The objective is to coordinate concurrent actions within each fleet while minimizing the overall mission completion time. Addressing this subproblem requires a formal and expressive problem description.

As PDDL 2.1 is well-suited for expressing time-constrained planning problems, we adopt it to formally define this problem. PDDL describes planning problems through domain and problem files: the domain file specifies components such as types, predicates, functions, and actions, while the problem file defines the objects, initial state, and goal conditions. In our model, the domain includes core predicates for task states and agent availability, numeric functions for distance, task size, and agent speeds, and four main durative actions: `move`

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

(RA relocation), `launch` (TA deployment), `execute` (task execution by a TA), and `retrieve` (TA recovery). These key elements are summarized in Appendix A. With this formulation, a PDDL planner can then generate executable plans. Representative temporal planners compatible with PDDL 2.1 include POPF [28], COLIN [29], and OPTIC [30].

V. Simulation Results

We apply the proposed hierarchical planning framework to the undersea survey mission introduced in Fig. 1, and evaluate the performance of each method through qualitative and quantitative analysis. For the qualitative analysis of the mission planning results, we visualize the USV's route to assess routing optimization and represent the plan using a Gantt chart to evaluate how effectively the agents within each fleet are temporally coordinated. For the quantitative analysis, we perform Monte Carlo simulations by generating 100 different scenarios for each number of tasks, where the tasks are randomly distributed across the mission area.

For all scenarios, the size of the mission area is 400x400 km², and the tasks are randomly distributed within this area. Each task involves surveying undersea regions using an AUV. All regions are assumed to be of equal size to ensure fair and consistent evaluation across scenarios. A marsupial fleet consists of one USV and three AUVs. The USV operates at an average speed of 8 knots, each AUV survey takes 20 hours, and deploying or retrieving an AUV takes 1 hour.

We first verify the suitability of the proposed homogeneity index for the hierarchical clustering approach to mission partitioning by comparing it to existing clustering indices, as discussed in Section V-A. Next, the effectiveness of the hierarchical planning framework with mission partitioning are evaluated under a single-fleet configuration, which is detailed in Section V-B. Finally, Section V-C presents an evaluation in a multi-fleet setting to assess the impact of the node-weighted cost, used in the routing subproblem, on the overall improvement of mission planning performance.

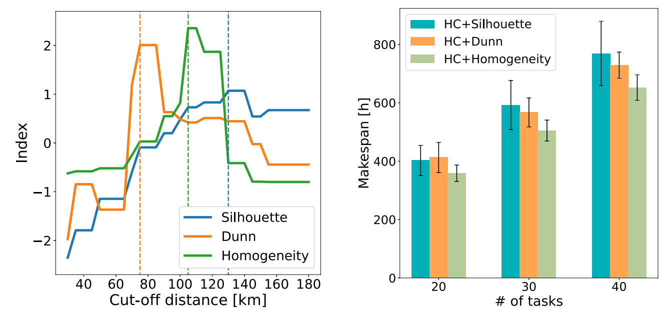
TABLE III
The planning time limits.

	Time limit [s]
Total planning time	5*(# of tasks)
Partitioning	0.5*(# of tasks)
Routing	0.5*(# of tasks)
Scheduling	remaining time

For all tests, OPTIC [30] is used as the PDDL planner, and CPLEX [31] with default parameters is used as the MILP solver. The time limits for each component of the planning process are summarized in Table III. For clarity, the methods evaluated in our experiments are summarized in Table IV. All experiments are conducted on a computer equipped with an Intel i7-12700KF CPU, an NVIDIA GeForce RTX 3070 GPU, and 32GB of RAM.

TABLE IV
Summary of methods used in the experiments.

Method	Description
PDDL-only	Mission planning solved solely using PDDL
PDDL-TSP [14]	PDDL with a TSP-based constraint
HP-HC	Hierarchical planning with hierarchical clustering
HP-BC	Hierarchical planning with balanced clustering



(a) (b)

Fig. 6. Clustering index analysis. (a) Index values versus cut-off distance in a 20-task scenario, with optimal cut-off distances identified for each index. (b) Resulting hierarchical planning outcomes guided by each index, showing that the proposed homogeneity index achieves superior performance.

