Machine learning for Combinatorial Optimization

Thesis are sorted in increasing order of difficulty.

Learning to branch for the identical machines job scheduling

In Identical Parallel Machines Job Scheduling (IPMJS), jobs are assigned to multiple machines with identical processing capabilities, meaning that each job takes the same amount of time on any machine. The objective is to distribute jobs across machines to minimize the maximum completion time.

The IPMJS problem is commonly addressed using the branch-and-bound (B&B) method, an optimization technique that systematically explores all possible job assignments to find the optimal solution. B&B operates by branching—dividing the problem into smaller subproblems—and bounding, which involves setting limits to discard branches that cannot yield better solutions than the current best.

Within this framework, a branching rule acts as a heuristic, providing a practical, experience-based approach to prioritize which subproblems to explore first. Although this heuristic may not guarantee finding the optimal solution faster, it helps prioritize promising paths, improving efficiency in reaching good solutions.

Thus, it is worthwhile to examine the impact of a learned heuristic in the B&B method for this specific problem, compared with heuristics that rely on theoretical insights or long-standing empirical evidence.

High-Level plan for the Thesis

- Initial meeting with the group to discuss and define objectives
- Study of B&B and related branching rules, [1]
- Review of IPMJS and current state-of-the-art algorithms, Chap. 10 of [3]
- Study PySCIPOpt for B&B applications, [2]
- Development of a suitable learning procedure for the task and analysis of its specifics
- Comparison with state-of-the-art methods for B&B
- **Extra:** Thoroughly evaluate the ML model's recommendations and assess if they provide theoretical guarantees

Neural generation of hard SAT instances

TBD

Generating vertices of 0-1 polytope

TBD

References

[1] Michele Conforti, Gérard Cornuéjols, Giacomo Zambelli, Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming models*. Springer, 2014.

[2] Stephen Maher, Matthias Miltenberger, João Pedro Pedroso, Daniel Rehfeldt, Robert Schwarz, and Felipe Serrano. PySCIPOpt: Mathematical programming in python with the SCIP optimization suite. In *Mathematical Software – ICMS 2016*, pages 301–307. Springer International Publishing, 2016.

[3] Vijay V Vazirani. Approximation algorithms, 2001.