Optimization Techniques

George Zois, Adjunct Professor, Co-Head of the Advanced Manufacturing and Optimization research group, ELTRUN, AUEB, georzois@aueb.gr, 210 8203663

Overview

The course provides an overview of optimization tools and techniques, motivated by large-scale problems that arise in Data Science, and real-life optimisation problems from industrial environments such as transportation, logistics and manufacturing. Optimization is at the heart of various critical tasks related to handling big data and enhancing decision making and a great number of methodologies have been developed over the years. The course aims to make students acquainted with modeling problems as optimization tasks and solving them using a range of exact and near-optimal solution methods.

In the first part, we will study fundamental mathematical modeling techniques, including linear and convex programming, and analyze key solution algorithms such as the simplex method, interior point method, and gradient descent. We will also explore their applications in core machine learning problems (e.g., regression, classification).

In the second part, we will extend our focus to the mathematical modeling of practical combinatorial optimization problems and examine their solutions using: a) exact methods, including branch and bound, decomposition methods, and constraint programming, and b) (meta-)heuristics, such as simulated annealing, genetic algorithms, as well as heuristic approaches that integrate combinatorial optimization with machine learning techniques, such as reinforcement learning (RL). The presentation of these methods will be accompanied by representative case studies demonstrating their application in real-world industrial problems.

Key Outcomes

By completing the course the students will be able to:

- Argue about developing an algorithmic solution to various optimization problems.
- Formulate mathematical models of optimisation problems arising in various practical applications.
- Employ optimization methodologies and models, along with problem-specific fine tuning.
- Carry out a fully-fledged implementation of optimization tasks, using state-of-the-art libraries and tools.

Requirements and Prerequisites

There are no formal prerequisites. However, it is expected that the students should have familiarity and basic knowledge of algorithmic principles. It is also expected that the students will spend some amount of time working with relevant libraries and tools (including Pyomo and DOcplex) which will be introduced during the lectures. Finally, mathematical maturity and a basic background on calculus, algebra and discrete math is encouraged.

Required Course Materials

There is no required textbook. All course material will be provided in class and available for downloading.

Bibliography

The course lectures are mainly based on relevant chapters of the following books:

Lieberman, G.J. and Hillier, F.S., 2005. *Introduction to operations research* (Vol. 8). New York, NY, USA: McGraw-Hill.

Wolsey, L.A., 2020. *Integer programming*. John Wiley & Sons.

Boyd, S., Boyd, S.P., and Vandenberghe, L., 2004. *Convex optimization*. Cambridge university press.

Michel Gendreau, M., Potvin, J.-Y., 2019. *Handbook of Metaheuristics, International Series in Operations Research & Management Science*.

Hooker, J.N., 2012. *Integrated methods for optimization.* New York: Springer.

Sutton, R.S. and Barto, A.G., 1999. Reinforcement learning. *Journal of Cognitive Neuroscience*, *11*(1), pp.126-134.

Grading

Students will be graded based on: i) an individual assignment (15%), ii) a group project (35%) and iii) a written examination at the end of the course (50%).

Software/Computing requirements

There are no such requirements

Attendance Requirements

Class attendance is essential to succeed in this course and is part of your grade. In-class contribution is also a significant part of our shared learning experience. When possible, please notify the instructor in advance of an excused absence.

Students are responsible for keeping up with the course material. It is the student's obligation to bring oneself up to date on any missed coursework.

Please arrive to class on time and stay till the end of the class period. Chronically arriving late or leaving class early is unprofessional and disruptive to the entire class. Please also turn off all electronic devices prior to the start of class. Cell phones tablets and other electronic devices are a distraction to everyone.

Code of Ethics

Students may not work together on assignments unless the instructor gives permission to do so.

Exercise integrity in all aspects of one's academic work and do not engage in any method or means that provides an unfair advantage. In any case of doubt, students must be able to prove that they are the sole authors of their work by demonstrating their knowledge to the instructor.

Clearly acknowledge the work and efforts of others when submitting written work as one's own. Ideas, data, direct quotations (which should be designated with quotation marks), paraphrasing, creative expression, or any other incorporation of the work of others should be fully referenced. No plagiarism of any sort will be tolerated. This includes any material found on the internet. Reuse of material found in question and answer forums, code repositories, other lecture sites, etc., is unacceptable. You may use online material to deepen your understanding of a concept, not for finding answers.

Please report observed violations of this policy. Any violations will incur a fail grade at the course and reporting to the senate for further disciplinary action.

Course Syllabus

The course comprises ten three-hour units which are taught in a weekly basis.

Part 1: Solving convex optimisation problems

Units 1-3: Linear and Integer Programming

We will start with an introduction to optimization and discuss simple examples. We will then focus on one of the fundamental topics in optimization, namely optimizing linear objectives under linear constraints (linear programming). We will overview known methods, such as simplex and interior point methods along with software tools for solving linear programs. We will briefly also discuss a class of more difficult problems that are modelled using integer linear programming. The latter class will be revisited in Unit 6.

Units 4-5: Convex Optimization and Applications to Machine Learning

The next form of optimization we will discuss concerns convex objective functions. We will study the gradient descent method as well as applications of convex optimization to machine learning and other problems concerning big data. These include, among others, linear regression, the least square method, and support vector machines.

Part 2: Solving combinatorial optimization problems

Unit 6-7: Combinatorial optimization

In these units we will discuss how to formulate mathematical models for discrete optimization problems, i.e., problems with a finite solution space defined over combinatorial domains. We will refer to some classic examples including problems defined on graphs, such as the Traveling Salesman problem and the Vehicle Routing problem, as well as on scheduling problems on parallel machine environments. We will revisit Mixed Integer Linear Programming from Units 1-3 as a way of formulating and solving such problems and discuss the standard branch and bound method for obtaining exact optimal solutions.

Units 8-10: Exact and heuristic methods

Our next objective is to introduce a set of efficient candidate methods, including both exact and near-optimal heuristic approaches, for solving practical instances of various combinatorial optimization problems. To this end, we will first examine two exact methodologies: Constraint Programming (CP) and Benders Decomposition. Extensions of the latter facilitate the seamless integration of Mixed-Integer Linear Programming (MILP) and CP in a compact manner and have demonstrated high efficiency across multiple applications. Additionally, we will review the relevant software tools for implementing these techniques, with a particular focus on Pyomo and DOcplex.

Then, we will present efficient heuristics such as local search meta-heuristic methods (including simulated annealing and genetic algorithms) and Reinforcement Learning (RL) that leverages learning and reward-driven adaptation to obtain near-optimal solutions. These methods will be complemented by real-world applications including a transportation problem where we want to optimize last-mile delivery with the use of mobile depots, and production planning and scheduling problems in metal and textile industries to optimize the production of aluminum rolls and noble woolen fabrics.