Combinatorial Optimization

MS-EV0022

Lecturer Info

Fernando H. C. Dias

Office Hrs: Wednesday 14:00-15:00

Otakaari 1, Y214

fernando.dias@aalto.fi

Course Info

Monday

14:15-16:00

U5 - U157

Study Session Info

Wednesday

16:15-18:00

U5 - U157

TA Info

Overview

Mathematical optimization is a subfield from Applied Mathematics in which one seeks to find variable values (representing decisions) within a domain that maximise (or minimise) the value of a given function (a performance measure). Its applications lie even beyond mathematics and engineering, reaching transportation, air traffic control and bioinformatics. A special branch called combinatorial optimization regards problems in which the optimal solution is found from a finite set of potentially feasible solutions. It is a framework composed of problems in network modelling, greedy algorithms, heuristics, meta-heuristics and many others.

In this course, the student will learn the fundamentals of combinatorial optimization theory and practice and how they can be applied to real-world optimization problems. By the end of the course, it is expected that the student will be capable of analysing the main characteristics of an optimization problem, modelling using standard techniques and deciding the most suitable method for its solution.

Material

Required Texts

Lectures Notes and Lecture Slides. Both are available on "MyCourse" Page.

Text Book

Korte, B., and Vygen, J. (2011). *Combinatorial optimization*, volume 1. Springer.

Additional Material

Any extra material suggested throughout lectures will be available on "MyCourse" Page.

Learning Objectives

Upon completing this course, the student should be able to:

- Apply the theory and the algorithm from the course as a tool to solve real and more complex problems involving combinatorial and integer optimization;
- Familiarise themselves with classical problems involving graphs;
- Learn the difference between approximation, heuristics and greedy algorithms and know where to apply them wisely;
- Learn complexity theory and its application;
- Understand whether a given combinatorial optimization problem can be solved efficiently (in polynomial time);
- Know the main techniques for modelling and solving different combinatorial problems and how to apply them in practice;
- Know how to use optimization software for implementing and solving combinatorial optimization problems.

Teaching Methods

The course will be taught by a composition of the following methods:

- lectures;
- theoretical and computational exercises;
- project assignment and feedback.

The lectures will be in person and exercise sessions. To address general questions about the course content or administration, we will use Zulip. Zulip is the standard chat platform in the Department of Mathematics and Systems Analysis and offers the possibility for calls and using Latex on messages.

Remark: Zulip link: `https://combinatorics.zulip.aalto.fi/join/jcgepynco6d6v3f45rx7ksjq/`

As preparation for the lectures (Monday 14:15-16:00pm), the students will be encouraged to study and familiarise themselves with the lecture notes (about 10 pages per lecture) beforehand and formulate questions to be submitted before (via Zulip) or during the lectures. Lecture notes for each week will be available every Monday (prior to the weekly lecture).

The exercise sessions (Wednesday 16:15-18:00pm) will happen as self-study sessions, in which the students are expected to study the exercises and by the end, the teaching assistant will help and provide solutions to such exercises. In addition, whenever possible, due to time constraints, additional support on additional projects and homework will also be provided.

The time slot of the exercise sessions will work as office hours for the students to clarify questions related to content, projects and assignments. We will monitor the course's Zulip chat whenever possible to answer any questions. However, for timely support, we recommend that the students join the exercise sessions on Zulip and ask questions there.

Remark: Reception hour:
Lecturer: at 13:00 - 14:00 in room Y214 (Otakaari 1). Please confirm the appointment by contacting the lecturer via email first.

Assessment

The final grade of the course is composed of four components:

- Project I (P-I);
- Homework assignment I (HA-I);
- Homework assignment II (HA-II);
- Homework assignment III (HA-III);

Each component will be graded individually in a scale of 0-100. The final grade FG will be calculated as 60% for Project I and 45% from assignments.

The conversion scale for to the 1-5 scale is as follows.

1-5	0-100
Fail	0-49
1	50-59
2	60-69
3	70-79
4	80-89
5	90-100

Table 1: Conversion from 0-100 to 1-5 scale

Homework assignments

A total of 3 homework assignments will be handed out. Combined, the homework assignments will count for 40 points. The homework will be available on specific Mondays (see schedule below) from "MyCourse" and will have a deadline of three to four weeks (depending on the assignment - see schedule below). The submission of the solutions must be made through "MyCourse". Homework submissions after the deadlines will have a 3-point discount plus an extra 1 point each day (24 hours) after the deadline.

