Hierarchical reinforcement learning for vehicle routing problems with time windows

Wang, Yunli; Sun, Sun; Li, Wei

Publisher's version / Version de l'éditeur:
https://doi.org/10.21428/594757db.f0516e23
Proceedings of the 34th Canadian Conference on Artificial Intelligence, 2021-06-08

Hierarchical Reinforcement Learning for Vehicle Routing Problems with Time Windows

Yunli Wang[†,*], Sun Sun[†], Wei Li[‡]
[†] National Research Council Canada
[‡] University of Ottawa

Abstract

Vehicle routing problem with time windows (VRPTW) is a practical and complex vehicle routing problem (VRP) which is faced by thousands of companies in logistics and transportation. Usually, VRP is solved by traditional heuristic algorithms. Recently, deep learning models under the reinforcement learning (RL) framework have been proposed to solve variants of VRP. In our study, we propose to use the hierarchical RL to find an optimal policy for generating optimal routes in VRPTW. The hierarchical RL structure includes a low level which generates feasible solutions and a high level which further searches for an optimal solution. Experimental results show that the proposed hierarchical RL model outperforms the non-hierarchical RL model and the heuristic algorithms Google OR-Tools. The proposed model also shows generalization capability in three different scenarios: varied time window constraints, from small-scale to large-scale problems, and generalization across different datasets. The flexible framework of hierarchical RL can also be applied to solve other complex VRPs with multiple objectives.

Keywords: Vehicle routing problem with time windows; Hierarchical reinforcement learning; Generalization

1. Introduction

Standard vehicle routing problem (VRP) aims to find an optimal routing from the depot to multiple customers so as to minimize the cost, and can be formulated as a combinatorial optimization problem. Practically, VRP has found many important applications in logistics and transportation. Depending on different applications there are several variants of VRP such as capacitated vehicle routing problem (CVRP), split delivery VRP (SDVRP), VRP with time windows (VRPTW), and VRP pick and delivery (VRPPD). VRP generalizes Traveling Salesman Problem (TSP) and is NP-hard. To solve VRP traditional methods can be categorized into exact solutions, classical heuristic algorithms, and meta-heuristic algorithms [1]. Exact solutions only exist in small-scale problems. Heuristic and meta-heuristic algorithms are usually designed for one particular variant of VRP, and thus cannot be easily adjusted for other VRP problems. Recently deep learning approaches have been proposed to solve large-scale combinatorial optimization problems with less inference time [2–5]. Previous work that applies reinforcement learning (RL) on VRP usually adopts a flat RL structure [3–5] and designs problem-specific network architectures [6]. Very few studies have focused on more practical and complex VRPs such as VRPTW. VRPTW has found many applications in distribution management such as beverage and food delivery, bank and postal delivery, and waste collection [1].

In this paper, we focus on the study of VRPTW based on deep learning approaches. Exact algorithms and heuristics for VRPTW have been reported in operation research community [1]. Recently a deep learning approach has been proposed to solve VRPTW by using the attention on sequence construction of multiple routes [6]. Differently, in our study we instead use a hierarchical RL model to solve VRPTW. A hierarchical RL algorithm could, in principle, efficiently find solutions to complex problems and reuse representations between related tasks [7].

[*] Yunli.Wang@nrc-cnrc.gc.ca

2. Methods

VRPTW can be formulated as a complete undirected graph $G = (V, E, A)$. The set $V = \{v_0, ..., v_n\}$ is a vertex set. The vertex v_0 represents the depot, and the vertex $v_i, i \neq 0$, represents a customer with the demand q_i. Each edge $e \in E = (i, j), i, j \in V$, is associated with the travel cost c_{ij}. All vertexes have the node attribute $A = [X, T, D]$, where X represents the location, T represents the time window, and D represents the demand of customers (except the depot). We assume that there are M vehicles each with the capability Q available at the depot for delivery. The goal of VRPTW is to minimize the total cost such that the demand of each customer should be met within the time window. Customers can be visited only once.

