

Master 2 Research Internship

Graph Sparsification for Combinatorial Optimization

Advisor: Hamida SEBA (hamida.seba@univ-lyon1.fr)

Location : Univ. Lyon 1, LIRIS UMR 5205, Graphs and Applications research team

Context and motivation

Many fundamental problems on graphs are computationally difficult. Problems such as Maximum Independent Set (MIS), Minimum Vertex Cover (MVC), Maximum Cut (MaxCut), Minimum Dominating Set (MDS) or graph coloring are NP-hard and become particularly challenging on large graphs. Exact algorithms may become impractical, while approximation algorithms and heuristics often have to explore a very large search space.

Recent progress in Artificial Intelligence has significantly changed the way such problems can be approached. Machine learning does not change their worst-case computational complexity, but it provides new ways of learning heuristics, guiding exact algorithms and exploiting structures that repeatedly occur across problem instances.

A seminal example is the work of Khalil et al. [1], which combines graph embeddings and reinforcement learning to learn constructive heuristics for Minimum Vertex Cover, Maximum Cut and Traveling Salesman Problem. Instead of manually designing all the decisions of the heuristic, the algorithm learns a policy from training instances.

Machine learning can also be integrated into exact optimization algorithms. Gasse et al. [2], for example, use Graph Neural Networks (GNNs) to learn branching decisions in branch-and-bound algorithms. Their results show that learned policies can generalize to instances larger than those encountered during training and can outperform expert-designed branching rules on some large instances. These developments have contributed to a broader research field combining machine learning and combinatorial optimization [3,4].

More recent work investigates whether learning can also improve the reduction of the problem before it is solved. Großmann et al. [5] use GNNs to accelerate reduction rules for the Maximum Weight Independent Set problem. The objective is not to replace the combinatorial algorithm, but to use learning to identify promising reductions more efficiently.

Another recent development concerns transfer between combinatorial problems. Cantürk et al. [6] investigate transfer between MVC, MIS and Maximum Clique, as well as multi-task learning involving MaxCut, MDS and graph coloring. Their results suggest that representations learned for one combinatorial problem can accelerate learning for related problems when known computational relationships between the problems are exploited.

These developments raise a natural question: can AI also learn how to reduce the graph itself before the combinatorial problem is solved?

Very recent results suggest that this direction is feasible. GES-TSP [9], for example, proposes a learning-based edge sparsification approach specifically for large Euclidean Traveling Salesman Problem instances. The learned sparsification exploits instance-specific information before the optimization problem is solved. The authors report pruning up to 95% of the edges on one benchmark while maintaining an optimality gap below 1%, with even higher pruning rates on some large TSPLIB instances.

Objectives

The main objective is to study the relationship between graph sparsification and combinatorial optimization.

For a combinatorial problem (P), we want to understand which edges of (G) are relevant to the computation of an optimal or near-optimal solution. In particular, we will investigate whether it is possible to construct a substantially smaller graph (G_P) such that solving (P) on (G_P) provides sufficient information to recover a good solution on (G).

This raises several fundamental questions:

- How sparse can (G_P) be?
- Which structural properties must be preserved?
- Can approximation guarantees be derived from properties of the sparsifier?
- Can a solution obtained on (G_P) be efficiently transformed into a solution on (G)?
- How does the answer depend on the considered combinatorial problem?

Candidate profile

The internship is intended for a Master 2 student with a strong background in theoretical computer science.

Knowledge of graph algorithms, complexity theory, combinatorial optimization and approximation algorithms is particularly relevant. Knowledge of machine learning or Graph Neural Networks is useful but not mandatory.

The primary focus of the internship remains on the algorithmic and combinatorial structure of the problem. Machine learning will be used as a tool when it provides new possibilities for identifying and exploiting reductions that are difficult to design manually.

References

[1] Khalil, E.B., Dai, H., Zhang, Y., Dilkina, B., Song, L. *Learning Combinatorial Optimization Algorithms over Graphs*. NeurIPS, 2017.

- [2] Gasse, M., Chételat, D., Ferroni, N., Charlin, L., Lodi, A. *Exact Combinatorial Optimization with Graph Convolutional Neural Networks*. NeurIPS, 2019.
- [3] Bengio, Y., Lodi, A., Prouvost, A. *Machine Learning for Combinatorial Optimization: A Methodological Tour d’Horizon*. European Journal of Operational Research, 290(2), 405–421, 2021.
- [4] Mazyavkina, N., Sviridov, S., Ivanov, S., Burnaev, E. *Reinforcement Learning for Combinatorial Optimization: A Survey*. Computers & Operations Research, 134, 105400, 2021.
- [5] Großmann, E., Langedal, K., Schulz, C. *Accelerating Reductions Using Graph Neural Networks for the Maximum Weight Independent Set Problem*. SIAM Conference on Applied and Computational Discrete Algorithms (ACDA), pp. 155–168, 2025.
- [6] Cantürk, S., Sabourin, T., Wenkel, F., Perlmutter, M., Wolf, G. *Can Computational Reducibility Lead to Transferable Models for Graph Combinatorial Optimization?* ICML, 2026.
- [7] Spielman, D.A., Teng, S.H. *Spectral Sparsification of Graphs*. SIAM Journal on Computing, 40(4), 981–1025, 2011.
- [8] Spielman, D.A., Srivastava, N. *Graph Sparsification by Effective Resistances*. SIAM Journal on Computing, 40(6), 1913–1926, 2011.
- [9] Chen, T., Li, X. *GES-TSP: Graph Edge Sparsification for TSP*. Preprint, 2026.