COURSE DESCRIPTION CARD - SYLLABUS

Course name
Biologically-inspired Algorithms and Models [S2SI1E>AMIB]

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
30

Laboratory classes
30

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

5,00

Coordinators

dr hab. inż. Maciej Komosiński prof. PP
maciej.komosinski@put.poznan.pl

Lecturers

Prerequisites

Students starting this course should have a basic knowledge of the concept of computational complexity and optimization problems posed as search problems. They should have the ability to model and solve simple optimization problems, possess programming skills and the ability to obtain information from the indicated sources. They should also understand the need to expand their competences. In addition, in terms of social competences, students should present attitudes such as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, and respect for other people.

Course objective

1. Students acquire knowledge about advanced optimization algorithms, including biologically inspired ones such as evolutionary algorithms. 2. Students acquire knowledge about common features and a unified approach to all optimization algorithms. 3. Students develop skills in efficient implementation and evaluation of the effectiveness of optimization algorithms – both in terms of time and quality. 4. Students learn to draw conclusions from self-conducted research, to create reports on computational experiments and to properly visualize the results.

Course-related learning outcomes

Knowledge
K2st_W1: has advanced and in-depth knowledge of widely understood artificial intelligence systems and optimization algorithms, 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 artificial intelligence with the emphasis on biologically-inspired optimization algorithms
K2st_W3: has advanced detailed knowledge regarding selected issues in artificial intelligence and related fields
K2st_W4: has knowledge about development trends and the most important cutting edge achievements in computer science, artificial intelligence and optimization algorithms
K2st_W6: knows advanced methods, techniques and tools used to solve complex engineering tasks and conduct research in the field of artificial intelligence and related fields

Skills
K2st_U1: is able to obtain information from literature, databases and other sources (both in Polish and English), integrate them, interpret and critically evaluate them, draw conclusions and formulate and fully justify opinions
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 engineering problems and research problems
K2st_U4: can use analytical, simulation and experimental methods to formulate and solve engineering problems and 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_U8: can carry out a critical analysis of existing technical solutions and propose their improvements
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_K2: understands the importance of using the latest knowledge in the field of computer science and artificial intelligence in solving research and practical problems in modeling and optimization
K2st_K4: is aware of the need to develop professional achievements and comply with the rules of professional ethics
Methods for verifying learning outcomes and assessment criteria Learning outcomes presented above are verified as follows: Lecture: assessment exam is conducted after the last lecture. Students answer a number of questions regarding material presented during lectures and laboratory classes. Reaching above 50% of total points is sufficient to get a "3" grade, and this is scaled linearly to "5" for 100% of points. Laboratory classes: During the semester, students complete individual reports related to the main topics discussed during the classes. The reports include theoretical and practical verification of knowledge, modeling, and programming skills. The final grade is calculated based on the average of the points gained from these reports. Reaching above 50% of total points is sufficient to get a "3" grade, and this is scaled linearly to "5" for 100% of points.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture: assessment exam is conducted after the last lecture. Students answer a number of questions regarding material presented during lectures and laboratory classes. Reaching above 50% of total points is sufficient to get a "3" grade, and this is scaled linearly to "5" for 100% of points.

Laboratory classes: During the semester, students complete individual reports related to the main topics discussed during the classes. The reports include theoretical and practical verification of knowledge, modeling, and programming skills. The final grade is calculated based on the average of the points gained from these reports. Reaching above 50% of total points is sufficient to get a "3" grade, and this is scaled linearly to "5" for 100% of points.

Programme content

The classes cover advanced topics in optimization, in particular concerning evolutionary algorithms, including selection, crossover, mutation, and fitness scaling techniques. Variants like genetic algorithms, evolutionary strategies, differential evolution, and genetic programming are explored. Co-evolutionary architectures, both cooperative and competitive, are studied along with their problems and ways to mitigate them. Evolutionary design principles, genotype-phenotype mapping and modularity are introduced. Optimization challenges, fitness landscape analysis, epistasis detection, and search space transformations are also discussed.

Course topics

Lecture:
Local search, tabu search and simulated annealing algorithms. Genetic and evolutionary algorithms. Selection and fitness scaling. Crossover, mutation, schema theorem, epistasis, MDP, NFL. Hierarchical GA and decomposition, evolutionary strategies, differential evolution, real-number representation. Global convexity. Genetic programming. Evolutionary design. Surrogate functions. Biologically-inspired mechanisms. Diversity control: fitness-based, phenotype-based, MAP-Elites. Lamarckian and Baldwinian approach. Directed and undirected, closed- and open-ended evolution. Coevolution: cooperative and competitive.

Laboratory classes:
Implementation of local search, simulated annealing and tabu search algorithms. Comparison of the quality of solutions they achieve on combinatorial problems, time spent, and efficiency. Analysis of the similarity of discovered local optima and best solutions. Evolutionary design:
- Mutation intensity and its impact on the optimization process,
- Modifying the fitness landscape, making it smoother, creating a gradient,
- Modifying the way the topology of solutions is searched. Evolution of designs and their control,
- Discovering the fitness landscape: measures of ruggedness and convexity,
- Detecting, estimating and visualizing epistasis,
- Transformation between search spaces: from the GP space to the evolutionary design space,
- Case study: optimize a shape or behavior.

Teaching methods

Lecture: slide shows and script-based presentations, whiteboard sketches with discussions, occasional demonstrations of programs.
Laboratory classes: whiteboard-sketches-based presentation, interaction with biologically-inspired models, solving illustrative examples on the board and coding problem solutions, conducting computational experiments, discussions, teamwork.

Bibliography

Basic
Okwu, M. O., & Tartibu, L. K. (2020). Metaheuristic Optimization: Nature-Inspired Algorithms, Swarm and Computational Intelligence, Theory and Applications. Springer International Publishing.
Vasuki, A. (2020). Nature-Inspired Optimization Algorithms. CRC Press.

Additional
Blum, C., Roli, A., & Sampels, M. (2008). Hybrid Metaheuristics: An Emerging Approach to Optimization. Springer Berlin Heidelberg.
Cotta, C., & van Hemert, J. (2008). Recent Advances in Evolutionary Computation for Combinatorial Optimization. Springer.
Glover, F. W., & Kochenberger, G. A. (2006). Handbook of Metaheuristics. Springer US.

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	65	2,50