POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

COURSE DESCRIPTION CARD - SYLLABUS

Course name
Multiobjective Optimization [S2SI1E>OPW]

Course

Field of study
Artificial Intelligence

Year/Semester
1/1

Area of study (specialization)
–

Profile of study
general academic

Level of study
second-cycle

Course offered in
English

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
15

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr inż. Michał Tomczyk
michal.tomczyk@put.poznan.pl

Lecturers

Prerequisites

Mathematical knowledge from the secondary school. Programming skills. Knowledge of the Python language.

Course objective

The course aims to introduce the students to the main topics in Operational Research (OR). These include linear programming, simplex algorithm, dual programming, sensitivity analysis, network optimization models, dynamic programming, integer programming, nonlinear programming, job scheduling, and heuristics. The students should get to know the basic methods, techniques, and algorithms for each of these sub-fields to use them for practical problem-solving.

Course-related learning outcomes

Knowledge
K2st_W1: has advanced and in-depth knowledge of widely understood information systems, artificial intelligence systems, theoretical foundations of their construction and methods, tools and programming environments used to implement them.
K2st_W2: has a structured and theoretically founded general knowledge related to key issues in the field of

computer science with the emphasis on the artificial intelligence and related fields.
K2st_W3: has advanced detailed knowledge regarding selected issues in artificial intelligence and related fields.

Skills
K2st_U3: is able to plan and carry out experiments, including computer measurements and simulations, interpret the obtained results and draw conclusions and formulate and verify hypotheses related to complex engineering problems and simple research problems.
K2st_U4: can use analytical, simulation and experimental methods to formulate and solve engineering problems and simple research problems.
K2st_U5: can - when formulating and solving engineering tasks - integrate knowledge from different areas of computer science and artificial intelligence (and if necessary also knowledge from other scientific disciplines) and apply a systemic approach, also taking into account non-technical aspects.
K2st_U10: is able - using among others conceptually new methods - to solve complex IT tasks, including artificial intelligence tasks, atypical tasks and tasks containing a research component.
K2st_U12: can communicate both in Polish and English using different techniques in a professional environment and in other environments, also using IT tools.
K2st_U13: is able to prepare and present a scientific study in Polish and English, presenting the results of scientific research or oral presentation on specific issues in the field of computer science and artificial intelligence.

Social competences
K2st_K1: understands that in the field of IT with particular emphasis on the artificial intelligence, the knowledge and skills quickly become obsolete.
K2st_K2: understands the importance of using the latest knowledge in the field of computer science and artificial intelligence in solving research and practical problems.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture: Assessment test is conducted at the last lecture. The students need to solve several computational tasks concerning the subjects presented during all lectures. Each task is evaluated individually, being allocated a certain number of points. The points are summed up and a standard scale is used to derive the final marks: <50% - 2.0, [50%, 60%) - 3.0, [60%,70%) - 3.5, [70%,80%) - 4.0, [80%, 90%) - 4.5, and [90%, 100%] - 5.0.
Laboratory classes. After each class, students solve practical, programming assignments and report their solutions to the instructors leading the laboratory classes within two weeks. Each assignment is evaluated on a scale from 2.0 to 5.0. The final grade is computed as an average from the individual marks, with the proviso that the two worst out of ten marks obtained throughout the semester may be neglected.

Programme content

Linear programming, the simplex method, duality theory, sensitivity analysis, network optimization models, dynamic programming, integer programming, nonlinear programming, metaheuristics, job scheduling, queueing theory.

Course topics

Linear programming: basic notation and possible transformations of constraints; modeling an optimization problem using linear functions; solving the problem using a graphical approach; formulating a matrix representation of the model.
The simplex method: obtaining the augmented representation of the model; introduction to the simplex method; solving the problem algebraically and using the simplex tableau; introduction to the Big M method.
Duality theory: the matrix form of the optimization model; the fundamental insight for solving problems using the simplex algorithm; duality theory and interpretation of the dual problem; specifying primal-dual relationships.
Sensitivity analysis: using the fundamental insight for revising the simplex tableau; the general procedure for the sensitivity analysis, analyzing how changes in the model can potentially affect the

solution; introduction to the dual-simplex algorithm.
Network optimization models: the transportation and assignment problems; the transportation simplex method and the Hungarian algorithm; the shortest path problem; the Dijkstra's algorithm; the minimum spanning tree problem; the maximum flow problem; the minimum cost flow problem; introduction to the network simplex method.
Dynamic programming: solving typical for the scope of OR problems using deterministic dynamic programming, concerning discrete and continuous decision variables; probabilistic dynamic programming.
Integer programming: applications; branch and bound algorithm for solving pure binary integer problems and mixed-integer problems; the branch-and-cut algorithm.
Nonlinear programming: graphical illustration, types of nonlinear problems, the Karush-Kuhn-Tucker condition for constrained optimization; quadratic programming.
Metaheuristics: solving typical for the scope of OR problems using tabu search, simulated annealing, ant colony optimization, and evolutionary algorithms.
Job scheduling: single and multi-stage job scheduling problems, open-shop problems, flow-shop problems, job-shop problems.
Queueing theory: basic queueing system, M/M/s processes.

Teaching methods

Lecture: slide show presentations on different sub-fields of Operational Research, illustrated with examples and practical assignments that serve as a summary of the lectures and preparation for the assessment test.
Laboratory classes: solving illustrative examples on board and coding problem solutions in Python, conducting computational experiments, discussion on the chosen methods, teamwork.

Bibliography

Basic:
Introduction to Operations Research, F. S. Hiller, G. J. Lieberman, McGraw-Hill, 2021.
Linear and nonlinear programming, D. G. Luenberger, Y. Ye., Springer, cop. 2008.

Additional:
Introduction to Stochastic Models in Operations Research, F. S. Hiller, G. J. Lieberman, McGraw-Hill, 1990.
Introduction to Operations Research, G. J. Ecker, M. Kupferschmid, John Wiley, 1988.
Linear programming : basic theory and applications, L. W. Swanson, McGraw-Hill Book Company, cop. 1980.

Breakdown of average student's workload

	Hours	ECTS
Total workload	75	3,00
Classes requiring direct contact with the teacher	30	1,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	45	1,50