CS 538 Projects

A component of the course is an implementation of an Optimization-related project (or a research on your own). The objective is to develop 1) a software product that solves some real life problems by adapting/designing/improving algorithms into your implementation or 2) improvement or survey study on existing techniques for solving optimization problems, especially NP-Complete or NP-Hard problems.. During the semester, the project team will work together through the full development.

General Information about the Projects

Choosing a Project

Currently students are advised to explore their own interest and figure out what optimization (approximation)-related project they would like to implement. Had the students not identified topic of interest, a list of suggested projects will be introduced.

Deliverables (if coding project)

Since every project is different, there is not a set list of deliverables that every project must provide. Part of your task is to manage the progress of the project such that the project can succeed, instead of failure. Typical deliverables (at each stage) include formal proposal (after the first presentation), progress report, working code, documentation and final report, etc. Further details will be given as the semester goes on (that means this document will be constantly updated based on our progress.

These are group projects, but you will also be rewarded individually for special contributions to the project. At the end of the semester, a survey will be delivered to you electronically. You have the opportunity to evaluate your team members in terms of their contribution to the project.

Technical environment

Most projects will use C++, Java, Python, Ruby, PHP, or Perl, on Unix, Linux, Windows, or Macintosh computers, tablets, or smartphones, but you are encouraged to use whatever is right for your particular product.

Schedule

Task 1: March 29

- Identify your interest (see footnote 1) and talk about it:
    - Think about possible projects that can be implemented by the existing skill sets of your team
    - 1 page project proposal
        1. Description of the problem
        2. Sketch of the algorithm of interest (steps and complexity)
        3. Schedule (what to deliver at what date)

Task 2: Progress Report: April 17

As discussed in class, we will need a brief update on your project progress. Please prepare accordingly. All teams must report in class on **April 17**. Please have a 5-minute presentation on (a) **what has been implemented** so far (b) what has been **changed** since your last proposal (c) what **difficulties** are you facing and what are your **strategies** to to address these issues (d) can you **estimate how likely you will deliver the results** that you were expecting during your previous presentation?

Task 3: Group Presentation
https://sunypoly.edu/home5/cmsfiles/Spring17FinalExamSchedule.pdf

Date: 04/26, 05/01

Team	Date	Topic
G2		Improving TSP tours using Dynamic Programming
G3		Edge Elimination in TSP Instances
G4		Maximum Flow and Minimum Cut Problem
G5		Edge Elimination in TSP Instances
G6		Edge Elimination in TSP Instances
G7		Vertex Cover Problem
G8		Improving TSP tours using Dynamic Programming
RedTeam		Mixed-Integer Linear Programming Solver
TeamRedundantTeam		Approximating the regular graphic TSP in near linear time
MJ		Edge Elimination in TSP Instances

Quora		Integer Factorization

Each team would give a presentation of their project, the content should include

1. Brief overview of the project (what it does, what techniques are used)
2. Brief overview of your almost-finished manual (if any)
3. Demonstration of your project
4. Brief description of each person's contribution
5. Q+A section

. Aim for a 20-minute presentation, all members must participate and present

Task 4: Your material

Date: May 1st

Final product [Source code + full-fledged Manual + presentation slides]

Resources:

1. Find your match (seek on arxiv.org for optimization, TSP, NP-complete, NP-hard)

2. Optimal Lower Bounds for Universal and Differentially Private Steiner Trees and TSPs
http://www.microsoft.com/en-us/research/publication/optimal-lower-bounds-universal-differentially-private-steiner-trees-tsps/

3. Improving TSP tours using dynamic programming over tree decomposition
https://arxiv.org/pdf/1703.05559.pdf

4. Improved Approximations for Cubic Bipartite and Cubic TSP
https://link.springer.com/chapter/10.1007/978-3-319-33461-5_21
http://arvanzuijlen.people.wm.edu/cbTSP.pdf

4-1: A 9/7-Approximation Algorithm for Graphic TSP in Cubic Bipartite Graphs
http://dl.acm.org/citation.cfm?id=2957058
https://arxiv.org/pdf/1311.3640.pdf

5. Approximating the Regular Graphic TSP in near linear time
http://drops.dagstuhl.de/opus/volltexte/2015/5626/pdf/10.pdf

6. Almost Optimal Streaming Algorithms for Coverage Problems
https://arxiv.org/pdf/1610.08096v1.pdf

7. Minimax Optimal Algorithms for Unconstrained Linear Optimization
https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41859.pdf

8. Edge Elimination in TSP Instance
https://arxiv.org/pdf/1402.7301.pdf

https://link.springer.com/chapter/10.1007/978-3-319-12340-0_23