**PhD title: "Embedding Structured Machine Learning in Decomposition Techniques"**

**Advisor:** Roberto Wolfler Calvo, LIPN (roberto.wolfler@lipn.univ-paris13.fr)
**Co-advisor:** Antonio Frangioni, Università di Pisa (frangio@di.unipi.it)
**Co-advisor:** Mathieu Lacroix, LIPN (lacroix@lipn.univ-paris13.fr)

# Research proposal

State-of-the-art solvers for hard optimization problems are highly complex software systems, composed by many sophisticated algorithms that have to work synergistically in order to reach maximum efficiency. Machine Learning (ML) has been recently widely used in order to improve mixed integer programming (MIP) solvers [1]. For instance, ML has been successfully applied to improve branch-and-bound methods by helping to choose the variable to branch on or the node of the branch-and-bound tree to process [2] or to decide when to search primal bounds (feasible solutions) [3]. ML is also used to guess whether a problem can be efficiently tackled by a MIP solver [4, 5], allowing dynamic restarts in branch-and-bound methods [5], or to automatically derive useful Gomory cuts to strengthen the linear relaxation [6].

The vast majority of combinatorial optimization algorithms present a certain degree of *modularity*, i.e., they consists of several blocks/routines that solve simpler problems repetitively. A well developed class of algorithms capable of exploiting modularity are decomposition methods, among which chiefly Lagrangian relaxation/Dantzig-Wolfe reformulation/Column Generation and Benders' decomposition As the name suggests, decomposition algorithms rely on decomposing a problem into distinct subproblems linked by a master problem. The resolution method alternates between solving the master problem and the subproblems until the optimum is found. The subproblems are solved a large number of times during the resolution process; each time, they may differ "little" from the problems solved at the previous iterate.

Up to now, ML is barely used to improve the performance of decomposition based algorithms. The only related works we are aware of are the followings: learning whether to apply Dantzig-Wolfe decomposition [7], learning to select columns to add in the master problem in column generation [8], or learning a performance function which is then used to define an optimization problem whose solution gives an optimal configuration (with respect to the approximated performance function) [9, 10, 11]. The objective of this PhD is to enhance decomposition algorithms with the use of machine learning. More specifically, the objective is to focus on exact methods based on Lagrangian relaxation. Such algorithms are widely used to solve complex combinatorial optimization problems having a block structure and give satisfactory computational results in terms of running time and size of instances solved to optimality. For instance, such methods are the best at solving unit commitment problem, multicommodity network design problems, quadratic knapsack problem, etc [12, 13, 14, 15, 16, 17].

The objective of this PhD is to understand what are good parameter choices that are specific for Lagrangian based algorithm using machine learning. This project can be decomposed into two main research questions. The first one focuses on using of machine learning to speed up the computation of the Lagrangian relaxation whereas the second extends the use of ML when Lagrangian relaxation latter is embedded within a branch-and-bound framework.

## Q1: How ML can help solving Lagrangian relaxations?

Given a mixed-integer programming linear problem, Lagrangian relaxation consists in dualizing constraints in order to obtain a relaxed problem whose resolution, parametrized by Lagrangian multipliers, gives a lower bound (in case of minimization) of the optimum. The dual Lagrangian problem reduces to determining the value of the Lagrangian multipliers giving the highest bound. This dual Lagrangian problem is solved using iterative methods such as subgradient or bundle methods: starting from an initial value for the Lagrangian multipliers, the relaxed problem is solved to optimality; from this optimal solution, new values are computed for the multipliers. This process is repeated until optimal Lagrangian multipliers are found.

The efficiency of the algorithm then strongly relies on the choice of the initial values used for the Lagrangian multipliers and additional parameters such that the policy used to update the step size and the precision required at each iteration. Different approaches exist for setting these values based on human knowledge and experience. This PhD aims at providing a framework to learn how to set these values from the previous resolutions of instances of the same type.

There is an ongoing work in this direction in collaboration with Antonio Frangioni. The objective is to learn how to initialize the Lagrangian multipliers of a Lagrangian relaxation for the multicommodity capacitated fixed-charge network design problem (MCND). In this problem, given a graph and a set of commodities, a subset of arcs must be selected and the commodities must be routed from their origin to their destination using these arcs. The selection of an arc incurs a fixed cost and provides a predefined capacity available to all commodities. In addition, a cost stems from routing a unit of commodity on an arc which depends on the arc and the commodity. The problem is to find a minimum cost solution such that the demand for each commodity is met and the capacity on each arc is not exceeded.

