MAPDP :
Cooperative Multi-Agent Reinforcement Learning to Solve Pickup and Delivery Problems

Wangxianyi

LZU

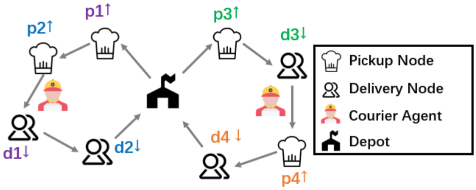
2024 年 5 月 14 日

Background Introduction

- Vehicle Routing Problem (VRP) is crucial in various real-world applications such as express systems, industrial warehousing, and on-demand delivery.
- Cooperative Pickup and Delivery Problem (PDP) is a variant of VRP that plays a significant role in applications like on-demand delivery and industrial logistics.
- Challenges in solving cooperative PDP include structural dependency between pickup and delivery pairs and the need for effective cooperation among different vehicles.

Mathematical Modeling of Cooperative PDP

$$\min \sum_{k=1}^{K} \sum_{i=0}^{2N} \sum_{j=1}^{2N+1} e_{ij} x_{ijk} \quad (1)$$

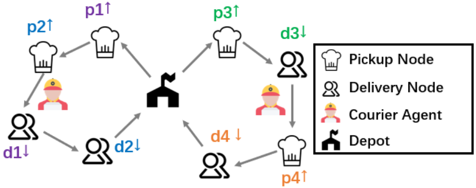


- $x_{ijk} \in \{0,1\}$: whether the vehicle k travels directly from node v_i to node v_j.
- e_{ij}: spatial distances.

$$\sum_{k=1}^{K} \sum_{j=1}^{2N+1} x_{ijk} = 1, \forall i \in [0, 2N], \quad (2)$$

$$\sum_{k=1}^{K} \sum_{i=0}^{2N} x_{ijk} = 1, \forall j \in [1, 2N+1], \quad (3)$$

$$\sum_{i \in S'} d_i \leq C_k, \forall S' \subseteq S, \forall k \in [1, K], \quad (4)$$

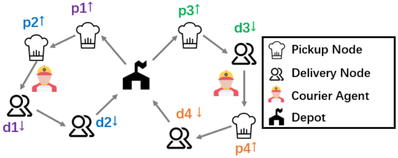


- d_i: Each pickup order has a demand volume.
- C_k: Capacity of the k-th vehicle.
- S: A consecutive routing sequence from v_0 and ends at v_{2N+1}.

$$\sum_{j=1}^{2N+1} x_{i,jk} = \sum_{j=0}^{2N+1} x_{i+N,jk}, \forall k \in [1, K], i \in [1, N], \quad (5)$$

$$T_i \leq T_{i+N}, \forall i \in [1, N] \quad (6)$$

- State: At step t, agent k's state includes its remaining capacity C_k^t and current trajectory S_k^t. The current location, the last visited node, is denoted by $v_{I_k^t}$, where I_k^t is the node index.
- Action: The action at step t for vehicle agent k is to determine a node as its next target, represented as $v(k, t)$.

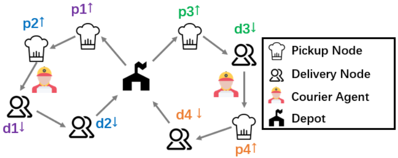


[a]

[a] All vehicles can communicate centrally, ensuring full observability in the cooperative PDP setting.

Explanation of Transition, Reward

- Transition: For each agent: $S_k^{t+1} = (S_k^t; \{v_{I_k^t}\})$, $C_k^{t+1} = C_k^t - d_{I_k^t}$, where ; means concatenating the partial solution with the new selected node.
- Reward: Minimize the total travel distance. At each step, the reward $r_k^t = -e_{I_k^t}$ is the negative length of the newly traveled arc, and the final episode reward $R = \sum_{k=1}^{k=K} \sum_{t=0}^{T-1} r_k^t$ [1]is the sum of all individual rewards r_k^t.



