Easily generate hard instances

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, especially SAT instances. 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 (starting from SAT problems) to the continuous space and try different approaches like classical supervised Machine Learning techniques (to predict how to construct hard instances) or Reinforcement Learning (Deep Q-learning, curiosity-driven learning...) and to compare it to purely combinatorial generators (and to random solutions). We can also think of starting from already known hard instances (SAT challenge) and try to improve its hardness by incremental changes (hill climbing) or Monte Carlo Search. We can also exploit known structures of SAT instances (backdoors, backbones, treewidth...). We will also need scalability, i.e. learning on small instances and be able to generate large instances (transfer).

Required skills

- Advanced knowledge in Machine Learning and Reinforcement Learning (Master level)
- Good knowledge in Combinatorial Optimization
- 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.

The student will have access to the LAMSADE computing ressources.

References

[1] Carlos Ansótegui, Maria Luisa Bonet, Jordi Levy, and Felip Manyà. Measuring the hardness of SAT instances. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 222–228. AAAI Press, 2008. URL: http://www.aaai.org/Library/AAAI/2008/aaai08-035.php.

[2] 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. doi:10.1090/dimacs/035/01.

[3] 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. URL: https://openreview.net/forum?id=kI4rsNdppSs.

[4] Guillaume Escamocher, Barry O'Sullivan, and Steven David Prestwich. Generating difficult SAT instances by preventing triangles. *CoRR*, abs/1903.03592, 2019. URL: http://arxiv.org/abs/1903.03592, arXiv:1903.03592.

[5] Marc Goerigk and Stephen J. Maher. Generating hard instances for robust combinatorial optimization. *Eur. J. Oper. Res.*, 280(1):34–45, 2020. doi:10.1016/j.ejor.2019.07.036.

[6] Satoshi Horie and Osamu Watanabe. Hard instance generation for SAT (extended abstract). In Hon Wai Leong, Hiroshi Imai, and Sanjay Jain, editors, *Algorithms and Computation, 8th International Symposium, ISAAC '97, Singapore, December 17-19, 1997, Proceedings*, volume 1350 of *Lecture Notes in Computer Science*, pages 22–31. Springer, 1997. doi:10.1007/3-540-63890-3“4.

[7] Yang Li, Xinyan Chen, Wenxuan Guo, Xijun Li, Wanqian Luo, Junhua Huang, Hui-Ling Zhen, Mingxuan Yuan, and Junchi Yan. Hardsatgen: Understanding the difficulty of hard SAT formula generation and A strong structure-hardness-aware baseline. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye, editors, *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 4414–4425. ACM, 2023. doi:10.1145/3580305.3599837.

[8] 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. URL: http://proceedings.mlr.press/v100/mazoure20a.html.

[9] Nathan Mull, Daniel J. Fremont, and Sanjit A. Seshia. On the hardness of SAT with community structure. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 141–159. Springer, 2016. doi:10.1007/978-3-319-40970-2“10.

[10] 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. URL: http://proceedings.mlr.press/v119/rendsburg20a.html.

[11] Milo Roucairol and Tristan Cazenave. Refutation of spectral graph theory conjectures with monte carlo search. In Yong Zhang, Dongjing Miao, and Rolf H. Möhring, editors, *Computing and Combinatorics - 28th International Conference, COCOON 2022, Shenzhen, China, October 22-24, 2022, Proceedings*, volume 13595 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 2022. doi:10.1007/978-3-031-22105-7“15.

[12] 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. URL: http://proceedings.mlr.press/v101/sato19a.html.

[13] 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. doi:10.1016/j.cor.2020.105184.

[14] 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.

[15] Kate Smith-Miles and Leo Lopes. Measuring instance difficulty for combinatorial optimization problems. *Comput. Oper. Res.*, 39(5):875–889, 2012. doi:10.1016/j.cor.2011.07.006.

[16] Stefan Szeider. On fixed-parameter tractable parameterizations of SAT. In Enrico Giunchiglia and Armando Tacchella, editors, *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, pages 188–202. Springer, 2003. doi:10.1007/978-3-540-24605-3“15.

[17] Jano I. van Hemert. Evolving combinatorial problem instances that are difficult to solve. *Evol. Comput.*, 14(4):433–462, 2006. doi:10.1162/evco.2006.14.4.433.

[18] Adam Zsolt Wagner. Constructions in combinatorics via neural networks. *CoRR*, abs/2104.14516, 2021. URL: https://arxiv.org/abs/2104.14516, arXiv:2104.14516.

[19] 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. URL: http://ijcai.org/Proceedings/05/Papers/0989.pdf.

[20] Jiaxuan You, Haoze Wu, Clark W. Barrett, Raghuram Ramanujan, and Jure Leskovec. G2SAT: learning to generate SAT formulas. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10552–10563, 2019. URL: https://proceedings.neurips.cc/paper/2019/hash/daea32adcae6abcb548134fa98f139f9-Abstract.html.