Combinatorial Optimization

[View PDF](#)

Instructor(s):

Dávid Szeszlér
Péter Pál Pach

Short Description of the Course:
Combinatorial optimization is the art and science of finding the best solution out of a large but finite set of possible solutions. While in most practical applications scanning through all cases is only a theoretical possibility due to their enormous number, combinatorial optimization offers more sophisticated methods and algorithms resulting in reasonable running times. One of the most powerful tools of combinatorial optimization is *linear and integer programming*; this is a general framework capable of modeling and efficiently solving many problems arising in real-life applications. The objective of this course is to get acquainted with the basic notions and methods of this theory and use it for various applications.

Aim of the Course:
The course starts by introducing the necessary graph-theoretic background: efficient algorithms for the maximum size bipartite matching and the maximum network flow problems are covered. (Students should be aware that these two topics are also covered by the Graph Theory course offered by AIT; that is, the two courses share an overlap of not more than 10%).

Then the focus is moved to the main topic of the course: the basic notions of linear and integer programming are introduced and a glimpse of their range of applicability is given. Students are given the chance to model and solve miniatures of real-life problems that stem from the world of business and industry.

The next part of the course gives a deeper insight into the theory of linear programming. In particular, the fundamental notion of linear programming duality is introduced.

The last part of the course is dedicated to more advanced applications. First it covers a short introduction into the theory of two-player, zero sum games and then it puts graph-theoretic results on bipartite matchings and network flows into a more general context and addresses various generalizations.

Although the applications and optimization problems covered by the course are structured around the theory of linear and integer programming, a thorough description of the relevant cutting-edge algorithms is not covered.

Learning Objectives:
By completing the course the students

- will learn to use linear and integer programming to formulate various problems arising in real-life circumstances;
- will learn to identify the range and the limits of applicability of linear and integer programming;
- will learn to use certain efficient algorithms for some bipartite matching and network flow problems;
- will get a chance to further develop their mathematical skills.

Prerequisites:

The course builds on the following topics from linear algebra:

- elementary matrix operations (addition and multiplication) and their properties;

- Gaussian Elimination (also known as Row Reduction) Algorithm for solving systems of linear equations.

Furthermore, students are required to have sufficient mathematical experience and maturity that enables them to follow and understand somewhat more advanced proofs. Therefore students planning to take the course will be asked to write a **placement test** to assess their math skills before the term starts. This test does not require any kind of preparation, nor does it rely on the knowledge of any specific field of mathematics. Based on the result of the placement test students will receive a personal recommendation on whether or how much it is advised for them to take the course. In some cases students will even be asked not to take the course at all if the result of the placement test suggests that it would be unsuitable for them.

Detailed Program and Class Schedule:

1. Matchings in bipartite graphs, the augmenting path algorithm.
2. The maximum network flow problem, the Ford-Fulkerson algorithm.
3. The basic problem of linear programming.
   Graphical solution for two-variable problems.
4. Modeling practical problems as multivariable LP instances.
   Solving LP problems with Microsoft Excel.
5. The notion of integer programming.
   Modeling practical problems as IP instances. Using decision variables, incorporating logical constraints.
   Solving IP problems with Microsoft Excel.
6. Linear algebra revision: fundamental operations on matrices, determinants.
   The matrix form of LP problems.
7. Solving systems of linear inequalities with the Fourier-Motzkin elimination.

   Midterm test.

8. A necessary and sufficient condition for the solvability of systems of linear inequalities: the Farkas-lemma.
9. Boundedness of the objective function of a linear program.
   The concept of duality in linear programming.
10. The duality theorem of linear programming.
    Complementary slackness.
    Algorithmic complexity of the linear and integer programming problems.
11. Applications: the Ford-Fulkerson theorem for the maximum flow problem; the minimum cost flow and the multi-commodity flow problems; two-player, zero sum games.
12. The branch and bound method for integer programming.
    Integer programming with a totally unimodular coefficient matrix.
13. Applications: the minimum cost integer flow problem, the optimum assignment problem.
14. Egerváry's algorithm (the "Hungarian method") for the optimum assignment problem.

   Final exam.

Method of Instruction:
Each 100 minute class consists of two, roughly equal parts. In the first part of the class some new theory is introduced while the second part is dedicated to solving problems. Problems constitute an inherent part of the learning process: they give the students the opportunity to better understand and deepen their knowledge of the notions, theorems and algorithms covered in the first part of the class.

Students are given homeworks at the end of each class. These can be of two types: the first type is to be solved individually and then handed in. These problems are intended to ensure that the student has acquired the necessary command of the material to be able to further follow the course. Their solution requires

nothing but the adaptation of methods already covered in class. The second type of homework might require some original ideas and insights and students are encouraged to work together on these if they wish. The solutions of these problems often provide the grounds for getting acquainted with the new material next class.

Grading:
The final grade is made up of the following four components.

Component	Percentage	Explanation
Class Participation	10%	A high grade is achieved by showing positive, cooperative attitude towards common work in class and by arriving fully prepared at every class. Besides that, each missed class or late arrival to a class (without a legitimate reason) results in a loss of 2% or 1%, respectively from the class participation component.
Homeworks	30%	The maximum grade is achieved by doing all required hand ins. However, students are encouraged to reach above the "maximum" grade by solving and handing in extra problems.
Midterm Test	30%	The midterm test consists of problem solving only. The problems are similar, but not identical to the ones already solved in class. A sample midterm test is available here.
Final Exam	30%	The final exam consists of both problem solving and theory. Problems are similar to the ones in the midterm, however, they focus more on the second part of the semester. Theoretical questions either test the knowledge of definitions, theorems and algorithms covered or require reproducing a proof presented in class. A sample final exam is available here.

The maximum cutoffs for letter grades will be at the traditional 90%, 80%, etc. with plus and minus grades given at the top or bottom 3% of the intervals.

Textbook:
Combinatorial Optimization – handouts

Instructors' bio:

Dávid Szeszlér (born 1975) is an associate professor of the Department of Computer Science and Information Theory, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics (BME). In 1997 he was awarded as "Excellent Teacher of the Department", based on student feedback surveys. He graduated from Eötvös Loránd University as a mathematics teacher and English language technical translator in 1998. He obtained his Ph.D. in the field of VLSI routing in 2005; in the same year, he was awarded the "Farkas Gyula prize" of the János Bolyai Mathematical Society for applications of mathematics. His Erd?s number is 3.

Péter Pál Pach (born 1985) is an associate professor of the Department of Computer Science and Information Theory, Budapest University of Technology and Economics (BME) and the head of the MTA-BME Lendület Arithmetic Combinatorics research group. He received his M.Sc. in Mathematics (2009) and his Ph.D. in Mathematics and Computer Science (2013) from Eötvös Loránd University. His main research topic is Arithmetic Combinatorics. He won the Erd?s Pál Young Researcher Scholarship in 2014 and the Bolyai Research Scholarship of the Hungarian Academy of Sciences in 2015. As a high school student he participated twice at the International Mathematical Olympiad, and received a silver and a gold medal. His Erd?s number is 2. His hobby is long-distance running, in 2014 he finished 10th at the Marathon National Championship.