COURSE DESCRIPTION CARD - SYLLABUS

Course name

Optimization methods

Course

Field of study	Year/Semester
Mathematics in Technology	3/6
Area of study (specialization)	Profile of study
	general academic
Level of study	Course offered in
First-cycle studies	Polish
Form of study	Requirements
full-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	30	
Tutorials	Projects/seminars	

Number of credit points

4

Lecturers

Responsible for the course/lecturer:

dr inż. Karol Gajda

email: karol.gajda@put.poznan.pl

tel. 61665 2805

Faculty of Control, Robotics and Electrical Engineering

Piotrowo 3A, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

The student starting this subject should have knowledge and skills of the linear and quadratic programming course. Should know the limits of their own knowledge and understand the need for further education.

Course objective

Presentation of selected optimization methods, graph algorithms, task scheduling and network flows.

Course-related learning outcomes

Knowledge

1. has extended and in-depth general knowledge of various branches of higher mathematics, including theorems and proofs, and advanced detailed knowledge about the application of mathematical techniques, methods and tools in engineering and technical sciences

2. has extended and in-depth knowledge of mathematical modelling

3. has systematized knowledge of terminology in mathematics and selected issues in the field of engineering and technical sciences related to the field of study, also in a foreign language

4. has deepened and theoretically founded knowledge of computer science, including numerical methods; knows at least one software package or a programming language in detail

Skills

1. can use knowledge of higher mathematics;

2. can build and analyse simple mathematical models;

3. can use mathematical tools and methods, including numerical ones, to solve engineering problems;

4. is able to construct an algorithm for solving a simple engineering task as well as implement and test it in a selected programming environment;

5. knows how to use a foreign language sufficiently to communicate, as well as read and understand mathematical texts, technical documentation and similar documents;

6. can work individually and in a team and interact with other people; knows how to estimate the time needed to complete the task; is able to develop and implement a work schedule to ensure that the deadline is met;

7. is able to plan and implement self-education independently in order to raise and update their competences.

Social competences

1. is aware of the level of their knowledge in relation to research in exact and natural sciences as well as engineering and technical sciences;

2. is aware of the deepening and expanding knowledge to solve new technical problems;

3. is able to work as a team; understands the need for systematic work on any projects that have a long-term nature.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Knowledge acquired during the lecture is verified by a 45-minute colloquium consisting of variously scored questions (test and open). Passing threshold: 50% of points. Final issues on the basis of which questions are prepared will be forwarded to students during the lecture preceding the colloquium, or sent by e-mail using the university's e-mail system.

Skills acquired as part of the laboratory are verified on the basis of developed projects and final test. Passing threshold: 50% of points.

Programme content

Undirected graphs: searching in depth and in breadth; shortest paths from one source.

Directed graphs: Reachability from a single source and from multiple sources; paths from a single source; shortest paths from a single source; detection of directed cycles; vertex cleanup when searching in depth; scheduling with priority constraints; topological sort.

Minimal spanning trees: Prim's algorithm; Kruskal's algorithm; Fredman-Tarjan algorithm.

Algorithms for determining the shortest paths: Dijkstra; topological sort; Bellman-Ford.

Algorithms for flow networks: Ford-Fulkerson algorithm for determining the maximum flow.

Teaching methods

1) lectures:

- lecture with presentation supplemented with examples given on the board,

- a lecture conducted in an interactive manner with formulating questions to a group of students or to specific students indicated,

- students' activity during classes is taken into account when issuing the final mark,

- during the lecture initiating the discussion,

- theory presented in close connection with practice,

- theory presented in connection with the current knowledge of students,

- presenting a new topic preceded by a reminder of related content known to students in other subjects.

2) laboratory:

- laboratories supplemented with multimedia presentations,

- detailed reviewing of reports by the laboratory chair and discussions on comments,

- using tools that enable students to perform tasks at home (eg open source software),

- demonstrations,

- work in teams,

- computational experiments.

Bibliography

Basic

1. Sedgewick R., Wayne K., Algorithms, 4th Edition, Addison-Wesley Professional 2011,

2. Kusiak J., Danielewska-Tułecka A., Oprocha P., Optymalizacja. Wybrane metody z przykładami zastosowań, PWN, 2019,

3. Horla D., Metody obliczeniowe optymalizacji w zadaniach, WPP, Poznań, 2016,

4. Jędrzejczyk Z., Kukuła K., Skrzypek J., Walkosz A., Badania operacyjne w przykładach i zadaniach, PWN, Warszawa, 2016.

Additional

1. Kincaid D., Cheney W., Numerical Analysis: Mathematics of Scientific Computing (The Sally Series; Pure and Applied Undergraduate Texts, Vol. 2),

2. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C., Introduction to Algorithms.

Breakdown of average student's workload

	Hours	ECTS
Total workload	102	4,0
Classes requiring direct contact with the teacher	62	2,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests, project preparation) [1]	40	2,0

[1] delete or add other activities as appropriate