



Programme of Integrated course "Network design"

This course is composed of 2 Modules: 1) Network Flows, 2) Network Optimization

Programme of Module "Network Flows"

- Code: DT0059
- Type of course unit:
- Level of course unit:
- Semester: 2

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

Teachers: Fabrizio Rossi (fabrizio.rossi@univaq.it)

1	<b>Course objectives</b>	Ability to recognize and formulate network flow problems Knowledge of basic and advanced network flow algorithms Ability to design resolution approaches to solve non standard network flow problems
2	<b>Course content and learning outcomes (dublin descriptors)</b>	<p>Topics of the module include:</p> <ul style="list-style-type: none"> <li>• Network Flows Problem: introduction and definitions</li> <li>• Maximum Flows and the path packing problem. Flows and cuts: Max-Flow/Min-Cut theorem. Augmenting path algorithms: Ford and Fulkerson algorithm, Edmonds and Karp algorithm. Generic Preflow-Push algorithm. Flows with lower bounds.</li> <li>• Maximum Flows: additional topics and applications. Flows in Unit Capacity Networks. Flows in Bipartite Networks. Network Connectivity.</li> <li>• Minimum Cuts. Global Minimum Cuts. Node Identification Algorithm. Random Contraction. Applications.</li> <li>• Minimum-Cost Flow Problems. Definition and applications. Optimality Conditions. The Ford-Bellman algorithm for the shortest path problem. Primal algorithms: Augmenting Circuit Algorithm for the Min Cost Flow Problem.</li> <li>• Network Simplex Algorithms. Applications of Min Cost Flows.</li> </ul> <p>On successful completion of this module, the student should :</p> <ul style="list-style-type: none"> <li>• Know and formulate network flow problems</li> <li>• Model decision problems as network flow problems Use base and advanced algorithms to solve network flow problems</li> <li>• Ability to identify network flow models scope</li> <li>• Ability to explain network flows models and algorithms</li> <li>• Ability to learn state-of-art algorithms for network flow problems</li> </ul>
3	<b>Course prerequisites</b>	Basic knowledge of: Discrete Mathematics, Linear Programming, Algorithms and Data Structures, Computational complexity
4	<b>Teaching methods and language</b>	<p>Lectures</p> <p><b>Language:</b> English</p> <p><b>Reference textbooks</b></p> <ul style="list-style-type: none"> <li>• Cunningham, Pulleyblank, Schrijver , <b>Combinatorial Optimization</b>.</li> <li>• Ahuja, Magnanti, Orlin, <b>Network Flows</b>.</li> </ul>
5	<b>Assessment methods</b>	Written text exam

Programme of Module "Network Optimization"

- Code: DT0060
- Type of course unit: Compulsory (Master Degree in Computer Science curriculum SEAS)
- Level of course unit: Postgraduate Degrees
- Semester: 2

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

Teachers: Fabrizio Rossi (fabrizio.rossi@univaq.it)

1	<b>Course objectives</b>	Ability to recognize and model network optimization problems as Integer Linear
---	--------------------------	--------------------------------------------------------------------------------

		Programming problems. Knowledge of fundamental algorithmic techniques for solving large scale Integer Linear Programming problems. Knowledge of commercial and open source Integer Linear Programming solvers.
2	<b>Course content and learning outcomes (dublin descriptors)</b>	<p>Topics of the module include:</p> <ul style="list-style-type: none"> <li>• Formulations of Integer and Binary Programs: The Assignment Problem; The Stable Set Problem; Set Covering, Packing and Partitioning; Minimum Spanning Tree; Traveling Salesperson Problem (TSP); Formulations of logical conditions.</li> <li>• Mixed Integer Formulations: Modeling Fixed Costs; Uncapacitated Facility Location; Uncapacitated Lot Sizing; Discrete Alternatives; Disjunctive Formulations.</li> <li>• Optimality, Relaxation and Bounds. Geometry of <math>R^n</math>: Linear and affine spaces; Polyhedra: dimension, representations, valid inequalities, faces, vertices and facets; Alternative (extended) formulations; Good and Ideal formulations.</li> <li>• LP based branch-and-bound algorithm: Preprocessing, Branching strategies, Node and variable selection strategies, Primal heuristics.</li> <li>• Cutting Planes algorithms. Valid inequalities. Automatic Reformulation: Gomory's Fractional Cutting Plane Algorithm. Strong valid inequalities: Cover inequalities, lifted cover inequalities; Clique inequalities; Subtour inequalities. Branch-and-cut algorithm.</li> <li>• Software tools for Mixed Integer Programming</li> <li>• Lagrangian Duality: Lagrangian relaxation; Lagrangian heuristics.</li> <li>• Network Problems: formulations and algorithms. Constrained Spanning Tree Problems; Constrained Shortest Path Problem; Multicommodity Flows; Symmetric and Asymmetric Traveling Salesman Problem; Vehicle Routing Problem Steiner Tree Problem; Network Design. Local Search Tabu search and Simulated Annealing MIP based heuristics</li> <li>• Heuristics for network problems: local search, tabu search, simulated annealing, MIP based heuristics.</li> </ul> <p>On successful completion of this module, the student should :</p> <ul style="list-style-type: none"> <li>• Know and define single objective network optimization problems</li> <li>• Use and design exact or heuristic algorithms to solve single objective network optimization problems</li> <li>• Ability to judge models and methods to tackle network optimization problems</li> <li>• Ability to explain the models, the algorithms and the computational complexity needed to solve network optimization problems</li> <li>• Ability to learn state-of-art algorithms for network optimization problems</li> </ul>
3	<b>Course prerequisites</b>	Basic knowledge of: Discrete Mathematics, Linear Programming, Algorithms and Data Structures, Computational complexity. Knowledge of at least one programming language.
4	<b>Teaching methods and language</b>	<p>Lectures and software training</p> <p><b>Language:</b> English</p> <p><b>Reference textbooks</b></p> <ul style="list-style-type: none"> <li>• L.A. Wolsey, <i>Integer Programming</i>. Wiley. 1998.</li> </ul>
5	<b>Assessment methods</b>	Written text exam and assignment