COURSE DESCRIPTION CARD - SYLLABUS

Course name
Evolutionary computation

Course

Field of study
Artificial Intelligence

Year/Semester
4/7

Area of study (specialization)

Profile of study
general academic

Level of study
First-cycle studies

Course offered in
English

Form of study
full-time

Requirements
elective

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
22	22	0
Tutorials	**Projects/seminars**	
0	0	

Number of credit points

4

Lecturers

Responsible for the course/lecturer:
Prof. Andrzej Jaszkiewicz

email: andrzej.jaszkiewicz@put.poznan.pl

tel. 61 665 3420

Faculty of Computing and Telecommunications

Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

Student entering this course should have a basic knowledge about discrete mathematics, algorithms and data structures or pracical algorithmics, operational research, combinatorial optimization, statistics and data analysis, programming.

Course objective

The objective of this course is to give the students knowledge about evolutinary optimization methods with a focus on discrete/combinatorial problems. After completion of this course student should have skills allowing designing and implementing an efficient optimization method for a given optimization problem. The student should also be able to seek possibilities for further improvements of this method in the scientific and technical literature.

Course-related learning outcomes

Knowledge

K1st_W2: has a basic, ordered, and well-grounded knowledge essential for important areas of computer science such as evolutionary computations

K1st_W3: has a well-grounded knowledge of fundamental computer science problems within the scope of evolutionary computations

K1st_W4: knows and understands the basic techniques, methods, algorithms, and tools used for solving computer problems as well as problems in evolutionary computations

K1st_W7: has a basic knowledge of the life cycle and processes taking place in computers and, in particular, evolutionary computations-based software and hardware

Skills

K1st_U3: can formulate and solve complex data mining, optimization, and decision problems within the scope of computer science and, in particular, evolutionary computations, by applying appropriately selected methods

K1st_U7: can carry out a critical analysis and an assessment of the functioning of both computer systems and EC methods

K1st_U8: can design - following a pre-defined specification - and create an EC system by first selecting and then using the available methods, techniques, and computer tools (including programming languages)

K1st_U9: can adapt the existing algorithms as well as formulate and implement the novel algorithms, including the algorithms typical for different streams of EC

K1st_U10: can retrieve, analyze and transform different types of data, and carry out data synthesis to knowledge and conclusions useful for solving a variety of decision problems

K1st_U11: can adapt and make use of the models of intelligent behavior (e.g., evolutionary algorithms) as well as computer tools simulating such a behavior

K1st_U16: can plan and carry out life-long learning, and is aware of the possibilities of MSc studies

Social competences

K1st_K1: understands that knowledge and skills quickly become outdated in EC, and perceives the need for constant additional training and raising one's qualifications

K1st_K2: is aware of the importance of scientific knowledge and research related to EC in solving practical problems which are essential for the functioning of individuals, firms, organizations as well as the entire society within such example application fields as transport, healthcare, education, home/service robots, public safety, and entertainment

K1st_K5: can think and act in an enterprising way, finding the commercial application for the created EC-based systems, having in mind the economic benefits as well as legal and social issues

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:

a) lectures:

- based on student's activity during lectures and answers to lecturer's questions concerning material from the previous classes,

b) laboratories:

- based on student's activity and assesment of on-goind progress in realization of laboratory tasks

Summary assessment:

a) lectures:

- assessment of knowledge and skills through a written colloquium composed of open and closed questions, and tasks,

- discussion on colloquium results

b) laboratories:

- constant assesment during each class (oral answers), special bonuses for improvement of skills in applying learned rules and methods

- assesment of reports preaperd partially during classes and partially after classes, this assesment involves also assesing team work skills

Additional points could be obtained for additional activity during classes, in particular

- demonstration of additional interesting skills beyond course program,

- discussion about additional aspects of tasks,

- remarks and improvemnt suggestions about didactic materials,

- team work skills during laboratory tasks.

