7th Interdisciplinary Conference on Production, Logistics and Traffic
March 18-19, 2025 | Darmstadt, Germany

Using Neural Combinatorial Optimisation for Solving Time-Constrained Vehicle Routing Problems

Elija Deineko[1*] and Carina Kehrt[2]

[1] *German Aerospace Centre (DLR), Institute of Transport Research, Rudower Chaussee 7, 12489 Berlin*
[*] *Correspondence: elija.deineko@dlr.de; Tel.: +49 (30) 67055253*

Abstract. Developing fast optimisation algorithms and decision support frameworks that can produce accurate solutions in short computation time is crucial for integrated and real-time optimisation in the 21st century. Recently, a new trend in optimisation science has emerged: Neural Combinatorial Optimisation (NCO), where researchers apply generative Artificial Intelligence (AI) models to combinatorial problems. This paper presents an end-to-end NCO model designed to address the Vehicle Routing Problem (VRP) with time constraints and a finite vehicle fleet. The NCO model developed in this study incorporates and extends various state-of-the-art frameworks and algorithms, utilising Reinforcement Learning (RL) approach to train the attention-based encoder-decoder model. We validate our approach by comparing its constructed solutions with state-of-the-art VRP heuristics, demonstrating its performance and scalability to larger problem instances. This study highlights the ability of the NCO model to generalise, even to instances with unknown customer distributions and varying problem sizes. Our findings indicate that NCO offers promising prospects for solving complex, multi-objective optimisation problems in transport logistics, offering an effective solution strategy for solving highly complex sequential decision problems.

1 MOTIVATION

Vehicle Routing Problem (VRP) is a well-known NP-hard combinatorial problem that has been a subject of substantial research over the past decades. However, due to the exponential growth in computational complexity, dynamic environment, numerous constraints, and stochasticities in real-world scenarios, the application of numerical combinatorial methods become almost obsolete for large-size VRPs. Recently, a new paradigm in optimisation science has emerged – Neural Combinatorial Optimisation (NCO). The combination of Reinforcement Learning (RL) and generative Deep-Learning (DL) models, such as Transformers [1], offers a vital perspective for solving various sequential decision problems, including VRP. Since Bello et al. [2], and later Kool et al. [3] have demonstrated the effectiveness of attention-based encoder-decoder models for routing, researchers increasingly leveraged, adapted, and extended this paradigm to related combinatorial tasks [4-6]. Although these studies successfully adopt and evaluate NCO paradigm, the hard-coded integration of constraints, intricate implementation, and intensive computational training limit their broader application. This study presents an end-to-end NCO approach for solving the time-constrained VRP (TCVRP) with a homogeneous, finite vehicle fleet. Concretely, our contribution can be summarised as follow:

- We extend the RL for Operational Research (RLOR) framework [7], adopting it to multi-objectives problems and implementing additional constraints, such as dynamical time windows and finite vehicle fleet, applying state-of-the-art algorithms, and techniques in this field.
- We modify and fine-tune the reward function to optimise for multiple objectives – maximising fleet utilisation and minimising total mileage.
- The method for time constraints is integrated into the decoder network to dynamically prioritise nodes with urgent services.
- We evaluate the developed NCO approach by benchmarking it against conventional techniques and analyse its transferability and generalisation.

