Cooperative Multi-Agent Reinforcement Learning for Heterogeneous Vehicle Routing

Junkai Zhang[1,2], Yifan Zhang[1,2,3], Jinmin He[1,2], Yifan Zang[4], Yang Wu[1,2], Jian Cheng[1,2,5]
[1] C[2]DL, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences, Nanjing
[4]Beijing Institute of Astronautical Systems, [5]AiRiA
{zhangjunkai2021, hejinmin2021, zangyifan2019, wuyang2023@ia.ac.cn}@ia.ac.cn,
{yfzhang, jcheng}@nlpr.ia.ac.cn

***Abstract*—Combinatorial Optimization (CO) is fundamental to numerous fields with Heterogeneous Capacitated Vehicle Routing Problem (HCVRP) standing out as a critical application, which optimizes delivery fleet operations. Existing deep learning based (DRL-based) methods show promising performance but suffer from high action space. In this work, we propose Cooperative Multi-Agent Reinforcement Learning for Heterogeneous Vehicle Routing (MAVR) to enhance the performance. We decouple the routing problem into a two-stage learning process: (1) Node Allocation, where customer nodes are assigned to vehicles; (2) Node Traversal, where each vehicle searches for its shortest path. For Task (2), the reduced number of assigned nodes reduces the search space, making the task can be effectively solved using existing DRL-based methods. Consequently, our focus shifts to the Node Allocation task, which can be naturally seen as a cooperative multi-agent task. Specifically, we model each vehicle as an agent within a cooperative multi-agent framework. The process sequentially relays node information to each vehicle agent, allowing them to decide whether to select it. We further implement an adaptive process that dynamically adjusts the vehicle trip number based on a theoretical analysis which narrows down the practical vehicle trip number. We conducted experiments on different routing problems and the results showcase our improved performance.**

***Index Terms*—Heterogeneous-CVRP, MARL, COP**

I. Introduction

The Heterogeneous Capacitated Vehicle Routing Problem (HCVRP) is a fundamental combinatorial optimization peoblem, where a fleet of vehicles with diverse capacities and travel speed is deployed to service a set of customer nodes [1], [2]. Despite intensive research has been implemented in the Operations Research community, HCVRP remains challenging due to its NP-hard nature [3], [4]. Conventional solvers primarily fall into two categories: exact algorithms , such as Branch-and-Cut and Branch-and-Price, which guarantee optimality but scale poorly due to exponential complexity [5], [6]; and meta-heuristics including Large Neighborhood Search (LNS) [7], simulated annealing [8] and genetic algorithm [9], which offer better efficiency but rely on handcrafted rules, limiting their adaptability to diverse problem distributions.

Corresponding author

Recently, deep learning-based (DRL-based) methods have emerged as a promising frontier for routing optimization. While initial works focus on homogeneous routing problems [10]–[12], a growing body of literature has begun to develop specialized neural architectures to tackle HCVRP [13]–[15]. These methods employ reinforcement learning to train an agent that constructs solutions auto-regressively. At each timestep t, the agent selects an action $a_t = (v_t, u_t)$ from a joint space defined by the Cartesian product of vehicles and nodes, $\mathcal{V} \times \mathcal{U}$. Specifically, the agent maps the current state and partial trajectory to a probability distribution over $\mathcal{V} \times \mathcal{U}$, from which the vehicle-node pair is sampled. When a vehicle's remaining capacity becomes insufficient, the agent returns to the depot, thereby completing a *trip*.

Despite showing promising performance, existing works suffer from low data efficiency, which worsens as the number of customer nodes and vehicles increases. This issue arises because they use a single agent to learn two essential HCVRP tasks simultaneously: (1) Assigning customer nodes to vehicles. (2) Determining the shortest path for each vehicle's assigned nodes. For task (1), considering that HCVRP requires multi-vehicle to cooperative, most existing methods use a monolithic single-agent framework, requiring the agent to output a coupled (vehicle, customer node) pair at each decision step, which creates an excessively large action space. Besides, because the agent only focuses on one vehicle at a time, it lacks the multi-vehicle cooperation from a global perspective. For task (2), the node trajectory of each vehicle should be independent of the others and each vehicle should focus on the assigned subset of customer nodes. However, in previous works, simultaneously outputting both vehicles and nodes creates an entangled action space, which obstructs the learning of specialized traversal policies for each vehicle.

Following the idea of two decoupled tasks, we propose Cooperative **M**ulti-**A**gent Reinforcement Learning for Heterogenous **V**ehicle **R**outing, denoted by MAVR. Since two tasks can be learned independently in stages, we convert the HCVRP into a two-stage learning process: Node Allocation and Node Traversal, illustrated in Figure 1. For Node Traversal, it essentially degenerates into a simpler problem