Programme content

Elements of optimization tasks. Classification of optimization methods. Sources of difficulty of optimization tasks. Examples of optimization problems with the focus on discrete/combinatorial problems. Exhaustive search. The idea of branch-and-bound method. Complexity of black box search with quantum computers. Random search. Greedy heuristics. Randomization of greedy heuristics.

Regret heuristics. The ida of neighborhood. Local search in greedy and steepest versions. Improvements of local search efficiency: using delta of objectve function, the use of moves evaluations from previous iterations, candidate moves, global memory of (components of) moves evaluations, advanced techniques. Multiple start local search. Variable neighborhood local search. Iterated local search. Adaptive local search. Large scale neigjborhood search. Simulated annealing and related algorithms. Tabu search. Long term memory.

Population and biologically inspired algorithms. Ant colony algoithms. Genetic and evolutionary algorithms. Crossover and recombination. The idea and role of schemata. Selection methods. Solutions encoding. Indirect encoding. Hybrid evolutionary algorithms. Hyper-heuristics and genetic hyper-heuristics. Approaches for handling constratints.

General scheme of evolutionary optimization methods. No free lunch theorem - assumptions, outline of the proof, scope. practical conclusions. Measures of optimization tasks difficulty. Objective function landscape analysis. Systematic design of optimization methods for particular problems. Examples of applications of this systematic approach. Experimental evaluation of evolutionary optimization methods. Recent trends in evolutionary optimization methods..

Teaching methods

1. Lectures: multimedia presentations, demonstration, discussion

2. Laboratory classes: oral introduction, programming, realization and analysis of results of computational experiments, discussion. During laboratory classes students work in pairs on a selected optimization problem elaborating more and more advanced evolutionary optimization methods, based on results from a previous classes.

Bibliography

Basic

1. Jarosław Arabas, Wykłady z algorytmów ewolucyjnych, WNT, 2006.

2. Zbigniew Michalewicz, Algorytmy genetyczne + struktury danych = programy ewolucyjne, Helion, 2003.

3. Z. Michalewicz, Jak to Rozwiązać, czyli Nowoczesna Heurystyka, WNT, 2006

Additional

1. Jaszkiewicz A., Distance preserving recombination operator for earth observation satellites operations scheduling, Journal of Mathematical Modelling and Algorithms, Volume 7, Issue 1, March 2008, Pages 25-42.

2. Lust, T., Jaszkiewicz, A., Speed-up techniques for solving large-scale biobjective TSP, 2010, Computers and Operations Research, 37(3), pp. 521-533.

3. Deep Infeasibility Exploration Method for Vehicle Routing Problems / Piotr Beling, Piotr Cybula, Andrzej Jaszkiewicz (WIiT), Przemysław Pełka, Marek Rogalski, Piotr Sielski // W: Evolutionary

Computation in Combinatorial Optimization : 22nd European Conference, EvoCOP 2022, Held as Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022 : Proceedings / red. Leslie Pérez Cáceres, Sébastien Verel - Cham, Switzerland : Springer, 2022 - s. 62-78.

4. Evolutionary Algorithm for Vehicle Routing with Diversity Oscillation Mechanism / Piotr Cybula, Andrzej Jaszkiewicz (WIiT), Przemysław Pełka, Marek Rogalski, Piotr Sielski // W: Parallel Problem Solving from Nature – PPSN XVII : 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I / red. Günter Rudolph - Cham, Switzerland : Springer, 2022 - s. 279-293.

5. Effective recombination operators for the family of vehicle routing problems / Piotr Cybula, Marek Rogalski, Piotr Sielski, Andrzej Jaszkiewicz (WIiT), Przemysław Pełka // W: GECCO '21: Proceedings of the Genetic and Evolutionary Computation Conference Companion / red. Francisko Chicano, Krzysztof Krawiec (WIiT) - New York, United States : Association for Computing Machinery (ACM), 2021 - s. 121-122.

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	44	2,0
Student's own work (literature studies, preparation for laboratory classes, preparation for the assessment test, project preparation - solving programming assignments, solving practical exercises) [1]	56	2,0

[1] delete or add other activities as appropriate