**Università degli Studi di L'Aquila - Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica**

# Course catalogue

*Page compiled on 21/07/2018*

| Programme of Integrated course "Ricerca operativa e Ottimizzazione" |
|---|
| This course is composed of 2 Modules: 1) Ottimizzazione Combinatoria, 2) Ricerca Operativa |
| Programme of Module "Ottimizzazione Combinatoria" |

- Code: F0141
- Type of course unit: Compulsory (Bachelor Degree in Computer Science curriculum General)
- Level of course unit: Undergraduate Degrees
- Semester: 2

Number of ects credits: (Bachelor Degree in Computer Science) 6 (workload 150 hours)

Teachers: Claudio Arbib (Claudio.Arbib@univaq.it)

| | | |
|---|---|---|
| **1** | **Course objectives** | Learn algorithmic techniques for some combinatorial optimization problems. Being able to formulate and solve combinatorial optimization problems using integer linear programming. Understand the complexity of the problems studied. |
| **2** | **Course content and learning outcomes (dublin descriptors)** | Topics of the module include:<br>• Prerequisites: graphs, definitions and basic properties. Elements of computational complexity: classes P, NP; polynomial-time reduction and the class NP-complete. Examples. Combinatorial optimization problems. Basic definitions. Examples: Transversal (vertex-cover), stable set, dominating set, edge-cover, (perfect) matching, knapsack, graph colouring (chromatic number and index), shortest path, spanning subgraph etc. Formulation as 0-1 linear programming.<br>• Unimodular and totally unimodular matrices. Necessary conditions, sufficient conditions. Hoffmann-Kruskal's Theorem. Convex hull and ideal formulation of an integer LP. Examples: shortest path, min-cost flow, bipartite matching, interval graphs.<br>• Recursion in CO: paths on directed acyclic graphs (DAG) and dynamic programming. Examples of application: optimal paths in a DAG, 0-1 Knapsack. [Projection of polyhedra, systems of linear inequalities, Fourier-Veronese's Theorem, Fourier-Motzkin's Method. Fundamentals of duality theory in LP. The Primal-Dual Method. Example of application: Dijkstra's Algorithm].<br>• Matroids and the Greedy Algorithm. Definition and characterization of a matroid (Rado's Theorem). Basics of linear algebra: linear/affine dependence-independence, bases, uniqueness of representation, Steinitz's (or exchange) Theorem. Examples: trivial matroid, graphical matroid, partition matroid, vector matroid, [matching matroid]. Matroid representation. [An industrial application]. [Rank function, submodular and supermodular set functions. Rank of a matroid. Support function of a convex set. Submodular polyhedron. Lòvasz's extension. Matroid polyhedron, matroid intersection, intersection of two submodular polyhedra].<br>• Guaranteed approximation algorithms for CO. Definitions. Polynomial approximation schemes (PTAS, EPTAS, FPTAS). Examples: TSP: double tree and Christofides; Knapsack 0-1.<br>• Matching (perfect/non-perfect, weighted/unweighted, bipartite/non-bipartite). Weak dual inequalities: transversal, stability number, edge-cover and matching. The bipartite case: Gallai's and König's Theorem. [Bi-stochastic matrices and perfect bipartite matching. Non-bipartite matching: matching polyhedron, Edmonds' Theorem].<br>• Linear relaxation of a (mixed) integer linear program. Implicit enumeration: the Branch-and-Bound Method. Linear bounds, example: integer Knapsack, 0-1 Knapsack. Dichotomy on a fractional variable. Combinatorial bounds, example: TSP.<br>• [Separation of an integer polyhedron. Tips on the Ellipsoid Method. Separation and optimization: Finding valid inequalities. Solution of LP with exponentially many variables, example: Min-cut, TSP, 0-1 Knapsack, Stable Set].<br><br>On successful completion of this module, the student should :<br>• Being able to evaluate the general complexity of combinatorial optimization problems. |