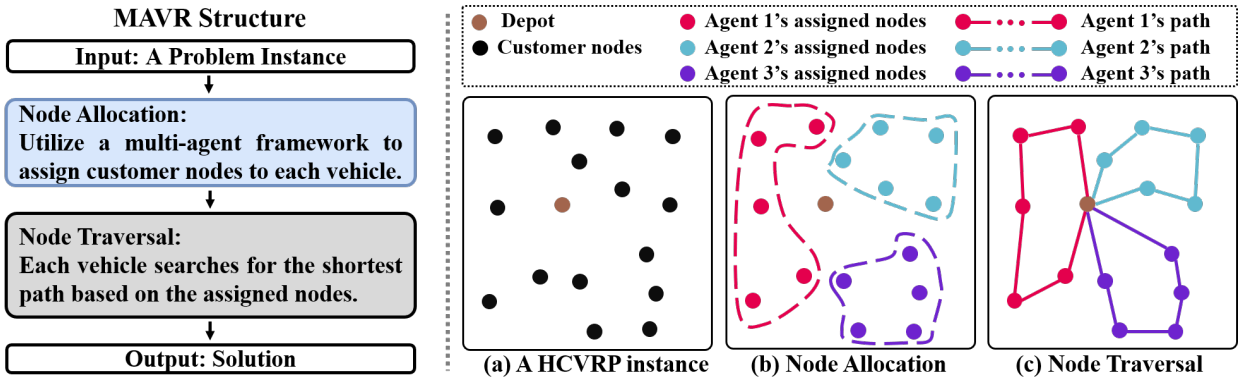

Fig. 1. The overall structure of our method.

with fewer nodes called Traveling Salesman Problem (TSP). Since it can be pre-trained, the key to the HCVRP lies in the Node Allocation. It determines the set of nodes each vehicle will visit. Each vehicle must coordinate to select nodes for overall objective. Therefore we model each vehicle as an agent and formulate Node Allocation as a *Cooperative Multi-Agent Reinforcement Learning* task. At each timestep, a node from the HCVRP is selected and input to vehicle agents, who then take a binary action: whether to select it. A global action mask ensures only one agent obtains the node. The episode ends when all nodes are visited. The reward from Node Traversal serves as feedback to guide MARL learning.

Furthermore, we provide a theoretical analysis highlighting that the optimal number of vehicle trips in HCVRP is dynamic and cannot be pre-determined. To optimize this, we introduce an intrinsic reward within the MARL framework. This mechanism enables an adaptive process that narrow down the practical number of trips.

Our proposed MAVR advances in three aspects. (1) We directly address the Node Allocation task from a global perspective, utilizing a MARL framework that enables vehicle agents to collaborate in assigning customer nodes. (2) The reduced action space simplifies the problem, allowing the existing TSP DRL-based agent to efficiently learn the Node Traversal task with greater effectiveness. (3) Node Traversal can utilize a simpler TSP solver to provide precise and dense rewards, enabling efficient MARL training for Node Allocation. Consequently, MAVR allows us to focus exclusively on the Node Allocation task without repeatedly learning the Node Traversal, thereby enhancing overall training efficiency.

To demonstrate the efficiency of our method, we conduct experiments in different HCVRPs with varied vehicle capacity and speed with varied node numbers. The results demonstrate that our method can rapidly learn the Node Allocation skill for improved data efficiency and has outstanding performance.

II. Related Works

A. Homogeneous Routing Problems

There are some seminal works on solving hemogeneous routing problems. Vinyals et al. [16] propose the Pointer Network as a neural solver to directly construct NCO solutions in an autoregressive manner, which is trained by supervised learning. Bello et al. [17] consider the PointerNet as the policy and propose a RL training strategy. On this basis, Kool et al. [10] utilizes the Transformer architecture [18] and accomplishes improved performance above PointerNet. After that, AM becomes a popular modeling choice and many AM variants are proposed [11] Sym-NCO [19]. However, they treat the entire customer node set as the action space and they overlook the heterogeneity of vehicles.

B. Heterogeneous Routing Problems

A few DRL-based studies have specifically targeted the HCVRP. As a pioneer, Vera et al. [20] trained a node-selection policy, which is applied iteratively to each vehicle. Alternatively, Qin et al. [21] utilized reinforcement learning to dynamically select among multiple meta-heuristics with diverse characteristics. Besides, Li et al. [13] introduced a dual-decoder framework that decomposes the decision process into two sequential selection steps. More recently, Liu et al. [15] developed the 2D Array Pointer network (2D-Ptr), which jointly select (vehicle, node) pairs as a single action. However, these methods are essentially extensions of homogeneous routing solvers. By employing a single-agent framework that selects from the customer and vehicle spaces simultaneously, they suffer from a prohibitive expansion of the action space.

III. Background

A. Traveling Salesman Problem (TSP)

The TSP is a well-known combinatorial optimization problem. Formally, a TSP instance is defined on a graph with M city nodes $\mathcal{U} = \{1, ..., M\}$. The objective is to determine the optimal route r that minimizes the total distance while ensuring that each city is visited only once.

B. Heterogeneous Capacitated Vehicle Routing Problem

Formally, an HCVRP instance is defined by a set of customer nodes $\mathcal{U} = \{u_1, \ldots, u_N\}$ with corresponding demands $d(u_i)$ and a depot u_0 with $d(u_0) = 0$. The problem involves heterogeneous vehicles $\mathcal{V} = \{v_1, \ldots, v_M\}$, where each vehicle $v_m \in \mathcal{V}$ is characterized by two distinct attributes: its load

