On Learning Node Selection in a Branch and Bound Algorithm

Marc Etheve[1,3] Zacharie Alès[2,3] Côme Bissuel[1] Olivier Juan[1]
Safia Kedad-Sidhoum[3]

[1] EDF R&D, France
{marc.etheve, come.bissuel, olivier.juan}@edf.fr
[2] ENSTA Paris, Institut Polytechnique de Paris, France
zacharie.ales@ensta-paris.fr
[3] CNAM Paris, CEDRIC, France
safia.kedad_sidhoum@cnam.fr

Mots-clés : *Machine Learning, Mixed Integer Linear Programming, Branch and Bound, Node Selection Strategy*

1 Introduction

Over the past few years, the study of interactions between Combinatorial Optimization and Machine Learning has been a growing field of research. In particular, learning the heuristically made choices along a Branch and Bound algorithm has been targeted by the literature (see [1] for a survey). The majority of the approaches is dedicated to learning to imitate the behaviour of solver's heuristics (see for instance [2, 3, 4, 5, 6]), using imitation learning.
On the contrary, quite few address the task of discovering new policies, undoubtedly due to the difficulty of the challenge. [7] learns a cutting strategy amongst Gomory's cutting planes, In this preliminary work, we extend the methodology presented in [8] to the node selection strategy, using reinforcement learning to discover such strategy. An alternative approach, using supervised learning, is also proposed.

2 Framework

Our goal is to design a node selection policy in a Branch and Bound procedure. The considered instances should come from a same given Mixed Binary Linear Problem. As a consequence, they share the same structure, *i.e.* the same dimensions and null coefficients. The objective of the designed node selection policy is to build trees of minimal size on such instances.
As in [9], we focus on learning the selection of one of the two node's children, once the branching variable has been selected. To ensure the coherence of the proposed methods, we enforce the use of Depth-First Search (DFS) and solve instances to optimality.

3 Node selection using Supervised Learning

Let us make two hypotheses in this section :
H1 : the branching strategy is only dependent on the current branching state.
H2 : the optimal integer solution is unique.

When dealing with the node selection strategy in a simple B&B procedure, one can first acknowledge that, under H1, the node selection strategy has no impact on the tree size once the optimal solution of the instance has been found.
As a consequence, when the optimal integer solution is unique (H2), choosing at each node the

child leading to the optimal solution (if possible) will minimize the number of nodes processed to reach it, hence the size of the entire tree.
Our approach is then to predict, at each node, the optimal integer solution of the according sub-problem. Once the branching variable has been selected, we then visit first the child node corresponding to the predicted solution. A classifier is trained to predict the optimal integer solution (if any) at each node, the training data being drawn from the according node selection strategy.

4 Node selection using Reinforcement Learning

In addition to the hypotheses previously mentioned, a limit to the supervised approach is the absence of the notion of risk. As a classifier cannot predict perfectly the optimal node, it seems to be important to limit the occurrence of harmful errors.
Under DFS, the optimal choice presented above is equivalent to visiting first the child which will minimize the subtree size rooted in the current node. On the contrary, a harmful error would be choosing a child which leads to a large subtree.
Using Approximate Q-learning, we train an agent to design the node selection policy. The Q-value at current node is defined as a discounted version of the subtree size and the policy is given by the minimization of the predicted Q-value. Doing so, the agent can quantify the cost of a bad node selection and hence avoid making such choices.

5 Discussion

Both approaches have different advantages and drawbacks. On the one hand, supervised learning allows to avoid the sample complexity inherent to the reinforcement learning approach, as the targets are stationary. However, it may suffer from problem's symmetries and approximation errors. On the other hand, reinforcement learning quantifies the risk and handles symetries, but at the price of a higher sample complexity.
Experiments are made on simple real-world problems provided by Electricité de France.

Références

[1] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization : a methodological tour d'horizon," *arXiv preprint arXiv :1811.06128*, 2018.

[2] H. He, H. Daume III, and J. M. Eisner, "Learning to search in branch and bound algorithms," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3293–3301, Curran Associates, Inc., 2014.

[3] G. Zarpellon, J. Jo, A. Lodi, and Y. Bengio, "Parameterizing branch-and-bound search trees to learn branching policies," *arXiv preprint arXiv :2002.05120*, 2020.

[4] E. B. Khalil, P. Le Bodic, L. Song, G. Nemhauser, and B. Dilkina, "Learning to branch in mixed integer programming," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[5] M.-F. Balcan, T. Dick, T. Sandholm, and E. Vitercik, "Learning to branch," *arXiv preprint arXiv :1803.10150*, 2018.

[6] M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi, "Exact combinatorial optimization with graph convolutional neural networks," *arXiv preprint arXiv :1906.01629*, 2019.

[7] Y. Tang, S. Agrawal, and Y. Faenza, "Reinforcement learning for integer programming : Learning to cut," in *Proceedings of the 37th International Conference on Machine Learning*

(H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 9367–9376, PMLR, 13–18 Jul 2020.

[8] M. Etheve, Z. Alès, C. Bissuel, O. Juan, and S. Kedad-Sidhoum, "Reinforcement learning for variable selection in a branch and bound algorithm," in *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pp. 176–185, Springer, 2020.

[9] K. Yilmaz and N. Yorke-Smith, "Learning efficient search approximation in mixed integer branch and bound," *arXiv preprint arXiv :2007.03948*, 2020.