POZNAN UNIVERSITY OF TECHNOLOGY
EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)
pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name
Multiobjective Optimization

Course

Field of study
Artificial Intelligence

Year/Semester
1/1

Area of study (specialization)

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
English

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
15

Other (e.g. online)

Tutorials

Projects/seminars

Number of credit points

3

Lecturers

Responsible for the course/lecturer:
Michał Tomczyk, Ph. D.

e-mail: michal.tomczyk@cs.put.poznan.pl

tel: +48 61 665 3020

Faculty of Computing and Telecommunications
Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

Basic mathematical knowledge from the secondary school. Basic programming skill in Python developed at the course on "Introduction to programming."

Course objective

The course aims to introduce the students to the main topics in Multiple Objective Optimization (MOO). These include classical methods for multiple objective optimization, evolutionary methods for multiple objective optimization, visualization of non-dominated solutions, quality assessment, discussion on problem properties & challenges, preference-based methods, and discussion on real-world applications.

The students will learn the basic methods, techniques, and algorithms and how 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: Lecture: The assessment test is conducted at the last lecture. The students are asked several theoretical and practical questions. 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 the next week. 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.

Programme content

Introduction & real-world applications: introduction to multiobjective optimization, presenting basic concepts, discussion on several selected real-world case studies.

Classical methods: introduction to classical methods based on linear programming for solving multiple objective optimization problems, e.g., the Weighted Sum Method or the Epsilon-Constrained Method; discussion on their strengths and weaknesses.

Evolutionary methods: introduction to evolutionary methods for solving multiple objective optimization problems, e.g., NSGA-II, MOEA/D, discussion on their strengths and weaknesses, discussion of different classes of methods in this stream such as, e.g., the indicator- or the decomposition-based methods.

Visualization & quality assessment: introduction of different ways of assessing the performance of methods for multiple objective optimization problems, including quantitative measures and visualization techniques.

Problem properties & challenges: introduction of different properties of multiple objective optimization problems, e.g., multi-modality or bias, and discussion about possible ways of handling them.

Preference-based methods: introduction to hybrid methods that, instead of approximating a whole Pareto frontier, incorporate the decision maker's preferences to identify their highly preferred solutions.

Teaching methods

Lecture: slide show presentations on different topics of multiobjective optimization, illustrated with examples and practical assignments that serve as a summary of the lectures and preparation for the assessment test.

Laboratory classes: coding problem solutions in Python, conducting computational experiments, discussion on the chosen methods, teamwork.

Bibliography

Basic

Multiobjective Optimization: Ineractive and Evolutionary Approaches, J. Branke, K. Deb, K. Miettinen, R. Słowiński, 2008.

Additional

Recent Advances in Evoutionary Multi-objective Optimization, S. Bechikh, Rituparna Datta, Abhishek Gupta, 2017.

Multi-Objective Optimization using Evolutionary Algorithms, K. Deb, 2001.

Breakdown of average student's workload

	Hours	ECTS
Total workload	75	3,0
Classes requiring direct contact with the teacher	30	1,5
Student's own work (preparation for laboratory classes, preparation for tests) [1]	45	1,5

[1] delete or add other activities as appropriate