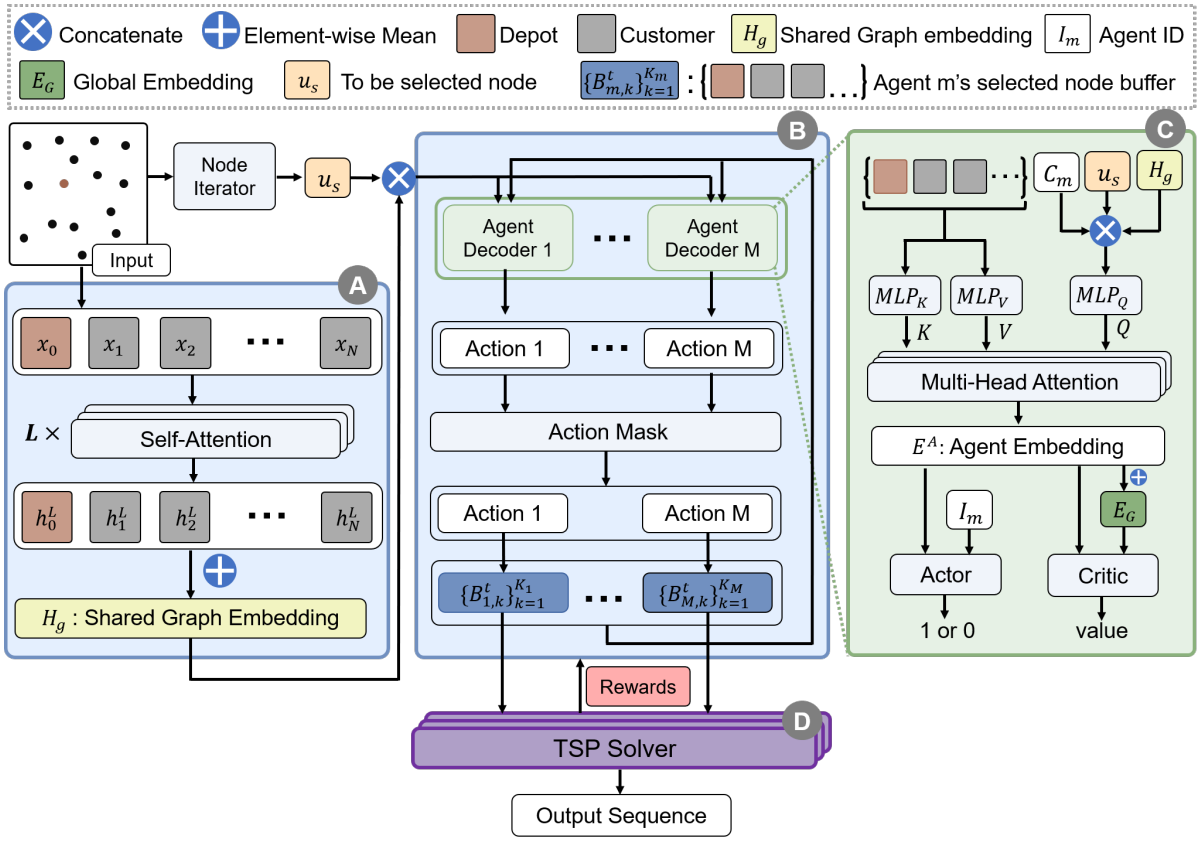

Fig. 2. The MAVR structure, where (A), (B), and (C) constitute the MARL framework for the Node Allocation task, and (D) performs the Node Traversal task. (A). Policy encoder of multi-agents. (B). Action outputs of multi-agents. (C). Policy decoder of multi-agents. (D). The TSP solver to feedback rewards.

capacity ρ_m and its travel speed χ_m. The objective is to determine a set of routes $\mathcal{R} = \{r_1, ..., r_M\}$ that minimizes the maximum travel time across all vehicles. Each customer node in $\mathcal{U}$ is visited exactly once. Let $\text{dist}(u^i_m, u^j_m)$ denote the Euclidean distance between two sequential nodes of vehicle m. The optimal solution $\mathcal{R}^*$ is defined as:

$$\mathcal{R}^* = \arg\min_{\mathcal{R}} \left(\max_{r_m \in \mathcal{R}} \sum_{j=1}^{|r_m|} \frac{\text{dist}(u_{r^{j-1}_m}, u_{r^j_m})}{\chi_m} \right) \quad (1)$$

We distinguish between a *trip* and a *route* of vehicles.

- A *trip* τ_m is defined as a sequence starting and ending at the depot u_0 that services a set of customer nodes, where the cumulative demand must not exceed its capacity ρ_m.
- A *route* r_m represents the entire travel for a specific vehicle v_m, which may encompass $K_m (K_m \geq 1)$ trips as $r_m = (\tau_{(m,1)}, \tau_{(m,2)}, \ldots, \tau_{(m,K_m)})$.

IV. Method

In this section, we introduce **M**ulti-**A**gent Reinforcement Learning for Heterogeneous **V**ehicle **R**outing (**MAVR**), which is illustrated in Figure 2. We believe that solving HCVRP involves completing two essential tasks: (1) Node Allocation: Assigning customer nodes to heterogeneous vehicles; (2) Node Traversal: Determining the shortest path for each vehicle based on its assigned nodes. These two tasks can be learned independently in stages, allowing us to convert the HCVRP into a two-stage learning process. For Task (2), Node Traversal essentially degenerates into a simpler TSP problem with fewer nodes. Since it's a universal task for all vehicles, a generic TSP solution can be pre-trained in advance. Therefore, the key to the HCVRP lies in learning the Node Allocation task.

Node allocation determines the set of nodes each vehicle will visit, influencing the trip length in task (2). Each vehicle must coordinate with one other to optimize the Node Allocation, aiming to minimize the maximum distance rather than their individual distances. Therefore we model each vehicle as an agent and take the Node Allocation as a *Cooperative Multi-Agent Reinforcement Learning* task, where collaboration among agents is essential for the joint objective.

