Project 1: Optimization

- **Due: September 30**
- Ideally done in a two-person team; ask me if you want help finding a partner.

Steps

1. Jointly decide on a problem class within the broad category of discrete or continuous optimization. Decide on a programming language and an interface between problems and solution algorithms.
2. Each partner should find a domain (e.g., locating firehouses or solving puzzles or docking molecules) and code up at least two instances of different size or difficulty (or a random problem generator).
3. Each partner should implement an algorithm for solving problems in this class.
4. Each team should collectively run at least 8 experiments, running each algorithm on each problem instance.
5. Each team should hand in a short paper (roughly 5 - 8 pages) explaining your domains, algorithms, and results. Explain your results. Append your code, but I probably won't look closely at it.

You can use any algoirthm we've talked about in lecture. If you want advice about more sophisticated algorithms, or extra reading material, or feedback about your choice of domains and/or algorithms, just email me.