A deep learning approach for the Team Orienteering Problem

Iván Peña-Arenas[1], Rym Guibadj[1], Cyril Fonlupt[1]

Université du Littoral Côte d'Opale, EA 4491 – LISIC – Laboratoire d'Informatique Signal et Image de la Côte d'Opale, F-62228 Calais, France

{ivan.pena-arenas, rym.guibadj, cyril.fonlupt}@univ-littoral.fr

Keywords : *Team Orienteering Problem, Combinatorial optimization, Deep learning.*

1 Introduction

The Orienteering Problem, a combinatorial challenge, is recognized as NP-Hard [1], and has various practical applications in fields such as logistics and telecommunications. Among its different variants, the Team Orienteering Problem (TOP) (see, [2]), [3], [4]) is particularly notable. Recent advances in machine learning techniques, especially in deep learning model architectures like Pointer Networks and Graph Neural Networks, have demonstrated their effectiveness in addressing complex combinatorial optimization issues [5]. These results have developed the interest in applying such techniques to combinatorial problems.

The heuristics, though efficient, are usually handcrafted and specialized and lack generalizability, require a great deal of expertise for development. In contrast, machine learning models make decisions and solve problems, extracting useful information directly from the data without specific problem knowledge. However, as the problem size and complexity increase, the performance of neural networks may decrease if we wish to keep a reasonable computational time.

To take advantage of both types solution methods, we propose an innovative approach that integrates an efficient splitting algorithm for the TOP within a deep learning framework. This scheme operates in two stages: first, a deep neural network generates a *giant tour* (a sequence of customers/locations), and subsequently, it is evaluated using the split algorithm. In addition, in the best of our knowledge these hybrid approach has not been proposed to solve the TOP.

2 Deep learning approach

In the first step, our approach utilizes the neural network to generate the giant tour. This process involves a Graph Attention Network functioning as a system encoder-decoder [6]. The encoder processes the input instance and produces the embeddings for all inputs nodes which include clients and depot. This change of representation of the instance is intended to capture the hidden structures that would be useful to solve the problem. The decoder takes as an input the encoder embeddings and the history of the already visited clients, to compute a single vector that summarizes the current solution. After, a multi-head attention module takes this vector in order to compute a probability distribution function to select the next client. By repeating this process, the decoder builds the candidate solutions step by step, selecting at each time the next client in the sequence, until all of them are visited.

In the second step, we apply an optimal split algorithm to find the set of sub-tours such that the sum of their profits is maximized. This procedure guarantees that if a set of sub-tours forming an optimal solution for the TOP is currently present as sub-sequences in the giant tour $\boldsymbol{\pi}^*$, the application of the split procedure on $\boldsymbol{\pi}^*$ will return the optimal TOP solution. Figure (1) presents the solution schema. In this sense, the neural network works under a reduced solution space, determined by the set of giant tours result of the split procedure.

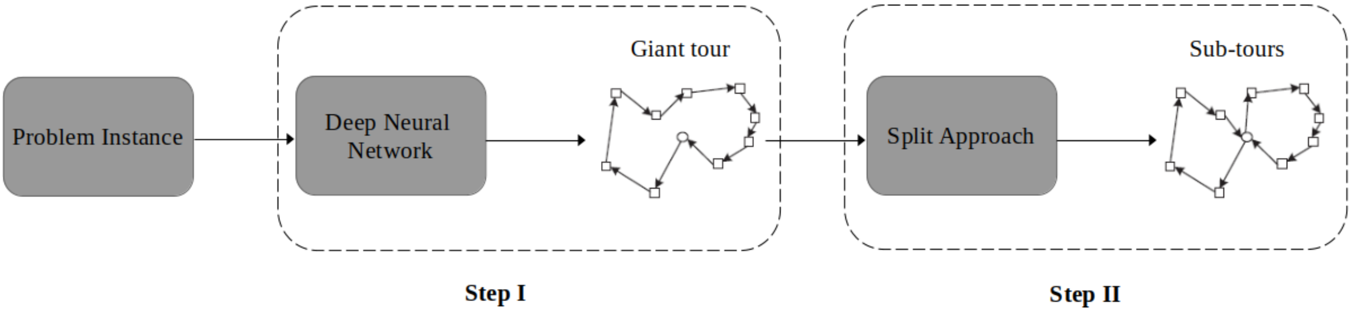

FIG. 1: Deep learning solution approach.

During the learning process the Graph Attention Network defines a stochastic policy $p_\theta(\boldsymbol{\pi}|s)$ for selecting a *tour (sequence)* $\boldsymbol{\pi}$ given a TOP instance s and the parameters θ. Later, by applying the split procedure this sequence $\boldsymbol{\pi}$ is decomposed in a set of optimal sub-tours considering the maximum length duration for a tour L. Moreover, we define $J(\theta|s)$ as the policy objective function, which is the total expected score of the sub-tours evaluated by the split procedure given the instance s.

$$J(\theta|s) = \mathbb{E}_{p_\theta(\boldsymbol{\pi}|s)}\left[(split(\boldsymbol{\pi}|s)\right]$$

We use policy gradients methods to search for a local maximum in $J(\theta|s)$ by ascending gradient policy, w.r.t parameters θ. Finally, to optimize the expected score we use REINFORCE gradient estimator, and a greedy roll-out baseline.

3 Conclusion

This work introduces a two-step deep learning approach to address the Team Orienteering Problem. The method is compared against a customized meta-heuristic, and the results demonstrate its good performance, solving instances with 50 clients on average under 8.3×10^{-3} seconds with a relative GAP of 10.42%.

References

[1] Golden, B. and Levy, L. and Vohra, R. The orienteering problem. *Naval Research Logistics*, 34 : 307-318, 1987.

[2] I-Ming Chao and Bruce L. Golden and Edward A. Wasil. The team orienteering problem. *European Journal of Operational Research*, 88(3):464-474, 1996.

[3] Hao Tang and Elise Miller-Hooks. A TABU search heuristic for the team orienteering problem. *Computers & Operations Research*, 32(6):1379-1407, 2005.

[4] Duc-Cuong Dang and Rym Nesrine Guibadj and Aziz Moukrim. An effective PSO-inspired algorithm for the team orienteering problem *European Journal of Operational Research*, 229(2):332-344, 2013.

[5] Yoshua Bengio and Andrea Lodi and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405-421, 2021.

[6] Wouter Kool and Herke van Hoof and Max Welling. Attention, Learn to Solve Routing Problems! *International Conference on Learning Representations,*1-25, 2018.