However, there are two obstacles when utilizing MARL for Node Allocation. (1) Modeling Challenge: How to design MARL frameworks that promote effective agent cooperation, thereby mitigating the issue of large action spaces typically found in previous works. (2) Vehicle Trip Number Uncertainty: We analyze the theoretical requirements for the number of vehicle trips for an optimal solution and find that it cannot be determined in advance. We will address them sequentially.

A. Node Allocation: How to Assign Customer Nodes to Vehicles

To mitigate the inefficiency of using the joint vehicle and customer node as the action space, we introduce a new

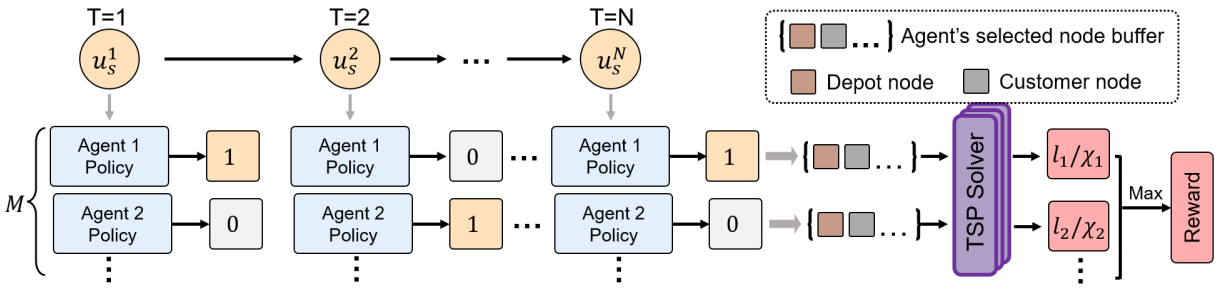

Fig. 3. The MDP pipeline of MARL. There are M agents that participate in the node's assignment.

modeling method. At each timestep, we designate a *to be selected node* u_s from the HCVRP and input its information into each agent. The agents then leverage their respective selected nodes to determine whether to select u_s. This iterative process is repeated until all nodes have been visited. Each vehicle that returns to the depot can restart for a new trip. The Multi-Agent Markov Decision Process is described below.

Observations. At the timestep t, the observation of agent m includes its selected node buffer B_m^t, the entire node information, and the node u_s^t.

Actions. At the timestep t, each agent takes their observations as input and takes a binary action $\{a_m^t\}$ for u_s^t : 1 for selection or 0 for rejection. It is possible that multiple agents may choose the same node u_s^t. To resolve this conflict, we implement an action mask that assigns u_s to the agent with the most deterministic policy distribution, i.e., $\arg\max_{m \in M} \pi_m(a_m = 1)$.

Transitions. After agents take actions at the timestep t, we add u_s^t to the corresponding agent m's selected node buffer $\{B_{m,k}^t\}_{k=1}^{K_m}$, which may contain K_m trips ($K_m \geq 1$). Then the node iterator selects the next node u_s^{t+1} from the neighboring nodes of u_s, When all customer nodes are selected, the episode ends. The number of execution steps T equals N.

Rewards. The length of trip k for vehicle m is given by $\tau_{m,k}^t = \mathrm{TSP}(B_{m,k}^t)$, where $\mathrm{TSP}(\cdot)$ is a function that calculates the shortest traversal path. And the route length for vehicle m is $l_m = \sum_{k=1}^{K_m} \tau_{m,k}^t$. Given that the agents cooperate to minimize the maximum travel time, the team reward is $r = -\max_m(l_m/\chi_m)$. The overall objective of MARL is:

$$\max \ \mathbb{E}_{\pi_1,\ldots,\pi_M}\left[\sum_{t=1}^{N} \gamma^t r_t\right] \tag{2}$$

The modeling design necessitates that agents select u_s based on their observations. To facilitate this, we present the policy structure illustrated in Figure 2. Each agent's policy consists of an encoder and a decoder. While the encoder is shared across all agents, the decoder is unique to each agent.

Policy Encoder. The node n of HCVRP has two features: demand $d(u_n)$ and 2-D location L_{u_n}. We concatenate these features to form initial node embeddings as $h_n^0 = W^h x_n + b^h, n \in [0, N]$, where $x_n = \mathrm{Concat}[L_{u_n}, d_{u_n}]$. After that, we adopt the Multi-Head Attention encoding structure [18]. Formally, the initial node embeddings h_n^0 are processed through L layers, detailed in Eq.(3), (4). After that, we average the node embeddings as the graph embedding $H_g = \sum_{n=0}^{N} h_n/(N+1)$.