2.1. Graph pointer network

Graph pointer network (GPN) has been proposed to solve TSP with time windows (TSPTW) [8]. In our study, GPN is adopted to create routes for VRPTW. Compared to TSPTW, VRPTW additionally includes the vehicle capacity constraint. As shown in Figure 1, the input to GPN is a graph with the unordered vertex set V, and the output includes multiple routes. Each route consists of a sequence of ordered v_k indicating the visiting order. GPN uses the encoder-decoder architecture. The encoder has two components: the context vector which represents the global representation of VRPTW, and the pre-context vector which captures the local context of the current route. The context vector is generated from a graph convolutional network (GCN) [9]. The sequence of previous nodes $(v_1, v_2, ..., v_{k-1})$ in the current route is used to generate the pre-context embedding. In the decoder, the context vector and the pre-context vector are combined using the attention mechanism to predict the probability distribution of the next customer v_k on the route.

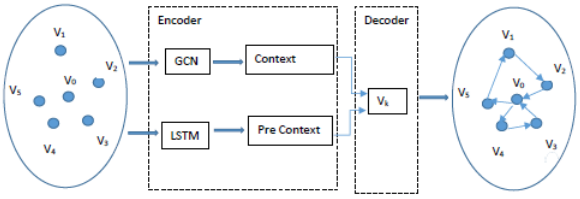

Figure 1. The encoder-decoder architecture in graph pointer network. The input to the encoder is the vertex set V with nodes $v_0, v_1, v_2, ...$ and the output of the decoder is one solution of two routes with ordered sequence of nodes v_0, v_4, v_3, v_0 and v_0, v_5, v_1, v_2, v_0.

2.2. Hierarchical reinforcement learning

RL for VRPTW Denote one solution for VRPTW as σ with $\sigma_k = v_i, k \in 1, 2, ..., N$. This consists of multiple routes and each route starts from the depot and ends at the depot using one vehicle. We formulate VRPTW as a Markov Decision Process (MDP). Under MDP an agent aims to minimize the cost by interacting with the environment. For VRPTW, the environment s_t includes the vehicle capacity, the locations of all nodes, as well as the demands and the time windows of all customers. Each state is defined as the set including all previously visited customers $s_t = \{\sigma_k\}_{k=1}^{t}$. The action a_t is to add a specific customer in the route (i.e., $a_t = \sigma_{k+1}$) or to return to the depot (i.e., $a_t = \sigma_1$). The action a_t is thus discrete and the action space will increase with the customer size. If the demand of a customer is satisfied, the customer will be masked so it will not be selected in the future. The cost $c(s_t, a_t)$ includes the travel cost $c_{k,k+1}$, the time penalty $t_{k,k+1}$, and the number

of the vehicles. Given the existing customers, the agent aims to find a policy π which is a distribution over the next visiting customer. The MDP problem for VRPTW is thus to find an optimal policy to minimize the expected cost: $\pi^* = \arg\min_{\pi} \mathbb{E}\left[\sum_{k=1}^{N} c(s_t, a_t)|\pi\right]$.

Hierarchical RL for VRPTW Based on hierarchical RL, we transform VRPTW into a multi-objective optimization problem with two layers. Specifically, we first find feasible solutions in the low level and then search for an optimal solution in the high level. The model trained at the low level is used as a hidden variable $H, h_t \in H$, at the high level. The input to the high level consists of a graph $G = (V, E, A)$ and the hidden variable H. In each state, the action a_t is sampled from a policy $\pi(a_t|s_t, h_t)$ in the decoder of GPN at the high level. To address the constraints of the time windows, we focus on the soft penalty and penalize the time window violations in the objective function. In our study, minimizing the number of vehicles is used as an auxiliary task in the objective function for VRPTW.

