Final Project
IE 407 – Computational Methods in Optimization
Dr. Ralphs

The project assignment is to do an in-depth computational study of a problem class or algorithm class relevant to the content of the course. This study may be one of two basic types: (1) a parallel scalability analysis of a single or multiple algorithms or (2) a study of different sequential algorithms or different implementations for a given problem class. This could include a study of how to cache optimize or other wise improve a base implementation. More details regarding suggested topics to be addressed as part of the study and suggested focus areas are listed below.

Broadly, the requirements are as follows.

- You are to make a formal project proposal by **Thursday, September 26**. The proposal should include details of your proposed study, including what problem you want to study, what algorithms you plan to implement, and what research questions you will answer. You should use good software engineering practices, including object-oriented design and your plan should indicate what language you are proposing to use and why you are proposing to use that language.
- A preliminary literature survey will be due on **Thursday, October 10**. This should be a brief summary of relevant papers that will inform your development or to which you will compare your final product.
- You should have a preliminary design for your software (file structure, class structure, API, etc.) by **Thursday, October 24, 2019**.
- You should have working beta code in a git repository shared with me and the TA by **Thursday, November 14** and a written plan for what computational experiments you will do.
- The final product of the project will be (1) a detailed report presented as a research paper that includes a literature review and detailed computational results, (2) well-commented source code and instructions on how to build it, (3) the data files you used to do your testing. Due date is TBD.

Your research plan should be well thought out and you should be able to justify all of your choices, including the algorithms, languages, testing platforms, and data sets you have chosen to focus on in your study.

It is required that you use git for your development and I will check your progress periodically by checking out your code from your repository. Part of your grade will depend on your development practices, including good documentation and testing.

For parallel scalability analysis, the types of questions that you might choose answer in your study are as follows.

1. Analyze in detail the effectiveness and scalability of the same algorithm running on different architectures.

   - For the case of shared memory, the study could examine the effects of details in the architecture, such as the size of the cache and the configuration of the memory.
   - For distributed memory, it could examine such things as memory configuration, bandwidth, and latency.
   - The study could compare a shared memory implementation with a distributed memory implementation or even a hybrid implementation.
   - Another option would be to experiment with a GPU-based architecture.

2. Analyze the differences in scalability of parallel versions of different sequential algorithms.

For sequential algorithm analysis, the types of questions that you might consider addressing are as follows (note that these might also apply to the parallel case as well).

1. Compare different algorithms for the same problem class.

2. Compare the same algorithm on different architectures, taking into account such things as the size of the cache and the specific configuration of the memory.

3. Compare the same algorithm implemented in different ways (cache optimized versus not).

4. Study the effect of the structure of different instances on running times in practice and try to develop some rules of thumb about what makes a particular instance difficult to solve.

Some suggested topic areas are listed below. You may choose a topics not listed here if you wish, but you are encouraged to consider these topics first. Regardless of what topic you choose, your research plan must be approved before starting the project.

1. **Parallel Scalability**

   (a) **Matrix multiplication**. Attempt to parallelize matrix multiplication (Strassen's algorithm is an option, but we have a homework assignment on it already) and compare this implementation to other possible approaches.

   (b) **Component labeling**. Parallelize algorithms for component labeling and test them in the context of either a parallel algorithm spanning tree or image analysis.

   (c) **Dynamic programming**. Implement parallel algorithms for dynamic programming and apply it to a combinatorial problem such as elementary shortest path with resource constraints or the knapsack problem.

   (d) **Shortest path**. Implement parallel algorithms for the shortest path problem.

   (e) **SAT**. Implement parallel algorithms for solving the satisfiability problem.

(f) **Randomized algorithms**. Implement randomized parallel algorithms per the paper of Rolim (1999) for simple problem classes and analyze the results.

2. **Combinatorial Optimization**

   (a) **Max flow**. There are a wide range of different approaches to solving this classical combinatorial problem. Implement and compare those considered state-of-the-art.

   (b) **Matching**. Compare different algorithms for the matching problem, including both combinatorial and cutting plane approaches.

3. **Numerical Algorithms**

   (a) **Gaussian elimination**. Implement methods for solving systems of equations, testing the various heuristic techniques for improving speed and accuracy.

   (b) **Matrix factorization**. Implement methods for solving one or more matrix factorization problems.