2 METHODOLOGY

Designing an NCO algorithm involves (i) formulating the problem as a Markov Decision Process (MDP) and constructing the MDP environment (commonly aligned with [8], (ii) developing a multi-layer encoder-decoder policy network [3], and (iii) adapting a RL algorithm to train the policy network. Moreover, the problem constraints can be implemented either by designing and tuning an appropriate reward function or through a mask – effectively prohibit infeasible actions. For our TCVRP formulation, we first model the problem environment as a tuple of state $\boldsymbol{S}$, action a, reward r and transition $\boldsymbol{Tr}$, as $MDP = (\boldsymbol{S}, a, r, \boldsymbol{Tr})$, where each state $\boldsymbol{S}$ contains the information about the customers $\boldsymbol{N} = \{n_0, n_1, \dots, n_i\}$. Each customer n contains the information about the location (x^n and y^n coordinates), time windows tw^n, demand d^n, the capacity l_t^v, and the number of available vehicles in each time step $\boldsymbol{V_t}$, such as $n = (x, y, tw, d, l_t^v, \boldsymbol{V_t})$. Additionally, we include mask $\boldsymbol{M}_t^N$, so the state $\boldsymbol{S}$ is a tuple $\boldsymbol{s}_t = (\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{M})$. The objective is to construct tours, i.e., a sequence of visited nodes for each vehicle v_i. The action a is therefore the index of the next node to visit, which is based on the context problem information and all previous states in this episode. An episode is terminated if all customers are masked $m_{t+1}^n = 1 \; \forall \; N$ (i.e., if all customers are visited or unreachable due to expired time windows $tw_t^n < 0$), or if there are no more available vehicles, i.e., $V = 0$. The transition function simply determines the rules, how the state $\boldsymbol{s}_t$ transitions to the next state $\boldsymbol{s}_{t+1}$. To encourage the agent to visit as many customers as possible and to prevent it from returning to the depot after visiting the first customer, our reward function is designed as follows:

$$r_t = min(dist(n_t, n_{t+1}) + P * l_t^v) \;\; \text{for} \;\; t \leq N, \quad (1)$$

where $dist$ represents the Euclidean distance between the next unvisited node n_{t+1} and the current node n_t. We set the penalty factor $P = 10$, effectively penalising the agent for returning to the depot with a full vehicle capacity $l \to 1$, and do not when the capacity of the vehicle $l \to 0$. To handle time constrains (note that $tw_t^n \; \forall \; \boldsymbol{N}$ decreases with each time step t), we incorporate our time windows directly in the decoder by rescaling the attention score in the pointer network, defining the urgency factor as follows:

$$f_v^n = \frac{dist(n_t, n_{t+1})/S^v}{tw^n - (d_t^V/S^v)}, \forall \; N, \; 0 < f_u < 1. \quad (2)$$

f_v^n expresses the urgency for serving the unvisited nodes $\boldsymbol{N}$ for the vehicle v being in the current node n. In other words, Equation 2 computes the ratio between the anticipated travel times and the total travelled times, which are updated in each time step. td_t^V/S^v is the total time travelled so far by the vehicle fleet $\boldsymbol{V_t}$. The factor S^v represents the vehicle's velocity which is set to $S^v = 50\frac{km}{h} = 0.014 \; \frac{distance \; units}{s}$ for the normalised instances: $x, y \in [0,1]$. Therefore, if $f_u \to 0$, the customer's requested end-time is far ahead, meaning there is no urgency. If $f_u \to 1$, the service becomes time critical. For $f_u > 1$ or $f_u < 0$ the services are considered as missed.

For our policy network, we enhance the methods introduced by [7], which in turn are based on the well-proven encoder-decoder model developed by [3]. We further incorporate our urgency factor f_u directly in the decoder by rescaling the attention score in the pointer network [9], analogous to Wang et al. 2024. The training procedure follows the Proximal Policy Optimisation (PPO) – a state-of-the-art actor-critic RL algorithm for training in continuous and discrete environments [10].

3 RESULTS

We generate the training and evaluation problem instances for each episode by following a procedure analogous to [3,7]. Each environment episode is initialised with 50 customer locations and a depot, randomly distributed in the normalised Euclidean space, i.e., $x, y \in [0,1]$. Each problem instance is parametrised by the vehicle capacity $C^v = 40$, which is then normalised to $l_{t=0}^v = 1$. We further distribute the demand uniformly: $d_n \sim P_u(1,10)$ for $n = 1,2, \dots, N$ and normalise the demand by $\frac{d^n}{C^v} \forall N$. The total number of vehicles in each instance is set to $V = 5$. Service time windows constraints tw^n are expressed in time units and are randomly sampled from the range $50 \leq tw^{n \neq 0} \leq 10{,}000$ for the customers, and $tw^{n=0} = 10{,}000$ for the depot. Again, this range defines the minimum and maximum time windows for 50 customers in a normalised instance, where the vehicle is deployed at 50 km/h. The results of our model are benchmarked against the PyVRP [11] – a state-of-the-art optimisation framework for multi-constrained VRP based on the genetic metaheuristics. Table 2 shows the results in terms of solution quality for PyVRP and the trained model.

