Supplementary Information

Cooperate to Generalize: Deep Reinforcement Learning for Real-time Ad Hoc Team Routing

Contents

1 Supplementary figures

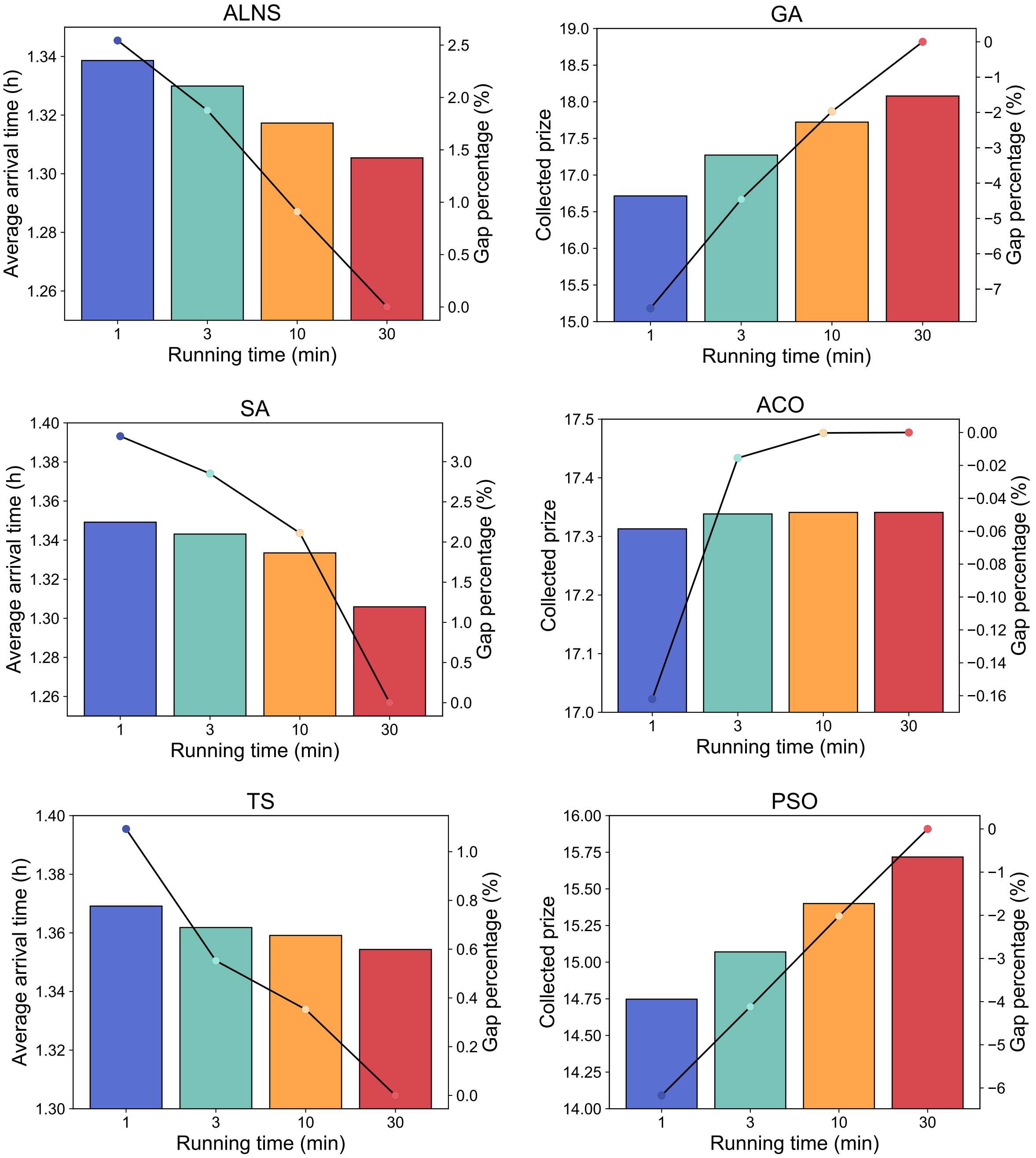

Fig. 1: **Results of heuristic-based methods under various time points.** The three figures on the left show the results of PDP under ALNS, SA, and TS algorithms; The three figures on the right show the results of TOP under GA, ACO, and PSO algorithms. The maximum operating times are restricted to 1 minute, 3 minutes, 10 minutes, and 30 minutes.

