SYLLABUS

1. Information on the study programme

1.1. Higher education institution	West University of Timisoara
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Study program field	Computer Science
1.5. Study cycle	Master
1.6. Study programme / Qualification	Artificial Intelligence and Distributed Computing

2. Information on the course

2.1. Course title			Metaheuristic Algorithms				
2.2. Lecture instructor			Daniela Zaharie				
2.3. Seminar / laboratory instructor			Daniela Zaharie				
2.4. Study year	2	2.5. Semester	1	2.6. Examination type	E	2.7. Course type	DI

3. Estimated study time (number of hours per semester)

3.1. Attendance hours per week	3	out of which: 3.2 lecture	2	3.3. seminar / laboratory	1
3.4. Attendance hours per semester	42	out of which: 3.5 lecture	28	3.6. seminar / laboratory	14

Distribution of the allocated amount of time*	**hours**
Study of literature, course handbook and personal notes	15
Supplementary documentation at library or using electronic repositories	16
Preparing for laboratories, homework, reports etc.	40
Exams	6
Tutoring	6
Other activities…	0

3.7. Total number of hours of individual study	83
3.8. Total number of hours per semester	125
3.9. Number of credits (ECTS)	5

4. Prerequisites (if it is the case)

4.1. curriculum	Artificial Intelligence, Numerical Calculus, Programming, Probability and Statistics, Operations Research and Optimization
4.2. competences	Knowledge of numerical algorithms, statistics, artificial intelligence, optimization and programming abilities

5. Requirements (if it is the case)

5.1. for the lecture	Lecture room with whiteboard and projector/ Online:

	https://classroom.google.com/c/MjI2NjMyMjMwMzA2?cjc=u75tn72
5.2. for the seminar / laboratory	Lab room with computers having Python and R software installed/ Online: Online: https://classroom.google.com/c/MjI2NjMyMjMwMzA2?cjc=u75tn72

6. Specific acquired competences

Professional competences	• Ability to identify the metaheuristic technique appropriate for a specific problem • Ability to implement and validate a computational model based on metaheuristic algorithms • Ability to solve a real-world problem using computational intelligence tools.
Transversal competences	• Ability to conduct research activity and to prepare reports on a given topic • Team work ability

7. Course objectives

7.1. General objective	Providing knowledge on Computational Intelligence methods, particularly metaheuristics designed for solving different types of optimization problems.
7.2. Specific objectives	*Knowledge objectives (OC)*: (1) to present basic principles of metaheuristic techniques; (2) to describe local search algorithms; (3) to describe global search algorithms; (4) to present examples of metaheuristics for global, multi-modal, multi-criteria and dynamic optimization. *Abilitation objectives (OAb)*: (1) to identify the techniques appropriate to a given problem ; (2) to use software tools which are specific for metaheuristics; (3) to implement efficient metaheuristic algorithms; *Attitude objectives (OAt)*: (1) to argue the utility of metaheuristic algorithma in solving real-world problems.

8. Content

8.1. Lecture	Teaching methods	Remarks, details
L1. Introduction. Classes of difficult problems (planning, assignment, selection, adaptation, prediction) and corresponding search spaces. Classes of metaheuristics. The overall structure of a metaheuristic algorithm. *(OC1, OAb1)*	Discourse, conversation, illustration by examples	2 hours ([1]- ch. 0, [4] – ch. 1)
*L2-3. Trajectory-based metaheuristics. L*ocal search (Pattern Search, Nelder Mead etc). Global search (restarted local search, Iterated Local Search, Simulated Annealing, Tabu Search, Variable Neighborhood Search etc).(OC2,OC3, OAb1)	Discourse, conversation, illustration by examples	4 hours ([1]-ch.2; [2]-ch.2, [4]-ch.2-3)
L4-6. Population-based metaheuristics. Overall structure. Main components (exploration and	Discourse, conversation,	6 hours ([1]-ch. 3,4; [2] –ch. 3; [3] – ch.

exploitation operators). Encoding types. Operators for evolutionary algorithms: mutation, crossover, selection. Genetic algorithms, evolution strategies, evolutionary programming, genetic programming. (OC3, OAb1)	illustration by examples	8-12, [4]-ch.8-9)
L7. Swarm Intelligence. Particle Swarm Optimization. Ant Colony Optimization. Artificial Bee Colony. (OC3, OAb1)	Discourse, conversation, illustration by examples	2 hours ([1] – ch. 8, [2]-ch. 6, [3] – ch. 16-17, [4] -ch. 5-6)
L8. Difference-based and Probabilistic Algorithms. Differential Evolution, Population Based Incremental Learning, Estimation of Distribution Algorithms, Bayesian Optimization Algorithms. (OC3, OAb1)	Discourse, conversation, illustration by examples	2 hours ([1]-ch. 9; [2]-ch. 5, [3]-ch.13)
L9-10. Multi-objective/multi-modal/dynamic optimization. Particularities of multi-objective optimization (non-domination, Pareto front etc) and specific methods (non-dominated sorting, decomposition-based models). Quality metrics. Multi-modal optimization and specific approaches (niching, sharing etc). Techniques for dynamic optimization (hyper-mutation, random immigrants, ageing mechanisms). (OC4, OAb3)	Discourse, conversation, illustration by examples	4 hours ([1] – ch. 7, [5])
L11. Scalability of Metaheuristic Algorithms. Cooperative coevolution. Parallel models for population-based metaheuristics (master-slave, island, cellular). (OC3, OAb3)	Discourse, conversation, illustration by examples	2 hours ([1] – ch. 5,6, [3] – ch. 15)
L12. Analysis of the performance of metaheuristic algorithms. Design of experiments. Benchmark problems. Statistical tests.	Discourse, conversation, illustration by examples	2 hours ([6])
L13-14. Applications of metaheuristic algorithms for: neural networks design, data mining, scheduling. (OC4, OAb3, Oat1)	Discourse, conversation, illustration by examples	4 hours ([2]-ch. 8; [3] –ch. 2,3)

