Combinatorial optimization	
Offered by	Department of Mathematics
Language	English
Primarily interesting for	Bachelor Applied Mathematics students
Prerequisites	Required courses: Linear algebra, analysis, set theory and algebra Recommended courses: -
Contact person	Dr. J.C.M. Keijsper (J.C.M.Keijsper@tue.nl)

Content and composition

Combinatorial optimization is concerned with selecting the best from a finite set of discrete alternatives. A classical example is the travelling salesman problem, where the goal is to determine the shortest tour passing through a given set of cities. Even when it is clear that there is just a finite set of possible routes, it is not practically feasible to select the best by enumerating all possibilities. Problems such as this are common in science and business, arising as scheduling, routing, packing and rostering problems. The key problem of combinatorial optimization is to develop algorithms for solving such problems efficiently or for finding a reasonable approximation to the best possible solution in limited time. This package gives an introduction to several basic mathematical tools for analyzing and solving such combinatorial optimization problems: graph theory, linear and integer linear optimization, and a catalogue of standard algorithms together with an analysis of their efficiency.

Course code	Course name	Level classification
2WO20	Linear optimization	3. Advanced
2WF60	Graph theory and combinatorics	2. Deepening
2ILC0	Algorithms	3. Advanced

Note that also the combination of only the first two courses from this package, 2WO20 and 2WF60, is considered a coherent elective package (of 10 ECTS).

Students who would like to take the full package of (15 ECTS) are advised to take Graph theory and combinatorics (2WF60) already in the second year of the bachelor, as it contains some prerequisites for Algorithms (2ILCO) (for students of Applied Mathematics, the course 2WF60 can replace the course Datastructures (2IL50) as a prerequisite for 2ILC0).

Course description

Linear optimization (2WO20)
This course treats a key method for solving (discrete) optimization problems. Linear optimization is deciding the values of several unknown quantities, so as to maximize a given linear function of these unknowns, subject to given linear inequalities. There are outstanding algorithms for solving problems of this type. Many real-life problems can be cast as a linear optimization problem and solved as such. On the theoretical side, it is taught how to attain certainty about the optimality of solutions of linear optimization problems. This involves studying linear inequalities, the separation theorem for convex sets, Farkas' lemma, and the duality theorem for linear optimization. We describe the simplex method for solving linear optimization problems, and we prove that it works.

Combinatorial optimization

Finally, we discuss two extensions of linear optimization: convex optimization and integer linear optimization. On the practical side, you will learn how to model real-world problems as linear optimization problems and to use existing solvers for evaluating the resulting models.

Graph theory and combinatorics (2WF60)
In the combinatorics part subjects include counting problems, generating functions, recurrence relations and analysis of sorting algorithms. In the graph theory part, first structural properties of graphs are studied, and then several (algorithmic) graph problems are considered, such as Euler tour, Hamilton circuit, graph coloring, graph search, the shortest path problem, the minimum spanning tree problem, the maximum flow problem, and the maximum (weight) bipartite matching problem. We also touch upon the subject of computational complexity theory, by looking at possible (computer) representations of graphs and by analyzing the efficiency of the graph algorithms that are considered in this course. Graph theory and combinatorics is a level 2 course, which means that it can be taken in the second year of the bachelor applied mathematics (the only prerequisites are first year bachelor courses). The other courses in this elective package are level 3 courses.

Algorithms (2ILC0)
This course consists of three parts. In the first part, three general techniques for solving optimization problems are studied: backtracking, dynamic programming, and greedy algorithms. The second part of the course deals with algorithms for optimization problems in graphs: shortest paths, maximum flows and matchings. In the third part of the course we study computational complexity theory by looking at NP-completeness, which investigates the limits of what is efficiently computable.