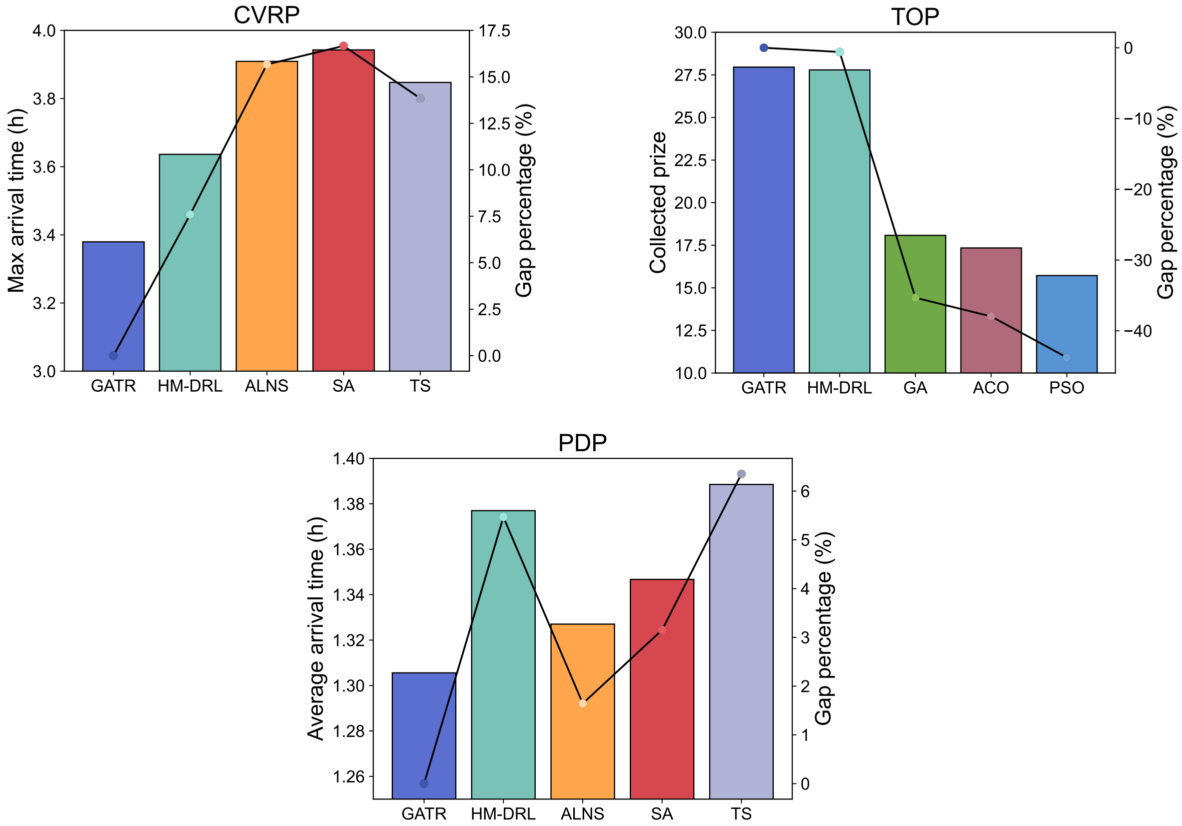

Fig. 2: **Out-of-distribution results.** During the training phase, we employ uniform distributions to simulate team configurations; While in the testing phase, normal distributions are utilized to mimic real team configurations. In all models, GATR still maintains its advantages, which demonstrates its strong generalization capability when facing out-of-distribution teams.

2 Supplementary tables

Table 1: Notation table for CVRP.

Sets	Description
N	The demand node set
N_0	The node set, $N_0 = N \cup \{0\}$
K	The ad hoc team set
Parameters	Descriptions
0	The depot of the team
q_i	The demand at node i
d_{ij}	The distance from node i to node j
s_i	The service time spent at node i
c_0	the loading time in depot to replenish supplies
F^k	The speed of team member k
l_i^k	Remaining capacity of member k after leaving node i
Q^k	The nominal capacity of the member k
a_i^k	The time when member k arrives at demand node i
a_0^k	The updated time when member k arrives at depot 0
T	Maximum arrival time of all team members
M_1, M_2, M_3	Large positive constant numbers
Decision Variables	Descriptions
x_{ij}^k	1 iff member k travels from node i to node j

Table 2: Notation table for TOP.

Sets	Description
C	The data collection point set
N	The node set, $N = C \cup \{0\}$
K	The ad hoc team set
Parameters	Descriptions
0	The depot of the team
d_{ij}	The distance from node i to node j
s_i	The service time spent at node i
F^k	The speed of member k
T^k_{max}	The travel time limit of each team member k
p_i	The profit (data collected) of node i
a^k_i	The time when member k arrives at node i
M	A large positive constant number
Decision Variables	Descriptions
x^k_{ij}	1 iff member k travels from node i to node j
z^k_{0j}	1 iff member k departs from the depot to node j on its initial trip

Table 3: Notation table for PDP.

Sets	Description
P	Set of pickup points, $P = \{1, ..., n\}$
D	Set of delivery points, $D = \{n+1, ..., 2n\}$
K	Set of the ad hoc team
N	Set of all points, $N = P \cup D \cup \{0\}$
Parameters	**Description**
n	Number of pickup-delivery pairs
0	The depot of the team
d_{ij}	The route distance between points i and j
ps_i	The service time spent at pickup node i
ds_i	The service time spent at delivery node i
a_i^k	The arrival time at point i of team member k
f_i^k	Cumulative in-trip flight time of member k upon reaching node i
F^k	Travel speed of the member k
L^k	Flight time cap for the member k
sw_0	The time to replace the battery in depot
M_e	$e = \{1, 2, 3, 4, 5, 6\}$, large positive constant numbers
Decision Variables	**Description**
x_{ij}^k	1 iff member k traverses arc (i, j)
z_{0j}^k	1 iff member k departs from the depot to node j on its initial trip

Table 4: Team configuration distributions of ad hoc teams.