A. Verification of the homogeneity index

A clustering index serves as a quantitative metric to evaluate clustering quality. That is, the higher the value of the clustering index $I(d)$, which is a function of the cut-off distance d, the better the resulting clustering. Accordingly, the optimal cut-off distance can be determined as:

$$d_{opt} = \arg\max_{d} I(d). \tag{11}$$

To illustrate this, Fig.6a presents the index values for each cut-off distance in a representative scenario with 20 tasks. The optimal cut-off distances are 130 km for the silhouette index, 75 km for the Dunn index, and 105 km for the proposed homogeneity index. Fig. 6b presents the hierarchical planning results following mission partitioning via hierarchical clustering guided by each index, evaluated through a Monte Carlo simulation. Among the three, the homogeneity index yields the best overall performance. These results suggest that the proposed homogeneity index is well-suited for guiding hierarchical clustering in the context of mission partitioning.

B. Evaluation in a single-fleet setting

To evaluate the effectiveness of the proposed hierarchical planning framework, we conduct experiments in a single-fleet setting. Figure 7 and 8 show the planning results for a scenario with 20 tasks. The PDDL-only method shows poor optimization in both routing and temporal coordination, while PDDL-TSP improves routing but fails to optimize the scheduling of tasks across agents. In contrast, HP-HC and HP-BC significantly improve both aspects, generating more efficient routes and compact Gantt charts.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

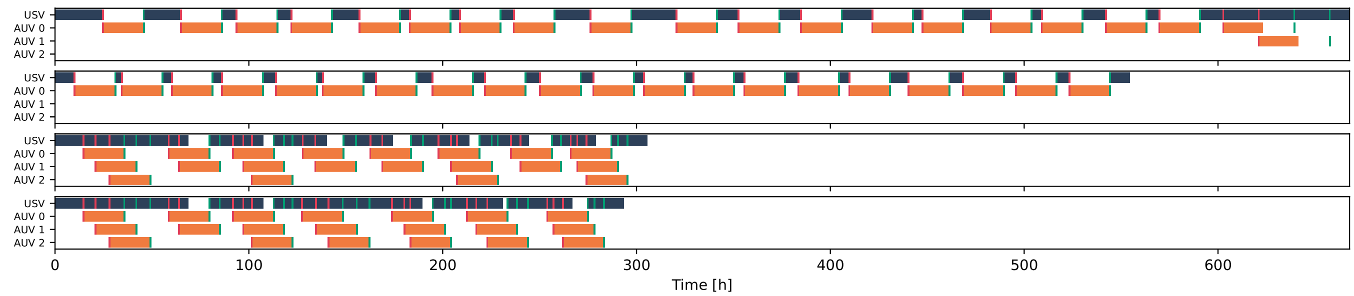


Fig. 7. Gantt charts showing temporal action sequences of each agent. Actions are color-coded: **move** (dark gray), **launch** (pink), **retrieve** (green), and **execute** (orange). From top to bottom: PDDL-only, PDDL-TSP, HP-HC, and HP-BC. Corresponding makespans are 668, 555, 306, and 293 hours, respectively.

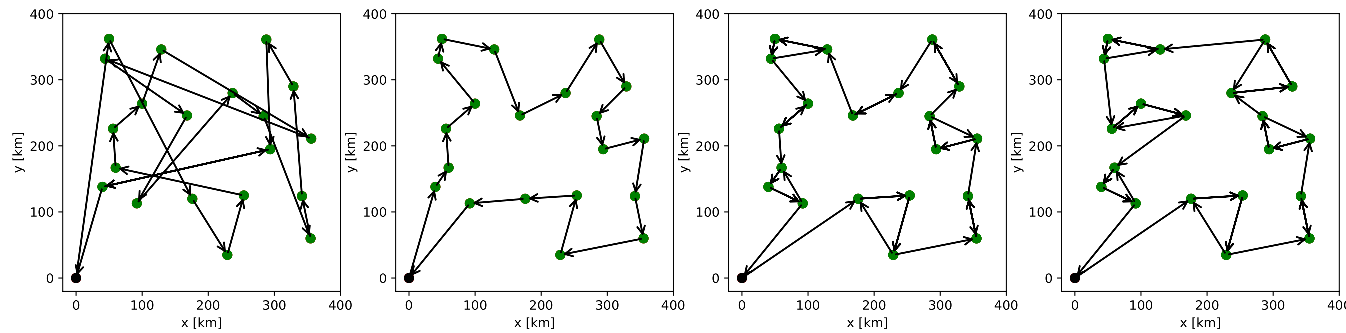


Fig. 8. Visualization of USV paths between tasks. The starting point is marked in black, task locations are shown in green, and arrows represent the paths taken by the USV. From left to right: PDDL-only, PDDL-TSP, HP-HC, and HP-BC.