The objective function of VRPTW is a linear combination of the travel cost, the time penalty, and the number of vehicles. α, β, and γ are the coefficients associated with the time cost C_σ, the time penalty T_σ, and the number of vehicles M_σ, respectively. In particular, the objective of the low level model is to minimize the time penalty and the number of vehicles. The high level RL model is trained to minimize the travel cost, the time penalty, and the number of vehicles. We formulate the optimization at the low and high levels as follows:

$$\min_{\sigma} L_l(\sigma) = \sum_{k=1}^{N} \beta T_\sigma + \gamma M_\sigma, \qquad \min_{\sigma} L_h(\sigma) = \sum_{k=1}^{N} \alpha C_\sigma + \beta T_\sigma + \gamma M_\sigma, \tag{2.1}$$

where the total time penalty for violating the time windows is defined as $T_\sigma = \sum_{k=1}^{N} t_{v_k}$ with t_{v_k} being the accumulated time violations, and the total travel cost is defined as $C_\sigma = \sum_{k=1}^{N} c_{(k,k+1)}$, where $c_{k,k+1}$ is the travel cost from v_k to v_{k+1}. Note that we use the number of vehicles M_σ in the objective at both the low and high levels to reduce the number of vehicles used.

In the above hierarchical RL framework, any RL algorithm can be used and we adopt REINFORCE [10] in our study. To compare with the proposed hierarchical RL model, we also create a non-hierarchical RL model, which is based on GPN and only optimizes the objective at the high level.

3. Experiments

To test the performance of our proposed algorithm, we compare it with non-hierarchical RL algorithm and Google OR-Tools. As far as we know, only one previous work focuses on VRPTW using deep learning method [6], and the linear combination of the time cost and the accumulated time violation is used as the single performance metric in [6]. Since the experimental setup is different, for comparison, we perform experiments on the same Solomon dataset used in [6] with a more strict vehicle capacity. To capture both the feasibility and the efficiency for VRPTW, we adopt three metrics for performance evaluation: the accuracy, the time cost, and the number of vehicles. All evaluation is performed on 1000 test instances. Concretely, the accuracy denotes the percentage of problems with feasible solutions, the time cost denotes the average time cost for all solutions, and the number of vehicles is the average number of vehicles used in these solutions.

We test all algorithms on the Solomon dataset that is commonly used in literature and also generate a new datatset 2-opt for VRPTW. The results of Solomon dataset are shown in a long version of this paper (https://github.com/yunliwang/HRL_VRPTW/blob/main/Hierarchical_RL_for_VRPTW.pdf). 2-opt dataset is generated using the heuristic algorithm 2-opt to ensure the existence of feasible solutions [11]. Given the number of vehicles

M and the time window relaxation factor tw, at first, all customer locations are randomly sampled, then K-Means is used to cluster customers into multiple routes, and 2-opt is used to find feasible solutions on each route, at last the demands of each customer on each route are sampled randomly. A larger value of tw indicates an easier VRPTW. The goal of 2-opt is to both create a feasible solution and to provide a baseline performance. In training the hierarchical and non-hierarchical RL models, the α is set as 1, and β and γ are set as 10 at the low and high levels as the hyper-parameters used in 2-opt datasets.

3.1. **Hierarchical RL vs. non-Hierarchical RL**

In Table 1, we compare the performance of hierarchical RL, non-hierarchical RL, the traditional heuristic algorithm 2-opt, and Google OR-Tools on VRPTW with 20 customers. For the hierarchical RL, we consider three different objectives at the high level: the time cost, the linear combination of the time cost and the time penalty, and the suggested objective in Section 2.2. We found that neither of the hierarchical and the non-hierarchical RL can reach 100% feasibility on the test set. The hierarchical RL models perform much better than the non-hierarchical RL model in terms of the time cost and the number of vehicles. For the hierarchical models, the proposed one leads to a higher accuracy. Overall, the hierarchical RL model reaches comparable performance with traditional methods on problem size 20 (i.e., 20 customers).

Table 1. Performance of hierarchical RL, non-hierarchical RL, and baselines on 2-opt dataset of VRPTW with problem size 20.

