Lecture 1 Introduction to Computational Optimization

Can Li

ChE 597: Computational Optimization
Purdue University

Course Info

- Instructor: Can Li
- Email: canli@purdue.edu
- Classroom: Hampton Hall 2102
- Time: Tuesday and Thursday, 4:30 pm - 5:45 pm
- Make up lecture: BHEE 236 (occasionally on Monday and Wednesday)
- Office: Forney Hall of Chemical Engineering, Room G027A
- Office Hours: Wednesday 5 pm - 6 pm

Course setup

- Instructor: Can Li
- Course Website: https://canli1.github.io/courses
- Notes, homework, and videos will be posted on the course website
- Brightspace will be used as a gradebook and making announcements
- Prerequisites:
  - Calculus, linear algebra
  - Programming in python
  - Formal mathematical thinking

Evaluation

- 10 homework (each 5 problems) (10%)
- 1 midterm (40%), 1 final (40%)
- 1 course project (10%)
- Bonus points: course evaluation (1%). Find a "significant" mistake (to be determined by the instructor) in the homework solution we posted (each 2% up to 10%).
- Homework load will be heavier than most elective courses. If your research is not in optimization, you can pick at least 60% problems in each homework. However, you are strongly encouraged to do all the homework.
- Homework won't be graded based on correctness. We only check whether you make a serious attempt or not (submitting a blank sheet or rephrasing the problem does not count).

Textbooks

No Required Textbooks: Course slides alone will suffice. For additional references, the following textbooks are recommended, listed in ascending order of mathematical difficulty.

- Grossmann, I. E. (2021). Advanced optimization for process systems engineering. Cambridge University Press.
- Boyd, S. P., & Vandenberghe, L. (2004). Convex optimization. Cambridge university press.
- Wolsey, L. A. (2020). Integer programming. John Wiley & Sons.
- Bertsimas, D.& Tsitsiklis, J. N. (1997). Introduction to linear optimization (Vol. 6, pp. 479-530). Belmont, MA: Athena Scientific.
- Tawarmalani, M., & Sahinidis, N. V. (2013). Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications (Vol. 65). Springer Science & Business Media.
- Horst, R., & Tuy, H. (2013). Global optimization: Deterministic approaches. Springer Science & Business Media.
- Conforti, M., Cornuéjols, G., Zambelli, G (2014). Integer programming. Graduate Texts in Mathematics
- Ben-Tal, A., & Nemirovski, A. (2001). Lectures on modern convex optimization: analysis, algorithms, and engineering applications. Society for industrial and applied mathematics.

The Ubiquity of Optimization

- **Engineering:** Chemical process control/design, resource allocation, power systems, scheduling.
- **Transportation:** Route planning, traffic flow optimization, logistics.
- **Health Care:** Treatment planning, hospital resource management, medical imaging.
- **Science:** Material design, protein structure prediction
- **Machine Learning:** Almost all the machine learning models are formulated as optimization problems

Linear Regression Example with Three Variables

Consider fitting a linear model to the following data points with three features:

Observation	x_1	x_2	x_3	Response (y)
1	1.0	0.5	1.2	2.0
2	2.0	1.0	2.1	3.9
3	3.0	1.5	2.9	6.1
4	4.0	2.0	3.8	8.0
5	5.0	2.5	4.5	9.8

Table: Data for Linear Regression with Three Features

The goal is to find the model $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ that best fits this data in the least squares sense.

Linear Regression Least Squares Quadratic Programming (QP) Formulation

We want to minimize the sum of squared residuals. The objective function is:

$$\text{Minimize} \quad S(\beta_0, \beta_1, \beta_2, \beta_3) = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3}))^2$$

Where:

- y_i is the observed response for observation i.
- x_{i1}, x_{i2}, x_{i3} are the observed values of the three features for observation i.
- β_0 is the intercept of the hyperplane.
- $\beta_1, \beta_2, \beta_3$ are the coefficients for the features.
- n is the number of observations.

This model is known as multiple linear regression and the parameters are determined by least squares estimation.

Diet Problem Example with Additional Nutrients

A dietitian is planning a meal that meets the daily nutritional requirements for calories, protein, and vitamins at a minimum cost.

Food Item	Cost ($)	Calories	Protein (g)	Vitamins (% Daily)
Apple	1	100	0.5	2
Bread	0.50	200	4	0
Milk	2	150	8	10
Egg	0.30	70	6	0

Table: Cost and nutritional content of food items

Daily nutritional requirements: 500 calories, 50g protein, 100% vitamins.

Linear Programming Formulation for Diet Problem

Define decision variables: y_1 for Apples, y_2 for Bread, y_3 for Milk, y_4 for Eggs. y_i represents the quantity of each food item.

$$\text{Minimize} \quad y_1 + 0.5y_2 + 2y_3 + 0.3y_4$$

$$\begin{aligned} \text{Subject to} \quad 100y_1 + 200y_2 + 150y_3 + 70y_4 &\geq 500 \\ 0.5y_1 + 4y_2 + 8y_3 + 6y_4 &\geq 50 \\ 2y_1 + 0y_2 + 10y_3 + 0y_4 &\geq 100 \\ y_1, y_2, y_3, y_4 &\geq 0 \end{aligned}$$

Ensure all dietary requirements for calories, protein, and vitamins are met.

