

CSE543T: Algorithms for Nonlinear Optimization

Course Project

The objective of the course project is to gain experiences in developing nonlinear optimization methods and/or applying nonlinear optimization techniques to solve real-world problems.

The final result of the project will be a technical report and an implementation package. You should state clearly in your technical report about the algorithm/system: 1) background 2) related theory and practice 3) your new idea 4) implementation details 5) experimental results 6) discussions and conclusions. If you don't have any implementation, your report should have original and publishable theoretical developments.

Forming a group and choosing a topic (due March 6, 11:59pm)

You can work by yourself, or work in a team with up to five members. You should start forming your team and choosing a topic immediately. By Mar 6, visit the following table

https://docs.google.com/spreadsheets/d/1tQyjAzleN2r1GvgARHu_HUtrRoJAtnNxCumgEJABUG4k/edit?usp=sharing

and enter the names of ALL the group members and a title of the project. If you do not respond by Mar 6, you will risk losing all credits for the course project.

Titles of some course projects in previous semesters are listed below to give you some examples:

- Detecting polynomials in image
- A study of optimizing time use for college students to maximize life satisfaction
- Designing popup craft automatically by nonlinear optimization
- Optimizing card counting strategies in blackjack
- Learning and optimizing in document classification
- DNA Motif finding via a two component mixture model
- A study of nonlinear optimization for modeling metabolic networks
- Meal planning for nutritional optimization
- Estimating missing data and classification in DNA microarray datasets
- Predict the 2016 NCAA Basketball Tournament
- Optimal locations of monitoring stations in drinking water distribution systems
- Advertisement validation and optimization

The topic can be a temporary one and changed later if necessary.

Project presentation (April 19, 24, 26, Schedule TBD)

Each team needs to give a 15-20 min presentation about their project. Details will be announced later.

Technical report and software (due May 5, 11:59pm)

Each course project will result in a detailed 8-12 page written technical report (letter-size, single spaced). The project report should be a self-contained scientific report with conference-paper quality. Also, it should be written with the readers in mind. All class members should be able to understand your report, and benefit from the results you obtain. Therefore, you should include adequate references and background materials and you should use tables, diagrams, graphs, figures, and portions of printouts to enhance readers' comprehension of your project. You must also clearly state the partition of work and the contributions of each team member.

The following format is suggested. You don't have to follow it exactly. Some sections may not be needed, or additional sections may be necessary.

- (1) Abstract. Gives succinct information on the purpose, methods, results and conclusions reported.
- (2) Introduction. Include background materials and define the problem and scope of your project.
- (3) Related theory and practice.
- (4) Technical details. Be sure to fully describe any figures, tables or diagrams you include.
- (5) Experimental results.
- (6) Discussions, observations, and comparisons.
- (7) Recommendations, especially for future work and unsolved problems.
- (8) Contributions of team members. You must clearly state the partition of work and the specific contributions of each team member. Failure to do so may lower the scores of all team members.
- (9) References.

Submission instructions:

Submit the entire report in a single **PDF file via Canvas** before **May 5, 11:59pm**.

Do not submit lengthy computer program outputs. Relevant excerpts from program listings or output should be included, but reduced to the size of the rest of the report and containing either as figures or tables in the text or as an appendix.

You also need to submit the software package you have implemented. Include a README file with clear instructions on how to duplicate the results in the report. **Email the package to the TA (liuhao@wustl.edu), cc all group members.**

No lateness or extension will be allowed.

Post-project considerations (Optional)

For high-quality projects at my discretion, the team may submit the technical report to a major conference and publish the results. Please contact me for details. This part will not affect the grading of the project.

IMPORTANT DATES

Team and topic registration:
Final report and software due:

March 6, 2023
May 5, 2023

GENERAL GUIDELINES

Grading of the project will consider the following aspects. a) Project idea and contributions, including: motivations, relevance to the class, clarity of the project's contributions, and applicability of the results. b) Technical strength, including: technical soundness, correctness of the approach, implementation efforts. c) Presentation quality, including: organization report, clarity, proper use of references, analysis of experimental results, and insights.

Suggested topics:

A. Theory and Algorithm

1. ***Removing duality gaps.*** For nonconvex problems, there are generally non-zero duality gaps when a traditional Lagrangian function is used (Section 5.1). Develop novel generalizations/variations of the Lagrangian function so that there is no duality gap even for nonconvex problems. There are a number of existing work on this topic, find and study them before proposing your own solution. Theoretically prove your results. It will be a plus if you can experimentally evaluate your approach.
2. ***Experimental evaluation of duality gaps.*** Implement a program to empirically evaluate the existence of duality gap for nonconvex problems. Your program needs to find Lagrange multipliers that maximize the dual function in the outer loop and minimize the Lagrangian function in the inner loop, which can be done by calling existing unconstrained global minimization software. You may not need to find the exact optimal points. Good approximate solutions are acceptable. Try to find out some interesting classes of nonconvex problems where the duality gaps are particularly large or particularly small or even zero. (For example, read Section 5.1.6 for a class of problems where the duality gaps are small.) Theoretical explanations will be a plus. <http://www.ampl.com/EXAMPLES/> contains many test problems.
3. ***Design and implementation of an improved optimization algorithm.*** Propose a new optimization algorithm. It can be just a small change to an existing algorithm, but it should be well motivated and target at addressing an existing limitation. You need to implement the algorithm by writing your own, modifying an existing package, or developing a meta code on top of Matlab or AMPL, and show some empirical results. Some recent hot topics include stochastic gradient descents (with parallelization), semi-definite programming, and sub-gradient methods for non-smooth optimization (such as l-1 regularization).

B. Applications

Model a problem in computer science or other areas as a nonlinear optimization problem and solve it using an existing solver or a customized solver. Suggested applications include but are not limited to:

- Planning and scheduling
- Markov Decision Processes (MDP) and partially-observable MDP

- Combinatorial optimization on trees, graphs, networks, etc.
- Network management (flow routing, bandwidth allocation, P2P, etc.)
- Sensor networks (placement and power management)
- Neural network training
- Supporting vector machine
- Conditional random fields
- Data mining (clustering, classifications, time-series prediction)
- Biosequence alignments
- Hard-particle simulations
- Investment portfolio optimization

It is required to: a) formulate a well-defined application problem, b) survey existing optimization models for this application and discuss their pros and cons, c) develop an optimization model for the application, d) use an existing solver to obtain the solution, and e) verify the solution quality using application data.

It is also required that you perform one or more of the following studies:

- Compare different models for the problem, in terms of optimization difficulty, computational cost, and solution quality.
- Modify and customize an exiting optimization package to make it more efficient and effective for the target application.
- Incorporate domain-knowledge to improve the optimization speed and quality.