Team parameters	**VRP**	**TOP**	**PDP**
Member speed (km / h)	$U(20, 40)$	$U(50, 100)$	$U(50, 100)$
Member capacity (t)	$U(2, 5)$	N/A	N/A
Member endurance (h)	N/A	$U(2, 4)$	$U(2, 4)$
The number of members	$U(3, 6)$	$U(5, 10)$	$U(2, 5)$

Table 5: **Settings of task scenarios.**

Scenario parameters	**VRP**	**TOP**	**PDP**
The number of demands	$U(60, 80)$	$U(100, 120)$	$U(30, 40)$
Service time (h)	1/6	0.25	1/12
Loading time (h)	1/6	N/A	N/A
Battery change time (h)	N/A	N/A	1/6

Table 6: **Hyperparameter settings of networks.**

Hyperparameter	**Value**
Dimension of node embedding layers	128
Dimension of hidden layers	512
Dimension of context embedding layers	128
Dimension of query, key, and value	16
Number of heads	8
Number of encoder layers	6
Number of context layers	1
Clipping factor	10

Table 7: **Hyperparameter settings of training.**

Hyperparameter	**Value**
Batch size	32
Pomo size	10
Number of epochs	$2e^3$
Number of episodes per epoch	$1e^5$
Initial learning rate	$1e^{-4}$
Weight decay	$1e^{-6}$
Milestones	$[501, 1001, 1501]$
Decay rate γ_{lr}	0.5

Table 8: Team configuration of the out-of-distribution ad hoc team.

Team parameters	VRP	TOP	PDP
Member speed (km / h)	$\mathcal{N}(30, 10)$	$\mathcal{N}(75, 20)$	$\mathcal{N}(75, 20)$
Truncation range of member speed (km / h)	$(10, +\infty)$	$(40, +\infty)$	$(40, +\infty)$
Member capacity (t)	$\mathcal{N}(3.5, 1)$	N/A	N/A
Truncation range of member capacity (t)	$(1, +\infty)$	N/A	N/A
Member endurance (h)	N/A	$\mathcal{N}(3, 1)$	$\mathcal{N}(3, 1)$
Truncation range of member endurance (h)	N/A	$(1, +\infty)$	$(1, +\infty)$

3 Optimization models

We rigorously present the mathematical optimization models for the three scenarios under consideration, namely, the min-max Capacitated Vehicle Routing Problem (CVRP), the Team Orienteering Problem (TOP), and the Pickup and Delivery Problem (PDP). Considering the establishment of ad hoc teams in our work, the distinction between our models and the standard formulations lies in the heterogeneity among team members in our setting. The corresponding notation tables, mathematical equations, and detailed descriptions are provided below.

3.1 Capacitated Vehicle Routing Problem (CVRP)

In the CVRP model, the heterogeneity between team members lies in capacity and speed. The objective is to minimize the maximum arrival time among all trucks while fulfilling all supply demands. The mathematical model is formulated below.

$$\min \ T \tag{1}$$

s.t.

$$\sum_{k\in K}\sum_{i\in N_0} x_{ij}^k = 1 \quad \forall j \in N, \tag{2}$$

$$\sum_{i\in N_0} x_{ij}^k = \sum_{h\in N_0} x_{jh}^k \quad \forall j \in N,\ \forall k \in K, \tag{3}$$

$$\sum_{j\in N} x_{0j}^k = \sum_{i\in N} x_{i0}^k \quad \forall k \in K, \tag{4}$$

$$l_0^k = Q^k \quad \forall k \in K, \tag{5}$$

$$l_j^k \leq l_i^k - q_j + Q^k(1 - x_{ij}^k) \quad \forall i \in N_0, \forall j \in N, \forall k \in K, \tag{6}$$

$$0 \leq l_i^k \leq Q^k \quad \forall i \in N_0, \forall k \in K, \tag{7}$$

$$a_j^k \geq a_i^k + \frac{d_{ij}}{F^k} + s_i - M_1(1 - x_{ij}^k) \quad \forall i \in N, \forall j \in N, \forall k \in K, \tag{8}$$

$$a_j^k \geq a_0^k + (1 - z_{0j}^k)c_0 + \frac{d_{0j}}{F^k} - M_2(1 - x_{0j}^k) \quad \forall j \in N, \forall k \in K, \tag{9}$$

$$a_0^k \geq a_i^k + s_i + \frac{d_{i0}}{F^k} - M_3(1 - x_{i0}^k) \quad \forall i \in N, \forall k \in K, \tag{10}$$

$$T \geq a_i^k \quad \forall i \in N, \forall k \in K, \tag{11}$$

$$x_{ij}^k \in \{0, 1\} \quad \forall k \in K, \forall i \in N, \forall j \in N. \tag{12}$$

