Deep reinforcement learning for combinatorial optimization: case study orienteering problems

Iván Guillermo Peña Arenas, Rym Nesrine Guibadj, Cyril Fonlupt

Deep reinforcement learning for combinatorial optimization: case study orienteering problems

BY

Ivan Peña-Arenas

Rym Guibadj

Cyril Fonlupt

Postdoctoral Progress Report

Laboratory of Computer Science, Signal, and Image of the Côte d'Opale

Project Title	Deep reinforcement learning for combinatorial optimization: case study orienteering problems
By	Ivan Peña-Arenas
Field of Study	Computer science
Project Advisors	Rym Guibadj
	Cyril Fonlupt
Academic Year	2023

ABSTRACT

In the Team Orienteering Problem (TOP) a set of routes is determined with the objective of maximizing the profit gained from visiting customers without exceeding a travel cost/time limit. Currently, the use of machine learning approaches along with optimization techniques in order to solve complex combinatorial problems has been proved successfully. We present an hybrid algorithm which combines a deep learning neural network with an efficient splitting algorithm to solve the TOP. The objective is to compare the efficiency of the proposed method in terms of both, quality of the solution and running times, against previous optimization methods proposed in the literature.

Keywords: : Team orienteering problems, deep learning, optimal split, decoder-encoder mechanisms.

Introduction

The Orienteering Problem is a combinatorial optimization problem that has been proven NP-Hard (Golden, Levy, & Vohra, 1987), and which has different practical applications ranging from logistics to telecommunications. There are different variants of the problem such as, the Team Orienteering Problem (see, Chao, Golden, and Wasil (1996), Tang and Miller-Hooks (2005), Dang, Guibadj, and Moukrim (2013)), the Orienteering Problem with Time Windows (see, Yu, Fang, Zhu, and Ma (2019), Roozbeh, Hearne, and Pahlevani (2020), Saeedvand, Aghdasi, and Baltes (2020)) and Team Orienteering Problem with Time Windows (see, Vansteenwegen, Souffriau, and Oudheusden (2011), Amarouche, Guibadj, Chaalal, and Moukrim (2020)). And several heuristics have been proposed to obtain high quality solutions.

Recent advances in machine learning techniques, particularly on deep learning model architectures like Pointer Networks and Graph Attention Networks, have shown a good performance in a wide variety of fields. These results have encouraged the interest in using those kind of techniques to tackle combinatorial problems. Some examples of this type of strict machine machine learning approaches can be seen in Vinyals, Fortunato, and Jaitly (2015), Khalil, Dai, Zhang, Dilkina, and Song (2017) and Nowak, Villar, Bandeira, and Bruna (2018).

The heuristics are tailored methods that work case-by-case, thus not easy to generalize and which usually require a great deal of expertise to be developed. However, they are efficient solution techniques that rely on their programmed behavior de Costa, Rhuggenaath, Zhang, Akcay, and Kaymak (2021). Machine learning models, make decisions and solve problems, extracting useful information directly from the data without having a particular knowledge of the problem. Nevertheless, as the size and complexity of the problems increases, the performance of the neural network diminishes because the computational time of the learning process should be reduced.

To take advantage of both solution methods, we propose to integrate an efficient splitting algorithm to solve the TOP with a machine learning technique. Unlike previous hybrid approaches, the giant tour (a sequence of customers/locations) is created at once by the neural network and in a second step it is evaluated by the heuristic. In

addition, in the best of our knowledge these hybrid approach has not been proposed to tackle the TOP and the TOPTW.

This document is organized as follows. First, we make a brief discussion of machine learning and hybrid methods applied to combinatorial problems. Later, we introduce the general architecture of the Graph Attention Network (GAN). And finally, some preliminary results and perspectives are presented.

