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 \( P = NP \)). 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, \emph{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


