COURSE DESCRIPTION

1. Program details

University	"Lucian Blaga" University of Sibiu
Faculty	Engineering Faculty
Department	Department of Computer Science and Electrical and Electronics Engineering
Main field of study	Computer Science and Information Technology
Level of education	Master
Specialization	Advanced Computing Systems

2. Course details

Course title	Evolutionary computing			
Course code	Type of course	Year of study	Semester	Number of credits
	Optional	2	1	10
Evaluation type	Type of course (FD=fundamental discipline.; DD=domain discipline; SD=specialized discipline; CD=complementary discipline)			
Exam *The final assessment will include written work consisting of (partial) grid tests and problems.*	FD			
Course instructor	Conf. dr. ing. Adrian FLOREA			
Seminar/lab/project instructor	Şef lucr.dr.ing. Adalbert GOLOMETY			

3. Estimated time

Course duration in the curriculum – number of hours per week				
Lecture	Seminar	Lab	Project	Total
3	-	2	-	**5**
Course duration in the curriculum - Total of hours curriculum				
Lecture	Seminar	Lab	Project	Total ($NOAD_{sem}$)
42	-	28	-	**70**

Distribution of hours for individual study	No. hours
Individual study using course handbooks, bibliography and notes	100
Additional documentation in library and on specialized electronic platforms	30
Preparing seminars / labs, homework, essays and portfolios	20
Tutoring	15
Exam preparation	15
Total hours for individual study ($NOSI_{sem}$)	180
Total hours per semester ($NOAD_{sem} + NOSI_{sem}$)	**250**

4. Prerequisites (if applicable)

Curriculum	Basic (undergraduate) courses in informatics (programming, algorithms) and mathematics (statistics).
competencies	

5. Conditions (if applicable)

course materials	Lectures, conversations, Projects, exercises, individual study, homework assignments, Smart Board, videoproiector.
sem/lab/project materials	Laboratory with computers.

6. Specific competences acquired

Professional competence	• Equipping you with the knowledge and skills necessary to apply evolutionary computing techniques to the solution of complex optimisation and machine learning problems. • Identifying how to represent the solution, selection, recombination and mutation operators for a certain evolutionary algorithm. Attainment capacity for integrate obtained knowledge from others classes. • Providing the master student with the basics for studying evolutionary computing (EC): metaphor linking natural evolution to problem solving, the history of EC, the motivation and applicability for EC. • Describing of advanced models and techniques in evolutionary computing and their relevance for addressing complex problems; • Developing skills for addressing complex real-world problems; • Supply the background for advanced studies and research in the field; • Understanding the ideas that underlie scientific research. • Develop individual research skills and acquiring the ability to write and publish scientific papers.
Transversal competences	• Professional approach and ethical conduct in scientific research. • Developing team-work ability in order to perform some relatively complex applications. • Collaborating with experts in other fields.

7. Objectives (based on the specific grid for the accumulated competences)

General objective	• Identify the main sources of information. • Critical analysis of theoretical models, ideas and approaches established. • Introduce the main concepts, techniques and applications in the field of evolutionary computation. • Give students some experience on when evolutionary techniques are useful, how to use them in practice and how to implement them with different programming languages.
Specific objectives	Evolutionary Computing represents a relatively new research field belonging to Artificially Intelligence. It deals with a range of problem-solving techniques based on principles of biological evolution, such as natural selection and genetic inheritance. The aims of this course are to make students understand the principles, the basic paradigms, techniques, and

	algorithms of evolutionary computing. Evolutionary models are applied for solving some typical NP-hard problems The course presents algorithms that involve techniques implementing mechanisms such as reproduction, mutation, recombination, natural selection and survival of the fittest. In this course, we will study some basic principles of genetic algorithms, evolutionary programming, evolution strategy, genetic programming and learning classifier systems and swarm intelligence (ant colony and wasp-like models for optimization and particle swarm optimization) as well as new paradigms as membrane computing. In addition, an enhanced attention will be on multi-objective optimization methods and searching techniques.

8. Contents

Course		No. hours
Course 1	Introduction to evolutionary computing (EC). Applications of EC. Positioning of EC and the basic EC metaphor. Historical perspective. Biological inspiration. Motivations for EC.	3
Course 2	Basic schema and components / operators of an evolutionary algorithm (EA): Representation / Evaluation / Population / Parent Selection / Recombination / Mutation / Survivor Selection / Termination. Application of EC in optimization.	4
Course 3	Informed[1] search strategies (heuristics): *Global* Search Algorithms (Best first search – Greedy, A*) and *Local* Search Algorithms (Hill Climbing, Simulated Annealing, Tabu Search).	3
Course 4	Genetic algorithms. Representations, mutations, crossovers, selection mechanisms. Examples. Memetic algorithms.	6
Course 5	Holland's Schema Theorem. Schema Fitnesses, Schema Disruption, Schema Survival. Landscape Metrics. Stochastic analysis techniques. Markov Chain Analysis. Examples.	3
Course 6	Evolutionary strategies. Theoretical Aspects. Algorithm. Representation and operators. Characteristics. Examples (1+1), (μ+1), (μ+λ), (μ,λ).	3
Course 7	Midterm Review from taught lessons.	2
Course 8	Evolutionary programming. Characteristics. Applications. Examples.	2
Course 9	Genetic programming. Characteristics. Applications. Examples.	2
Course 10	Pareto optimality. Non-Pareto Classification Techniques (Weighting Method, Lexicographic Order Method, Vector Evaluated Genetic Algorithm etc). Multi-objective optimization methods: Multi-objective Genetic Algorithm (MOGA), Non-dominated Sorting Genetic Algorithm (NSGA), Strength Pareto Evolutionary Algorithm (SPEA), Pareto Archived Evolution Strategy (PAES).	6
Course 11	Methodological aspects in working with EA. Quality Indicators useful in multi-objective optimization techniques. Experiment design. Algorithm design. Testing and validation problems. Using *jMetal* library (http://jmetal.sourceforge.net/index.html) make comparisons between Evolutionary Algorithms previously discussed.	2

[1] Se bazează pe informaţii (disponibile) specifice problemei încercând să restrângă căutarea prin alegerea inteligentă a nodurilor care vor fi explorate.