Method	Objectives	Accuracy	Time Cost	# Vehicles
2-opt	time cost	**1.000**	7.44	**5.00**
OR-Tools	time cost	**1.000**	**7.03**	**5.00**
Non-hier RL	time cost + time penalty + #vehicles	0.986	12.42	6.25
Hier RL	time cost (high)	0.830	7.19	5.03
Hier RL	time cost + time penalty (high)	0.970	7.27	**5.00**
Hier RL *	time cost + time penalty + #vehicles (high)	**0.995**	**6.78**	**5.00**

3.2. **Easy vs. Difficult Problems**

Furthermore, we test the hierarchical RL on easy and difficult problems by changing the time window relax factor tw. The comparison among the hierarchical RL, the non-hierarchical RL, and Google OR-Tools on problem size 20 is shown in Table 2. We consider two types of TW constraints in VRPTW: 1) the hard constraint in which solutions that violate the time windows are deemed as infeasible, and 2) the soft constraint in which the accumulated time violation T_σ is added to the objective. We test both the hierarchical and non-hierarchical RL models with the soft constraint. Google OR-Tools is tested on VRPTW with the hard (OR-Tools_hard) and the soft (OR-Tools_soft) constraints using a fixed number of vehicles 5. Therefore, RL models are comparable to OR-Tools_soft. Overall, RL models show better performance than OR-Tools on both the easy and the difficult VRPTW problems. The accuracy of OR-Tools obviously deceases when the problems become difficult, while the accuracy of RL models does not change much. The running time of OR-Tools indicates that OR-Tools is not able to find feasible solutions for more difficult cases. Also, the hierarchical RL with comparable accuracy, a lower time cost, and a less number of vehicles performs better than the non-hierarchical RL. On the difficult setting with more strict time windows, the RL models show more resilience and outperform Google OR-Tools.

3.3. **Generalization**

Generalization is important for real applications of VRP. Very often in practice the test distribution is different from the training distribution due to, e.g., the change of customers

Table 2. Performance of hierarchical RL on easy and difficult problem settings on 2-opt dataset with problem size 20.

Setting	Method	Accuracy	Time Cost	# Vehicles	Time (s)
Easy problem	OR-Tools_hard	1.000	7.091	5.000	22
$tw = 2$	OR-Tools_soft	**1.000**	9.117	5.000	33
	Non_hier RL	0.997	7.371	5.031	19
	Hier RL	0.959	**5.663**	**4.166**	14
Hard problem	OR-Tools_hard	0.910	7.163	5.000	2240
$tw = 1$	OR-Tools_soft	0.910	9.064	5.000	36
	Non_hier RL	0.992	8.397	5.799	18
	Hier RL	**0.993**	**6.570**	**4.854**	15

Table 3. Performance of hierarchical RL models trained on mix-up dataset ($tw = 1 \sim 3$) on 2-opt dataset with problem size 20 and generalization of the hierarchical RL models on VRPTW from problem size 20 to 50.

Model		Test data	Accuracy	Time Cost	# Vehicles
Hier RL		$tw = 4$	1.000	7.230	6.6
($tw = 3$)		$tw = 3$	1.000	7.300	6.6
Easy		$tw = 2$	0.992	7.490	6.6
		$tw = 1$	0.888	8.212	6.6
Hier RL		$tw = 4$	1.000	8.084	6.1
($tw = 1$)		$tw = 3$	1.000	7.722	6.2
Difficult		$tw = 2$	0.998	7.595	6.4
		$tw = 1$	0.998	8.052	6.6
Hier RL		$tw = 4$	1.000	7.138	6.4
($tw = 1 \sim 3$)		$tw = 3$	1.000	7.169	6.4
Mix-up		$tw = 2$	0.999	7.325	6.4
		$tw = 1$	0.997	7.878	6.5
vrptw20	vrptw20	$tw = 4$	1.000	7.303	6.6
($tw = 4$)		$tw = 3$	1.000	7.297	6.5
		$tw = 2$	0.999	7.560	6.6
		$tw = 1$	0.909	8.606	7.0
vrptw20	vrptw50	$tw = 4$	0.997	11.161	7.3
($tw = 4$)		$tw = 3$	0.967	11.509	7.5
		$tw = 2$	0.833	12.467	8.0
		$tw = 1$	0.259	17.331	10.3

