

Project 0: Discrete Optimization

This project is intended to be a warm-up for future projects, which will have a similar form, but involve more complex algorithms and domains. It shouldn't take more than one or two evenings. You can do it in any software environment you like. You're welcome to do it in a small group or individually.

Grading: If you hand this project in, in class, on **Thursday, September 16**, then I will write comments and a grade on it and give it back to you. This will help you understand the level of expectations for work in this course, and contribute to your understanding of the material. It won't factor directly into your grade.

1 Algorithm

Implement at least one discrete search algorithm, such as hill-climbing, simulated annealing, threshold acceptance, local beam search, genetic algorithm.

2 Domain

Code up a discrete search domain, such as traveling sales-thing (TSP), bin-packing, or circuit partitioning. You can make up your own problem instances, or probably find some standard ones online.

Feel free to use some other domain, if it interests you.

3 Experiment

Compare two or more approaches to solving problems in your domain. Either vary the algorithm, or parameters of the algorithm (e.g. annealing schedule or step size), or the problem formulation (e.g. definition of neighbors).

Run your algorithms several times; report average and standard deviation of both running times and solution quality. Draw some conclusions from the results.