



Master: ADVANCED COMPUTING SYSTEMS - ACS

Course: EVOLUTIONARY COMPUTING - EC

Time/Room

- Course: IE 102, Wednesday 17:00-20:00 PM
- Lab: IM 219, Wednesday 20:00-22:00 PM

http://webspaces.ulbsibiu.ro/adrian.florea/html/Planificari/EvolutionaryComputing/Planificari/EvolutionaryComputing_ACS_2.pdf

Motto: "It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change." - Charles Darwin

▪ **Course 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 swarm intelligence (ant colony and wasp-like models for optimization and particle swarm optimization), the convergence analysis of genetic algorithms and Holland's Schema Theorem. In addition, an enhanced attention will be on multi-objective optimization methods and searching techniques.

- ✓ Evolutionary computing is a new research direction of great importance both from theoretical point of view and especially from the point of view of applications.
- ✓ Evolutionary computing began by lifting **ideas from biological evolutionary theory into computer science**, and continues to look toward new biological research findings for inspiration.
- ✓ Darwin's principles "**Survival of the fittest!**" and „**Natural selection and genetic inheritance!**" can be used as a starting point in introducing evolutionary computation.
- ✓ Although the history of evolutionary computation dates back to the 1950s and 1960s, only within the last decade have evolutionary algorithms become practicable for solving real-world problems on desktop computers.

- **Course structure / description:** (detailed in the analytical program of the courses Evolutionary Computing [MB ACS 2 09o Evolutionary Computing Florea Adrian](#))

Lectures

1. Meet the genetics! The 100 largest discoveries in genetics (video).
 - How evolution really works (Part I)?! ([video](#))
 - Introduction to evolutionary computing (EC). Applications / Necessity of EC (searching / optimizing techniques in a huge design space). Positioning of EC and the basic EC metaphor. Historical perspective. Biological inspiration. Motivations for EC ([pdf](#)).

[Introduction.ppt](#) → (A.E. Eiben¹, J.E. Smith // Amsterdam)

[Intro2EC.ppt](#) → (G. Vernon Dozier // North Carolina)

- Assigning [project themes](#).
- 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 (Queens / [Knapsack problems](#)), ([pdf](#)).
[What is an EA.ppt](#)
[EA_other.pptx](#)
- 3. *Local Search Algorithms* (Hill Climbing ([video](#)), [Simulated Annealing](#), Tabu Search).
[LocalSearch.ppt](#)
[Hill_Climbing.ppt, hillclimb02.pdf](#)
[02_search_informed.pdf](#)
- 4. Genetic algorithms. Representations, mutations, crossovers, selection mechanisms. Examples ([video](#), [pdf](#)).
[Genetic_Algorithms.ppt](#) → (A.E. Eiben, J.E. Smith)
Problem discussion (source code): *Maximizing a function's value* (e.g. $f(x)=x^2$)
Problems proposal: *Finding the minimum value of a function; Traveling Salesman Problem*
[GAs.ppt](#) → (G. Vernon Dozier)
[IA04_Optimizare.pdf](#)
- 5. 5.1. Holland's Schema Theorem. Examples.
[Theory_schemata_Eiben_modify.ppt](#)
5.2. Evolutionary strategies. Characteristics. Examples (1+1), ($\mu+1$), ($\mu+\lambda$). Differences ES and GAs.
[ES-Intro.ppt](#)
[Evolution_strategies.ppt](#)
[ES.ppt](#)
- 6. Pareto optimality. 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).
Non-pareto techniques and Bio-inspired methods.
[Multi-Objective_OptimizationMOEA.ppt](#)
[tutorial-panama-2007.pdf](#)
- 7. 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>, <http://jmetalnet.sourceforge.net/>) make comparisons between Evolutionary Algorithms previously discussed.

