A Deep Reinforcement Learning Method for Solving the Multi-depot Vehicle Routing Problem

Haixin Xu, Rong Hu(✉), Bin Qian, Ziqi Zhang, Qingxia Shang, Huaiping Jin

School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China
ronghu@vip.163.com

Abstract. In this paper, a deep reinforcement learning optimization algorithm combined with the clustering decomposition strategy (DRLA_CD) is proposed for solving the multi-depot vehicle routing problem (MDVRP). First, taking into account the NP-hard and strong coupling characteristics of MDVRP, an improved K-means algorithm (IKA) is designed to decompose MDVRP into several single-depot vehicle routing subproblems, thereby rationally reducing the search space and improving the search efficiency of the algorithm. Second, the deconstructed subproblems are solved using the deep reinforcement learning technique, and then the obtained solutions of subproblems are combined to form the whole solution of MDVRP. Finally, to confirm the efficacy of the proposed DRLA_CD, the comparative and simulation tests are carried out on instances with different scales.

Keywords: Mult-depot vehicle routing problem, Deep reinforcement learning, Cluster of decomposition, Attention mechanism.

1 Introduction

The vehicle routing problem (VRP), which was proposed by Ramser and Dantzing in 1959[1], has become a research hotspot in the field of modern operations research and is widely applied in the logistics and distribution industry. With the rapid development of the logistics industry, the traditional single-depot distribution model is no longer able to meet the increasingly diverse needs of customers, and it also affects the economic benefits of logistics companies. Therefore, Gillett et al.[2] proposed the Multi-depot Vehicle Routing Problem (MDVRP), which involves two decision steps: allocating customers to depots and planning the routes from depots to customers. Clearly, MDVRP is a more complex variant of VRP and belongs to an NP-hard problem. MDVRP is crucial in various transportation processes like cargo distribution, waste collection, and industrial manufacturing. Thus, researching its modeling and solution algorithms holds both theoretical and practical importance.

For MDVRP, some scholars have adopted exact algorithms for solving it. For example, Baldacci et al.[3] designed a column generation algorithm to solve MDVRP. In addition, Bettinelli et al.[4] utilized a branch-and-cut-and-price algorithm to solve the multi-depot heterogeneous vehicle routing problem with time windows. However, the computational complexity involved in solving NP-hard problems like MDVRP using exact algorithms is extremely high, and the solution quality is not satisfactory. Therefore, most studies resort to heuristic algorithms for solving such problems. Cordeau et al.[5] proposed a tabu search heuristic algorithm to solve the MDVRP and periodic VRP. Vidal et al.[6] proposed an efficient hybrid genetic algorithm to solve the MDVRP, periodic VRP, and multi-depot periodic VRP with capacitated vehicles and constrained route duration. Oliveira et al.[7] utilized clustering algorithms to decompose MDVRP into multiple single-depot VRPs, enabling the use of a cooperative co-evolutionary algorithm for solving. Hu Rong et al.[8] proposed an enhanced ant colony optimization based on clustering decomposition to solve the low-energy-consumption multi-depots heterogeneous-fleet vehicle routing problem with time windows. Sadati et al.[9] proposed a variable neighborhood tabu search algorithm, which can solve MDVRP as well as MDVRP with time windows and multi-depot open VRP. However, heuristic algorithms are iterative search-based optimization algorithms, and when dealing with large-scale problems, a significant amount of iterative searching can still result in substantial computational time consumption.

Deep reinforcement learning (DRL), a key branch of deep learning, has shown promise in solving NP-hard problems. Inspired by the success of AlphaGo Zero[10] in mastering Go and Atari[11] games, researchers have applied end-to-end DRL algorithms to tackle classic NP-hard problems like the Traveling Salesman Problem (TSP) and the VRP[12]. This approach utilizes a trained deep neural network (DNN) to directly output solutions without iterative search, resulting in fast solving speeds and strong generalization capabilities across problem instances with similar distribution characteristics.