TABLE V
AVERAGE MAKESPAN AND COMPUTATION TIMES OF ALL METHODS WITH 95% CONFIDENCE INTERVALS.

Method	Task	Makespan [h]	Solve time [s]		
			Scheduling	Partitioning	Routing
PDDL-only	10	370.09±3.80	50.00±0.00	-	-
	20	738.78±6.52	100.00±0.00	-	-
	40	*	200.00±0.00	-	-
	80	*	400.00±0.00	-	-
PDDL-TSP	10	308.79±1.44	45.00±0.00	-	0.068±0.03
	20	551.52±1.65	90.00±0.00	-	0.41±0.19
	40	1028.66±1.35	180.00±0.00	-	7.24±3.55
	80	1956.04±1.18	360.00±0.00	-	37.03±3.54
HP-HC	10	215.46±4.21	33.69±2.99	0.002±0.00	0.002±0.00
	20	353.18±6.66	72.21±3.58	0.002±0.00	0.01±0.00
	40	718.67±11.75	136.82±5.69	0.003±0.00	0.11±0.05
	80	**1296.07±15.89**	263.68±9.02	0.005±0.00	5.36±4.78
HP-BC	10	**202.89±3.19**	30.19±0.09	0.03±0.00	0.001±0.00
	20	**300.36±3.33**	79.11±0.79	0.34±0.05	0.01±0.00
	40	**593.29±5.90**	151.71±1.53	19.41±1.14	0.07±0.03
	80	1375.25±17.97	320.00±0.00	40.00±0.00	0.52±0.20

Note. "*" indicates that no feasible plan was found within the time limit, "–" denotes that the component is not used in the corresponding method, and **boldface** highlights the best performance (smallest makespan) for each task size.

For the quantitative evaluation, we conduct Monte Carlo simulations using task sizes of 10, 20, 40, and 80, generating 100 randomized scenarios per configuration. The outcomes are summarized in Table V, which reports the average makespan and solve times of all methods with 95% confidence intervals. Solve times are further decomposed into scheduling, partitioning, and routing components.

As shown in the results, the hierarchical methods (HP-HC and HP-BC) consistently outperform the baselines across all task sizes. In particular, HP-BC achieves superior performance in small-scale missions (10–40 tasks), while HP-HC shows clear advantages in large-scale scenarios, notably with 80 tasks. This trend reflects differences in computational cost: although BC can generate high-quality partitions, its optimization-based nature leads to rapidly increasing partitioning overhead as the problem size grows. By contrast, HC relies on a lightweight heuristic-based clustering procedure that keeps partitioning time negligible regardless of problem scale, thereby enabling stable performance as the number of tasks increases. Overall, BC is preferable when solution quality is paramount in small missions, while HC is better suited for large missions where timely planning is critical.