$$\hat{h}_n = BN^l\left(h_n^{l-1} + MHA^l\left(h_0^{l-1}, h_1^{l-1}, ..., h_N^{l-1}\right)\right) \tag{3}$$

$$h_n^l = BN^l\left(\hat{h}_n + FF^l(\hat{h}_n)\right) \tag{4}$$

Policy Decoder. To take the binary action selection for node u_s, agent m is required to consider its current loading capacity ρ_m, its selected node buffer $\{B_{m,k}^t\}_{k=1}^{K_m}$, and the whole graph information H_g. They are integrated into a final agent embedding E_m^A by the Transformer's Attention mechanism [18]. Specifically, we concatenate ρ_m, u_s, and, H_g and then map them as a *Query*. We take buffer $\{B_{m,k}^t\}_{k=1}^{K_m}$'s corresponding node embeddings $\{h_u^L\}, u \in B_{m,k}^t$ and map them to *Keys* and *Values*. Then every agent utilizes the agent embedding E_m^A and agent n's ID i_n to output the action probability and actions $\{a_m\}_{m=1}^{M}$ by Actor function. Formally, the decoder processes are detailed in Eq. (5) to (7). For centralized training, we average the agents embedding as the global embedding $E_G^A = \sum_{m=1}^{M} E_m^A/M$ and take E_m^A and global E_G as Critic's input to obtain value functions.

$$Q_m = W_q^m[H_g, C_m, u_s] + b_q^m \tag{5}$$

$$K_m, V_m = W_k^m(h_v^L), W_v^m(h_u^L), v \in B_{m,k}^t \tag{6}$$

$$E_m^A = \mathrm{Attention}(Q_m, K_m, V_m) \tag{7}$$

$$\pi_n(a|i_m) = \mathrm{Actor}(E_m^A, i_m), V(i_m) = \mathrm{Critic}(E_m^A, E_G) \tag{8}$$

Our modeling method reduces the action space of each agent from joint vehicle and customer nodes to two, which standardizes the process across different HCVRP and eases the training difficulty in large-scale problems. Furthermore, as the node allocation skills developed for different HCVRP are universal, our strategic design enables the method to generalize effectively across problems of varying scales.

B. Node Allocation: How to optimize the Vehicle Trip

In this section, we highlight the crucial role of the vehicle trip number in solving the HCVRP. Consequently, it becomes challenging to determine a vehicle trip number for a type of HCVRP with a fixed number of nodes. To elucidate this, we first present the following theorem. The proofs and details for this section are provided in the Appendix.

Theorem 1: Consider an optimal solution $s \in S$ for an HCVRP. Let route $r_m = (\tau_{(m,1)}, \ldots, \tau_{(m,K_m)})$ denote the sequence of K_m trips performed by vehicle m, and and $C_{m,k}$

be the total demand served during trip $\tau_{m,k}$. For any two distinct trips $i, j \in \{1, \ldots, K_m\}$, $C_{(m,i)} + C_{(m,j)} > \rho_m$ holds, where ρ_m is the maximum loading capacity of vehicle m.

Theorem 1 implies that a route r is sub-optimal if there exist two distinct trips i and j such that $C_{m,i} + C_{m,j} \leq \rho_m$. In such cases, merging these two trips into a single one would yield a shorter total distance. Consequently, an excessive number of trips per vehicle results in redundant travel costs. To prevent it, we introduce an episodic team reward as defined in Eq. (9). Let $C^{\min}_{(m,1)}$ and $C^{\min}_{(m,2)}$ represent the two smallest loads among all trips performed by vehicle m. To penalize trip redundancy, we implement a feedback mechanism that discourages configurations where any two trips could potentially be merged.

$$r_{int} = \frac{1}{M} \sum_{m=1}^{M} \frac{-\mathrm{ReLU}(C^{\min}_{(m,1)} + C^{\min}_{(m,2)} - \rho_m)}{\chi_m} \tag{9}$$

C. Node Traversal: How to Guide MARL Training

The Node Traversal task aims to provide the team reward for MARL. Formally, we parameterize agents' policy as $\{\theta_m\}_{m=1}^{M}$ and average E_m^A as global embedding E_G and denote $s = \{E_m^A, E_G\}$ as a centralized embedding. We follow the MAPPO [22] and utilize a CTDE training paradigm. We let agents individually take actions and perform centralized training by using a centralized critic network $V_w(s_t)$. Agent m's objective $\mathcal{L}_{\pi_{\theta_m}}$ is as follows, where $\delta_t^m = r_t + \alpha r_{int} + \gamma V_w(s_{t+1}) - V_w(s_t)$.

$$\mathcal{L}_{\pi_{\theta_m}} = \mathbb{E}[\min\left(\eta_t^{\theta_m} A_t^m, \mathrm{clip}\left(\eta_t^{\theta_m}, 1-\epsilon, 1+\epsilon\right) A_t^m\right)] \tag{10}$$

$$\eta_t^{\theta_m} = \frac{\pi_{\theta_m}(a_t^m)}{\pi_{\theta_m^{old}}(a_t^m)}, \quad A_t^m = \sum_{l=0} (\gamma\lambda)^l \delta_{t+l}^m \tag{11}$$

V. Experiments

In this section, we empirically compare our MAVR with the DRL baselines on different HCVRP instances. In terms of computational resources, all experiments are conducted on GPU *RTX 3090* and CPU *Intel(R) Xeon(R) Gold 5220*.

Problem Setting For a fair comparison, we adopt the problem settings from [15]. Specifically, node locations L_m are sampled uniformly from a $[0, 1]^2$ unit square, and demands $d(u)$ are sampled from $\{1, \ldots, 9\}$. Vehicle capacities and speeds are sampled uniformly from $\{20, \ldots, 40\}$ and $[0.5, 1.0]$, respectively. We evaluate MAVR on 12 problem instances denoted as VM-UN, where $M \in \{3, 5, 7\}$ and $N \in \{40, 60, 80, 100\}$. Each test set contains 1,280 instances.

