SYSEN 5800 / CHEME 6800: Computational Optimization
Fall 2026

Lectures: 2:55 - 4:10 PM on Mondays and Wednesdays in Upson Hall 222 and via Zoom

Recitations: 2:30 - 3:20 PM on Fridays in Rhodes Hall 253 and via Zoom

Author/Instructor: Professor Fengqi You

Course Aids: Nathan Preuss (nnp25@cornell.edu), Wenhao Yuan (wy337@cornell.edu), Rahul Sheshanarayana (rs2246@cornell.edu)

Ed Discussion: https://edstem.org/us/courses/100322/

Example Codes: https://github.com/PEESEgroup/SysEn6800

Course Objectives

- Learn systems optimization modeling, computational methods, and applications
- Understand the basics of advanced optimization theory, concepts, and algorithms
- Develop ability to formulate large-scale optimization models for real-world applications
- Apply systems optimization techniques to solve application problems
- **A balance between theory, computation and applications**
    - Optimization theory (30%)
    - Computational modeling, algorithms, and software (30%)
    - Applications (40%)

Catalog Description

Advanced systems optimization modeling, computation, and applications. Includes theory and algorithms of linear, nonlinear, mixed-integer linear, mixed-integer nonlinear, and deterministic global optimization, as well as stochastic programming, robust optimization, and optimization methods for machine learning and big-data analytics. Real-world applications of large-scale computational optimization in manufacturing, aerospace, control, management, infrastructure, energy systems, sustainability, and data science. This course has 4 credits.

Prerequisites:

Multivariable calculus (MATH 2220 or equivalent) and elementary linear algebra (MATH 2940 or equivalent). Background on basic probability (CEE 3040/MATH 4710/ORIE 3500/CHEME 5740 or equivalent) is encouraged but not required.

Attestation

By registering for this class and accessing course materials through Canvas, students agree to abide by University, College, Department, and Course policies.

Academic Integrity

Each student is expected to abide by the Cornell University **Code of Academic Integrity** (https://cuinfo.cornell.edu/aic.cfm). You may discuss homework with each other, but you must solve the problems yourself and everything you turn in must be your own work. List at the top of the HW assignment the names of anyone you discussed the HW problems with. To be ready for the exams you'll need to be able to do all the problems on your own.

Copying from solution manuals or from each other is strictly forbidden. The way to learn mechanics (or pretty much anything) is to practice solving and applying basic ideas, so except where a problem explicitly permits it, the use of Generative AI (GAI) tools (e.g., ChatGPT, Claude, Gemini, Copilot) to produce answers to homework or exam questions is not allowed. Letting AI do your work shortchanges your education, and GAI often returns confident but incorrect optimization models. To help you learn to use these tools critically, each homework assignment includes one clearly marked AI exercise for which using an AI assistant is required: you solve the designated problems yourself, use an AI tool to reproduce them, and then verify and critique its output, submitting your prompt(s), the AI response, and your critique. Using GAI on any problem that is not explicitly designated as an AI exercise is a violation of this academic integrity policy. The textbooks are accurate and have been vetted and are your go-to source for information relevant to the course.

Course materials posted on Canvas are intellectual property belonging to the author. Students are not permitted to buy, sell, trade or share any course materials without the express permission of the instructor. Such unauthorized behavior constitutes academic misconduct. It is a violation of the Cornell Code of Academic Integrity to utilize a site such as chegg.com or coursehero.com to upload, download or receive help on exams or assignments that do not permit consultation. Please note that Cornell has resources that enable instructors to trace posts (both the original poster and those who read the posts) from study sites. Using these sites and other forms of cheating is unfair to your classmates.

Violations of the above policies will be dealt with through the Academic Integrity Hearing process.

Diversity Statement

Every student in this course is expected to contribute to an inclusive and respectful class environment. It is my intent that students from all diverse backgrounds and perspectives be well-served by this course, that students' learning needs to be addressed both in and out of class, and that the diversity that students bring to this class be viewed as a resource, strength and benefit. It is my intent to present materials and activities that are respectful of diversity: gender, sexual orientation, disability, age, socioeconomic status, ethnicity, race, culture, perspective, and other background characteristics. Civil discourse, reasoned thought, sustained discussion, and constructive engagement without degrading, abusing, harassing, or silencing others is required of all students in this class. Your suggestions about how to improve the value of diversity in this course are encouraged and appreciated. Please let me know ways to improve the effectiveness of the course for you personally or for other students or student groups.