C. Evaluation in a multi-fleet setting

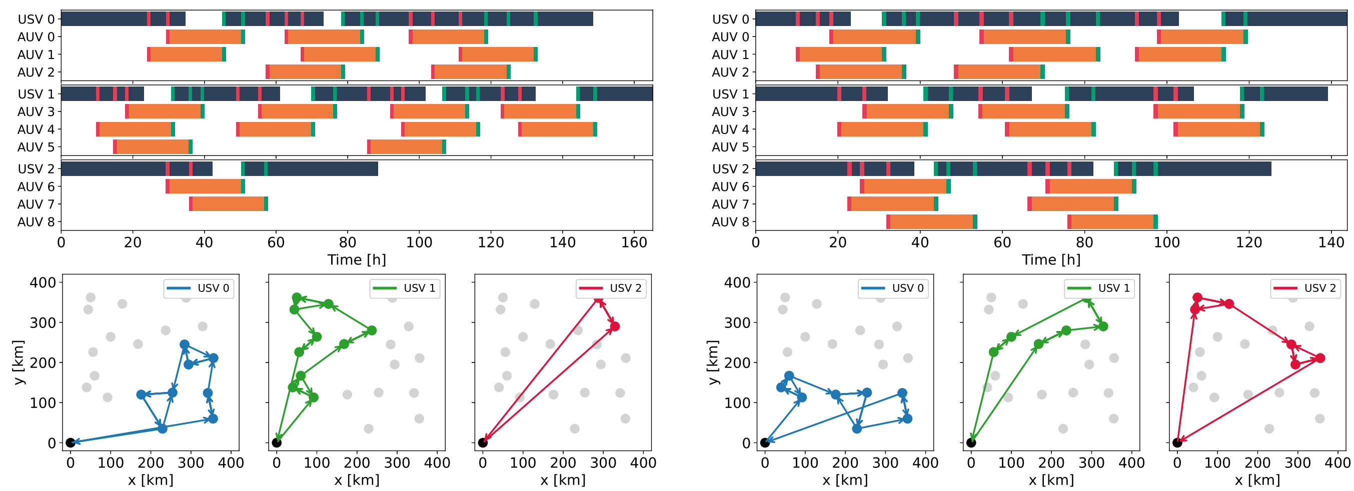
In this section, we evaluate the impact of the node-weighted cost model on mission planning performance and analyze scalability with respect to fleet size and task density. To isolate the effect of the routing cost model, the mission partitioning method is fixed to hierarchical clustering. Figure 9 presents mission planning results for a representative scenario with 20 tasks, obtained using two different routing cost models: Euclidean cost and node-weighted cost. Under the Euclidean cost model, one fleet finishes much earlier than the others, indicating an unbalanced workload distribution. In contrast, the node-weighted cost yields a more balanced plan in which all fleets finish at approximately the same time.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE VI
AVERAGE MAKESPAN, GAP AND COMPUTATION TIME OF ALL METHODS WITH 95% CONFIDENCE INTERVALS

Method	Task	3 fleets			5 fleets			10 fleets		
		Makespan [h]	Gap [h]	Solve time [s]	Makespan [h]	Gap [h]	Solve time [s]	Makespan [h]	Gap [h]	Solve time [s]
HP-HC (Euclidean)	50	374.11±22.48	171.80±24.74	186.36±9.90	266.56±21.01	141.56±26.73	185.68±8.62	226.86±30.04	178.27±30.26	187.32±9.88
	100	624.06±25.06	184.81±53.47	390.91±18.44	447.29±42.21	205.21±61.41	387.46±16.47	397.44±37.57	338.96±43.86	385.45±14.81
	200	1397.3±186.4	553.68±300.3	778.99±21.98	1040.4±283.5	645.67±304.8	768.36±25.22	626.36±391.8	431.43±424.8	749.11±52.98
HP-HC (Node-weighted)	50	**315.18±13.37**	**47.227±18.11**	187.47±9.68	**232.19±6.327**	**67.44±11.63**	186.68±9.81	**145.45±4.891**	**48.28±7.03**	185.96±9.92
	100	**592.90±24.19**	**83.21±24.19**	389.71±15.06	**406.87±10.81**	**91.95±10.77**	383.66±19.28	**254.73±16.84**	**124.85±34.82**	384.60±16.26
	200	**1158.9±19.80**	**135.5±33.50**	760.03±23.36	**782.02±53.42**	**195.76±106.9**	755.61±24.50	**334.79±78.61**	**208.31±72.10**	790.28±50.82

Note. **Boldface** highlights the best performance for each task size (i.e., the smallest makespan and the smallest gap). The gap refers to the difference between the maximum and minimum mission completion times across fleets. Solve time represents the total computation time, including partitioning, routing, and scheduling.



(a) (b)

Fig. 9. Qualitative mission planning results with three fleets for a 20-task scenario, showing USV paths and Gantt charts. (a) Using Euclidean cost, the fleets travel similar distances but one fleet finishes much earlier, revealing unbalanced workload distribution (makespan: 165.19 h). (b) Using node-weighted cost, the workload is more evenly distributed and all fleets complete their missions at nearly the same time (makespan: 143.86 h).

To generalize these observations and examine scalability, Monte Carlo simulations are conducted with 50, 100, and 200 tasks across multiple fleets. The results, summarized in Table VI, report the average makespan, completion gap (the difference between the longest and shortest fleet completion times), and solve time with 95% confidence intervals. Compared with Euclidean cost, the node-weighted model consistently achieves smaller gaps and lower variance, leading to more balanced coordination and shorter makespans overall. These findings indicate that incorporating node-weighted cost into the routing subproblem effectively promotes workload balance, minimizes fleet idle time, and enhances the overall quality of mission planning.