Literature review

Among strict machine learning approaches applied to solve combinatorial problems, Vinyals et al. (2015) present a Pointer Network (PN) architecture to solve the planar Travel Salesman Problem. In this work they apply supervised learning to train the Neural Network (NN). The main issue with this type of training, is that it relies on the quality of the solutions which are used. Thus, there is a good performance for small size problems, where optimal or near to optimal solutions are known, however for large problems the performance is bounded by the quality of the training set solutions. Afterwards, Bello, Pham, Le, Norouzi, and Bengio (2016) apply the same architecture however, they propose a different training method called Reinforcement Learning (RL). In this type of training, it is not necessary an external input (solution) to train the NN. Moreover, the NN adapts its behaviour at each step, receiving positive or negative rewards whether the movements (decisions) increase or decrease the objective function value. They empirically demonstrate that RL significantly improves over supervised learning.

An alternative deep learning architecture based on a decoder coupled with an attention mechanism is presented in Nazari, Oroojlooy, Snyder, and Takac (2018) applied to the Capacitated Vehicle Routing Problem (CVRP). Furthermore, they use RL to train the NN; during in this process the algorithm computes the reward and verifies the feasibility at the same time. The comparison is done against *Google's OR-Tools*, and for instances of 50 and 100 customers in roughly 61% of the cases the algorithm provides shorter tours. In addition, the same architecture can solve the CVRP with stochastic demands and allowing split deliveries.

Kool, van Hoof, and Welling (2018) present an encoder-decoder NN coupled with an Attention Mechanism (AM). In order to improve the training algorithm they test different baselines, such as, roll-out, exponential and using a critic. Moreover, the AM is compared with a Pointer Network (PN) under different baselines. The results show that the AM outperforms the PN using any baseline, and the roll-out baseline improves the quality and the convergence speed of both AM and PN.

Recently there have been presented different hybrid approaches combining heuristics with machine learning approaches to solve combinatorial problems. Hottung and Tierney (2019) propose a Neural Large Neighbourhood Search (NLNS) method to solve the CVRP and the Split Delivery Vehicle Routing Problem (SDVRP), in which the neighbourhood is explored by a Recurrent Neural Network (RNN). The learning mechanism is based on a deep neural network with an attention mechanism, and it is designed to perform the complex task of developing repair operators. The experiments show that on SDVRP instances, the NLNS is able to outperform the state-of-the-art method SplitILS on instances with 100 customers.

A RNN is trained to learn a neighbourhood search mechanism based on 2-Opt operators to solve vehicle routing problems in de Costa et al. (2021). They present a new approach to explore the neighbourhood, where the 2-Opt method is modeled as a Markov Decision Process (MDP), where the states are defined by the tuple S, S' which are the the current and lowest-cost solution, and the transitions are given by the tuple of actions $(a1, a2)$ which are the index positions to exchange. The handicap of this method is that it requires a large amount of iterations during the training to achieve the performance of the classical methods.

Gama and L. Fernandes (2021) present a RNN based on Pointer Networks applied to the Orienteering Problem with Time Windows (OPTW) and the Tourist Trip Design Problem (TTDP). Their model is based on the PN architecture proposed in Vinyals et al. (2015), albeit they introduce three different aspects with respect to previous PN architectures: New set representation at each iteration, transformer with recursion and masked self-attention. The algorithm significantly outperforms the standard competitive heuristic Iterated Local Search, with inference times that are suitable for real

time on-line applications. However, since the feasibility is verified while creating the sequence, the method requires long training times.

Methods

We propose a Graph Attention Network integrated to an efficient splitting algorithm to solve the TOP. Several methods that combine machine learning and heuristics have been proposed in the past Gama and L. Fernandes (2021). However, most the previous approaches build a sequence in a constructive way, that is to say, inserting a new client/location into the sequence after imposing feasibility at each step, making the learning process difficult and slow. Our main idea is to use the GAN to create a permutation (sequence) of all accessible locations (usually referred as *giant Tour*) at once and afterwards, call the splitting algorithm to construct the set of optimal tours. In this sense, striving to accelerate the learning process, while keeping good quality solutions.