Constraint (2) stipulates that each demand node must be served exactly once by a member. Flow balance at each demand node for team members is enforced by constraint (3), and similarly, constraint (4) specifies the balance of team members entering and leaving the depot, with both the starting and ending points fixed at the depot. Multi-trip operations are allowed, and each team member is permitted to replenish their supply to full capacity upon returning to the depot, as indicated in constraint (5). Constraints (6) and (7) delineate the evolution of load carried by team members when visiting any demand node, while ensuring compliance with their nominal capacity; it is noteworthy that constraint (6) has been linearized. The temporal ordering of node visits by team members is defined in constraints (8) to (10), which correspond respectively to the time transitions from one demand node to another, from the depot to a demand node, and from a demand node back to the depot. These constraints incorporate both the service times s_i at each node and the loading time c_0 at the depot, and all have been linearized using the big-M method. It is noteworthy that when a team member departs from the depot for the first time, this moment is considered as time zero, and no additional loading time is incurred. Consequently, we introduce an extra decision variable z_{0j}^k in (9) to indicate whether team member k directly proceeds to demand node j after the initial departure from the depot. Equation (11) establishes the relationship between the arrival time at each demand node and the overall optimization objective T. The binary characteristics of the decision variables are specified in equation (12).

3.2 Team Orienteering Problem (TOP)

In the TOP model, the heterogeneity between team members lies in speed and endurance. The ad hoc team is hoped to collect maximum profits within a limited time frame, which reflects the collected valuable disaster-related information.

$$\max \quad \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} p_j x_{ij}^k \tag{13}$$

$s.t.$

$$\sum_{i\in N} x_{im}^k = \sum_{j\in N} x_{mj}^k \quad \forall m \in C, \forall k \in K, \tag{14}$$

$$\sum_{k\in K}\sum_{i\in N} x_{ij}^k \leq 1 \quad \forall j \in C, \tag{15}$$

$$\sum_{j\in N} x_{0j}^k = \sum_{j\in N} x_{j0}^k \leq 1 \quad \forall k \in K, \tag{16}$$

$$a_i^k + \frac{d_{ij}}{F^k} + s_i - a_j^k \leq M(1 - x_{ij}^k) \quad \forall k \in K, \forall i \in N, \forall j \in N, \tag{17}$$

$$a_i^k \leq T_{max}^k \quad \forall k \in K, \forall i \in N, \tag{18}$$

$$x_{ij}^k \in \{0, 1\} \quad \forall k \in K, \forall i \in N, \forall j \in N. \tag{19}$$

Constraint (14) establishes the flow balance at the data collection nodes. In the TOP problem, due to constraints such as team members' working time, not all nodes are necessarily visited, as indicated by constraint (15). Similarly, constraint (16) ensures that each team member departs from and returns to the depot at most once. The temporal ordering of node visits by team members is specified in constraint (17), which incorporates the service time s_i required for collecting information at each node (with $s_0 = 0$). The big-M method is employed to linearize this constraint. Constraint (18) stipulates an upper bound on the travel time for team members, denoted as T^k_{max}, requiring them to return to the depot within this limit; T^k_{max} is determined by the heterogeneous capabilities of team members. Finally, constraint (19) specifies the bounds for the decision variables.

3.3 Pickup and Delivery Problem (PDP)

In the PDP model, the heterogeneity between team members lies in speed and endurance. The team member must visit the corresponding pick points to get medical resources before visiting the delivery points. The ad hoc team is expected to minimize the average arrival time at delivery points while fulfilling all delivery demands.

$$\min \quad \frac{1}{n} \sum_{k\in\mathcal{K}} \sum_{i\in N} \sum_{j\in D} a^k_j x^k_{ij} \tag{20}$$

s.t.

$$\sum_{k\in K} \sum_{j\in N} x^k_{ij} = 1 \quad \forall i \in P \cup D, \tag{21}$$

$$\sum_{j\in N} x^k_{0j} = \sum_{i\in N} x^k_{i0} \quad \forall k \in K, \tag{22}$$

$$\sum_{j\in N} x^k_{ji} = \sum_{j\in N} x^k_{ij} \quad \forall i \in P \cup D,\ \forall k \in K, \tag{23}$$

$$x^k_{ij} = 0 \quad \forall i, j \in P,\ \forall k \in K, \tag{24}$$

$$x^k_{ij} = 0 \quad \forall i, j \in D,\ \forall k \in K, \tag{25}$$

$$x^k_{0j} = 0 \quad \forall j \in D,\ \forall k \in K, \tag{26}$$

$$a^k_j \geq a^k_0 + (1 - z^k_{0j})sw_0 + \frac{d_{0j}}{F^k} - M_1(1 - x^k_{0j}) \quad \forall j \in P,\ \forall k \in K, \tag{27}$$

$$a^k_j \geq a^k_i + ps_i + \tfrac{d_{ij}}{F^k} - M_2\big(1 - x^k_{ij}\big) \quad \forall i \in P,\ \forall j \in D \cup \{0\},\ \forall k \in K, \tag{28}$$

$$a^k_j \geq a^k_i + ds_i + \tfrac{d_{ij}}{F^k} - M_3\big(1 - x^k_{ij}\big) \quad \forall i \in D,\ \forall j \in P \cup \{0\},\ \forall k \in K, \tag{29}$$

