COURSE DESCRIPTION CARD - SYLLABUS

Course name
Artificial life

Course

Field of study
Computing

Area of study (specialization)
Intelligent information technologies

Level of study
Second-cycle studies

Form of study
full-time

Year/Semester
1/2

Profile of study
general academic

Course offered in
Polish

Requirements
compulsory

Number of hours

Lecture
30

Tutorials

Laboratory classes
30

Projects/seminars

Other (e.g. online)

Number of credit points

5

Lecturers

Responsible for the course/lecturer:
dr hab. inż. Maciej Komosiński
email: maciej.komosinski@put.poznan.pl
tel: 61 6652931
wydział: Instytut Informatyki
adres: ul. Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:
mgr inż. Konrad Miazga
email: konrad.miazga@put.poznan.pl
tel: 61 6653058
wydział: Instytut Informatyki
adres: ul. Piotrowo 2, 60-965 Poznań

Prerequisites

Students starting this course should have a basic knowledge of optimization and computational complexity, machine learning algorithms and artificial neural networks. They should be able to model and solve simple optimization problems, should have programming skills and the ability to obtain information from provided sources. They should also understand the need to expand their competences. Moreover, in terms of social competences, they should exhibit honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, and respect for other people.

Course objective

1.Provide knowledge about advanced optimization algorithms, primarily biologically inspired algorithms such as ant colony optimization, particle swarm, artificial bee colony algorithms.

2. Provide knowledge about common features and a uniform approach to all optimization algorithms.

3. Provide knowledge on game theory with particular emphasis on modeling the population of agents.

4. Provide knowledge in the field of artificial life and the simulation of biological models, including impulse neural networks.

5. Improve skills to draw conclusions from self-conducted research and to write reports on computational experiments as well as be able to visualize results properly.

Course-related learning outcomes

Knowledge

has in-depth knowledge about possible, natural inspirations for information systems and for the simulation of physical and biological processes [K2st_W1]

has a structured and theoretically founded general knowledge related to key issues in biologically-inspired algorithms [K2st_W2]

has advanced detailed knowledge regarding simulations and modeling of biological phenomena [K2st_W3]

has knowledge about development trends and the most important cutting edge achievements in computer science and other selected and related scientific disciplines [K2st_W4]

has advanced and detailed knowledge of the processes occurring in the life cycle of hardware or software information systems [K2st_W5]

knows advanced methods, techniques and tools used to solve complex engineering tasks and conduct research in biologically-inspired algoritms [K2st_W6]

Skills

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_U1]

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_U3]

can use analytical, simulation and experimental methods to formulate and solve engineering problems and simple research problems [K2st_U4] can - when formulating and solving engineering tasks - integrate knowledge from different areas of computer science (and if necessary also knowledge from other scientific disciplines) and apply a systemic approach, also taking into account non-technical aspects [K2st_U5]

is able to assess the suitability and the possibility of using new achievements (methods and tools) and new IT products [K2st_U6]

can carry out a critical analysis of existing technical solutions and propose their improvements [K2st_U8]

can assess the usefulness of methods and tools for solving an engineering task consisting in building a biological model or a biologically inspired algorithm [K2st_U9]

can, in accordance with a given specification, taking into account non-technical aspects, model and simulate a biological or sociological process using appropriate methods, techniques and tools, including adapting existing or developing new tools for this purpose [K2st_U11]

can prepare and present a scientific study in Polish and English, presenting the results of scientific research, or prepare an oral presentation on detailed issues in the field of computer science [K2st_U13]

can determine the directions of further learning and implement the process of self-education, including other people [K2st_U16]

Social competences

understands that in the field of IT the knowledge and skills quickly become obsolete [K2st_K1]

understands the importance of using the latest knowledge in the field of modeling and simulation for solving research and practical problems [K2st_K2]

is aware of the need to develop professional achievements and comply with the rules of professional ethics [K2st_K4]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:
a) lectures:
- based on answers to questions about the material discussed in previous lectures,
b) laboratories:
- based on an assessment of the current progress in the implementation of tasks.
Summative assessment:
a) lectures: verification of the learning outcomes is carried out by:
- assessment of the knowledge and skills demonstrated in a test consisting of several questions or short tasks. Exceeding 50% of the points allows to obtain a satisfactory grade.
- discussion of the test results,
b) laboratories: verification of the learning outcomes is carried out by:
- assessment of skills related to the implementation of laboratory exercises,

- continuous assessment during each class (oral answers) - rewarding the increase in the ability to use the learned principles and methods,
- evaluation of reports prepared partly during the classes and partly after their completion, with the possibility of using the Moodle platform,
- presenting the results of individual experiments.