and constraints. First, we evaluate whether the hierarchical RL model trained on easy and difficult sets respectively can be used to generate feasible solutions on a mixed dataset with both easy and difficult data samples. We call a mix-up strategy by mixing data samples from different distributions in training. We show the performance of generalization under mix-up in Table 3. Note that VRPTW becomes easier with a higher value of the time window factor tw. During the training of the hierarchical RL models, the time windows of all nodes are generated either using the same factor $tw = 1$ (difficult)/$tw = 3$ (easy) or a varied factor $tw = 1 \sim 3$. We found that the RL model trained under mix-up can reach a better performance than the one trained with only one type of dataset distribution. In particular, the performance of the hierarchical RL model trained with mix-up is much better than the one trained on easy problems regarding the three metrics considered, and the model also reaches a lower time cost than the hierarchical RL model trained in the difficult setting.

In real applications, the number of customers may vary. We evaluate the generalization of the hierarchical RL model trained on VRPTW problem size 20 to problem size 50 (see Table 3). The results show that the hierarchical RL model performs well on easy VRPTW problem size 50, but deteriorates dramatically on difficult cases. On easy VRPTW of problem size

50, the hierarchical RL model is able to find feasible solutions with only a moderately more number of vehicles. However, the accuracy of the hierarchical RL drops to 26% on difficult cases with $tw = 1$.

4. Conclusions

In this study, we proposed a new hierarchical RL framework for solving VRPTW. The goal of VRPTW is to find an optimal policy which generates optimal routes for visiting customers within the time windows. We adopted graph pointer networks (GPNs) in the hierarchical RL framework. We showed that the hierarchical RL model generally outperforms the non-hierarchical RL model and the traditional heuristic algorithm Google OR-Tools on two datasets. In addition, we showed that the hierarchical RL model reaches a better performance than Google OR-Tools especially on difficult VRPTW with more strict time window constraints. The hierarchical RL model we proposed also shows generalization from small-scale to large-scale datasets. Our study shows that the hierarchical RL is able to achieve superior performance for VRPTW. The hierarchical RL we considered is a general framework which is suitable for complex combinatorial optimization problems with multiple conflicting objectives. Currently, only a few classical RL algorithms such as REINFORCE, A3C, and Q-learning have been used for combinatorial optimization problems. We would like to adopt more advanced RL algorithms such as soft A3C in the hierarchical RL model to further boost the performance.

Acknowledgements

The authors would like to thank Artificial Intelligence for Logistics program at the National Research Council Canada for support.

References

[1] J.-F. Cordeau, G. Laporte, M. W. Savelsbergh, and D. Vigo. “Vehicle routing”. In: *Handbooks in operations research and management science* 14 (2007), pp. 367–428.

[2] O. Vinyals, M. Fortunato, and N. Jaitly. “Pointer networks”. In: *Advances in neural information processing systems*. 2015, pp. 2692–2700.

[3] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác. “Reinforcement learning for solving the vehicle routing problem”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 9839–9849.

[4] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song. “Learning combinatorial optimization algorithms over graphs”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 6348–6358.

[5] W. Kool, H. Hoof, and M. Welling. “Attention solves your TSP, approximately”. In: *Statistics* 1050 (2018), p. 22.

[6] J. K. Falkner and L. Schmidt-Thieme. “Learning to Solve Vehicle Routing Problems with Time Windows through Joint Attention”. In: *arXiv preprint arXiv:2006.09100* (2020).

[7] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine. “Latent Space Policies for Hierarchical Reinforcement Learning”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, pp. 1851–1860.

[8] Q. Ma, S. Ge, D. He, D. Thaker, and I. Drori. “Combinatorial Optimization by Graph Pointer Networks and Hierarchical Reinforcement Learning”. In: *AAAI Workshop on Deep Learning on Graphs: Methodologies and Applications*. 2020.

[9] T. N. Kipf and M. Welling. “Semi-supervised classification with graph convolutional networks”. In: *International Conference on Learning Representations*. 2017.

[10] R. J. Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine learning* 8.3-4 (1992), pp. 229–256.

[11] J.-Y. Potvin and J.-M. Rousseau. “An exchange heuristic for routeing problems with time windows”. In: *Journal of the Operational Research Society* 46.12 (1995), pp. 1433–1446.