---

where T is the decision step amount in a complete episode

Overview of MAPDP Framework

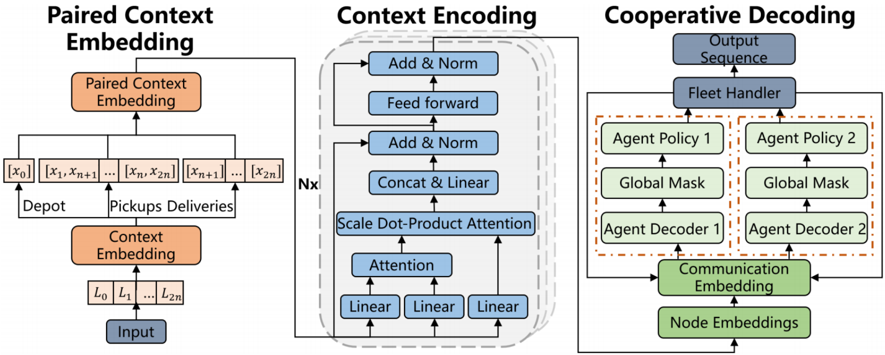
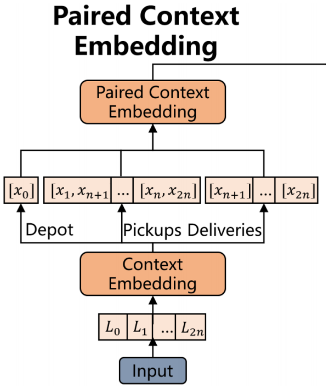


图 1: MAPDP Framework

Paired Context Embedding

- $\mathcal{L}_i$: Original 2-D location information.
- $x_i = W^x[\mathcal{L}_i, d_i] + b^x$: Concatenat the two features and map them into one dense vector.

$$h_i^0 = \begin{cases} W_0^x x_i + b_0^x, & i = 0, \\ W_p^x [x_i; x_{i+N}] + b_p^x, & 1 \leq i \leq N, \\ W_d^x x_i + b_d^x, & N+1 \leq i \leq 2N, \end{cases} \tag{7}$$



Context Encoding

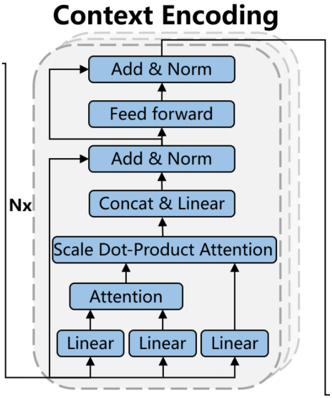
The initial paired context embedding h_i^0 is processed through L attention layers

- Multi-head attention layer (MHA).
- Skipconnection layer (He et al. 2016).
- Feed-forward (FF) layer.
- Batch normalization (BN) layers (Ioffe and Szegedy 2015).

$$Q_i^h, K_i^h, V_i^h = W_Q^h h_i, W_K^h h_i, W_V^h h_i, \tag{8}$$

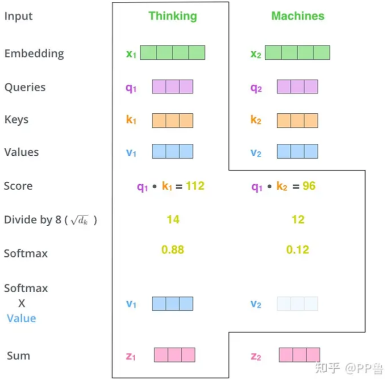
$$A_i^h = softmax(Q_i^h {K^h}^T / \sqrt{d_k}) V_j^h, \tag{9}$$

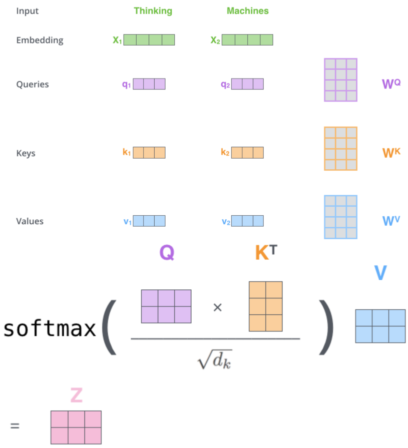
$$MHA_i = Concat(A_i^1, A_i^2, ..., A_i^H) W_O, \tag{10}$$



Attention

$$Q_i^h, K_i^h, V_i^h = W_Q^h h_i, W_K^h h_i, W_V^h h_i,$$
$$A_i^h = softmax(Q_i^h {K^h}^T / \sqrt{d_k}) V_j^h$$