Vinyals et al.[13] proposed a pointer network (PN) model for solving TSP, which for the first time applied deep learning to combinatorial optimization problems in an end-to-end manner. PN uses supervised learning for training, but this approach requires a large number of optimal solutions as the training label, which is difficult and time-consuming for NP-hard combinatorial optimization problems. Therefore, Bello et al.[14] proposed using an actor-critic reinforcement learning algorithm[15] instead of supervised learning to train PN. Nazari et al.[16] considered that the solution to the problem is independent of the order of input nodes, so they replaced the long short-term memory[17] of the input layer of the PN encoder with a simple linear embedding layer. This model can also be used to solve VRP with dynamic features. Kool et al.[12] improved the traditional PN model by borrowing from the transformer model[18]. The encoder of the model adopts the same structure as the transformer model, while the decoder considers the global graph embedding information, the decision made in the previous step, and the remaining capacity of the vehicle. In addition, some studies have utilized DRL to improve traditional heuristic algorithms. Li et al.[19] proposed a learning-based algorithm that divides large-scale problems into smaller, more easily solvable subproblems. Xin et al.[20] proposed the NeuroLKH algorithm for solving various routing problems including VRP, which combines deep learning with the traditional heuristic algorithm LKH. Kim et al.[21] proposed a learning-based heuristic algorithm that uses graph neural networks to predict the search results of the heuristic algorithm and uses the predicted results to guide the selection of sub-paths to exchange.

In summary, current research on MDVRP primarily focuses on heuristic algorithms, with most studies tending to encode and solve the problem as a whole. Inspired by the studies above, this paper proposes a DRLA_CD to solve the MDVRP, considering the complex solution space and the interdependence between the two stages. Firstly, an IKA algorithm is designed, which consists of K-means clustering and customer group assignment. This algorithm can effectively decompose the MDVRP into multiple VRPs. Secondly, the trained deep neural network (DNN) is used to directly output solutions for each decomposed sub-problem. These solutions are then combined to obtain the solution for the original problem. This approach significantly reduces the search area in the solution space, enabling the rapid solution of MDVRP. Finally, through simulation experiments and algorithm comparisons on different scale test sets, the effectiveness of the proposed DRLA_CD algorithm for solving MDVRP has been validated.

2 Problem Description & Modeling

2.1 Problem Description

The description of MDVRP is as follows: Given m depots and n customers, each depot is assigned a certain number of vehicles. The vehicles depart from the depots to deliver goods to customers, complete the service, and then return to the depots. The objective is to plan the optimal delivery routes to minimize the total transportation cost. As shown in Fig. 1, this problem can be represented by a directed graph $G=(V,E)$, where $V=\{D,C\}$ is the set of nodes, including m depot nodes and n customer nodes. $E=\{(i,j)\mid i,j\in V, i\neq j\}$ is the set of all edges. For the convenience of analysis and research, the following assumptions are made for this problem:

(1) The coordinates of all depots and customers, as well as the demands of customers, are known.

(2) Each customer node is exclusively served by one depot, and the demand of each customer can be fulfilled in a single service without the need for multiple services.

(3) Vehicles depart from a depot, and after completing the service, they need to return to the original depot.

(4) All customer demands are less than the maximum load capacity of the vehicle.

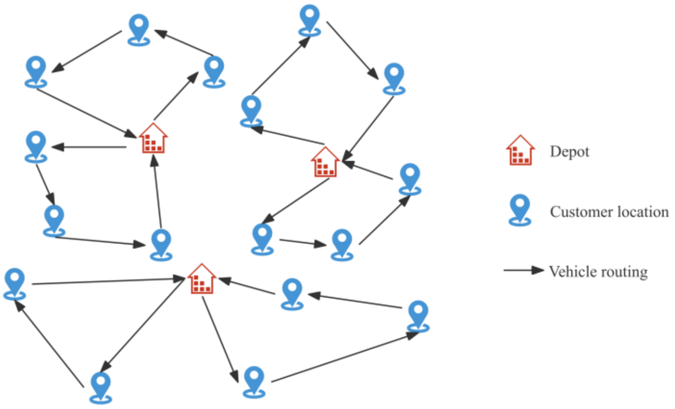

Fig. 1. Schematic diagram of MDVRP

2.2 Symbol Definition and Mathematical Model

The definitions of the relevant symbols involved in this article are shown in Table 1.