$$f^k_j \geq f^k_0 + (1 - z^k_{0j})sw_0 + \frac{d_{0j}}{F^k} - M_4(1 - x^k_{0j}) \quad \forall j \in P,\ \forall k \in K, \tag{30}$$

$$f^k_j \geq f^k_i + ps_i + \tfrac{d_{ij}}{F^k} - M_5\big(1 - x^k_{ij}\big) \quad \forall i \in P,\ \forall j \in D \cup \{0\},\ \forall k \in K, \tag{31}$$

$$f_j^k \geq f_i^k + ds_i + \tfrac{d_{ij}}{F^k} - M_6\big(1 - x_{ij}^k\big) \quad \forall i \in D,\ \forall j \in P \cup \{0\},\ \forall k \in K, \tag{32}$$

$$f_j^k \leq L^k \quad \forall j \in N,\ \forall k \in K, \tag{33}$$

$$f_0^k = 0 \quad \forall k \in K, \tag{34}$$

$$x_{ij}^k \in \{0,1\}, \quad a_i^k \geq 0, \quad f_i^k \geq 0 \quad \forall i,j \in N,\ \forall k \in K. \tag{35}$$

Constraint (21) ensures that each task node, whether a pickup or a delivery, is visited exactly once by a team member. Constraint (22) enforces flow balance as team members depart from and return to the depot, while also allowing multiple visits to the depot. Similarly, constraint (23) defines the flow balance for the task nodes. Constraints (24) to (26) restrict the visiting order among different types of nodes, requiring that team members execute tasks in a sequential pickup–delivery pair order. Constraints (27) to (29) maintain the temporal continuity of arrivals at each node, where a_i^k represents the arrival time of team member k at point i. In contrast, f_i^k denotes the cumulative flight time for a team member (UAV) during a single trip; notably, if a team member returns to the depot before reaching the time limit to change batteries, the cumulative flight time is reset to 0 (i.e., $f_0^k = 0$), and a new trip commences. The continuity of cumulative time is established in constraints (30) to (32). It is noteworthy that all three categories of nodes—those in the set P, the points in the set D, and the depot (node 0)—are associated with a service time, denoted by ps_i, ds_i and sw_0, respectively. Since no battery change is required when a team member departs from the depot for the first time, an additional decision variable z_{0j}^k is introduced to indicate whether team member k proceeds directly to task node j on the initial departure from the depot. The upper bound on the working time for a single trip by a team member is imposed by constraint (33), whereas constraints (34) and (35) delineate the permissible ranges for the variables including decision variables.

4 MDPs for ad hoc team routing problems

As a important branch of CO problems, The construction of a solution for the routing problems is usually modeled as a Markov decision process (MDP), which is defined by a tuple of components $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$. Suppose $\mathcal{M} = \{1, 2, \ldots, M\}$ is the set of the ad hoc team, and N is the set of nodes to serve. $\mathcal{P}$ is the state transition function, which we take a deterministic probability, that is, $\mathcal{P}(s_{t+1}|a_t, s_t) = 1$ if s_{t+1} is the result of s_t taking action a_t, otherwise $\mathcal{P}(s_{t+1}|a_t, s_t) = 0$. γ is the discount factor of the return.

- **State:** The state $s_t = (\mathbf{K}_t, \mathbf{V}_t) \in \mathcal{S}$ contains the current state of both team and nodes, where s_0, s_Θ are the initial empty state and final complete state, respectively. Here, $\mathbf{K}_t$ and $\mathbf{V}_t$ have different features according to different CO problems. For example, in CVRP, a team member state $k_t^j = (y_t^j, c_t^j, e_t^j, C^j, F^j) \in \mathbf{K}_t$ consists of the coordinates y_t^j, the remaining capacity c_t^j, cumulative travel time e_t^j, along with its nominal capacity C^j and speed F^j of the member j. The node state $v_t^i = (x_t^i, q_t^i) \in \mathbf{V}_t$ contains the coordinates x_t^i and the demand q_t^i of the unserved node i. As for TOP, the member state is represented as $k_t^j = (y_t^j, o_t^j, F^j)$, where y_t^j is the coordinates, o_t^j is the remaining time budget of a member since there is an upper time limit T_{max}^j in TOP, and F^j is the member speed. The node state is characterized as $v_t^i = (x_t^i, z_t^i)$, where x_t^i is the node coordinates, and z_t^i is the prize provided for visiting this node. In PDP, the member state $k_t^j = (y_t^j, e_t^j, L^j, F^j)$ contains its coordinates y_t^j, cumulative travel time e_t^j, maximum flight time L^j of a single trip, and the member speed F^j. The node state is characterized as $v_t^i = (x_t^i, g_t^i)$, where x_t^j is the node coordinate, g_t^i is the flag denoting the type of this node, i.e., pickup, delivery, or the depot node.
- **Action:** In routing problems involving only a single member, action typically consists of selecting a node from un-visited nodes. Some existing methods make decisions based on a predetermined order of team members, and the chosen node is for the member currently making the decision. Relatively, in ad hoc team settings, there is no predefined sequence of decisions for team members. For a single team member, its action is to select an available node, while in the view of the entire team, the current action includes choosing a team member and the node it intends to visit, i.e., $a_t = (k_t^j, v_t^i) \in \mathcal{A}$.
- **Reward:** Denote the solution of a problem instance $\mathcal{I}$ as $\mathbf{\Gamma} = (\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \ldots, \boldsymbol{\tau}^M)$ which contains the tours of all team members, the element of a member's tour $\boldsymbol{\tau}^k$ is a sequence of nodes visited by it. For the CVRP whose goal is to minimize the maximum arrival time among all team members, the reward in MDP is the negative value of that, i.e.,