Nonetheless, as the number of fleets and tasks grows, the time gap tends to increase. This reflects the limitation of solving the routing subproblem with exact optimization, which is effective for small problems but less scalable for larger ones. To address this, future work could employ local search–based metaheuristics such as tabu search, iterated local search (ILS), or large neighborhood search (LNS) to improve the scalability of the routing subproblem while preserving solution quality. Such methods can be incorporated without altering the overall hierarchical framework.

VI. CONCLUSION

This study addressed the MVRSP, focusing on efficiently performing spatially distributed tasks using a marsupial system, a specialized type of heterogeneous robotic system. By applying a hierarchical planning framework, the NP-hard problem is decomposed into routing and scheduling subproblems. To further enhance performance, clustering is employed for scalable mission partitioning and a node-weighted cost is introduced to balance workloads across fleets. Our mission planner significantly improves both planning efficiency and plan quality compared to the baseline, highlighting the effectiveness of our hierarchical approach in addressing the complexities of MVRSP.

While this study focuses on a simplified formulation to highlight the core routing and scheduling aspects of the MVRSP, real-world deployment would require handling additional factors. A promising extension is to incorporate battery constraints into the PDDL model by modeling capacity as a resource and adding a charging action, enabling more realistic planning under endurance limits. Similarly, other practical factors such as sensing uncertainty, communication delays, or vehicle failures could be handled through extensions or robust planning for more resilient deployment.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

APPENDIX A
EXCERPT OF PDDL MODEL FOR MVRSP

```
;; Domain
(define (domain mvrsp)
    (:types TA Task)
    (:predicates
        (RA−at ?t − Task) (TA−at ?a − TA ?t − Task)
        (RA−available) (TA−available ?a − TA)
        (on−RA ?a − TA) (done ?t − Task) (open ?t − Task))
    (:functions
        (dist ?ti ?tf − Task) (task−size ?t − Task)
        (RA−speed) (TA−speed) (launch−duration) (retrieve−duration))
    ;; Durative actions (skeletons)
    (:durative−action move
        :parameters (?ti ?tf − Task)
        :duration (= ?duration (/ (dist ?ti ?tf) (RA−speed)))
        ...)
    (:durative−action launch
        :parameters (?a − TA ?t − Task)
        :duration (= ?duration launch−duration)
        ...)
    (:durative−action execute
        :parameters (?a − TA ?t − Task)
        :duration (= ?duration (/ (task−size ?t) (TA−speed)))
        ...)
    (:durative−action retrieve
        :parameters (?a − TA ?t − Task)
        :duration (= ?duration retrieve−duration)
        ...)
)
;; Problem
(define (problem mvrsp−prob) (:domain mvrsp)
    (:objects a0 a1 − TA t0 t1 − Task)
    (:init
        (= (dist t0 t1) 200) (= (RA−speed) 15) (= (launch−duration) 1)
        (= (task−size t1) 100) (= (TA−speed) 5) (= (retrieve−duration) 1)
        (open t0) (open t1) (RA−at t0) (RA−available)
        (TA−available a0) (on−RA a0))
    (:goal (and (done t1) (on−RA a0) (RA−at t0)))
)
```

REFERENCES

[1] M. O. Anderson, M. D. McKay, and B. S. Richardson, "Multirobot automated indoor floor characterization team," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1996, pp. 1750–1753.

[2] S. Martinez-Rozas, D. Alejo, F. Caballero, and L. Merino, "Path and trajectory planning of a tethered uav-ugv marsupial robotic system," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6475–6482, 2023.

[3] R. R. Murphy, E. Steimle, C. Griffin, C. Cullins, M. Hall, and K. Pratt, "Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane wilma," *Journal of Field Robotics*, vol. 25, no. 3, pp. 164–180, 2008.

[4] B. Li, B. R. Page, B. Moridian, and N. Mahmoudian, "Collaborative mission planning for long-term operation considering energy limitations," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4751–4758, 2020.

[5] F. Marques, A. Lourenço, R. Mendonça, E. Pinto, P. Rodrigues, P. Santana, and J. Barata, "A critical survey on marsupial robotic teams for environmental monitoring of water bodies," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–6.

[6] R. Mendonça, M. M. Marques, F. Marques, A. Lourenco, E. Pinto, P. Santana, F. Coito, V. Lobo, and J. Barata, "A cooperative multi-robot team for the surveillance of shipwreck survivors at sea," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016, pp. 1–6.