Table 1. Notations applied in the model of MDVRP

Symbol	Constants
D	Depot node set $D=\{d_1,d_2,...,d_m\}$
C	Customer node set $C=\{c_1,c_2,...,c_n\}$
U	Vehicle set $U=\{u_1,u_2,...,u_m\}$,Where u_1 represents the set of vehicles in depot 1
Q	Maximum vehicle capacity
r_i	Demand of customer $i\in C$
G_u	Maximum number of vehicles
d_{ij}	The Euclidean distance between i and j
x_{iju}	Decision variable: 1 if vehicle $u\in U$ travels from customer i to j , otherwise 0
y_{ij}	Decision variable: 1 if customer $i\in C$ is served by depot $j\in D$, otherwise 0
z_{iu}	Decision variable: 1 if vehicle $u\in U$ belongs to depot $i\in D$, otherwise 0

Based on the above description, assumptions, and definitions, the mathematical model of MDVRP is as follows:

$$\min \sum_{i\in D}\sum_{j\in C}\sum_{u\in U} d_{ij}x_{iju} \tag{1}$$

Subject to:

$$\sum_{i\in D} z_{iu} = 1, \forall u\in U \tag{2}$$

$$\sum_{i\in D} y_{ij} = 1, \forall j\in C \tag{3}$$

$$\sum_{u\in U}\sum_{i\in D} z_{iu} \le G_u \tag{4}$$

$$\sum_{j \in C} x_{iju} = \sum_{j \in C} x_{jiu} = 1, \forall i \in D, \forall u \in U \tag{5}$$

$$\sum_{i \in C} r_i \sum_{j \in V} x_{iju} \leq Q, \forall u \in U \tag{6}$$

$$\sum_{i \in D} x_{iju} = 0, \forall j \in D, \forall u \in U \tag{7}$$

$$x_{iju} \in \{0,1\}, \forall i, j \in V, \forall u \in U \tag{8}$$

$$y_{ij} \in \{0,1\}, \forall i \in C, \forall j \in D \tag{9}$$

$$z_{iu} \in \{0,1\}, \forall i \in D, \forall u \in U \tag{10}$$

The objective function (1) aims to minimize the total transportation cost. Constraint (2) ensures that each vehicle is associated with a unique depot. Constraint (3) ensures that each customer is served by exactly one depot. Constraint (4) ensures that the total number of vehicles used does not exceed the maximum number of vehicles. Constraint (5) ensures that the starting and ending points of each route are at the same depot, and that each customer is served by only one vehicle. Constraint (6) ensures that the load of each vehicle during delivery does not exceed its maximum carrying capacity. Constraint (7) ensures that there are no connections between depots. Equations (8)-(10) represent the decision variables.

3 Proposed DRLA_CD Algorithm

The proposed DRLA_CD in this paper consists of a problem decomposition stage and a subproblem solving stage (see Fig. 2). After decomposition by IKA, the original MDVRP problem is transformed into multiple subproblems of VRP. Then, the attention model-vrp (AM-VRP) is used to decouple MDVRP, solving each VRP sequentially to obtain solutions and optimize the objective value of the original problem.