$\mathcal{R}(\mathbf{\Gamma};\mathcal{I}) = -\max_{k\in\mathcal{M}}\{\boldsymbol{\tau}^k\}_{\text{time}} = -\max_{k\in\mathcal{M}, i\in\boldsymbol{\tau}^k\setminus\{0\}}\{a_i^k\}$, where a_i^k refers to the time when the member k arrives at demand node i, index 0 is the depot. In terms of TOP, the objective is to maximize the total prize collected, so the reward is represented as the sum of prizes, i.e., $\mathcal{R}(\mathbf{\Gamma};\mathcal{I}) = \sum_{k\in\mathcal{M}}\{\boldsymbol{\tau}^k\}_{\text{prize}} = \sum_{k\in\mathcal{M}}\sum_{i\in\boldsymbol{\tau}^k} z^i$. For the PDP, we want to minimize the average response time for all delivery nodes. Therefore, the reward is represented as $\mathcal{R}(\mathbf{\Gamma};\mathcal{I}) = -\frac{1}{n}\sum_{k\in\mathcal{M}}\{\boldsymbol{\tau}^k\}_{\text{time}} = -\frac{1}{n}\sum_{k\in\mathcal{M}}\sum_{i\in\boldsymbol{\tau}^k\cap D} a_i^k$, where D stands for the set of delivery nodes.

Given the MDP model, the solution $\mathbf{\Gamma}$ is generated based on the policy π_θ with the trainable parameter θ which is optimized by the following equation:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \left[\mathbb{E}_{\mathcal{I}\sim\rho(\mathcal{I})}[\mathbb{E}_{\mathbf{\Gamma}\sim\pi_\theta}[\mathcal{R}(\mathbf{\Gamma};\mathcal{I})]\right], \tag{36}$$

where $\rho(\mathcal{I})$ is the problem distribution.

5 Comparison methods

We have analyzed the relevant work in the field and added more benchmark methods. Our baselines consist of two main categories: the DRL-based method and heuristic-based approaches. To the best of our knowledge, we are the first to generalize the routing problems to stochastic teams in the field of RL. Therefore, we have adapted mainstream methods designed for heterogeneous fleets [1], created the comparison method HM-DRL, and compared our approach with it. More details of our modifications will be provided below. Besides, we have chosen six popular heuristic-based approaches as benchmarks. Compared to DRL-based methods, these approaches offer greater flexibility in adapting to various team configurations, but generally entail longer search durations to achieve near-optimal solutions within the solution space. Moreover, the performance of heuristic-based methods is often sensitive to hyperparameter settings, which can affect the effectiveness of their search strategies across different team configurations.

5.1 Modified DRL-based method

Consistent with approaches for addressing heterogeneous CVRP [1], this method centers on hierarchical decision-making, whereby members are selected first, followed by their corresponding node selection. The node selection mechanism resembles that used in homogeneous CVRP solutions [2, 3]. Specifically, a query is constructed from the current state of the selected member and the location of its current position. Subsequently, node selection probabilities are derived by querying candidate nodes. In the member selection process, the linear layer was replaced with a self-attention mechanism, which allows the decision model to adapt to varying team configurations rather than being limited to fixed ones. The input to the member selection module comprises the current states and positions of all members. When learning the model, the joint probability of this decision-making process includes two parts: the member selection probability and the node selection probability, and the joint probability is the product of them. Furthermore, to ensure the adaptability of training baselines across various team configurations, we retain POMO as the training framework [4].

5.2 Heuristic-based method

5.2.1 Adaptive Large Neighborhood Search (ALNS)

ALNS is a metaheuristic that explores a wide solution space through an iterative destroy-and-repair methodology [5, 6]. The algorithm partially deconstructs a current solution using various ”destroy” operators (e.g., random or worst removal) and subsequently reconstructs it using ”repair” operators (e.g., greedy or regret insertion). A key feature of ALNS is its adaptive weight adjustment mechanism, which dynamically updates the selection probabilities of operators based on their historical success in improving solutions. Solution acceptance is often governed by a Simulated Annealing-like criterion, which allows for the probabilistic acceptance of inferior solutions to escape local optima. This framework was applied to solve both CVRP and PDP.

5.2.2 Simulated Annealing (SA)

SA is a probabilistic optimization technique inspired by the annealing process in metallurgy [7, 8]. It navigates the solution space by iteratively considering neighbor states and accepting them based on a temperature-dependent probability. While superior solutions are always accepted, inferior solutions may also be accepted with a probability that decreases as a ”temperature” parameter is gradually lowered according to a cooling schedule. This mechanism allows the search to initially explore broadly and later converge towards a local or global optimum, effectively avoiding getting trapped in poor local optima. Neighborhood moves can be generated by operators such as swaps, relocations, or 2-opt reversals. This algorithm was utilized in solving CVRP and PDP.