Recommended literature

1. Sean Luke: *Essentials of Metaheuristics,* Lulu, second edition, 2013, available for free at http://cs.gmu.edu/~sean/book/metaheuristics/
2. Jason Brownlee: *Clever Algorithms. Nature-inspired Programming Recipes*, 2011, available at http://www.CleverAlgorithms.com
3. A. Engelbrecht: *Computational Intelligence. An Introduction*, Wiley, 2007
4. B. Choppard, M. Tomassini, An Introduction to Metaheuristics for Optimization, Springer, 2018
5. Coello C.A., van Veldhuizen D.A., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer, 2002
6. J.Derrac, S.García, D.Molina, F.Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence

algorithms,Swarm and Evolutionary Computation,Volume 1, Issue 1,Pages 3-18,ISSN 2210-6502, 2011.

7. Classroom materials (https://classroom.google.com/c/MjI2NjMyMjMwMzA2?cjc=u75tn72)

8.2. Seminar / laboratory	Teaching methods	Remarks, details
L1. Examples of optimization problems. Getting started with implementation tools (Python, R) (Oab1)	Problem-based approach, dialogue, learning through collaboration	2 hours
L2. Combinatorial optimization. Trajectory and population-based metaheuristics (Simulated Annealing, Tabu Search) (OAb1, OAb2)	Problem-based approach, dialogue, learning through collaboration	2 hours
L3. Implementation of evolutionary algorithms: genetic algorithms for combinatorial optimization and evolution strategies for continuous optimization. (OAb1, OAb2, OAb2)	Problem-based approach, dialogue, learning through collaboration	2 hours
L4. Symbolic regression using genetic programming. Combinatorial optimization using Ant Colony Optimization. Continuous optimization using Particle Swarm Optimization. (OAb1, OAb2)	Problem-based approach, dialogue, learning through collaboration	2 hours
L5. Differential evolution, distribution estimation algorithms and cooperative coevolution. (OAb1, OAb2)	Problem-based approach, dialogue, learning through collaboration	2 hours
L6. Multi-objective and multi-modal optimization. (OAb1, OAb2, OAt)	Problem-based approach, dialogue, learning through collaboration	2 hours
L7. Evolutionary design of neural networks and applications of metaheuristics in data mining (OAb1, OAb2, OAt)	Problem-based approach, dialogue, learning through collaboration	2 hours

Recommended literature:

1. Rick Muller, *A crash course in Python for scientists*, https://nbviewer.jupyter.org/gist/rpmuller/5920182
2. Luis Marti, Advanced Evolutionary Computation: Theory and Practice, 2014, http://lmarti.com/aec-2014
3. DEAP: Distributed Evolutionary Algorithms in Python, https://pypi.org/project/deap/ , https://github.com/deap/deap
4. Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau and Christian Gagné, “DEAP: Evolutionary Algorithms Made Easy”, Journal of Machine Learning

Research, vol. 13, pp. 2171-2175, jul 2012
5. Classroom materials – Jupyter notebooks (https://classroom.google.com/c/MjI2NjMyMjMwMzA2?cjc=u75tn72)

9. Correlations between the content of the course and the requirements of the professional field and relevant employers.

The content is in accordance with the structure of similar courses offered by other universities and it covers the main aspects of using computational intelligence tools, particularly metaheuristic algorithms, in solving real world problems.

10. Evaluation

Activity	10.1. Assessment criteria	10.2. Assessment methods	10.3. Weight in the final mark
10.4. Lecture	Knowledge of the main principles of the metaheuristic algorithms	Written exam (open book)	20%
	The identification of the metaheuristic algorithm for a given problem	Project presentation (report, software application, oral presentation)	60%
10.5. Seminar / laboratory	Usage of software tools and implementation of metaheuristic algorithms	Applications at lab and homework	20%
10.6. Minimum needed performance for passing			

- Knowledge of the main concepts used in the design of metaheuristic algorithms for solving optimization problems.
- Ability to implement a simple metaheuristic algorithm.
- Ability to identify the optimization technique/ metaheuristic algorithm which is appropriate for solving a real-world problems.

The final mark is computed as weighted average of the marks corresponding to the components specified at 10.4 and10.5. The exam is considered passed if the average is at least 5 (it is not required that each mark is at least 5). In each session of exams (including re-examinations) the mark is computed using the same rule. The student can be re-examined only for the components for which the current mark is smaller than 5, excepting the cases when the student asks to be re-examined .

Online activities: all course/lab materials will be available on Google Classroom (https://classroom.google.com/c/MjI2NjMyMjMwMzA2?cjc=u75tn72) and the online activities will be organized using Google Meet.

Date of completion	Signature (lecture instructor)	Signature (seminar instructor)
13.09.2021	prof.dr. Daniela Zaharie	prof.dr. Daniela Zaharie

Date of approval	Signature (director of the department)