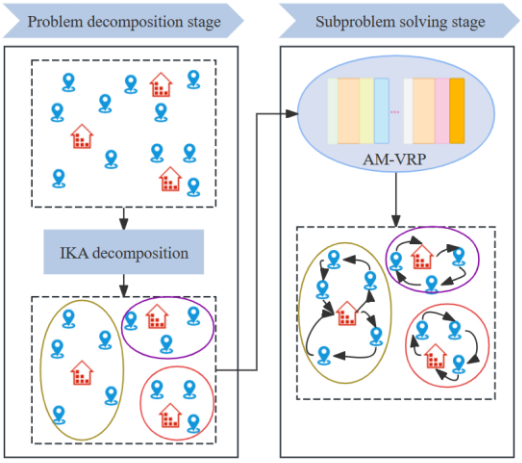

Fig. 2. Framework of DRLA_CD

3.1 DRLA_CD Problem Decomposition Stage

To effectively guide the algorithm to search in high-quality feasible solution space and improve the search efficiency of the algorithm, this paper designs the IKA algorithm to decompose MDVRP and obtain a series of VRPs.

Firstly, based on the distribution of customer locations, the K-means algorithm is utilized to cluster all customers into several groups equal to the number of depots. Secondly, by considering the distance between depot locations and cluster centers, distance-based rules are designed to effectively determine the customer groups served by each depot. Finally, the customers in these groups that do not meet the depot capacity constraints are adjusted, and these customers are reassigned to other depots. Through the decomposition using IKA, multiple single-depot VRP subproblems are obtained.

3.2 DRLA_CD Subproblem Solving Stage

For the decomposed VRP instance s, it is defined as a graph with T nodes. A Markov Decision Process is established for the VRP, and its agent, state, action, and reward are defined as follows:

Agent: The vehicles are regarded as intelligent agents. At each time step t, the agent selects actions based on the environment and learns to maximize cumulative rewards.

State: States include static and dynamic states. The static state is the overall image feature information output by the encoder, including customer location, requirements, and other information. The dynamic state is based on the characteristics of the visited nodes and the remaining capacity of the vehicle at the current time.

Action: At each time step t, the agent selects the next node to visit based on the probability distribution of unvisited nodes. The node selected for visitation at time step t is represented as action π_t.

Reward: For MDVRP, the objective function aims to minimize the total transportation cost. As indicated by equation (1), minimizing the route length is sufficient, hence the negative route length is defined as the reward.

The policy is defined as a mapping from states to actions, approximated by a neural network-based stochastic policy $p_\theta(\pi \mid s)$, which is parameterized by the neural network with parameters θ. This stochastic policy is modeled as:

$$p_\theta(\pi \mid s) = \prod_{t=1}^{T} p_\theta(\pi_t \mid s, \pi_{1:t-1}) \tag{11}$$

After the stochastic policy $p_\theta(\pi \mid s)$ samples a round of data and obtains corresponding rewards, the policy gradient estimation is performed based on the Reinforcement Learning algorithm to adjust the parameters θ.

3.2.1 Encoder

The encoder used in this article is similar to the encoder of the transformer model, and due to the input order independence of the VRP, positional encoding has been removed. Fig. 3 illustrates the specific structure of the encoder.

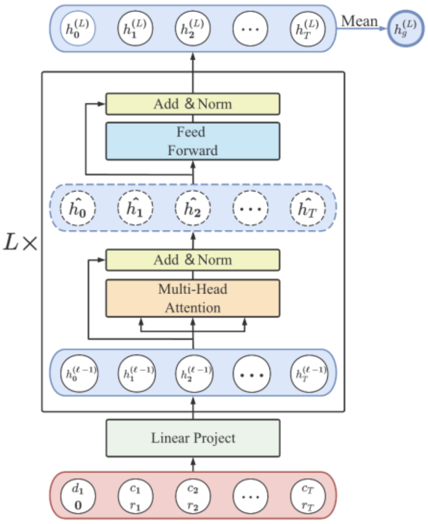

Fig. 3. Encoder structure diagram

To distinguish between depot and customers, separate parameters W_0^x and b_0^x are used to compute the initial embedding $h_0^{(0)}$ of depot node, as shown in Eq. (12):

$$h_i^{(0)} = \begin{cases} W_0^x[d_1, 0] + b_0^x \\ W^x[c_i, r_i] + b^x \qquad i = 1, ..., T \end{cases} \tag{12}$$