5.2.3 Tabu Search (TS)

TS is a metaheuristic that guides a search process to explore the solution space by employing memory structures [9, 10]. Its core component is a ”tabu list,” a short-term memory that records recently visited solutions or moves and temporarily forbids them to prevent cycling and encourage exploration of new areas. The algorithm iteratively examines neighbors of the current solution, selecting the best admissible move (i.e., not on the tabu list). An ”aspiration criterion” allows the tabu status of a move to be overridden if the move leads to a solution that is superior to the best-so-far solution. To avoid search stagnation, TS often incorporates diversification strategies, such as perturbations, and can feature adaptive tabu tenures. TS was implemented to address CVRP and PDP.

5.2.4 Genetic Algorithm (GA)

GA is an evolutionary algorithm that mimics the process of natural selection to find optimal or near-optimal solutions [11, 12]. The algorithm operates on a population of candidate solutions (chromosomes), which are encoded to represent a potential solution to the problem. For instance, a string representing the target sequence can be combined with another representing route divisions. Through iterative application of genetic operators such as selection (e.g., roulette wheel), crossover, and mutation, the population evolves over generations. A fitness function evaluates the quality of each solution, and fitter individuals have a higher probability of being selected to produce offspring for the next generation. This approach was applied to solve TOP.

5.2.5 Ant Colony Optimization (ACO)

ACO is a probabilistic metaheuristic inspired by the foraging behavior of ants [13, 14]. The algorithm utilizes a population of artificial ants to construct solutions to combinatorial optimization problems collaboratively. Ants build solutions incrementally, making probabilistic decisions at each step based on heuristic information and artificial pheromone trails. Pheromone levels, which represent the accumulated experience of the ant colony, are updated iteratively; paths leading to high-quality solutions receive more pheromone, making them more attractive to subsequent ants. This positive feedback mechanism guides the search towards promising regions of the solution space. ACO was one of the intelligent algorithms used to solve TOP.

5.2.6 Particle Swarm Optimization (PSO)

PSO is a population-based stochastic optimization technique modeled on the social behavior of bird flocking or fish schooling [15, 16]. The algorithm maintains a swarm of particles, where each particle represents a candidate solution and possesses a position and velocity within the multi-dimensional solution space. Particles ”fly” through this space and adjust their trajectories based on two main factors: their own best-known position (personal best) and the best-known position found by any particle in the entire swarm (global best). This collective movement, guided by shared information, allows the swarm to converge towards optimal solutions. This heuristic was implemented to address TOP.

5.2.7 Performance of heuristic-based methods under various time points

In the test and execution process, we display the performance of heuristic-based methods with a 30-minute time limitation. In this part, we prefer to explore the effect of time limitations on heuristic-based methods. For the test under regular resource deployment, we record the best results of heuristic-based methods under various time points: 1 minute, 3 minutes, 10 minutes, and 30 minutes. The Fig. 1 indicates that sufficient iteration time will improve the performance of heuristic-based methods. However, in time-sensitive scenarios, these methods may not be able to find near-optimal solutions within a limited timeframe, thereby limiting their effectiveness.

6 Scenario and experimental setting

In this section, we offer a comprehensive overview of the scenario and experimental setting. The scenario settings introduction encompasses the parameter ranges for team configuration and the main characteristics of the demand scenarios. The experimental setup primarily outlines the key hyperparameters employed during training and the setting for implementing instance augmentation technique.

6.1 Ad hoc team configuration settings

During the training process, we first determine the range of parameters for the team configuration, then construct distributions and sample from them to obtain specific data. The simulated distribution of team configurations for the training process is presented in Table 4. We use a uniform distribution to simulate the distribution of team configurations during the training process, with the number of members specified as integers.

6.2 Demand scenarios settings

To ensure that the route scheduling aligns with the actual needs of real-world scenarios, we establish the scope of demand quantities and the additional time needed to perform tasks, excluding travel time, based on experience and expert opinions. The settings are displayed in Table 5. Similarly, we use uniform distributions to simulate the demand quantities. In the VRP, the relief unloading time is set to 10 minutes, and the loading time after trucks return to the depot is also set to 10 minutes. In the TOP, the UAV data collection time at each point is 15 minutes. In the PDP, the pick time and delivery time for UAV are both 5 minutes, and the battery change time for UAV costs 10 minutes.

6.3 Hyperparameters for policy networks and experiments

Tables 6 and 7 display hyperparameter settings for our proposed policy networks and experiments. In addition, to facilitate better convergence of the policy, we dynamically adjust the learning rate. Specifically, the learning rate is decayed by γ_{lr} once the number of epochs reaches one of the milestones. To ensure fairness in comparison, the experiments and policy network hyperparameters of the RL methods used for comparison are kept consistent with those of our proposed RL method.