MHA

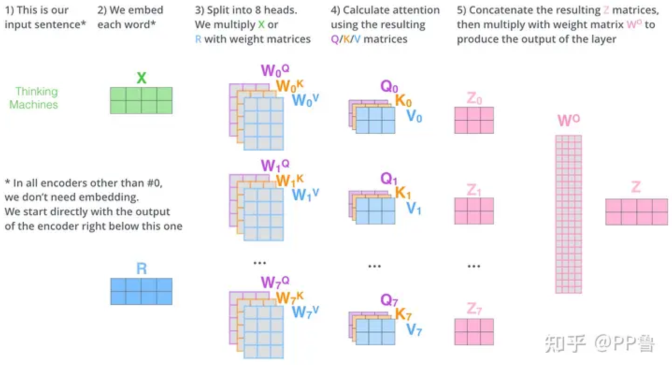


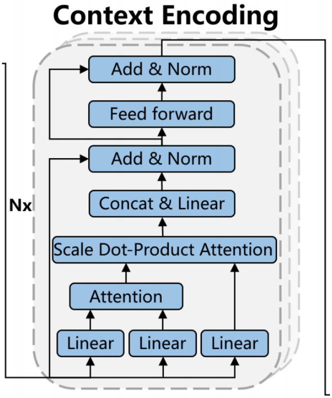
图 2: MAPDP Framework

$$MHA_i = Concat(A_i^1, A_i^2, ..., A_i^H)W_O$$

Context Encoding

$$\hat{h}_i = BN^{\ell}(h_i^{\ell-1} + MHA_i^{\ell}(h_1^{\ell-1}, h_2^{\ell-1}, \cdots h_{2N}^{\ell-1})), \tag{11}$$

$$h_i^{\ell} = BN^{\ell}(\hat{h}_i + FF^{\ell}(\hat{h}_i)). \tag{12}$$

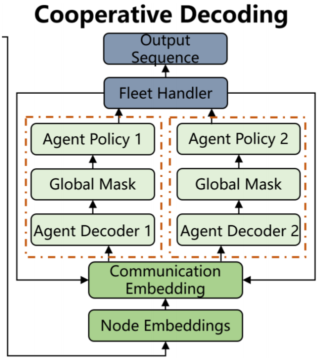


Cooperative Multi-Agent Decoders

A communication layer to record the updated states of different agents as follows:

$$Comm^t = [h_{I_1^t}; C_1^t; h_{I_2^t}; C_2^t; \ldots; h_{I_K^t}; C_K^t] \quad (13)$$

- $h_{k,(c)}^t = [\overline{h}\,^2; h_{I_k^t}; C_k^t; Comm^t]$: Agent k concatenates essential information for decision-making, including global static representation, its current state, and others'.
- $v_{I_k^t}$: Agent k selects the next node to visit at step t.



[2] $\overline{h} = \frac{1}{2N}\sum_{i=0}^{2N} h_i$: The average of all nodes

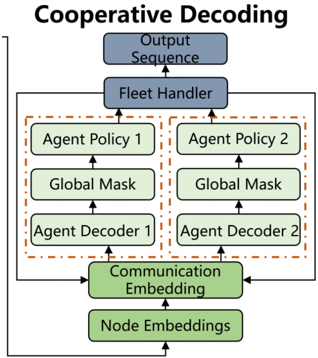
Cooperative Multi-Agent Decoders

$$g_k^t = MHA_{k,(c)}(h_1, h_2, ..., h_{2N}), \tag{14}$$

$$Q_k^t, K_{k,i}^t = W_{Q,k} g_k^t, W_{K,k} h_i, \tag{15}$$

$$u_{k,i}^t = Dtanh\left(\frac{{Q_k^t}^T K_{k,i}^t}{\sqrt{d_k}}\right), \tag{16}$$

$$p_{\theta_k,\phi}(v(k,t)) = softmax\left(Mask^t(u_{k,i}^t)\right) \tag{17}$$



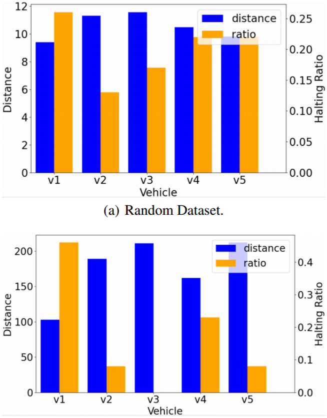
- $W_{Q,k}$ and $W_{K,k}$ are the weight matrices of the last single-head attention
- D=10 is the clip rate for better exploration (Bello et al. 2016).
- Fleet handler: Randomly maintains the action of one agent from all candidates to the node and keeps the others stay at their current location $v_{l_k^t}$.