Where W and b are the parameters to be learned, $h_i^{(0)}$ represents the initial embedding of node i, and $h_i^{(\ell)}$ represents the operation result in the ℓth encoding layer, where $\ell \in \{1, 2, ..., L\}$. Each attention layer consists of two sub-layers: multi-head attention (MHA) and feed-forward (FF). Additionally, skip-connection and batch normalization (BN) are introduced to ensure the stability of training deep neural networks, as shown in Eq. (13):

$$\begin{gathered} h_i = BN^{\ell}(h_i^{(\ell-1)} + MHA_i^{\ell}(h_1^{\ell-1}, ..., h_T^{(\ell-1)})) \\ h_i^{\ell} = BN^{\ell}(h_i + FF^{\ell}(h_i)) \end{gathered} \tag{13}$$

After L encoding layers, embeddings $h_i^{(L)}$ for T nodes are obtained and the graph embedding $h_g^{(L)}$ representing the global graph information is obtained according to Eq. (14).

$$h_g^{(L)} = \frac{1}{T}\sum_{i=0}^{T} h_i^{(L)} \tag{14}$$

3.2.2 Decoder

Decoding is performed in sequence. At time t, the decoder predicts the current output node based on the node embeddings $h_i^{(L)}$ and graph embedding $h_g^{(L)}$ from the encoder, as well as the previous output $\pi_{t'}(t' < t)$ and the remaining capacity Q_t of the vehicle at time t . The output node is then added to the solution sequence π , until the feasible solution is obtained. During decoding, a special context vector is used to represent the decoding context information. Taking the solution $\pi = (0,1,2,0)$ as an example, Fig. 4 shows the process of constructing the solution by the decoder.

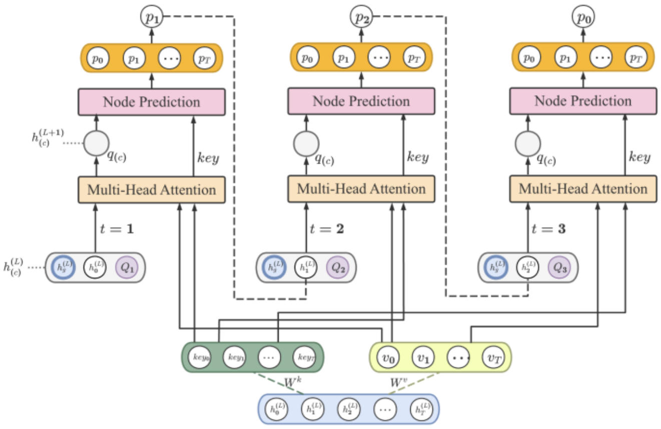
p_1
p_2
p_0
p_0 p_1 $\cdots$ p_T
Node Prediction
$h_{(c)}^{(L+1)}$
$q_{(c)}$
key
Multi-Head Attention
$t=1$
$t=2$
$t=3$
$h_{(c)}^{(L)}$
$h_3^{(L)}$ $h_0^{(L)}$ Q_1
$h_3^{(L)}$ $h_1^{(L)}$ Q_2
$h_3^{(L)}$ $h_2^{(L)}$ Q_3
key_0 key_1 $\cdots$ key_T
v_0 v_1 $\cdots$ v_T
W^k
W^v
$h_0^{(L)}$ $h_1^{(L)}$ $h_2^{(L)}$ $\cdots$ $h_T^{(L)}$

Fig. 4. Decoder structure diagram

Node prediction: the node prediction layer outputs node selection probabilities using a single-head attention layer:

$$u_{(c),i} = \begin{cases} clip \cdot \tanh(\frac{q_{(c)}^T k_i}{\sqrt{\dim_k}}) & If\ \ i\ feasible \\ \text{-}\infty & otherwise \end{cases} \tag{15}$$