Obtaining additional points for activity during classes, especially for:
- carrying out extended, non-obligatory experiments as part of laboratory tasks and describing them in the report,
- remarks to improve teaching materials.

Programme content

Lectures:

Reminder of the basic optimization algorithms: local, simulated annealing, tabu search, evolutionary. Schema theorem and epistasis. Messy genetic algorithm. Hierarchical genetic algorithm. Empirical and theoretical evaluation of genetic algorithms. Mechanisms inspired by nature. Evolutionary strategies. Differential evolution. Evolutionary programming: floating point representation, embryogenesis, crossing over and global convexity. Genetic programming and symbolic regression. Classifier systems: input and output, main loop, learning, classifier evaluation, Bucket Brigade algorithm, rule discovery. Knowledge seeding and handling constraints. Parallel evolutionary algorithms. Co-evolutionary architectures: cooperative and competitive. Problems and pathologies in coevolution and ways to minimize them. Discussion of other optimization techniques: ant colony optimization and stigmergy (AA/ACO), particle swarm algorithms (PSO), artificial immune systems (AIS), artificial bee colony algorithms (ABC), gravitational search algorithms (GSA), charged system search (CSS) and other biologically inspired algorithms. Different environments and computational paradigms (molecular, quantum, membrane) and their application in optimization. Artificial life; definitions of life and research areas. Evolutionary theories. Emergence. Spontaneous and directed evolution. Closed-ended and open-ended evolution. Cellular automata. L-systems. Examples of classical experiments in artificial life. Evolutionary design and evolutionary robotics; genotype-phenotype mapping, morphogenesis and modularity. Physics simulation engines. Agent and the environment; complex adaptive systems (CAS) and multi-agent systems (MAS). Levels of autonomy of agents. Formal descriptions of an evolving system. Elements of game theory and evolutionary games. Social behavior and dilemmas. Examples of biological life models and conclusions drawn from their simulations.

Laboratory classes:

Definitions of life and the field of artificial life; open problems. Cellular automata: elementary cellular automata, Langton's ant, Conway's Life Game. Cellular automata in art. Using Golly and DDLab. Using Repast. Multi-agent systems, modeling, creating a model of a selected problem. Swarm intelligence. Ant algorithms, particle swarm optimization, practical applications. NK model of adaptation landscapes. Creating your own implementation of a selected swarm intelligence algorithm. Evolutionary design and evolutionary robotics compared to classical combinatorial optimization problems. An agent and the environment: perception, stimuli, information representation, information processing, control. Using

Framsticks and Framscript programming. Practical experiments in the Framsticks environment. The role of genetic representation in evolutionary optimization. Effectiveness of optimization in evolutionary design. Genotype-phenotype mapping, global convexity. Spiking neurons: biological inspiration, principles, comparison with standard artificial neurons. Hebb's rule. Coevolution, Lotka-Volterra model, undirected evolution.

Teaching methods

Lecture: multimedia presentation illustrated with examples.
Laboratory exercises: presentation illustrated with examples; carrying out the tasks given by the teacher (practical exercises).

Bibliography

Basic

1. D.E. Goldberg, Algorytmy genetyczne i ich zastosowania, WNT, Warszawa, 2003.
2. Z. Michalewicz, Algorytmy genetyczne + struktury danych = programy ewolucyjne, WNT, Warszawa, 2003.
3. M. Komosinski, A. Adamatzky, Artificial Life Models in Software. Springer, wydanie drugie, 2009.
4. A. Adamatzky, M. Komosinski, Artificial Life Models in Hardware. Springer, 2009.

Additional

1. R. Dawkins, Ślepy zegarmistrz / Wspinaczka na szczyt nieprawdopodobieństwa / Samolubny gen.
2. George B. Dyson, Darwin wśród maszyn: rzecz o ewolucji inteligencji. Prószyński i S-ka, 2005.
3. Bernd-Olaf Kűppers, Geneza informacji biologicznej. Filozoficzne problemy powstania życia, PWN 1991.
4. C. G. Langton, Artificial Life: An Overview. MIT Press, 1995.

Breakdown of average student's workload

	Hours	ECTS
Total workload	132	
Classes requiring direct contact with the teacher	72	
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	60	

[1] delete or add other activities as appropriate