Finding Hard Instances is Hard

Your task is to make it easy

Most combinatorial optimization problems are NP-hard, meaning that there exists a family of instances which requires exponential-time to solve them exactly (unless $\mathsf{P} = \mathsf{NP}$). However, in practice, a large part of the instances can be solved efficiently. It is also known that random instances of a problem are often easy to solve (see also the Transition Phase phenomenon).

The goal of this project is to generate a family of hard (feasible) instances, *without* specific knowledge of the underlying problem (without expert rules or specific reductions), and understand why these instances are hard.

Hard instances are of importance to compare different algorithms solving the same problem or to improve the performances of an algorithm solving a specific problem (and understand where it struggles) or to disprove some graph conjectures.

Our approach will be to move these discrete problems to the continuous space and try different approaches like classical supervised Machine Learning techniques (to predict how to construct instances) or Reinforcement Learning (Deep Q-learning, curiosity-driven learning...). We can also think of starting from already known hard instances and try to improve its hardness by incremental changes (hill climbing) or Monte Carlo Search. We will have to test different "measure of hardness". This could be the time taken by an algorithm (but it is very dependent to a given solver and could not give an important information on the intrinsic difficulty of an instance), or the depth of a search tree in a solver, or the number of backtracks needed by a solver. A challenging issue will be to get enough data while the instances should be hard to solve (and therefore require time to compute).

Required skills

- Advanced knowledge in Machine Learning and Reinforcement Learning (M2 level)
- Basic knowledge in Combinatorial Optimization Problems (Knapsack, TSP, SAT...)
- Good knowledge in some programming language

Practical information

The internship will be in LAMSADE, inside Paris Dauphine University. It will be supervised by Florian Sikora, Benjamin Negrevergne and Florian Yger, *firstname.lastname*@dauphine.fr. Submit your application with your relevent grades. Salary will be according to the french internship rules.

The student will have access to the LAMSADE computing ressources.

References

[CM96] Stephen A. Cook and David G. Mitchell. Finding hard instances of the satisfiability problem: A survey. In Ding-Zhu Du, Jun Gu, and Panos M. Pardalos, editors, *Satisfiability Problem: Theory and Applications, Proceedings of a DIMACS Workshop, Piscataway, New Jersey, USA, March 11-13, 1996*, volume 35 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–17. DIMACS/AMS, 1996.

[EFK+20] Jorel Elmiger, Lukas Faber, Pankaj Khanchandani, Oliver Paul Richter, and Roger Wattenhofer. Learning lower bounds for graph exploration with reinforcement learning. In *Learning Meets Combinatorial Algorithms at NeurIPS2020*, 2020.

[GM20] Marc Goerigk and Stephen J. Maher. Generating hard instances for robust combinatorial optimization. *Eur. J. Oper. Res.*, 280(1):34–45, 2020.

[MDD+19] Bogdan Mazoure, Thang Doan, Audrey Durand, Joelle Pineau, and R. Devon Hjelm. Leveraging exploration in off-policy algorithms via normalizing flows. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 430–444. PMLR, 2019.

[RC22] Milo Roucairol and Tristan Cazenave. Refutation of spectral graph theory conjectures with monte carlo search. *CoRR*, abs/2207.03343, 2022.

[RHvL20] Luca Rendsburg, Holger Heidrich, and Ulrike von Luxburg. Netgan without GAN: from random walks to low-rank approximations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8073–8082. PMLR, 2020.

[SCM21] Kate Smith-Miles, Jeffrey Christiansen, and Mario Andrés Muñoz. Revisiting *where are the hard knapsack problems?* via instance space analysis. *Comput. Oper. Res.*, 128:105184, 2021.

[SMHL10] Kate Smith-Miles, Jano van Hemert, and Xin Yu Lim. Understanding tsp difficulty by learning from evolved instances. In *International conference on learning and intelligent optimization*, pages 266–280. Springer, 2010.

[SYK19] Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Learning to sample hard instances for graph algorithms. In Wee Sun Lee and Taiji Suzuki, editors, *Proceedings of The 11th Asian Conference on Machine Learning, ACML 2019, 17-19 November 2019, Nagoya, Japan*, volume 101 of *Proceedings of Machine Learning Research*, pages 503–518. PMLR, 2019.

[vH06] Jano I. van Hemert. Evolving combinatorial problem instances that are difficult to solve. *Evol. Comput.*, 14(4):433–462, 2006.

[Wag21] Adam Zsolt Wagner. Constructions in combinatorics via neural networks. *CoRR*, abs/2104.14516, 2021.

[XBHL05] Ke Xu, Frédéric Boussemart, Fred Hemery, and Christophe Lecoutre. A simple model to generate hard satisfiable instances. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 337–342. Professional Book Center, 2005.