$u_{(c),i}$ represents the unnormalized probability value, and the output probability p_i is obtained by applying softmax normalization to $u_{(c),i}$:

$$p_i = \mathrm{softmax}(u_{(c),i}) \tag{16}$$

3.2.3 Train with REINFORCE Algorithm

The negative path length is used as a reward and trained through reinforcement learning algorithms. The loss is defined as the expected path length $L(\pi)$ of the VRP instance to be solved, as shown in Eq. (17) :

$$\begin{gathered} \nabla_\theta \mathrm{L}(\theta) = \mathrm{E}_{p_\theta(\pi|s)}\left[L(\pi)\nabla \log p_\theta(\pi \mid s)\right] \\ \theta \leftarrow \theta + \nabla_\theta \mathrm{L}(\theta) \end{gathered} \tag{17}$$

The parameter θ is updated using gradient descent with the REINFORCE algorithm incorporating a baseline, as shown in Eq. (18):

$$\begin{gathered} \nabla_\theta \mathrm{L}(\theta) = \mathrm{E}_{p_\theta(\pi|s)}\left[(L(\pi) - b(s))\nabla \log p_\theta(\pi \mid s)\right] \\ \theta \leftarrow \mathrm{Adam}(\theta, \nabla_\theta \mathrm{L}(\theta)) \end{gathered} \tag{18}$$

3.2.4 Search Strategy

AM-VRP can rapidly generate delivery paths for MDVRP, but there is still room for improvement in the solution results. To enhance the delivery paths, this paper adopts two search strategies:

1) 2-opt search: applying the 2-opt operation to the model output solution to enhance the overall solution quality.

2) Sampling Search: To address the issue of models easily getting trapped in local optima caused by greedy strategies, the model employs a random sampling action strategy to repeatedly solve the same problem instance, obtaining multiple complete solutions. Selecting the optimal solution among them can to some extent avoid this problem.

4 Experimental Results and Discussion

In this section, numerical simulations and experiments are conducted using randomly generated instances of various sizes to validate the superiority and effectiveness of DRLA_CD. All algorithms are implemented using Python and PyTorch and run on a 12th Gen Intel(R) Core(TM) i5-12400f 2.5GHz CPU and GeForce RTX 3060 GPU.

4.1 Experiment Design

We evaluate the algorithm's performance on three datasets of different sizes: 50_2, 100_2, and 300_3. For each dataset, we test the algorithm's performance on 1000 randomly generated instances according to their corresponding distribution. The average value of all test cases is used as the performance metric for the model.

The proposed method is compared with PSO[22], CSOM&CW [23], HTS[24], and ML[25]. In addition, to demonstrate the contribution of key components to DRLA_CD, we implemented some variants of DRLA_CD. DRLA_CD1 removed the 2-opt search. DRLA_CD2 removed the sampling search strategy. DRLA_CD3 replaced AM-VRP with the traditional Clarke and Wright (CW) algorithm[26] in the Subproblem Solving Stage. Relative error (*RE*) is used to evaluate the effectiveness of scheduling algorithms and is directly used to evaluate the performance of algorithms, as shown in Eq. (19):

$$RE = \frac{f_i - f_{best}}{f_{best}} \tag{19}$$

Where f_{best} represents the minimum total cost obtained by running all comparison algorithms, and f_i represents the total cost obtained by running the *i*-th algorithm.To obtain more reliable experimental results, all algorithms are executed 20 times for each instance, with each execution time of $6 \times T_{time}$ (T_{time} is the execution time of DRLA_CD). The best *RE* (*BRE*), average *RE* (*ARE*), and worst *RE* (*WRE*) are used to measure the performance.

