Internship in Machine Learning and Mathematical Programming

Improving the representation of binary linear problems via graph neural networks using strengthening inequalities

Context Machine Learning (ML) for Combinatorial Optimization (CO) is a fast growing field with many applications [1, 5]. One consists in generating near-optimal solutions of Binary Linear Problems (BLP) by learning a score for each variable, and then applying a greedy algorithm to find a solution maximizing the total score instead of the original objective function. This allows to obtain solutions satisfying hard constraints, and if the score is related to the probability to be in an optimal solution, to get high quality solutions. The score prediction is usually computed using Graph Neural Networks (GNN) to get a representation of the BLP [2]. The computed feature of each variable is then independently used to obtain a 0/1 prediction via a Multi Layer Perceptron and a threshold [3].

Internship description Modern mathematical solvers use cutting plane based algorithms to solve BLP [4]. These consist in strengthening the continuous relaxation by introducing valid inequalities that are violated by the current fractional solution of the relaxation. These provide better dual bounds and add information linking the variables. The objective of this internship is to computationally evaluate the relevance of adding such strengthening inequalities in the formulation before the score prediction.

The evaluation will be done on the maximum independent set problem (MIS). This problem consists, given an undirected graph, in determining a subset of pairwise non-adjacent nodes of maximum cardinality. It can be formulated using a BLP containing a variable per node and a constraint per edge. However, one can add clique inequalities to reinforce the formulation. These inequalities ensure that at most one vertex per clique is selected in a solution. These inequalities correspond to adding new nodes in the GNN, and to densifying the resulted graph.

The internship is to computationally evaluate how adding these constraints improve the score prediction. Moreover, since there may exist an exponential number of such constraints, how can we choose the ones to incorporate in the model? A way could be to choose those violated by the optimal solution linear relaxation. However, since a drawback of GNN is the number of layers necessary to share the information over the whole graph, could we choose those that tend to decrease the number of necessary layers by decreasing the diameter of the GNN?

Application We are looking for a candidate with either CO background (master level) with very good knowledge of Machine Learning methods for CO, or with a strong ML background willing to adapt recent models to CO tasks. We expect proficiency with python and deep learning libraries such as pytorch. For additional information contact lacroix@lipn.fr. If you are interested please attach to your application a CV, a cover letter and a transcript of Master level marks.

References

[1] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost, *Machine learning for combinatorial optimization: A methodological tour d'horizon*, European Journal of Operational Research **290** (2021), no. 2, 405–421.

[2] Maxime Gasse, Didier Chetelat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi, *Exact Combinatorial Optimization with Graph Convolutional Neural Networks*, Advances in Neural Information Processing Systems (H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[3] Vinod Nair, Sergey Bartunov, Felix Gimeno, Ingrid von Glehn, Pawel Lichocki, Ivan Lobov, Brendan O'Donoghue, Nicolas Sonnerat, Christian Tjandraatmadja, Pengming Wang, Ravichandra Addanki, Tharindi Hapuarachchi, Thomas Keck, James Keeling, Pushmeet Kohli, Ira Ktena, Yujia Li, Oriol Vinyals, and Yori Zwols, *Solving Mixed Integer Programs Using Neural Networks*, arXiv:2012.13349 [cs, math] (2021), arXiv: 2012.13349.

[4] Integer Programming, *50 Years of Integer Programming 1958-2008*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[5] Jiayi Zhang, Chang Liu, Xijun Li, Hui-Ling Zhen, Mingxuan Yuan, Yawen Li, and Junchi Yan, *A survey for solving mixed integer programming via machine learning*, Neurocomputing **519** (2023), 205–217 (en).