COURSE DESCRIPTION

Computational intelligence models in solving optimization problems

Academic year 2026-2027

1. Programme-related data

1.1. Higher Education Institution	Babeș-Bolyai University of Cluj-Napoca
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Doctoral Shool	Doctoral School in Mathematics and Computer Science
1.4. Field of study	Computer Science
1.5. Level of study	Doctoral

2. Course-related data

2.1. Course title	**Computational intelligence models in solving optimization problems**			Course code	**MDE8176**
2.2. Course coordinator	Dr. habil. Gaskó Noémi				
2.3. Seminar coordinator	Dr. habil. Gaskó Noémi				
2.4. Year of study	1	2.5. Semester	1	2.6. Type of assessment	Exam
2.7. Course status	Optional			2.8. Course type	Core subject

3. Total estimated time (hours per semester of teaching activities)

3.1. Number of hours per week	**3**	of which: 3.2. course	**2**	3.3. seminar/ laboratory/ project	**1**
3.4. Total of hours in the curriculum	36	of which: 3.5. course	24	3.6. seminar/ laboratory	**12**
Time allocation for individual study (IS) and self-taught activities (ST)					**hours**
Learning from textbooks, course materials, bibliography, and notes (IS)					46
Additional research in the library, on subject-specific electronic platforms, and on-site					54
Preparing seminars/ laboratories/ projects, assignments, reports, portfolios, and essays					55
Tutoring (professional guidance)					30
Examinations					29
Other activities					
3.7. Total hours of individual study (IS) and self-taught activities (ST)				**214**	
3.8. Total hours per semester				**250**	
3.9. Number of credits				**10**	

4. Prerequisites (where applicable)

4.1. curriculum-related	Artificial intelligence
4.2 skills-related	Programming skills

5. Specific conditions (where applicable)

5.1. course-related	
5.2. seminar/laboratory-related	

6. Subject-specific learning outcomes

Knowledge
1. The student knows modelling methods, being able to analyse real-life problems and to translate them in concrete requirements and to design a corresponding software model
2. The student has knowledge related to specifying the requirements of research activities in the domain of computational optimization
Skills
1. The student can use specific language and terminology for the field of computational optimization, being able to communicate and interact with members of a team.
2. The student can apply computational optimization techniques to solve real-world problems.
Responsibility and autonomy
1. The student can work independently to obtain the knowledge necessary for designing, managing, and evaluating research activities in the field of computational optimization.
2. The student can model and design applications in the field of computational optimization.

7. Contents

7.1. Course	Teaching and learning methods	Remarks[1]
Week 1: Introduction. Taxonomy of optimization	• Interactive exposure • Explanation • Conversation	
Week 2: Mathematical methods	• Interactive exposure • Explanation • Conversation	
Week 3-4: Single state methods: hill climbing, local search methods, Simualted Annealing, Tabu Search	• Interactive exposure • Explanation • Conversation	
Week 5-7: Evolutionary algorithms – GA, ES, EP, GP	• Interactive exposure • Explanation • Conversation	
Week 8: Swarm Intelligence: Ant colony, Bee colony, Particle Swarm optimization techniques	• Interactive exposure • Explanation • Conversation	
Week 9: Multiobjective Optimization: multiobjective optimization problem, non-dominance, weighted sum methods, evolutionary multiobjective optimization.	• Interactive exposure • Explanation • Conversation	
Week 10: Hyperheurstics, co-evolution, hybrdization	• Interactive exposure • Explanation • Conversation	
Week 11: Neuroevolution	• Interactive exposure • Explanation • Conversation	
Week 12: Optimization in uncertainity	• Interactive exposure • Explanation • Conversation	
Week 13-14: Real-word applications	• Interactive exposure • Explanation • Conversation	
Bibliography 1. Mykel Kochenderfer, Tim Wheeler, Kyle Wray, Algorithms for decision making, MIT Press, 2022		

[1] For example, organisational aspects, recommendations for students, specific aspects relating to the course/seminar, such as inviting experts in the field, etc.