4.2 Experiment Results and Discussion

The comprehensive comparison results of all comparison algorithms are shown in Table 2 and Fig. 5. The minimum *BRE*, *ARE*, and *WRE* for each instance in Table 2 are highlighted in bold. According to Table 2, it can be seen that DRLA_CD has the lowest *BRE*, *ARE*, and *WRE*, which proves the superiority of the proposed DRLA_CD algorithm. In addition, in instances of different scales, DRLA_CD achieved the lowest total average values of *BRE*, *ARE*, and *WRE*, indicating that the average performance of DRLA_CD is better than the other seven comparison algorithms.

Table 2. Comparison results of DRLA_CD with the other methods.

Method	50_2			100_2			300_3		
	BRE	*ARE*	*WRE*	*BRE*	*ARE*	*WRE*	*BRE*	*ARE*	*WRE*
PSO	0.07	0.09	0.14	0.08	0.12	0.15	0.15	0.21	0.29
CSOM&CW	0.16	0.29	0.46	0.48	0.57	0.65	0.42	0.47	0.49
HTS	0.46	0.54	0.71	0.39	0.64	0.71	0.45	0.51	0.66
ML	0.12	0.14	0.17	0.08	0.11	0.14	0.14	0.17	0.21
DRLA_CD1	0.05	0.06	0.08	0.12	0.15	0.17	0.20	0.24	0.26
DRLA_CD2	**0.00**	0.06	0.13	0.01	0.08	0.15	0.03	0.07	0.12
DRLA_CD3	0.22	0.22	0.22	0.42	0.42	0.42	0.51	0.51	0.51
DRLA_CD	**0.00**	**0.01**	**0.02**	**0.00**	**0.02**	**0.04**	**0.00**	**0.02**	**0.04**

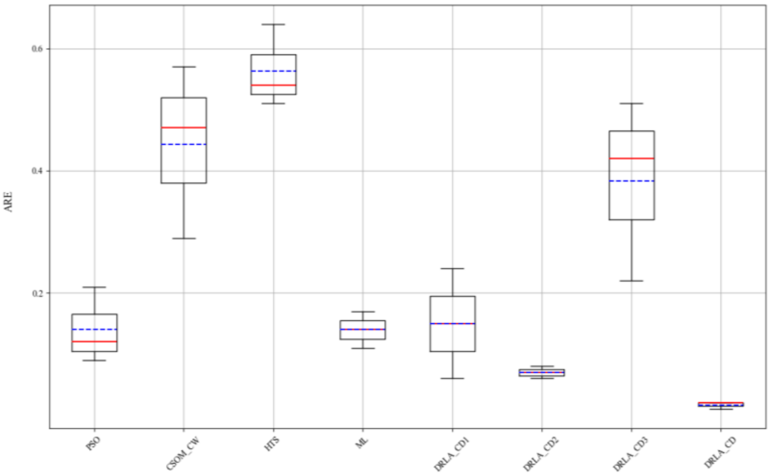

Fig. 5. Box plot of comparison results between SA-DRL and seven other algorithms

5 Conclusions

For the MDVRP, this paper proposes DRLA_CD for solving. DRLA_CD consists of two stages. The first stage is problem decomposition, where an IKA clustering decomposition strategy is designed to decompose the MDVRP, effectively reducing the solution space of the problem. The second stage is subproblem solving, where the trained AM-VRP is combined with multiple search strategies to solve each decomposed subproblem. Finally, the solutions of each subproblem are merged to obtain the solution of the original problem. Simulation experiments and algorithm comparisons prove that DRLA_CD is an effective algorithm for solving the MDVRP. The next step will be to extend DRLA_CD to the location routing problem and heterogeneous fleet vehicle routing problem.

Acknowledgement. This research was supported by the National Natural Science Foundation of China (62173169 and 72201115) and the Basic Research Key Project of Yunnan Province (202201AS070030).