A classic Lagrangian relaxation for the MCND problem leads to a relaxed problem which decomposes by arc and which can be solved by computing a continuous knapsack problem associated with each arc. We propose to learn the initialization using structured Learning. This latter is an umbrella term for supervised machine learning techniques that involves predicting structured objects, rather than scalar discrete or real values. In structured learning, the inference phase, which given an unseen instance consists in outputting the desired result (in our case the dual Lagrangian initial values), is performed by solving an auxiliary combinatorial problem. In this work, the learnt parameters are used to construct from an unseen MCND instance a set of minimum cost flow problems (one per commodity) which are solved to optimality. The Lagrangian multipliers for this new MCND instance are initialized with the potentials derived from the computed flows.

Initializing Lagrangian values with flow potentials without learning experimentally works [18, 19] but using learning to set the capacities in the flow problems should greatly improve this initialization as it would allow to take into account the other commodities when solving the flow problem of a commodity.

The first step of this PhD is to pursue this work by implementing the algorithm. A three months internship with a second year of engineering-school student will start soon under the supervision of Mathieu Lacroix and Roberto Wolfler Calvo whose objective is to provide training data points for the supervised learning. The next step is to continue this approach by learning the other parameters as well as to apply this approach to other combinatorial optimization problems such as unit commitment problem for instance.

## Q2: How ML can enhance Lagrangian based exact methods?

Lagrangian relaxation is usually used within a branch-and-bound procedure to produce a bound at each node of the branch-and-bound tree (instead of computing the linear relaxation).

In this context, a first step is to use the same procedure developed in Q1 in all the nodes of the branch-and-bound tree. However, one can exploit the peculiarities of the branch-and-bound algorithm. Indeed, in a Lagrangian based branch-and-bound algorithm, the Lagrangian Relaxation is computed many times, with "slight and localised" changes in the data. Hence, it could be useful to adopt an online learning approach. Online machine learning is a generic method of machine learning in which data becomes available in a sequential order and is used to update the model for future data at each step. In our situation, given an instance, the learning could take profit of the data generated by the resolution of the Lagrangian relaxation in the nodes of the branch-and-bound tree already processed for this instance in order to adjust the learnt model.

A second crucial parameter when incorporating Lagrangian relaxation into a branch-and-bound algorithm is which precision which level of precision to have when solving the Lagrangian dual problem at each node of the tree. The choice relative to the precision can be interpreted as a trade-off between the quality of the dual bound and the time required to compute it. A good dual bound would allow to improve the probability of pruning the nodes and keep the size of the branch-and-bound tree under control. Therefore, an interesting aspect to investigate is to learn what is the right level of precision to be imposed at each node to have a good probability of pruning the current node.

Finally, a key component of branch-and-bound algorithms are the procedures that derive from the current node of the branch-and-bound tree solutions of high quality. Even if these solutions are not optimal, they allow to speed up the overall resolution process by decreasing the size of the branch-and-bound tree thanks to pruning. Recent works [20, 21] have shown that ML can be used to produce very high quality solutions of combinatorial optimization problems by solving a relaxed version of the problem, this version being constructed using learning. A direction of research could be to use the Lagrangian problem as the relaxed version. In this case, the challenge is to learn how to construct the Lagrangian problem in such a way that its solution can be transformed in a feasible solution of the whole problem.

### Required skills

The PhD student should have a master degree in combinatorial optimization or operational research. A background in machine learning and programming skills in C++ will be much appreciated.

## References

[1] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405–421, 2021.

[2] A. Lodi and G. Zarpellon, "On learning and branching: a survey," *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, vol. 25, pp. 207–236, July 2017.

[3] E. B. Khalil, B. Dilkina, G. L. Nemhauser, S. Ahmed, and Y. Shao, "Learning to run heuristics in tree search," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017* (C. Sierra, ed.), pp. 659–666, ijcai.org, 2017.

[4] M. Fischetti, A. Lodi, and G. Zarpellon, "Learning milp resolution outcomes before reaching time-limit," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research* (L.-M. Rousseau and K. Stergiou, eds.), (Cham), pp. 275–291, Springer International Publishing, 2019.