2. Kochenderfer, Mykel J., and Tim A. Wheeler. Algorithms for optimization. Mit Press, 2019.
3. Yang, Xin-She. Nature-inspired optimization algorithms. Academic Press, 2020.
4. Simon, Dan. Evolutionary optimization algorithms. John Wiley & Sons, 2013.
5. Eiben A & Smith JE, Introduction to Evolutionary Computing. Springer-Verlag 2010.
6. Chiong, Raymond, and Thomas Weise. Variants of evolutionary algorithms for real-world applications. Edited by Zbigniew Michalewicz. Vol. 2. Berlin: Springer, 2012.
7. David E. Goldberg, The Design of Innovation: Lessons from the competent genetic algorithms. Springer-Verlag; 2002.
8. Sean Luke, Essentials of Metaheuristics. Freely available for download at http://cs.gmu.edu/~sean/book/metaheuristics/
9. Dumitrescu, D., B Lazzerini, Evolutionary Computation, CRC Press, New York, Boca Raton, 2000

7.2. Seminar/ laboratory	**Teaching and learning methods**	**Remarks**
Each student chooses a problem (it can be connected to the Ph.D topic), and implements three different optimization algorithms for the selected problem Hybridization Comparisons Discussion of the results	• Interactive exposure • Explanation • Conversation	-6 laboratories
Project presentation, documentation	• Interactive exposure • Conversation	-1 laboratories

Bibliography
1. Mykel Kochenderfer, Tim Wheeler, Kyle Wray, Algorithms for decision making, MIT Press, 2022
2. Kochenderfer, Mykel J., and Tim A. Wheeler. Algorithms for optimization. Mit Press, 2019.
3. Yang, Xin-She. Nature-inspired optimization algorithms. Academic Press, 2020.
4. Simon, Dan. Evolutionary optimization algorithms. John Wiley & Sons, 2013.
5. Eiben A & Smith JE, Introduction to Evolutionary Computing. Springer-Verlag 2010.
6. Chiong, Raymond, and Thomas Weise. Variants of evolutionary algorithms for real-world applications. Edited by Zbigniew Michalewicz. Vol. 2. Berlin: Springer, 2012.
7. David E. Goldberg, The Design of Innovation: Lessons from the competent genetic algorithms. Springer-Verlag; 2002.
8. Sean Luke, Essentials of Metaheuristics. Freely available for download at http://cs.gmu.edu/~sean/book/metaheuristics/
9. Dumitrescu, D., B Lazzerini, Evolutionary Computation, CRC Press, New York, Boca Raton, 2000

8. Evaluation

Type of activity	8.1 Evaluation criteria[2]	8.2 Evaluation methods[3]	8.3 Percentage in the final grade
8.4. Course	- presentation about the Ph.D topic, application possibilities of the course	presentation	40.00%
8.5. Seminar/ laboratory	-be able to implement course concepts and algorithms -be able to make a practical project during the semester	Laboratory work Practical project	60.00%
8.6 Minimum standard for passing			
At least grade 5 (from a scale of 1 to 10) at both presentation and laboratory work.			

[2] The evaluation criteria must directly reflect the learning outcomes targeted at the level of the degree programme respectively at the level of the subject. More specifically, the learning outcomes set out in the expected learning outcomes are assessed.

[3] Both final evaluation methods and ongoing evaluation strategies should be established.

9. SDG labels (Sustainable Development Goals)[4]

Not applicable.

Date of entry: 13.02.2026	Signature of course coordinator Conf.dr. Gaskó Noémi	Signature of seminar coordinator Conf.dr. Gaskó Noémi
Date of approval in the department: ...		Signature of the head of department .....

[4] Select a single label which, according to the *Implementation of SDG labels in the academic process*, best matches the subject. If the subject addresses sustainable development in a generic manner (i.e. by presenting/introducing the general framework of sustainable development, etc.), then the Sustainable Development generic label may be applied. If none of the labels describe the subject, select the last option: *"No label applies."*