Performance Comparison with Other Methods

Model	Random Dataset								
	2N = 20, K=2			2N = 50, K=5			2N = 100, K=10		
	Cost	Gap	Time	Cost	Gap	Time	Cost	Gap	Time
ACO (Gambardella, Taillard, and Agazzi 1999)	34.73	39.60%	6min	79.94	52.01%	32min	136.89	53.86%	51min
Tabu Search (Glover 1990)	29.76	19.67%	7min	64.57	22.78%	34min	112.38	26.31%	51min
OR-Tools (Google 2021)	25.91	4.18%	4min	54.64	3.90%	31min	94.25	5.93%	49min
RL-VRP (Nazari et al. 2018)	26.79	7.72%	1s	63.12	20.02%	5s	101.13	13.67%	9s
AM-VRP (Kool, van Hoof, and Welling 2019)	26.64	7.12.%	1s	67.41	28.18%	4s	105.91	19.04%	8s
MDAM (Xin et al. 2021)	25.98	4.46%	8s	67.24	27.86%	25s	105.11	18.14%	51s
MAPDP	**24.87**	**0.00%**	1s	**52.59**	**0.00%**	4s	**88.97**	**0.00%**	7s
Model	Real-World Dataset								
	2N = 20, K=2			2N = 50, K=5			2N = 100, K=10		
	Cost	Gap	Time	Cost	Gap	Time	Cost	Gap	Time
ACO (Gambardella, Taillard, and Agazzi 1999)	812	30.13%	6min	1205	35.39%	34min	2054	20.47%	53min
Tabu Search (Glover 1990)	834	33.65%	6min	1197	34.49%	34min	2033	19.24%	51min
OR-Tools (Google 2021)	749	20.03%	4min	1056	18.65%	31min	1811	6.22%	50min
RL-VRP (Nazari et al. 2018)	714	14.42%	1s	1130	26.97%	5s	1842	8.04%	9s
AM-VRP (Kool, van Hoof, and Welling 2019)	661	5.93%	1s	942	5.84%	4s	1759	3.17%	9s
MDAM (Xin et al. 2021)	638	2.24%	8s	941	5.73%	25s	1733	1.64%	52s
MAPDP	**624**	**0.00%**	1s	**890**	**0.00%**	4s	**1705**	**0.00%**	7s

图 3: Comparison of Different Models on Random and Real-World Datasets

Case Studies on Vehicle Cooperation



(a) Random Dataset.

(b) Real-World Dataset.

图 4: Case studies on vehicle cooperation analysis from two datasets.

- MAPDP-SP: The simplified model where all agent decoders share the same parameters.(Heterogeneous training can further slightly improve its effectiveness based on pure parameter sharing.)
- MAPDP-NC: The multi-agent framework without consideration on the communication embedding.(In a fully cooperative scenario, up-to-date communication with other agents is critical to effective coordination.)

Dataset	Model	2N=20	2N=50	2N=100
Random	MAPDP	24.87	52.59	88.97
	MAPDP-SP	24.99	53.61	89.78
	MAPDP-NC	26.89	68.78	108.12
Real	MAPDP	624	890	1705
	MAPDP-SP	639	943	1721
	MAPDP-NC	731	1033	1896

图 5: Case studies on vehicle cooperation analysis from two datasets.

Conclusion

- The proposed MAPDP framework leverages Multi-Agent Reinforcement Learning (MARL) to effectively solve the Cooperative Pickup and Delivery Problem (PDP) by capturing dependencies and promoting cooperation among multiple vehicles.
- MAPDP outperforms existing baselines by at least 1.64
- The centralized MARL framework, paired context embedding, cooperative decoders, and cooperative A2C algorithm collectively contribute to the success of MAPDP in addressing the challenges of PDP.
- Future research directions may include exploring scalability of MAPDP to larger problem instances, incorporating real-time constraints, and adapting the framework to dynamic environments.

Thanks!