Implement Details of MAVR. We pre-train an AM-based TSP-20 agent [10] for 2M timesteps to provide reward feedback for our MARL policy. To optimize the node iteration sequence and prevent large distances between consecutive nodes in a random order, we employ the same agent as the node iterator. It ensures that the model processes spatially proximate nodes sequentially. Following [10], [11], the TSP solver in MAVR can be executed multiple times to improve solution quality. We evaluate three inference strategies: MAVR-Greedy (MAVR-G), which generates a single trajectory per vehicle; MAVR-Multi-trajectory (MAVR-M), which samples 1280 trajectories to select the best candidate; and MAVR-LKH3 (MAVR-L), which combines 1280 trajectories with the LKH-3 solver for the TSP tasks.

a) Baselines : We evaluate our method against four DRL-based baselines AM [10], ET [23], DRL$_{Li}$ [13], and 2D-Ptr [15] and three heuristic algorithms GA [24], SA [25], and SISR [26]. To adapt ET and AM for HCVRP, we follow 2D-Ptr and modify their input layers and decoder masks to incorporate heterogeneous vehicle features. Additionally, since the original AM is a single-vehicle solver, we implement a turn-taking decision mechanism following 2D-Ptr [15].

A. Performance Comparisons

Table IV-B presents the main results for uniformly distributed HCVRP instances, including average distances and inference times. While single-agent DRL methods perform competitively on medium-sized instances (U40 and U60), their performance drops on larger scales (U80 and U100). In contrast, MAVR consistently achieves superior performance across all scales with reasonable computational overhead. This robust scalability stems from our two-stage decoupling design. Furthermore, given MAVR's reliance on a fixed TSP solver, these results highlight the specific effectiveness of our MARL framework in optimizing node allocation.

B. Strengths of MAVR

MAVR exhibits robust zero-shot generalization capabilities. Since the node allocation task remains is universal across varying vehicle fleets and customer nodes, MAVR generalizes effectively to unseen HCVRP instances. To validate this, we deploy MAVR models trained on V3-U100 and V7-U40 seperately, and directly test them to instances with different vehicle and node scales, as illustrated in Figure IV-C. Compared to 2D-Ptr which suffers from exponentially expanding action space but has generalization performance, MVAR leverages decoupled policies for node allocation and traversal. Consequently, MAVR maintains high efficiency and stability across diverse problem scales.

MAVR optimizes vehicle trip allocation for enhanced efficiency. To evaluate this, we conduct experiments across three independent seeds and monitor the evolution of practical vehicle trip numbers during the early training phase, as illustrated in Figure IV-C. The results show that the required trip count across various HCVRP instances gradually converges to a lower bound. This trend demonstrates the emergence of a cooperative policy within our MARL framework: agents learn to coordinate their routes, effectively eliminating redundant trips in accordance with Theorem 1. Consequently, MAVR can flexibly adapt to varying trip numbers across different HCVRP, allowing multi-agents to travel only the necessary trips.

MAVR performs better in Node Allocation. To quantify this, we define two vehicle trips as redundant if their combined load is less than ρ_m. Following Theorem 1, the frequency of such redundant trips serves as a critical metric for evaluating node allocation quality, which we define as the Redundant-Trip-Ratio (RTR). We monitor the RTR of MAVR and AM

TABLE I
PERFORMANCE COMPARISON. THE NOTATION (G.), (S.1280) AND (S.12800) REFER TO GREEDY, SAMPLE1280 AND SAMPLE12800, RESPECTIVELY.