Main textbook

- Lecture notes, handouts, and slides

Recommended textbooks and useful online resources

- Cornell University Computational Optimization Open Textbook: https://optimization.cbe.cornell.edu/
- R.J. Vanderbei, *Linear Programming: Foundations and Extensions*. Springer, 2008. (PDF at http://link.springer.com/book/10.1007/978-0-387-74388-2)
- S. Boyd, L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009. (PDF at https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf)
- J. Nocedal, S. J. Wright, *Numerical Optimization*. Springer, 1999. (PDF at http://link.springer.com/book/10.1007/b98874)
- J. Birge, F. Louveaux, *Introduction to Stochastic Programming*, Springer, 2011. (PDF at http://link.springer.com/book/10.1007/978-1-4614-0237-4)
- A. Ben-Tal, L.E. Ghaoui, A. Nemirovski. *Robust optimization*. Princeton University Press, 2009. (PDF at http://www2.isye.gatech.edu/~nemirovs/FullBookDec11.pdf)
- L.A. Wolsey, *Integer Programming*. Wiley, 1998.
- L.T. Biegler, I.E. Grossmann, A.W. Westerberg, *Systematic Methods of Chemical Process Design*. Prentice Hall Press, 1997.
- L.T. Biegler, *Nonlinear Programming: Concepts, Algorithms and Applications to Chemical Engineering*. SIAM, 2010. (PDF at https://epubs.siam.org/doi/book/10.1137/1.9780898719383)
- T.F. Edgar, D.M. Himmelblau, L.S. Lasdon, *Optimization of Chemical Processes*. McGraw-Hill, 2001.
- T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning*, Springer, 2009. (PDF at https://hastie.su.domains/Papers/ESLII.pdf)
- K. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. (PDF at https://github.com/probml/pml-book)

Course Topics

- Introduction & optimization basics
  - Course overview
  - Brief introduction of optimization methods and applications
  - Basic concepts in optimization
    - Objective function, constraints, variables, feasible region
    - Graphical solution approach
    - Convexity of functions and sets
  - Analytical solution of unconstrained optimization problems
  - Numerical algorithms for unconstrained optimization problems
    - Gradient descent
    - Newton's method
    - Quasi-Newton's method
    - Machine learning applications

    - Non-convexity and discrete variables
        - Modeling with integer variables and Boolean expressions
        - Logic proposition, modeling of disjunctions
- Linear programming (LP)
    - Applications:
        - Energy planning
        - Metabolic flux balance analysis
        - Resource allocation problem
        - Transportation planning
        - Multi-period optimization of supply chains
    - Theory & algorithms:
        - Simplex algorithm
        - Duality
        - Reformulation and LP problems with special structure
    - Computation:
        - Introduction to GAMS and Pyomo
        - LP Solvers
        - Modeling and solving LP problems
- Mixed-integer linear programming (MILP)
    - Applications:
        - Synthesis of energy systems
        - Reaction and conversion pathway/network analysis
        - Scheduling of batch manufacturing processes
        - Capacity planning of process networks
        - Unit commitment for electric power systems operations
        - Hospital operating room assignment and scheduling
    - Theory & algorithms:
        - Modeling with discrete decisions
        - Branch-and-bound algorithm
        - Valid inequalities and cutting planes
        - Branch-and-cut method
        - Disjunctive programming and logic optimization
    - Computation:
        - Computational issues on solving MILP problems
        - Introduction to MIP solvers
        - Modeling and solving MILP problems
- Nonlinear programming (NLP) and dynamic optimization (DO)
    - Applications:
        - Real-time optimization of dynamic energy systems
        - Gasoline blending optimization
        - Parameter estimation
        - Aircraft trajectory optimization
        - Optimal path planning of space, manned, and unmanned aircraft
        - Simulated moving bed and adsorption process optimization
        - Three-dimensional conflict resolution of multiple aircraft
    - Theory & algorithms:

        - Solution of nonlinear equations
        - Karush-Kuhn-Tucker optimality conditions
        - NLP algorithms
        - Modeling and solving NLP problems
        - Brief overview of dynamic optimization
    - Computation:
        - State-of-the-art NLP solvers
        - Modeling and solving NLP problems
- Mixed-integer nonlinear programming (MINLP) and deterministic global optimization
    - Applications:
        - Design and synthesis of manufacturing processes
        - Life cycle optimization and energy supply chain optimization
        - Aircraft design and optimization
        - Multiple-stage launch space vehicle ascent problem
    - Theory & algorithms:
        - MINLP algorithms, including branch & bound, outer approximation, generalized Benders decomposition and extended cutting plane
        - Reformulation and convexification of nonconvex NLP/MINLP problems
        - McCormick convex envelope
        - Spatial branch-and-bound
        - Special topic on mixed-integer fractional programming
    - Computation:
        - State-of-the-art MINLP solvers
        - Global optimization software packages
        - Modeling and solving MINLP problems
        - Modeling and solving deterministic global optimization problems
- Stochastic programming (SP)
    - Applications:
        - Agricultural production planning
        - Newsvendor problem and inventory management
        - Multistage portfolio selection
        - Offshore oilfield development planning under uncertainty
    - Theory & algorithms:
        - Decisions, stages, and recourse
        - Two-stage stochastic programming and its deterministic equivalent
        - Value of stochastic solution and expected value of perfect information
        - Monte Carlo sampling and Sample Average Approximation (SAA)
        - L-shaped method and multicut L-shaped method
        - Downside risk, VaR, CVaR, and risk averse optimization
        - Multi-stage stochastic programming
        - Chance constrained programming
    - Computation:
        - Modeling and solving stochastic programming problems
- Robust optimization (RO)
    - Applications:
        - Robust planning of cancer treatment

        - Portfolio optimization
        - Robust planning of product network
        - Multi-stage robust inventory control
    - Theory & algorithms:
        - Static robust optimization
        - Uncertainty set
        - Globalized robust counterpart representation
        - Two-stage adaptive robust optimization with recourse
        - Multi-stage adaptive robust optimization
        - Decision rules
    - Computation:
        - Modeling and solving robust optimization problems
- Optimization and machine learning
    - Applications:
        - Optimization for regression
        - Optimization for classifications
        - Optimization for dimension reduction
        - Optimization for clustering
        - Optimization for deep learning
    - Theory & algorithms:
        - Linear regression
        - Support vector machine
        - Logistic regression
        - Principal component analysis
        - Support vector clustering
        - K-means algorithm
        - Expectation-maximization algorithm
        - Optimization algorithms in deep learning
        - Surrogate-based optimization
        - Bayesian optimization
        - Data-driven robust optimization
    - Computation:
        - Modeling and solving machine learning and deep learning problems

Course Home Page

Important course announcements, lecture notes and slides, supporting materials and homework assignments will be placed on Canvas <https://canvas.cornell.edu/>. The electronic versions of homework assignments, reports and slides, will also need to be submitted through this website.

Weighting of Grades

The course will be graded based on six homework assignments and a team project. The weights are given below:

50% homework assignments
45% team project
5% recitation participation (for students in CHEME/SYSEN 6800)

Homework Assignments

- Please submit the electronic files through Canvas under the "Assignments" folder.
- Some homework problems might ask you to program in GAMS or Python. The source files/codes need to be submitted to Canvas electronically – you are allowed to prepare part of the homework in hard copies and the rest electronically.
- Late homework assignments incur a penalty of 5% per hour. This means that no credit will be received if it is more than 20 hours late.