| | | |
|---|---|---|
| | | Being able to formulate them as 0-1 linear programming. Know basic primal-dual relations. Know elementary matroid theory, dynamic programming. Have the notion and know the main properties of unimodular matrices. Have basic notions on algorithm and problem complexity. Know standard algorithms for spanning tree, bipartite matching, shortest path. Know heuristics for the TSP and Knapsack. Know general implicit enumeration, LP based algorithms (branch-and-bound)<br>• Understand the difference between an “easy” and a “hard” problem. Evaluate the complexity of an algorithm and, for simple cases, of a problem. Be able to recognize whether a matrix is, or is not, totally unimodular. Solve by standard algorithms the spanning tree, bipartite matching, knapsack, optimal path, travelling salesman problem. Formulate combinatorial optimization problems, or binary optimization problems derived from applications, as 0-1 linear programming.<br>• Understand when a problem has the shape of a matroid and what is the advantage. Know what is the advantage of a totally unimodular matrix. Decide when it is convenient to use dynamic programming. Realize the limits of heuristics.<br>• Prove rigorously a simple theorem. Prove or disprove (by counterexample) a simple conjecture. Understand the role played by combinatorial optimization in applications.<br>• Address submodular set functions, polyhedral combinatorics, advanced integer linear programming. |
| **3** | **Course prerequisites** | Notion on basic algorithms and linear algebra, linear programming |
| **4** | **Teaching methodsand language** | lectures, exercises<br>**Language**: Italian<br>**Reference textbooks**<br>• W.J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver, ***Combinatorial Optimization***. Wiley. 1997.<br>• C.H. Papadimitriou, K. Steiglitz, ***Combinatorial Optimization: Algorithms and Complexity***. Dover Books on Computer Science. 1998. *http://www.amazon.com/Combinatorial-Optimization-Algorithms-Complexity-Computer/dp/0486402584*<br>• A. Sassano,, ***Modelli e Algoritmi della Ricerca Operativa***. Franco Angeli Editore. |
| **5** | **Assessment methods** | written test, oral test |

## Programme of Module "Ricerca Operativa"

- Code: F0140
- Type of course unit: Compulsory (Bachelor Degree in Computer Science curriculum General)
- Level of course unit: Undergraduate Degrees
- Semester: 2

Number of ects credits: (Bachelor Degree in Computer Science) 6 (workload 150 hours)

Teachers: Stefano Smriglio (Stefano.Smriglio@univaq.it)

| | | |
|---|---|---|
| **1** | **Course objectives** | Introduce the student to the formulation of basic Optimization problems, particularly Linear Optimization problems, and train him/her to the related solution algorithms. |
| **2** | **Course content and learning outcomes (dublin descriptors)** | Topics of the module include:<br>• Optimization problems: decision varibles, objectives and constraints; modeling techniques and model classification<br>• Convex optimization problems; local and global optima<br>• Geometry of Linear Programming<br>• The Simplex method<br>• Duality theory in Linear Programming and its applications<br>• Dual interpretation of the simplex method and the dual simplex method<br><br>On successful completion of this module, the student should :<br><br>• Acquire the knowledge of Optimization problems and of the mathematical modeling techniques for complex decisions. Acquire the knowledge of some solution algorithms for Linear Programming problems.<br>• Acquire the ability to recognize optimization problems and develop mathematical |

| | | |
|---|---|---|
| | | models of decision-making problems. Acquire the ability of computing solutions of linear programming problems<br>• Acquire autonomy in modeling and algorithmic choices for problems related to complex decision-making<br>• Be able to hold a conversation and to read texts on topics related to the modeling of decision problems and Linear Programming<br>• Acquire the ability of upgrading flexible knowledge and skills in the field of Optimization and related problems that arise in various areas, such as mathematics, computer science and management science |
| **3** | **Course prerequisites** | Vector space, scalar product, matrix product, inverse matrix |
| **4** | **Teaching methodsand language** | teaching lessons<br>**Language**: Italian<br>**Reference textbooks**<br>• Dimitris Bertsimas and John N. Tsitsiklis, ***Introduction to Linear Optimization***. Athena Scientific. 1997.<br>• Matteo Fischetti, ***Lezioni di Ricerca Operativa***. Progetto Libreria Padova. 1995.<br>• Antonio Sassano, ***Modelli e Algoritmi della Ricerca Operativa***. Franco Angeli. 1992. |
| **5** | **Assessment methods** | 1. paper test consisting of various exercises (problem formulation, insights about algebraic or geometric problems properties, problem solution by know algorithms) 2. oral test about theoretical topics; this is accessible only for the students who earned a passing grade at the paper test; NOTE: a sufficient paper test allows the student only for the oral at the same date, but NOT for next dates. 3. A mid-term test is also planned: a positive grade to it allows the student to skip the corresponding topics in the final test. |