Our model is a Recursive Neural Network based on a Graph Attention Network, composed by an encoder-decoder mechanism (Kool et al., 2018). The encoder has three encoder layers connected in a cascade fashion. Therefore, the information treatment is independent, and once the information is treated by a layer it is sent to next layer, see Figure 1. Each encoder layer is composed by one Multi-head attention (MHA) and one Feed Forward Layer (FFL) linked by dropout and batch normalization functions.

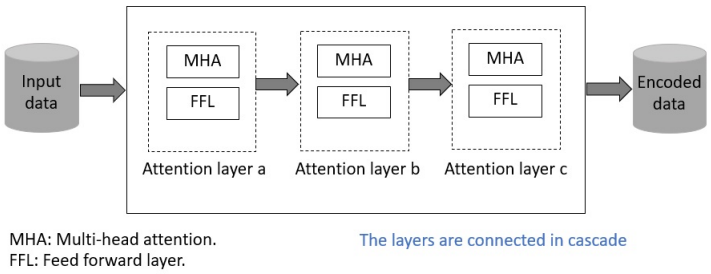
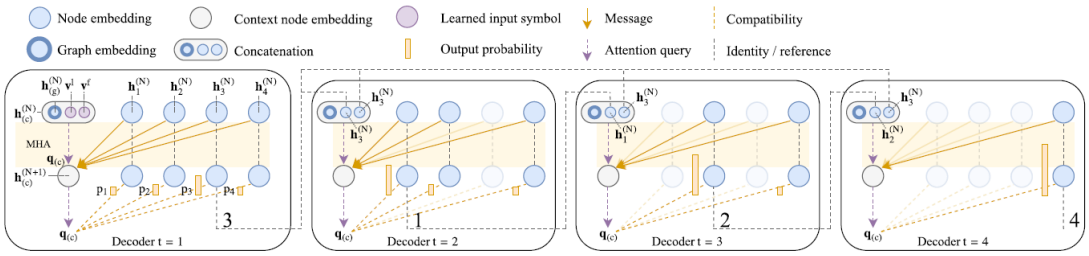


Figure 1: Encoder.

The decoder is composed by one MHA and a FFL, linked by $Thanh$ function. The decoding of the giant Tour is done sequentially, at each time step $t \in \{1, 2, .., n\}$, the selected node (location) π_t is decoded based on the embeddings from the encoder

and the previous outputs π'_t generated at time $t' < t$. In particular, at each iteration the context consists of the graph embedding and the embeddings of the first and the previous output (node) of the current tour. In addition, the final probabilities are computed using a single-head attention mechanism.

The context vector encodes all the information coming from the encoder in a sequence to sequence Neural Network. Furthermore, the attention mechanism applied to the context allows the decoder to focus on particular sections of the outputs from the encoder (Bahdanau, Cho, & Bengio, 2014). As mentioned, in the current architecture the context consists of the embedding of the graph, the previous (last) node π_{t-1} and the first node π_1. At each iteration π_{t-1} is updated and the nodes that cannot be visited (since they are already in the sequence) are masked. Figure 2 presents an example of decoding the process.



Source: Kool et al., 2018.

Figure 2: Example decoding mechanism.

In this architecture, the attention mechanism algorithm assigns the message weights between the nodes in the graph. Since the attention is linked to the changes in the context and the graph, it also has to be updated at each iteration. The weight of the message value that a node receives from a neighbor depends on the compatibility of its query with the key of the neighbor (see, Vaswani et al. (2017)).

We start with the OP version of the problem to later move forward to the TOP and the TOPTW. The attention Neural Network defines a stochastic policy $p_\theta(\boldsymbol{\pi}|s)$ for selecting a *tour (sequence)* π given the problem instance s and the parameters θ. Moreover, we define $J(\theta|s)$ as the policy objective function, which is the total score of the tour given the instance s. Also, we use policy gradients methods to search for a local

maximum in $J(\theta|s)$ by ascending gradient policy, w.r.t parameters θ, defined as:

$$\nabla_\theta J(\theta|s) = \mathbb{E}_{p_\theta(\boldsymbol{\pi}|s)} \left[\nabla_\theta \log p_\theta(\boldsymbol{\pi}|s)(G(\boldsymbol{\pi}) - b(s))\right]$$

The value of the gradient function $\nabla_\theta J(\theta|s)$ is equal to the expected value of the multiplication between the functions score and advantage. The gradient $(\nabla_\theta \log p_\theta(\boldsymbol{\pi}|s))$ is a measure of the movement of the function in the solution space *(score function)*. While, the difference between the score $G(\boldsymbol{\pi})$ of the tour π, and the baseline $b(s)$ is the *advantage function*. To optimize the expected score we use REINFORCE (Williams, 1992) gradient estimator, and a greedy roll-out baseline.