[7] B. Balaram, T. Canham, C. Duncan, H. F. Grip, W. Johnson, J. Maki, A. Quon, R. Stern, and D. Zhu, "Mars helicopter technology demonstrator," in *2018 AIAA atmospheric flight mechanics conference*, 2018, p. 0023.

[8] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[9] G. D. Konstantakopoulos, S. P. Gayialis, and E. P. Kechagias, "Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification," *Operational research*, vol. 22, no. 3, pp. 2033–2062, 2022.

[10] L. Antonyshyn, J. Silveira, S. Givigi, and J. Marshall, "Multiple mobile robot task and motion planning: A survey," *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–35, 2023.

[11] M. Fox and D. Long, "Pddl2. 1: An extension to pddl for expressing temporal planning domains," *Journal of artificial intelligence research*, vol. 20, pp. 61–124, 2003.

[12] F. Jovan and S. Bernardini, "Adaptive temporal planning for multi-robot systems in operations and maintenance of offshore wind farms," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 13, 2023, pp. 15 782–15 788.

[13] X. Zhang, H. Qin, F. Wang, Y. Dong, and J. Li, "Lamma-p: Generalizable multi-agent long-horizon task allocation and planning with lm-driven pddl planner," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 221–10 221.

[14] J. Jang, H. Do, and J. Kim, "Mission planning for underwater survey with autonomous marine vehicles," *Journal of Ocean Engineering and Technology*, vol. 36, no. 1, pp. 41–49, 2022.

[15] D. Kim, J. Jang, and J. Kim, "Mission planning for collaborative operations of autonomous marine systems," in *OCEANS 2023-Limerick*. IEEE, 2023, pp. 1–5.

[16] M. S. Mondal, S. Ramasamy, J. D. Humann, J.-P. F. Reddinger, J. M. Dotterweich, M. A. Childers, and P. Bhounsule, "Optimizing fuel-constrained uav-ugv routes for large scale coverage: Bilevel planning in heterogeneous multi-agent systems," in *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2023, pp. 114–120.

[17] J. Banfi, A. Messing, C. Kroninger, E. Stump, S. Hutchinson, and N. Roy, "Hierarchical planning for heterogeneous multi-robot routing problems via learned subteam performance," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4464–4471, 2022.

[18] J. J. Miranda-Bront, B. Curcio, I. Méndez-Díaz, A. Montero, F. Pousa, and P. Zabala, "A cluster-first route-second approach for the swap body vehicle routing problem," *Annals of Operations Research*, vol. 253, no. 2, pp. 935–956, 2017.

[19] B. Singh, L. Oberfichtner, and S. Ivliev, "Heuristics for a cash-collection routing problem with a cluster-first route-second approach," *Annals of Operations Research*, vol. 322, no. 1, pp. 413–440, 2023.

[20] D. Prajapati, A. R. Harish, Y. Daultani, H. Singh, and S. Pratap, "A clustering based routing heuristic for last-mile logistics in fresh food e-commerce," *Global Business Review*, vol. 24, no. 1, pp. 7–20, 2023.

[21] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 895–943, 2022.

[22] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[23] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[24] Y. Findik, H. Osooli, P. Robinette, K. Jerath, and S. R. Ahmadzadeh, "Influence of team interactions on multi-robot cooperation: A relational network perspective," in *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2023, pp. 50–56.

[25] J. F. Lucas, *Introduction to abstract mathematics*. Rowman & Littlefield, 1990.

[26] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[27] X. Bao, G. Wang, L. Xu, and Z. Wang, "Solving the min-max clustered traveling salesmen problem based on genetic algorithm," *Biomimetics*, vol. 8, no. 2, p. 238, 2023.

[28] A. Coles, A. Coles, M. Fox, and D. Long, "Forward-chaining partial-order planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 20, 2010, pp. 42–49.

[29] A. J. Coles, A. I. Coles, M. Fox, and D. Long, "Colin: Planning with continuous linear numeric change," *Journal of Artificial Intelligence Research*, vol. 44, pp. 1–96, 2012.

[30] J. Benton, A. Coles, and A. Coles, "Temporal planning with preferences and time-dependent continuous costs," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 22, 2012, pp. 2–10.

[31] C. U. Manual, "Ibm ilog cplex optimization studio," *Version*, vol. 12, no. 1987-2018, p. 1, 1987.