Course 12	Hybrid Evolutionary Algorithms: Evolutionary Algorithms assisted by Neural Networks, Evolutionary Algorithms assisted by Ant Colony Optimization, Evolutionary Algorithms assisted by Particle Swarm Optimization, Fuzzy Logic assisted Evolutionary Algorithms, Evolutionary Algorithms assisted by Bacterial Foraging.	2
Course 13	Parameters control. General aspects. Adaptive and self-adaptive parameter control. Robustness. Multimodal Problems and Spatial Distribution. Statistical mechanics models. Continuous Space models. Vose' Dynamical Systems Model. Membrane computing	2
Course 14	Final Review from taught lessons.	2
	Total course hours:	42

Laboratory		No. hours
Lab 1	Lindemayer Grammars-models for biological evolving systems	2
Lab 2	Overview of **Genetic Algorithms application** in diverse area of fields: Aerospace engineering, Astronomy and astrophysics, Electrical engineering, Financial markets, Game playing, Mathematics and algorithmic, Molecular biology, Pattern recognition and Data mining, Robotics, Routing and Scheduling, Systems engineering. Presenting the practical assignment and organizing the teams for implementation.	4
Lab 3	**Representation methods** for data encoding in genetic algorithms.	4
Lab 4	**Selection** methods in genetic algorithms implementation.	2
Lab 5	Methods of change for genetic algorithms: **Crossover** and **Mutation**	2
Lab 6	Apply evolutionary techniques to optimize the target imposed. Implementation of algorithms and methods in code.	10
Lab 7	Each team will deliver a technical report (TR), code and PowerPoint presentation (PPT) in which will present the strategy used. Based on the TR it will be developed a scientific paper.	4
	Total lab hours:	28

Teaching methods

• Lectures, conversations, Projects, exercises, individual study, homework assignments.	Language of instruction	English

References

Recommended reading	http://webspace.ulbsibiu.ro/adrian.florea/html/Planificari/Planif_EvolutionaryComputing_ACS_2.pdf
	1. A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.
	2. Adam Marczyk, *Genetic Algorithms and Evolutionary Computation*, 2004 http://www.talkorigins.org/faqs/genalg/genalg.html .
	3. John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (Complex Adaptive Systems), The MIT Press, 1992.
	4. Carlos A. Coello Coello, David A. Van Veldhuizen, Gary B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd Edition, Springer, 2007.
	5. Crina Grosan, Ajith Abraham and Hisao Ishibuchi, *Hybrid Evolutionary Systems*, Studies in Computational Intelligence, Springer Verlag, Germany, 2006.

	6. Oltean, M., *A-Brain: The multiple problems solver*, Symposium on Applied Computing, Vol. 2, pp. 955-959, The Association for Computing Machinery, 23-27 April 2006.
	7. Jose A., Lozano, Pedro, Larranaga, Inaki, Inza: Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms (Studies in Fuzziness and Soft Computing), Springer Verlag GmbH; 2007.
	8. S.N.Sivanandam, S.N.Deepa, *Introduction to Genetic Algorithms*, Springer Verlag GmbH; 2007.
More references	9. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," Evolutionary Computation, IEEE Transactions on, vol. 6, no. 2, pp. 182-197, 2002.
	10. E. Zitzler, M. Laumanns, L. Thiele, and others, "SPEA2: Improving the strength Pareto evolutionary algorithm," in Eurogen, pp. 95–100, 2001.
	11. M. R. Sierra and C. A. C. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and dominance," IN EMO'2005, PAGES 505–519. LNCS 3410, pp. 505--519, 2005.
	12. M. Reyes-Sierra and C. A. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," International Journal of Computational Intelligence Research, vol. 2, no. 3, pp. 287–308, 2006.

9. Linking course content with expectations of the epistemic community representatives, professional associations and employers' representatives in the field related to the program

10. Evaluation

Type	Evaluation criteria	Evaluation methods	Percentage in final grade	Obs.*
Course	Quiz during semester	Written exam	20%	RPE
	Final exam	Written exam	55%	RFE
	Other activities: class participation, attendance, engagement	-	5%	nRPE
Lab	Practical activities	Oral evaluation, testing developed applications	10%	nRPE
	Homeworks / Technical Reports	Weekly checked	10%	nRPE
Minimum standard of performance				
The final assessment will include written work consisting of (partial) grid tests and problems. • Knowledge, understanding and explaining the basics of evolutionary computing. • Constant interest to acquire discipline. • Partial fulfilment (50%) of homework, essays and tests given during the semester.				

(*) REP – required for exam participation; nREP – not required for exam participation; RFE – required for final evaluation.

Date of completion: …10/09/2014..
Date of approval in the Department:……

	Position, title, first name, surname	Signature

Course instructor	Conf. dr. ing. Adrian FLOREA	
Head of department	Prof. dr. ing. Daniel VOLOVICI	

Adresa: Str. Emil Cioran, nr. 4
Sibiu, 550025, România
e-mail: inginerie@ulbsibiu.ro
web: inginerie.ulbsibiu.ro

Tel: +40 (269) 217 928
Fax: +40 (269) 212 716