The homework will be composed of theoretical and computational exercises. The computational skills required to solve the exercises will be introduced in the exercise tutorials, but the student is expected to learn and practise the language independently. Supporting material for that will be provided. The programming language that will be used in this course is Python (python.org) (preferably) or Julia (julialang.org). Submissions using other languages will be allowed (but preferably use one of the suggested).

Project

The students will be requested to develop one single guided project worth 60 points. The objective of the project is to use the acquired knowledge in different combinatorial topics in practice and discuss related technical aspects.

The project can be conducted individually or in pairs. It will comprise an implementation using Python/Julia that addresses the requirements of the projects and a 3000-5000 word (approximately eight pages) report.

The project is organized in a way that overarches the entire course, meaning that as we progress through the course, different parts of the project will become more accessible to solve.

Workload estimate

The following breakdown shows an estimate of the total workload of the course. Recall that each ECT credit is equivalent to 28 hours of work; thus, 5 ECTs are equivalent to 140 hours.

- Pre-Lecture/Exercise Session preparations require 2 hour per week (as recommendation) (24 hours in total)
- Exercise Sessions require 2 hours per week (24 hours in total);
- Lectures requires 2 hours per week (24 hours in total);
- Homework assignments require 6 hours per assignment (18 hours in total);
- Project requires 40 hours per assignment (40 hours in total);
- The course requires 130 hours (5 ECT).

Deadline Extension Policy

Extension for the deadline for assignments and/or projects will only be allowed for students who have preemptively contacted the lecturer or teaching assistant *before the due date*. In case of extraordinary events, the student should notify either the lecturer or teaching assistant afterwards; then, a special exception will be discussed for each case.

Equality, Diversity and Inclusivity Statement

I consider this classroom to be a place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability - and other visible and non-visible differences. All class members are expected to contribute to a respectful, welcoming and inclusive environment for every other class member.

Accommodations for Students with Special Requirements

If you are a student with learning needs that require special accommodation, contact me as soon as possible to discuss how to improve your participation in this course. Please e-mail me as soon as possible to schedule a time to discuss your learning needs.

Academic Integrity

Research and Academic Integrity are central to the ideals of this course. Students are expected to be independently familiar with them and recognize that their work in the course will be their original work that truthfully represents the time and effort applied. Use of any AI-generated text or code is prohibited.

Lectures Schedule

Part I: Introduction to Combinatorics

Week 1	*Introduction* Admin. and Course Introduction Graph Basics & Optimization
Week 2	*Graph Problems I*: MST: Minimum Spanning Trees SP: Shortest Path
Week 3	*Graph Problems II*: Maximum Flow, Minimum Cost Flow
Week 4	*Graph Problems III*: Weighted Matching Maximum Matching

Part II: Complexity and Graphs

Week 5	*NP-Hard Problems I* TSP: Travelling Salesman Problem Vertex Cover Graph Colouring
Week 6	*NP-Hard Problems II* Cliques, Independent Set, Dominant Set Hamiltonian Path SAT, 3-SAT
Week 7	*Complexity* P vs. NP Polynomial Time-Reduction

Part III: Solutions Methods

Week 8	*Exact Methods*
	Enumeration
	Mathematical Modelling
	Dynamic Programming
Week 9	*Polyhedral Theory*
	Pohyhedral Theory
	Unimodularity
Week 10	*Approximations*
	Greedy Algorithms
	Local Search
	Exhaustive Search

Part IV: Real World

Week 11	*Real-World Applications*
	Flow Decomposition
	Portfolio Optimization
	Robust Combinatorial
	Transport
Week 12	*Guest*
	Guest Lecture
	(subject to change based on guest availability)

Teaching Sessions Schedule

Week 1	Combinatorics	Exercises session - Publication of HA-I
Week 2	Graph Problems I	Exercises session
Week 3	Graph Problems II	Exercises session - Publication of P-I
Week 4	Graph Problems III	Exercise session - Deadline of HA-I
Week 5	NP-Hard Problems I	Exercise session - Publication of HA-II
Week 6	NP-Hard Problems II	Exercise session
SPRING BREAK		
Week 7	Complexity	Exercise session
Week 8	Exact Methods	Exercise session - Deadline of HA-II
Week 9	Polyhedral Theory	Exercise session - Publication of HA-III
Week 10	Approximations	Exercise session
Week 11	Project Support	Exercise session
Week 12	Project Support	Exercise session - Deadline of HA-III - Deadline of P-I