	Methods	U40			U60			U80			U100		
		obj. ↓	gap(%)	time	obj. ↓	gap(%)	time	obj. ↓	gap(%)	time	obj. ↓	gap(%)	time
V3	SISR	4.77	0.00	5m	6.57	0.00	6m	8.52	0.00	13m	10.29	0.00	18m
	GA	6.35	33.22	5m	9.21	40.25	7m	12.32	44.68	11m	15.33	48.93	14m
	SA	5.07	6.27	4m	7.04	7.16	7m	9.17	7.60	9m	11.13	8.13	13m
	AM (g.)	6.32	32.57	<1s	8.49	29.20	<1s	10.81	26.93	<1s	12.68	23.23	<1s
	AM (s.1280)	5.48	14.93	<1s	7.62	16.07	<1s	9.92	16.45	<1s	11.82	14.82	<1s
	ET (g.)	5.60	17.35	<1s	7.58	15.46	<1s	9.76	14.61	<1s	11.74	14.08	<1s
	ET (s.1280)	5.20	9.01	<1s	7.14	8.64	<1s	9.19	7.88	<1s	11.20	8.84	2s
	DRLL$_i$ (g.)	5.47	14.66	<1s	7.43	13.07	<1s	9.64	13.22	<1s	11.44	11.13	<1s
	DRLL$_i$ (s.1280)	5.09	6.79	<1s	6.97	6.12	2s	9.10	6.78	2s	10.90	5.94	2s
	2D-Ptr (g.)	5.29	10.92	<1s	7.20	9.63	<1s	9.24	8.53	<1s	11.12	8.04	<1s
	2D-Ptr (s.1280)	4.96	3.91	<1s	6.82	3.87	<1s	8.85	3.86	<1s	10.71	4.11	<1s
	2D-Ptr (s.12800)	4.92	3.09	<1s	6.77	3.08	2s	8.78	3.13	2s	10.65	3.44	4s
	MAVR-G (g.)	5.20	9.01	<1s	7.13	8.52	<1s	9.02	5.68	<1s	10.93	6.21	<1s
	MAVR-M (s.1280)	4.89	2.51	<1s	6.76	2.89	<1s	8.76	2.81	<1s	10.52	2.23	3s
	MAVR-L (s.1280)	4.84	1.46	<1s	6.73	2.43	2s	8.73	2.46	2s	10.46	1.65	5s
V5	SISR	2.94	0.00	5m	4.00	0.00	9m	5.10	0.00	13m	6.17	0.00	18m
	GA	4.88	66.10	6m	6.89	72.29	10m	8.95	75.55	14m	10.93	77.24	18m
	SA	3.21	9.18	6m	4.39	9.74	9m	5.61	10.02	13m	6.80	10.20	16m
	AM (g.)	4.15	41.15	<1s	5.51	37.76	<1s	6.87	34.84	<1s	8.10	31.42	<1s
	AM (s.1280)	3.54	20.66	<1s	4.82	20.69	<1s	6.19	21.46	<1s	7.45	20.75	<1s
	ET (g.)	3.57	21.45	<1s	4.76	19.15	<1s	6.01	17.75	<1s	7.25	17.51	<1s
	ET (s.1280)	3.29	11.98	<1s	4.46	11.59	<1s	5.64	10.65	<1s	6.85	11.02	<1s
	DRLL$_i$ (s.1280)	3.24	10.28	<1s	4.34	8.47	<1s	5.54	8.64	2s	6.65	7.76	2s
	2D-Ptr (g.)	3.34	13.58	<1s	4.48	12.01	<1s	5.65	10.71	<1s	6.75	9.40	<1s
	2D-Ptr (s.1280)	3.09	5.27	<1s	4.20	5.01	<1s	5.36	5.11	<1s	6.46	4.73	<1s
	2D-Ptr (s.12800)	3.06	4.26	2s	4.16	4.02	2s	5.32	4.21	3s	6.41	3.92	4s
	MAVR-G (g.)	3.27	11.22	<1s	4.42	10.49	<1s	5.60	9.80	<1s	6.50	5.34	<1s
	MAVR-M (s.1280)	3.04	3.40	<1s	4.18	4.49	<1s	5.35	4.90	<1s	6.35	2.91	2s
	MAVR-L (s.1280)	3.02	2.72	<1s	4.17	4.24	2s	5.32	4.31	3s	6.33	2.59	4s
V7	SISR	2.29	0.00	5m	2.91	0.00	9m	3.69	0.00	13m	4.45	0.00	20m
	GA	4.35	90.36	8m	5.98	105.29	12m	7.58	105.30	17m	9.10	104.61	23m
	SA	2.50	9.24	7m	3.30	13.38	11m	4.17	13.09	15m	5.01	12.58	20m
	AM (g.)	3.15	37.79	<1s	4.15	42.45	<1s	5.18	40.54	<1s	6.13	37.81	<1s
	AM (s.1280)	2.66	16.29	<1s	3.63	24.68	<1s	4.64	25.80	<1s	5.58	25.46	<1s
	ET (g.)	2.69	17.69	<1s	3.58	22.93	<1s	4.43	19.98	<1s	5.23	17.59	<1s
	ET (s.1280)	2.46	7.65	<1s	3.33	14.49	<1s	4.17	12.88	<1s	4.98	11.99	<1s
	DRLL$_i$ (g.)	2.67	16.81	<1s	3.60	23.49	<1s	4.52	22.46	<1s	5.38	20.87	<1s
	DRLL$_i$ (s.1280)	2.43	6.06	<1s	3.25	11.59	2s	4.13	11.94	2s	4.98	11.82	3s
	2D-Ptr (g.)	2.50	9.33	<1s	3.31	13.81	<1s	4.14	12.26	<1s	4.92	10.49	<1s
	2D-Ptr (s.1280)	2.35	2.73	<1s	3.09	6.01	<1s	3.90	5.65	<1s	4.68	5.22	<1s
	2D-Ptr (s.12800)	2.33	2.02	2s	3.05	4.86	2s	3.86	4.58	3.43	4.64	4.30	5s
	MAVR-G (g.)	2.48	8.29	<1s	3.29	13.05	<1s	4.13	11.92	<1s	4.72	6.06	<1s
	MAVR-M (s.1280)	2.34	2.18	2s	3.18	9.27	2s	3.85	4.33	2s	4.61	3.59	3s
	MAVR-L (s.1280)	2.32	1.31	2s	3.17	8.93	3s	3.84	4.06	3s	4.58	2.91	4s

[10] during the training phase, as shown in Figure IV-C. The results indicate a strong positive correlation between solution quality and a lower RTR. Notably, while AM struggles to mitigate redundant trip occurrences, MAVR rapidly converges to a coordinated team policy that minimizes such cases. This suggests that MAVR not only effectively regulates the total number of trips but also optimizes the distribution of customer nodes among vehicles with high efficiency.

VI. CONCLUSIONS

In this work, we decouple HCVRP into 2 tasks: node allocation and node traversal. Since the latter is essentially an easier TSP task, we utilize it to provide rewards to train our proposed MARL for Node Allocation. In detail, we make a subtle design where each vehicle agent takes each node as input to determine whether to select it. Besides, we theoretically analyze the redundant vehicle trip number's effect on HCVRP's node allocation. Based on that we realize a dynamic learning process by which the vehicle trip number can be gradually narrowed down to a reasonable one. We conducted experiments in different size HCVRP within a limited training timestep. The efficiency outperforms the existing constructive DRL-based baselines. We will make our code available upon acceptance.