Knapsack Problem Example

Consider a hiker who needs to choose the most valuable items for a hike without overloading the backpack.

- Items: Tent (Value: \$120, Weight: 2kg), Stove (Value: \$80, Weight: 1kg), Food (Value: \$60, Weight: 1kg)
- Backpack capacity: 3.5kg

Objective: Maximize the value of items in the backpack.

Integer Program Formulation for Knapsack Problem

Define binary decision variables: x_1 for Tent, x_2 for Stove, x_3 for Food. $x_i = 1$ if the item is chosen, and 0 otherwise.

$$\text{Maximize } 120x_1 + 80x_2 + 60x_3$$

$$\text{Subject to } 2x_1 + x_2 + x_3 \leq 3.5$$

$$x_1, x_2, x_3 \in \{0, 1\}$$

Mixed-Integer Nonlinear Programming (MINLP)

A generic optimization problem is represented in the following succinct form:

$$\begin{aligned} \min_{x,y} \quad & f(x, y; \theta) \\ \text{s.t.} \quad & g(x, y; \theta) \leq 0 \\ & h(x, y; \theta) = 0 \\ & x \in \mathbb{R}^{n^x}, y \in \{0, 1\}^{n^y} \end{aligned} \tag{1}$$

- x are **continuous variables** with dimension n^x.
- y are **binary variables** with dimension n^y.
- θ represents the **parameters** of the problem.
- $g_i \; i = 1 \ldots, m^{\leq}$, $h_i \; i = 1, \ldots m^{=}$ are the inequality and the equality **constraints**
- f is the **objective function**

Mixed-Integer Nonlinear Programming (MINLP)

- an (x,y) that satisfies all the constraints are called a **feasible solution**
- (1) is called **infeasible** if there exists no feasible solution x, y
- If it is feasible, the minimizer of (1), x^*, y^*, is called the **optimal solution**. $f(x^*, y^*)$ is called the **optimal value**
- (1) is called **unbounded** if the optimal value is $-\infty$

Special Case of the MINLP

Depending on the forms of f, g, h, x, and y, the deterministic optimization problem (1) can be classified into several categories:

- If some of f, g, h are nonlinear functions, the problem is a **mixed-integer nonlinear program (MINLP)**.
- If f, g, h are all linear functions, it becomes a **mixed-integer linear program (MILP)**.
- If some of f, g, h are nonlinear and $n^y = 0$, it is a **nonlinear program (NLP)**.
- If some of f, g, h are quadratic and $n^y = 0$, it is a **quadratic constrained quadratic program (QCQP)**.
- If f, g, h are linear and $n^y = 0$, the problem is a **linear program (LP)**.

The choice among MINLP, MILP, NLP, QCQP, and LP depends on the nature of the problem.

Pyomo Basics: Sets, Parameters, Variables, Constraints

Pyomo is a Python-based, open-source optimization modeling language. Key components include:

- **Sets**: Collections of indices used to define parameters, variables, and constraints.
- **Parameters**: Data values (constants) used in the model, often defined over sets.
- **Variables**: Decision variables of the optimization problem.
- **Constraints**: Equations or inequalities that describe the relationships among variables and limit the feasible solution space.
- **Objective**: The function to be maximized or minimized in the optimization.

Optimization Solvers in Pyomo

Understanding Solvers: Pyomo itself is not a solver. It formulates optimization problems that can be solved using various external solvers. Here are some common types:

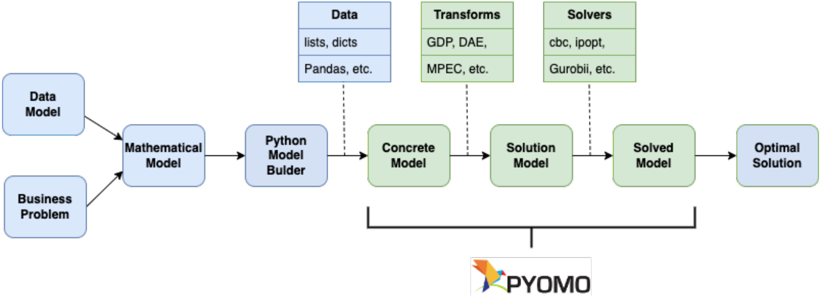
- **Open-Source Solvers:**
  - **CBC (Coin-or branch and cut)**: Mainly for linear and integer programming.
  - **IPOPT (Interior Point OPTimizer)**: For large-scale nonlinear optimization.
- **Commercial Solvers:**
  - **CPLEX**: High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming.
  - **Gurobi**: Advanced solver for linear, mixed-integer, and quadratic programming. Known for its performance and robustness. Added MINLP capability in 2024.
  - **BARON**: Generic MINLP solver with global optimality guarantee.

Link to a full list of solvers: see JuMP installation guide .

Optimization Solvers in Pyomo

Solver Selection: The choice of solver depends on the type of optimization problem (linear, non-linear, integer programming, etc.) and the problem size.

Integration with Pyomo: Solvers are typically implemented in compiled languages like C/C++/Fortran. Solvers are integrated with Pyomo through solver interfaces, enabling Pyomo to send models to these solvers and retrieve solutions.



Links to tutorials

- Installation
- Examples

References

1. ND Pyomo Cookbook
2. Grossmann, I. E. (2021). Advanced optimization for process systems engineering. Cambridge University Press.