PhD proposal on
Heuristics Learning for Solving Network Optimization Problems

Supervised by

Cristina Bazgan, Université Paris-Dauphine, bazgan@lamsade.dauphine.fr

Morgan Chopin, Orange, morgan.chopin@orange.com

Pierre-Henri Wuillemin, Université Paris-Sorbonne, pierre-henri.wuillemin@lip6.fr

With the deployment of a new generation of networks, the use of *Software Defined Networking* (SDN) and *network virtualization* will be democratized in a near future and brings new challenges from a combinatorial optimization perspective [9]. These new network technologies will provide more flexibility in terms of programmability, and one could imagine to deploy and execute optimization algorithms embedded directly into network controllers (SDN controller). This is the converse to what it is usually done *i.e.* the algorithms are executed offline and the solutions are then implemented in the network (*e.g.* as new MPLS tunnels or metrics). These new paradigms to control the networks behavior are more than welcome in a context where the quality of service (QoS) has to be maintained while networks size and traffic volume are constantly growing (*HD/4K Streaming*, *Cloud Gaming*,...). Leveraging on modern networks capabilities, this thesis aims at studying and designing new optimization algorithms that lie at the interface of *machine learning* (ML) and *combinatorial optimization* (CO). This emerging research domain may offer a promising way to develop efficient algorithms that scale well with network size and adapt well to the virtualization.

State of the art & expected work: In the last few years there has been a growing interest in the development of algorithmic techniques combining ML and CO. There are actually two approaches that exist in the literature (*i*) using ML techniques to improve the efficiency of combinatorial algorithms [2, 8] or, the other way around (*ii*) improving learning techniques with the use of combinatorial optimization algorithms [6]. From the first perspective, machine learning technique can be employed in different ways:

- Learning heuristics to improve the search of optimal solutions (branching strategy, greedy algorithm, ...) [7, 5].
- Detecting the parameters that makes an instance hard to solve [11].
- Learning approximation algorithms with performance guarantee [10].

In this thesis, we are interested in the first item, and more specifically in discovering new heuristics for network optimization problems. A recent result [7] tends to show that *reinforcement learning* is a promising technique to devise greedy algorithms for graph problems like Vertex Cover, Max Cut or TSP. A natural question would be to extend those results in the context of network problems. For instance, a central problem in *traffic engineering* is the determination of metrics in an IP network that induce shortest paths routing with minimum congestion [4]. Formally, the problem is to find a set of weights to assign to the links and a set of routing paths induced by those weights such that (*i*) there is a unique shortest path for each commodity according to the identified weights and (*ii*) the network congestion is minimum. The current state-of-the-art algorithms cannot solve the problem on graphs with more than a few dozen nodes. This is partly due to the difficulty of designing hand-made heuristics that guide the resolution of the models. Hence, this problem is a good candidate for applying machine learning techniques to get efficient heuristics or, at least, help to devise them (as suggested in [7]). From a more theoretical point of view, this is a routing problem that belongs to the broader class of inverse optimization problems. This suggests that the hybrid ML/OC techniques obtained to solve this particular problem may lead to a general approach to solve inverse problems in general. Additionally, the learning model developped in this thesis will have to take as input both a graph and a traffic matrix. Regarding the state of the art, it is an open question to find a suitable embedding of both inputs altogether.

Furthermore, one of our approach to solve this problem is to use a dynamic programming algorithm based on tree decomposition [3]. More generally, computing a graph decomposition (tree or branch decomposition) followed by dynamic programming is a standard method to devise exact algorithms. The efficiency of this kind of algorithms then heavily relies on the "quality" of the decomposition, and it is usually NP-hard to find such. Hence, learning heuristics that produce good quality graph decomposition would be an other interesting research direction to explore. Some works have already been undertaken in that direction [1] but learning a heuristic that builds a suitable graph decomposition is still largely unexplored, especially in the context of network problems.

Conditions: The PhD candidate is expected to have a Master's degree or equivalent in computer science with the following skills:

- A strong background in artificial intelligence and machine learning techniques (reinforcement learning, deep learning, ...)
- Knowledge in combinatorial optimization methodologies and algorithms, mathematical programming and graph theory.
- Programming experience in one or more of the following languages : Python, C, C++.
- Experience working with at least one machine learning framework (TensorFlow, PyTorch, ...)

The thesis is fully funded by Orange. The PhD candidate will join the MORE team (*Mathematical Models for Optimization and peRformance Evaluation*) located at Orange Gardens, Châtillon. The team is composed of researchers, PhD students and software engineers aims at developping mathematical models to evaluate the performance or to solve combinatorial optimization problems in networks related problems.

Additional informations:

Expected starting date: October 1st, 2020

Application deadline: Until the position is filled

Annual gross salary : 31k€ (1st and 2nd year), 35k€ (3rd year)

More information available at `https://orange.jobs/site/en-theses`

References

[1] M. Abseher, N. Musliu, and S. Woltran. Improving the Efficiency of Dynamic Programming on Tree Decompositions via Machine Learning. *Journal Artificial Intelligence Research*, 58:829–858, 2017.

[2] Y. Bengio, A. Lodi, and A. Prouvost. Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon. In *CoRR abs/1811.06128*. 2018.

[3] A. Benhamiche and M. Chopin. Toward Scalable Algorithms for the Unsplittable Shortest Path Routing Problem. *submitted*.

[4] A. Bley, B. Fortz, E. Gourdin, K. Holmberg, O. Klopfenstein, M. Pióro, A. Tomaszewski, and H. Ümit. *Optimization of OSPF Routing in IP Networks*, pages 199–240. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[5] P. Bonami, A. Lodi, and G. Zarpellon. Learning a classification of mixed-integer quadratic programming problems. In *15th International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, (CPAIOR 2018)*, volume 10848, pages 595–604, 2018.

[6] D. S. Hochbaum. Machine learning and data mining with combinatorial optimization algorithms. In *INFORMS TutORials in Operations Research*, pages 109–129, 2018.

[7] E. B. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song. Learning combinatorial optimization algorithms over graphs. In *30th Advances in Neural Information Processing Systems (NIPS 2017)*, pages 6348–6358, 2017.

[8] A. Lodi and G. Zarpellon. On learning and branching: a survey. *TOP*, 25:207–236, 2017.

[9] N. Perrot, A. Benhamiche, Y. Carlinet, and E. Gourdin. Future networks: Overview of optimization problems in decision-making procedures. In *IGI Global.*, pages 177–207, 2018.

[10] R. Sato, M. Yamada, and H. Kashima. Approximation ratios of graph neural networks for combinatorial problems. In *32nd Annual Conference on Neural Information Processing Systems (NIPS 2019)*, pages 4083–4092, 2019.

[11] R. Sato, M. Yamada, and H. Kashima. Learning to sample hard instances for graph algorithms. In *11th Asian Conference on Machine Learning (ACML 2019)*, pages 503–518, 2019.