Team Project

- The primary objective of the project is to make students familiar with the applications of systems optimization techniques by using/developing them.
- Each team will apply large-scale optimization modeling and solution techniques to an application that is of interest to the team member(s).
- The maximum size of a team is five (5) students; there is no problem at all to have only one member in a team.
- Every student in the same team gets the same score, regardless of the number of students per team.
- You are very welcome to consult with Professor You for suggestions on project topics if you are not sure which application/problem to work on.
- The project could deal with a deterministic optimization problem but influences of uncertainty and possible approach for optimization under uncertainty should be discussed quantitatively, or at least conceptually (implementation is not required).
- Please code/program the optimization models and/or algorithms in GAMS or Pyomo.
- Each team needs to submit a project plan, a final project report, and give a project presentation to the class.
- We encourage students to work out "team dynamic" problems for themselves, but if you feel some members of the team are not contributing their share, you could choose to split the team into two (or more). It is unacceptable to complain about such problems and request grade adjustments after the assignments are due. We reserve the right to deal with dysfunctional teams in extreme ways, e.g., assigning individual students to be one-member teams.
- All files need to be submitted through Canvas before the due date. Late submissions incur a penalty of 5% per hour. This means that no credit will be received if it is more than 20 hours late.
- Each team should sign up by the deadline posted on Canvas to secure a presentation slot.

Project Plan

- No more than 10 pages.
- Use font size Times New Roman 12; 1.5 line spacing; one blank line between paragraphs.
- Sources of all information borrowed from the literature or websites must be cited properly. Use the scientific journal citation method, i.e., cite where the information appears in the text.
- The **project plan** needs to include the following sections:

a. A title page: list a title of the project, name(s) of team member(s), and an abstract, which summarizes your plan in no more than 150 words.
b. Main body:
    i. **Introduction:** which application/problem to be addressed in this project, why is it important, what are the major challenges and how are you going to address them?
    ii. **Background:** a layman's description of the application problem.
    iii. **Literature review:** discussing the state-of-the-art and commenting on the pros and cons of the existing works.
    iv. **Problem statement:** roughly stating what are given (parameters), what are to be determined/optimized, and what is the objective.
    v. **References**

- Content and materials of the project plan could be re-used in the final project report, although improvements and updates are strongly encouraged.

Project Presentation

- Each team will have a total of **<u>10 minutes</u>** for the presentation and Q&A.
- Every member of the team is expected to give part of the presentation.
- Every student is expected to attend all the presentations. Team grades will be deducted for absence or lateness of team members.
- Questions from students will be *strongly* encouraged.
- The presentations will be graded based on 6 criteria listed below:
    - Quality of slides & delivery
    - Motivation & definition of the problem
    - Optimization modeling approach
    - Solution algorithm/approach
    - Case study and results
    - Q & A

Final Report

- No more than 30 pages plus supplementary materials.
- Use font size Times New Roman 12; 1.5 line spacing; one blank line between paragraphs.
- Sources of all information borrowed from the literature or websites must be cited properly. Use the scientific journal citation method, i.e., cite where the information appears in the text.
- The **final report** needs to include the following sections:
    a. A title page: list a title of the report, name(s) of team member(s), and an abstract, which summarizes your project in no more than 150 words.
    b. Main body:
        i. **Introduction:** which application/problem to be addressed in this project, why is it important, what are the major challenges and how are you going to address them?
        ii. **Background:** a layman's description of the application problem.
        iii. **Literature review:** discussing the state-of-the-art and commenting on the pros and cons of the existing works.

iv. **Problem statement:** clearly stating what are given parameters, what are to be determined/optimized, and what is the objective.
v. **Model formulation and solution algorithm:** providing a nomenclature, presenting the mathematical formulation of the model and discussing the corresponding solution strategy; please pay special attention to the convexity of the problem.
vi. **Results and discussion:** presenting at least one example or case study of the proposed modeling and optimization framework.
vii. **Conclusion and recommendations:** providing a summary of this project and discussing the potential future extensions of this work.
viii. **References**

c. Supplementary material, including the source code, and/or any other computer codes that you developed for this project.

Recitations

Recitation attendance will account for 5% of the final course grade. Students enrolled in CHEME/SYSEN 6800 are expected to attend recitations in person. To receive full credit, students must attend at least 10 of the 14 recitations. Students who attend fewer than 10 recitations will receive a proportionally reduced recitation grade.

SYSEN 5800 students will receive full recitation credit regardless of attendance because they are unable to attend in person. They are nevertheless welcome to participate via Zoom whenever their schedules allow.