References

1. G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Management Science,* vol. 6, no. 1, pp. 80-91, 1959.
2. B. E. Gillett and J. G. Johnson, "Multi-terminal vehicle-dispatch algorithm," *Omega-international Journal of Management Science,* vol. 4, pp. 711-718, 1976.
3. 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.
4. A. Bettinelli, A. Ceselli, and G. Righini, "A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows," *Transportation Research Part C Emerging Technologies,* vol. 19, no. 5, pp. 723-740, 2011.
5. J.-F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," Networks, vol. 30, pp. 105-119, 1997.
6. T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei, "A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems," *Operations Research,* vol. 60, no. 3, pp. 611-624, 2012.
7. F. B. D. Oliveira, R. Enayatifar, H. J. Sadaei, F. G. Guimares, and J. Y. Potvin, "A Cooperative Coevolutionary Algorithm for the Multi-Depot Vehicle Routing Problem," *Expert Systems with Applications*, vol. 43, no. C, pp. 117-130, 2015.
8. Hu Rong, Li Yang, Qian Bin, Jin Huai-Ping, Xiang Feng-Hong. An enhanced ant colony optimization combined with clustering decomposition for solving complex green vehicle routing problem. Acta Automatica Sinica, 2022, 48(12): 3006−3023.
9. M. E. H. Sadati, B. Catay, and D. Aksen, "An efficient variable neighborhood search with tabu shaking for a class of multi-depot vehicle routing problems," *Computers & operations research,* no. Sep., p. 133, 2021.
10. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature,* vol. 550, no. 7676, pp. 354-359, 2017.
11. Volodymyr *et al.*, "Human-level control through deep reinforcement learning," *Nature,* 2015.
12. W. Kool, H. V. Hoof, and M. Welling, "Attention, learn to solve routing problems!," in *International Conference on Learning Representations*, 2019.
13. O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer Networks," *Computer Science,* vol. 28, 2015.
14. I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. J. a. p. a. Bengio, "Neural combinatorial optimization with reinforcement learning," 2016.

15. V. R. Konda and J. N. Tsitsiklis, "Actor-Critic Algorithms," in *Neural Information Processing Systems*, 1999.
16. M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takáč, "Reinforcement Learning for Solving the Vehicle Routing Problem," 2018.
17. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, pp. 1735-1780, 1997.
18. A. Vaswani *et al.*, "Attention Is All You Need," *arXiv,* 2017.
19. S. Li, Z. Yan, and C. Wu, "Learning to Delegate for Large-scale Vehicle Routing," 2021.
20. L. Xin, W. Song, Z. Cao, and J. Zhang, "NeuroLKH: Combining Deep Learning Model with Lin-Kernighan-Helsgaun Heuristic for Solving the Traveling Salesman Problem," in *arXiv*, 2021.
21. M. Kim, J. Park, and J. Park, "L EARNING TO CROSS EXCHANGE TO SOLVE MIN - MAX VEHICLE ROUTING PROBLEMS," 2023.
22. S. Geetha, P. T. Vanathi, and G. Poonthalir, "METAHEURISTIC APPROACH FOR THE MULTI-DEPOT VEHICLE ROUTING PROBLEM," *Applied Artificial Intelligence,* vol. 26, pp. 878 - 901, 2012.
23. F. Oudouar, M. Lazaar, and Z. E. Miloud, "A novel approach based on heuristics and a neural network to solve a capacitated location routing problem," *Simul. Model. Pract. Theory,* vol. 100, p. 102064, 2020.
24. J. X. Cao, Z. Zhang, and Y. Zhou, "A location-routing problem for biomass supply chains," *Comput. Ind. Eng.,* vol. 152, p. 107017, 2021.
25. A. Arishi, K. K. Krishnan, and M. Arishi, "Machine learning approach for truck-drones based last-mile delivery in the era of industry 4.0," *Eng. Appl. Artif. Intell.,* vol. 116, p. 105439, 2022.
26. G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research,* vol. 12, no. 4, pp. 568-581, 1964.