6.4 Settings of instance augmentation

Instance augmentation is a data augmentation technique used to increase the diversity and quantity of training data, with the goal of improving model generalization and robustness. It has been widely used in various applications, including image classification, object detection, and speech recognition. Here, we utilize this technique to generate more equivalent instances during the decoding stage, thereby stimulating the learned policy to obtain more diverse solutions, which is expected to yield better results. Common methods to generate new instances includes rotation, scaling, and flipping. We have chosen the flipping method here, which is consistent with [4]. Specifically, we finish the $\times 8$ instance augmentation through the coordinate transformation operations: (x, y), (y, x), $(1 - x, y)$, $(1 - y, x)$, $(x, 1 - y)$, $(y, 1 - x)$, $(1 - x, 1 - y)$, and $(1 - y, 1 - x)$.

7 Out-of-distribution test for ad hoc team

In the main text, we consider extreme scenarios in situations where transportation resources may be either abundant or scarce, and we simulate this by imposing restrictions on the number of team members. In addition to the potential impact of the various numbers of members on the effectiveness of the policy network, shifts in the distribution of team configurations may also pose challenges. Therefore, to discuss the performance of our proposed method under different distributions of team configurations, we further conduct out-of-distribution testing. The test distributions of team configurations are presented in Table 8. Compared to the uniform distributions in Table 4, we reconstruct the test data using normal distributions. Additionally, to avoid obtaining inappropriate configuration information from sampling the normal distribution, we apply range truncation to the sampled data. For the new distributions, we generate 1,000 examples per scenario and evaluate the performance of various methods on these instances. The results are displayed in Fig. 7. The tests prove that our method retains good generalization capabilities when applied to teams with out-of-distribution configurations, while preserving its performance advantage. Moreover, this experiment demonstrates that the effectiveness of our trained policy is not overly sensitive to the training data settings. This attribute underscores the suitability of our method for real-world applications, where precise knowledge of team configuration distributions before tasks happens is often unavailable.

References

[1] Li, J., Ma, Y., Gao, R., Cao, Z., Lim, A., Song, W., Zhang, J.: Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. IEEE Transactions on Cybernetics **52**(12), 13572–13585 (2021)

[2] Kool, W., Van Hoof, H., Welling, M.: Attention, learn to solve routing problems! arXiv preprint arXiv:1803.08475 (2018)

[3] Nazari, M., Oroojlooy, A., Snyder, L., Takác, M.: Reinforcement learning for solving the vehicle routing problem. Advances in neural information processing systems **31** (2018)

[4] Kwon, Y.-D., Choo, J., Kim, B., Yoon, I., Gwon, Y., Min, S.: Pomo: Policy optimization with multiple optima for reinforcement learning. Advances in Neural Information Processing Systems **33**, 21188–21198 (2020)

[5] Pereira, V.G., Alves-Junior, O.C., Baldo, F.: An approach to solve the heterogeneous fixed fleet vehicle routing problem with time window based on adaptive large neighborhood search meta-heuristic. IEEE Transactions on Intelligent Transportation Systems (2024)

[6] Masson, R., Lehuédé, F., Péton, O.: An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transportation Science **47**(3), 344–355 (2013)

[7] Bräysy, O., Dullaert, W., Hasle, G., Mester, D., Gendreau, M.: An effective multirestart deterministic annealing metaheuristic for the fleet size and mix vehicle-routing problem with time windows. Transportation Science **42**(3), 371–386 (2008)

[8] Gutenschwager, K., Niklaus, C., Voß, S.: Dispatching of an electric monorail system: Applying metaheuristics to an online pickup and delivery problem. Transportation science **38**(4), 434–446 (2004)

[9] Gendreau, M., Hertz, A., Laporte, G.: A tabu search heuristic for the vehicle routing problem. Management science **40**(10), 1276–1290 (1994)

[10] Gmira, M., Gendreau, M., Lodi, A., Potvin, J.-Y.: Tabu search for the time-dependent vehicle routing problem with time windows on a road network. European Journal of Operational Research **288**(1), 129–140 (2021)

[11] Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA journal on computing **6**(2), 154–160 (1994)

[12] Mansfield, A., Manjanna, S., Macharet, D.G., Hsieh, M.A.: Multi-robot scheduling for environmental monitoring as a team orienteering problem. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6398–6404 (2021). IEEE

[13] Ke, L., Archetti, C., Feng, Z.: Ants can solve the team orienteering problem. Computers & Industrial Engineering **54**(3), 648–665 (2008)

[14] Verbeeck, C., Vansteenwegen, P., Aghezzaf, E.-H.: The time-dependent orienteering problem with time windows: a fast ant colony system. Annals of Operations Research **254**(1), 481–505 (2017)

[15] Dang, D.-C., Guibadj, R.N., Moukrim, A.: An effective pso-inspired algorithm for the team orienteering problem. European Journal of Operational Research **229**(2), 332–344 (2013)

[16] Xiao, K., Lu, J., Nie, Y., Ma, L., Wang, X., Wang, G.: A benchmark for multi-uav task assignment of an extended team orienteering problem. In: 2022 China Automation Congress (CAC), pp. 6966–6970 (2022). IEEE