Results and perspectives

As a preliminary result, we start with the implementation of the Graph Attention Network for the OP, and we compare their results against a MILP version of the problem (Vansteenwegen et al., 2011), using small size instances of five locations. During the training process, as the number the epochs increases, the GAN improves its performance achieving better scores as Figure 3 presents.

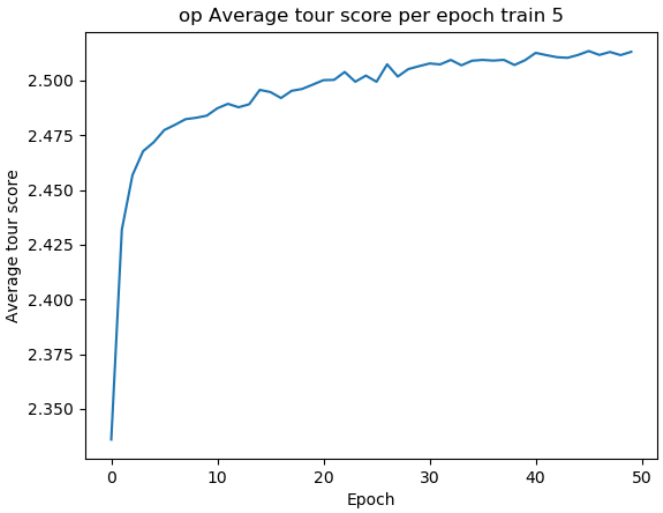


Figure 3: Average tour score per epoch during training.

Finally, both algorithms obtain the same tour scores, see Figure 4, Showing

to some extend the effectiveness of the GAN for small size instances.

	Solution (points)	
Instance	MILP	GAN
1	3	3
2	3	3
3	2	2
4	2	2
5	2	2
6	2	2
7	3	3
8	3	3
9	3	3
10	2	2

Figure 4: Comparison between MILP vs. GAN using instances of five locations.

We want to emphasize that implementing and training a sequence to sequence architecture requires a considerable investment of time. Neural Networks are powerful but complex models. Thus, the process of implementing and training these kind of models requires in-depth expertise and a great deal of experimentation. Finally, we would like to list some of the possible progress steps in the project:

1. Compare the GNA against the MILP model using instances of with $n = \{10, 50, 100\}$.
2. Compare the GNA implementation for the OP against the results reported in Kool et al. (2018).
3. Implement the hybrid algorithm to solve the TOP, and compare it against the results reported in Dang et al. (2013).
4. Modify the algorithm to solve the TOPTW, and compare it against the results reported in Amarouche et al. (2020).
5. Implement different ways to make the embeddings (GNN, GNC...etc) and compare with self-attention.

References

Amarouche, Y., Guibadj, R. N., Chaalal, E., & Moukrim, A. (2020). Effective neighborhood search with optimal splitting and adaptive memory for the team orienteering problem with time windows. *Computers & Operations Research*, *123*, 105039. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054820301568 doi: https://doi.org/10.1016/j.cor.2020.105039

Bahdanau, D., Cho, K., & Bengio, Y. (2014, September). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, arXiv:1409.0473. doi: 10.48550/arXiv.1409.0473

Bello, I., Pham, H., Le, Q. V., Norouzi, M., & Bengio, S. (2016). Neural combinatorial optimization with reinforcement learning. *ArXiv*, *abs/1611.09940*.