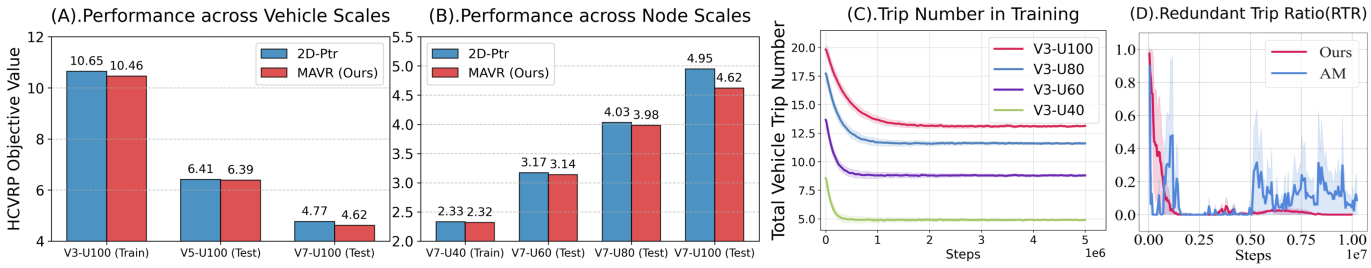

Fig. 4. (A),(B): The zero-shot performance across vehicles and nodes. (C): Vehicle Trip Number. (D): Redundant Trip Ratio.

REFERENCES

[1] R. Baldacci, M. Battarra, and D. Vigo, "Routing a heterogeneous fleet of vehicles," in *The vehicle routing problem: latest advances and new challenges*. Springer, 2008, pp. 3–27.

[2] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte, "Thirty years of heterogeneous vehicle routing," *European Journal of Operational Research*, vol. 249, no. 1, pp. 1–21, 2016.

[3] J. K. Lenstra and A. R. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[4] N. Vesselinova, R. Steinert, D. F. Perez-Ramirez, and M. Boman, "Learning combinatorial optimization on graphs: A survey with applications to networking," *IEEE Access*, vol. 8, pp. 120 388–120 416, 2020.

[5] R. Baldacci and A. Mingozzi, "A unified exact method for solving different classes of vehicle routing problems," *Mathematical Programming*, vol. 120, no. 2, pp. 347–380, 2009.

[6] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, "A generic exact solver for vehicle routing and related problems," *Mathematical Programming*, vol. 183, no. 1, pp. 483–523, 2020.

[7] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & operations research*, vol. 34, no. 8, pp. 2403–2435, 2007.

[8] F. Y. Vincent, P. Jewpanya, A. P. Redi, and Y.-C. Tsao, "Adaptive neighborhood simulated annealing for the heterogeneous fleet vehicle routing problem with multiple cross-docks," *Computers & operations research*, vol. 129, p. 105205, 2021.

[9] C. Prins, "Two memetic algorithms for heterogeneous fleet vehicle routing problems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 6, pp. 916–928, 2009.

[10] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" *arXiv preprint arXiv:1803.08475*, 2018.

[11] Y.-D. Kwon, J. Choo, B. Kim, I. Yoon, Y. Gwon, and S. Min, "Pomo: Policy optimization with multiple optima for reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 188–21 198, 2020.

[12] L. Xin, W. Song, Z. Cao, and J. Zhang, "Multi-decoder attention model with embedding glimpse for solving vehicle routing problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 12 042–12 049.

[13] M. LI JW, R. GAO *et al.*, "Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13 572–13 585, 2022.

[14] Y. Xu, M. Fang, L. Chen, G. Xu, Y. Du, and C. Zhang, "Reinforcement learning with multiple relational attention for solving vehicle routing problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 11 107–11 120, 2021.

[15] Q. Liu, C. Liu, S. Niu, C. Long, J. Zhang, and M. Xu, "2d-ptr: 2d array pointer network for solving the heterogeneous capacitated vehicle routing problem," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 2024, pp. 1238–1246.

[16] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[17] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] M. Kim, J. Park, and J. Park, "Sym-nco: Leveraging symmetricity for neural combinatorial optimization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1936–1949, 2022.

[20] J. M. Vera and A. G. Abad, "Deep reinforcement learning for routing a heterogeneous fleet of vehicles," in *2019 IEEE Latin American conference on computational intelligence (LA-CCI)*. IEEE, 2019, pp. 1–6.

[21] W. Qin, Z. Zhuang, Z. Huang, and H. Huang, "A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem," *Computers & Industrial Engineering*, vol. 156, p. 107252, 2021.

[22] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[23] J. Son, M. Kim, S. Choi, H. Kim, and J. Park, "Solving np-hard min-max routing problems as sequential generation with equity context," in *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023.

[24] S. Karakatič and V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem," *Applied Soft Computing*, vol. 27, pp. 519–532, 2015.

[25] İ. İlhan, "An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem," *Swarm and Evolutionary Computation*, vol. 64, p. 100911, 2021.

[26] J. Christiaens and G. Vanden Berghe, "Slack induction by string removals for vehicle routing problems," *Transportation Science*, vol. 54, no. 2, pp. 417–433, 2020.