Table 1. Solution costs from the benchmark heuristics PyVRP and the proposed model.

Instance	Nodes	PyVRP	NCO
N50	50	7.59	6.33
N100	100	7.70	6.25
N500	500	5.72	3.07
N1000	1000	7.56	4.48

The NCO model is trained on random instances with 50 customers and then transferred to oversized instances with 100, 500 and 1000 nodes. The policy training takes approximately one week on a CPU cluster machine. We do not compare the computation

time between the OR solutions and the inference time of the NCO solutions, as the tour construction in NCO does not rely on computing combinations but instead outputs the next node online based on the trained policy weights. Figures 1 and 2 present a direct comparison of the tours constructed by PyVRP and the NCO model for instance N100.

As shown in Table 1, the NCO method achieves better results in all cases examined. However, this can be attributed to the fact that the NCO model is trained with a penalty. Consequently, the trained model prioritises nodes with higher demand first and exploits the vehicle fleet faster. In contrast, the PyVRP agent, which is designed to solve prize-collecting VRP, aims to visit as many customers as possible, thereby reducing vehicle mileage by maximising rewards. Figures 1 and 2 show this behaviour by comparing the N100 instance solved by PyVRP (see Figure 1) and by the NCO model (see Figure 2).

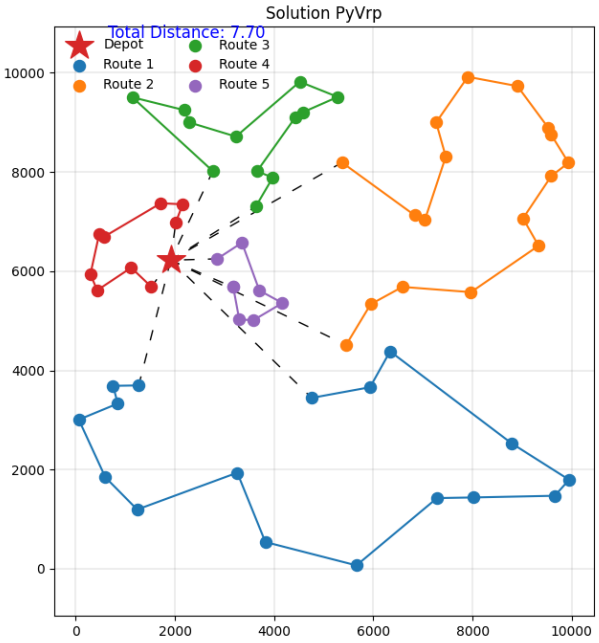

Figure 1. Solution computed by the OR model PyVRP.

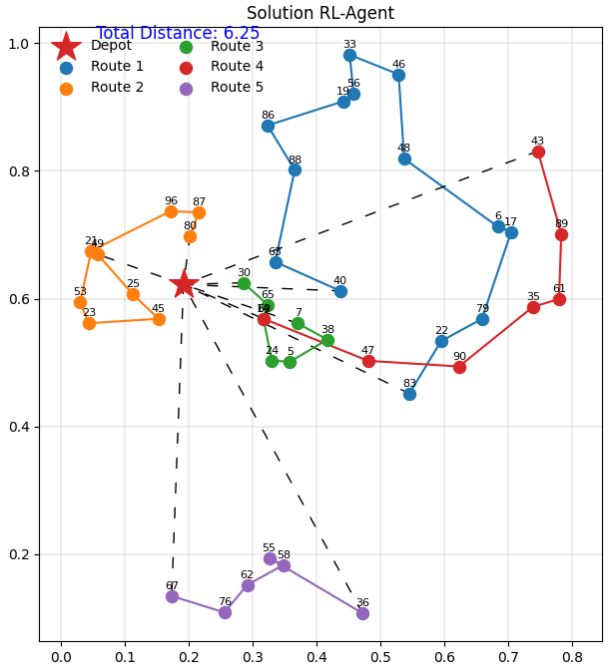

Figure 2. Solution constructed by the proposed NCO model.