Chao, I.-M., Golden, B. L., & Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research*, *88*(3), 464-474. Retrieved from https://www.sciencedirect.com/science/article/pii/0377221794002894 doi: https://doi.org/10.1016/0377-2217(94)00289-4

Dang, D.-C., Guibadj, R. N., & Moukrim, A. (2013). An effective pso-inspired algorithm for the team orienteering problem. *European Journal of Operational Research*, *229*(2), 332-344. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221713001987 doi: https://doi.org/10.1016/j.ejor.2013.02.049

de Costa, P., Rhuggenaath, J., Zhang, Y., Akcay, A., & Kaymak, U. (2021). Learning 2-opt heuristics for routing problems via deep reinforcement learning. *SN Computer Science*, *2*, 388-404. Retrieved from https://link.springer.com/article/10.1007/s42979-021-00779-2#citeas doi: https://doi.org/10.1007/s42979

-021-00779-2

Gama, R., & L. Fernandes, H. (2021). A reinforcement learning approach to the orienteering problem with time windows. *Computers & Operations Research*, *133*, 105357. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054821001349 doi: https://doi.org/10.1016/j.cor.2021.105357

Golden, B., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics*, *34*, 307-318. doi: https://doi.org/10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D

Hottung, A., & Tierney, K. (2019, November). Neural Large Neighborhood Search for the Capacitated Vehicle Routing Problem. *arXiv e-prints*, arXiv:1911.09539. doi: 10.48550/arXiv.1911.09539

Khalil, E., Dai, H., Zhang, Y., Dilkina, B., & Song, L. (2017). Learning combinatorial optimization algorithms over graphs. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30, p. 6348–6358). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2017/file/d9896106ca98d3d05b8cbdf4fd8b13a1-Paper.pdf

Kool, W., van Hoof, H., & Welling, M. (2018). Attention, learn to solve routing problems! In *International conference on learning representations* (p. 1-25).

Nazari, M., Oroojlooy, A., Snyder, L., & Takac, M. (2018). Reinforcement learning for solving the vehicle routing problem. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 31, p. 9860–9870). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2018/file/9fb4651c05b2ed70fba5afe0b039a550-Paper.pdf

Nowak, A., Villar, S., Bandeira, A. S., & Bruna, J. (2018). Revised note on learning quadratic assignment with graph neural networks. In *2018 ieee data science workshop (dsw)* (p. 1-5). doi: 10.1109/DSW.2018.8439919

Roozbeh, I., Hearne, J. W., & Pahlevani, D. (2020). A solution approach to the orienteering problem with time windows and synchronisation constraints.

Heliyon, *6*(6), e04202. Retrieved from https://www.sciencedirect.com/science/article/pii/S240584402031046X doi: https://doi.org/10.1016/j.heliyon.2020.e04202

Saeedvand, S., Aghdasi, H. S., & Baltes, J. (2020). Novel hybrid algorithm for team orienteering problem with time windows for rescue applications. *Applied Soft Computing*, *96*, 106700. Retrieved from https://www.sciencedirect.com/science/article/pii/S1568494620306384 doi: https://doi.org/10.1016/j.asoc.2020.106700

Tang, H., & Miller-Hooks, E. (2005). A tabu search heuristic for the team orienteering problem. *Computers & Operations Research*, *32*(6), 1379-1407. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054803003265 doi: https://doi.org/10.1016/j.cor.2003.11.008

Vansteenwegen, P., Souffriau, W., & Oudheusden, D. V. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, *209*(1), 1-10. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221710002973 doi: https://doi.org/10.1016/j.ejor.2010.03.045

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30, p. 5998–6008). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28, p. 2692–2700). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In R. S. Sutton (Ed.), *Reinforcement learning* (pp. 5–32). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/

978-1-4615-3618-5_2 doi: 10.1007/978-1-4615-3618-5_2

Yu, Q., Fang, K., Zhu, N., & Ma, S. (2019). A matheuristic approach to the orienteering problem with service time dependent profits. *European Journal of Operational Research*, *273*(2), 488-503. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221718306842 doi: https://doi.org/10.1016/j.ejor.2018.08.007