[5] D. Anderson, G. Hendel, P. L. Bodic, and M. Viernickel, "Clairvoyant restarts in branch-and-bound search using online tree-size estimation," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 1427–1434, AAAI Press, 2019.

[6] Y. Tang, S. Agrawal, and Y. Faenza, "Reinforcement learning for integer programming: Learning to cut," 2019.

[7] M. Kruber, M. E. Lübbecke, and A. Parmentier, "Learning when to use a decomposition," in *Integration of AI and OR Techniques in Constraint Programming* (D. Salvagnin and M. Lombardi, eds.), (Cham), pp. 202–210, Springer International Publishing, 2017.

[8] M. Morabit, G. Desaulniers, and A. Lodi, "Machine-learning-based column selection for column generation," *Les cahiers du GERARD*, vol. G-2020-29, 2020.

[9] A. Frangioni and L. Perez Sanchez, "Searching the best (formulation, solver, configuration) for structured problems," in *Complex Systems Design & Management: Proceedings of the First International Conference on Complex Systems Design & Management CSDM 2010*, pp. 85–97, Springer-Verlag, 2010.

[10] G. Iommazzo, C. D'Ambrosio, A. Frangioni, and L. Liberti, "A learning-based mathematical programming formulation for the automatic configuration of optimization solvers," in *Lecture Notes in Computer Science, 6th International Conference on Machine Learning, Optimization and Data science - LOD 2020*, Springer-Verlag, 2020.

[11] G. Iommazzo, C. D'Ambrosio, A. Frangioni, and L. Liberti, "Learning to configure mathematical programming solvers by mathematical programming," in *Lecture Notes in Computer Science, 14th Learning and Intelligent Optimization, LION 14*, Springer-Verlag, 2020.

[12] A. Borghetti, A. Frangioni, F. Lacalandra, A. Lodi, S. Martello, C. Nucci, and A. Trebbi, "Lagrangian relaxation and tabu search approaches for the unit commitment problem," in *2001 IEEE Porto Power Tech Proceedings (Cat. No. 01EX502)*, vol. 3, pp. 7–pp, IEEE, 2001.

[13] R. S. Shibasaki, M. Baïou, F. Barahona, P. Mahey, and M. C. de Souza, “Lagrangian bounds for large-scale multicommodity network design: a comparison between volume and bundle methods,” *International Transactions in Operational Research*, vol. 28, no. 1, pp. 296–326, 2021.

[14] T. G. Crainic, A. Frangioni, and B. Gendron, “Bundle-based relaxation methods for multicommodity capacitated fixed charge network design,” *Discrete Applied Mathematics*, vol. 112, no. 1-3, pp. 73–99, 2001.

[15] B. Ghaddar, J. Naoum-Sawaya, A. Kishimoto, N. Taheri, and B. Eck, “A lagrangian decomposition approach for the pump scheduling problem in water networks,” *European Journal of Operational Research*, vol. 241, no. 2, pp. 490–501, 2015.

[16] A. Billionnet and É. Soutif, “An exact method based on lagrangian decomposition for the 0–1 quadratic knapsack problem,” *European Journal of operational research*, vol. 157, no. 3, pp. 565–575, 2004.

[17] C. Zhang, Y. Gao, L. Yang, Z. Gao, and J. Qi, “Joint optimization of train scheduling and maintenance planning in a railway network: A heuristic algorithm using lagrangian relaxation,” *Transportation Research Part B: Methodological*, vol. 134, pp. 64–92, 2020.

[18] A. Borghetti, A. Frangioni, F. Lacalandra, and C. A. Nucci, “Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment,” *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 313–323, 2003.

[19] A. Frangioni and B. Gendron, “A stabilized structured dantzig–wolfe decomposition method,” *Mathematical Programming*, vol. 140, no. 1, pp. 45–76, 2013.

[20] A. Parmentier, “Learning to approximate industrial problems by operations research classic problems,” *HAL, archives ouvertes*, 2019.

[21] A. Parmentier and V. t’Kindt, “Learning to solve the single machine scheduling problem with release times and sum of completion times,” *arXiv preprint arXiv:2101.01082*, 2021.