As illustrated in Figure 2, the tours generated by our model are comparable to those found by the PyVRP heuristics. However, the NCO model constructs tours with fewer customers and, consequently, resulting in lower total mileage for the vehicle fleet. This effect was particularly evident in large instances (e.g., N1000) where the number of alternative nodes with higher demand increases. Surprisingly, despite being only trained on the 50N instances, the NCO-model generalise well to medium- and large-sized problems with 500N and 1000N, even with unknown node distribution.

4 CONCLUSION AND OUTLOOK

The objective of this study was to demonstrate a proof of concept and test the applicability and transferability of NCO for constrained, multi-objective optimisation problems in transport logistics. We developed, trained, and validated an end-to-end NCO model for a multi-objective, capacitated VRP, incorporating time constraints and a finite vehicle fleet. By incrementally adopting state-of-the-art NCO frameworks, our model effectively constructed tours and achieved solutions comparable to that of state-of-the-art heuristics. Furthermore, trained on medium-sized instances, our method was able to extrapolate the search procedure to the large-sized instances with unknown customer distributions. This highlights the ability of NCO models to generalise. However, assessing NCO models proved challenging, as their solution search logic demands novel validation methods, especially for complex, multi-objective tasks. In our further research, we plan to extend this methodology to other scenarios with additional constraints, such as service fees, customer preferences, and limited mileage for

electric vehicles. Furthermore, a real-world case study for CEP transport in Germany will be developed to apply the developed approach to a VRP with large instances.

REFERENCES

1. Vaswani, A., et al. "Attention is all you need in advances in *Neural information processing systems*." Search PubMed: 5998-6008. (2017). Available online: https://dl.acm.org/doi/10.5555/3295222.3295349 (accessed on 01.10.2024)
2. Bello, I., Pham, H., Le, Q. V., Norouzi, M., & Bengio, S. Neural combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1611.09940. (2016).
3. Kool, W.; Van Hoof, H.; Welling, M. Attention, learn to solve routing problems! arXiv:1803.08475. (2018).
4. Falkner, Jonas K., and Lars Schmidt-Thieme. "Learning to solve vehicle routing problems with time windows through joint attention." *arXiv preprint* arXiv:2006.09100. (2020).
5. Li, Kaiwen, et al. "Deep reinforcement learning for combinatorial optimization: Covering salesman problems." *IEEE transactions on cybernetics* 52.12: 13142-13155. (2021).
6. Gupta, Abhinav, Supratim Ghosh, and Anulekha Dhara. "Deep reinforcement learning algorithm for fast solutions to vehicle routing problem with time-windows." *Proceedings of the 5th Joint International Conference on Data Science & Management of Data* (9th ACM IKDD CODS and 27th COMAD). (2022).
7. Wan, Ching Pui, Tung Li, and Jason Min Wang. "RLOR: A Flexible Framework of Deep Reinforcement Learning for Operation Research." *arXiv preprint* arXiv:2303.13117. (2023).
8. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. Openai gym. *arXiv preprint* arXiv:1606.01540. (2016).
9. Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly. "Pointer networks." *Advances in neural information processing systems* 28. (2015). Available online: https://papers.nips.cc/paper_files/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf (accessed on 01.10.2024)
10. Huang, Shengyi, et al. "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms." *Journal of Machine Learning Research* 23.274:1-18. (2022).
11. Wouda, Niels A., Leon Lan, and Wouter Kool. "PyVRP: A high-performance VRP solver package." *INFORMS Journal on Computing*. (2024).

AUTHORS' BACKGROUND*

*This form helps us to understand your extended abstract better and will not be published.

Your Name	Title	Research Field	Personal website
Elija Deineko	PhD candidate	Transport Logistics, Multi-Agent Simulation, Optimisation, Deep Learning	
Dr.-Ing. Carina Kehrt	Research associate	Transport Logistics, Freight Transport Forecast, Freight Transport Modelling	
	Wählen Sie ein Element aus.		
	Wählen Sie ein Element aus.		