¹ A.E. Eiben is the Head of the Computational Intelligence Group, Department of Computer Science, Faculty of Sciences Vrije Universiteit Amsterdam - <http://www.cs.vu.nl/~gusz/>

Presentation Papers

1. Coello, Carlos A. Coello Coello, "[A short tutorial on evolutionary multiobjective optimization](#)", Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, 2001.
2. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "[A fast and elitist multiobjective genetic algorithm: NSGA-II](#)", IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.
3. Zitzler, Eckart, Kalyanmoy Deb, and Lothar Thiele, "[Comparison of multiobjective evolutionary algorithms: Empirical results](#)." Evolutionary computation 8.2 (2000): 173-195.
4. 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.
5. Zitzler, Eckart, and Lothar Thiele. "[Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach](#)" Evolutionary Computation, IEEE Transactions on 3.4 (1999): 257-271.
6. Díaz, J., Hidalgo, J. I., Fernández, F., Garnica, O., & López, S. (2009, July), [Improving SMT performance: an application of genetic algorithms to configure resizable caches](#). In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers (pp. 2029-2034). ACM.
7. Hindi, Musa M., and Roman V. Yampolskiy. "[Genetic algorithm applied to the graph coloring problem](#)." *Proc. 23rd Midwest Artificial Intelligence and Cognitive Science Conf.* 2012.
8. Tinggui Chen; Guangla Zhou, "[Vehicle Routing Optimization Problem with Time-windows and its Solution by Genetic Algorithm](#)", Journal of Digital Information Management; Apr2013, Vol. 11 Issue 2, p136.
9. Yu, H., and Nan Yu. "[Application of Genetic Algorithms to Vehicle Suspension Design](#)" Mechanical Engineering Department The Pennsylvania State University (2003).
10. Gellert, A., Calborean, H., Vintan, L., Florea, A., [Multi-Objective Optimizations for a Superscalar Architecture with Selective Value Prediction](#), IET Computers & Digital Techniques, Vol. 6, No. 4 (July), pp. 205-213, ISSN: 1751-8601, Stevenage, UK.
11. Florea, A., Gellert, A., [Different approaches for solving optimization problems using interactive e-learning tools](#), The 10th eLearning and Software for Education Conference - eLSE 2014 - organized by the Romanian Advanced Distributed Learning Association, pp.74-75(1-9), ISSN: 2066 – 026X; DOI 10.12753/2066-026X-14-081.
12. Roman L., Florea, A., Cofaru I.I., [Mathematical model and software simulation of suspension's system from OPEL cars](#), ANNALS OF THE ORADEA UNIVERSITY. Fascicle of Management and Technological Engineering, Vol. 23, No. 3 (December 2014), pp. 1-6, ISSN 1583 – 0691.

▪ Assignments

Students will be assigned to study a different theme about Evolutionary Algorithm (a small research exposed in front of your colleagues). They must prepare a paper presentation, a project on implementing and extending research described in one previously published paper. If anybody has a paper of particular interest for the project, please discuss it with the professor.

For every course session, starting from week 8th, student presentations of the final project will take place. The presentations will last for about 20-30 minutes. Presentations must be sent to adrian.florea@ulbsibiu.ro before sustaining the work according to the attached [schedule](#).

▪ **Labs topics:**

- Lindemayer Grammars-models for biological evolving systems
- Overview of Genetic Algorithms application in diverse area of fields: Aerospace engineering, Astronomy and astrophysics, Electrical engineering, 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.
- Representation methods for data encoding in genetic algorithms.
- Selection methods in genetic algorithms implementation.
- Methods of change for genetic algorithms: Crossover and Mutation
- Apply evolutionary techniques to optimize the target imposed. Implementation of algorithms and methods in code.

▪ **Evaluation (Grading policy):**

The class presentation of project assignment represents N1 mark which is 30% of final mark. It is based on accuracy of implementation, experimental results, and the quality of presentation.

Laboratory evaluation consists in 20% of the final mark.

Final exam represents 50% of final mark. It includes theoretical treating of some basic concepts of Evolutionary Computing and solving problems.

▪ **References**

1. A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003 (1st edition, 2003, ISBN: 3-540-40184-9).
2. S.N.Sivanandam, S.N.Deepa, *Introduction to Genetic Algorithms*, Springer Verlag GmbH; 2007.
3. Carlos A. Coello Coello, David A. Van Veldhuizen, Gary B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd Edition, Springer, 2007.
4. Mitsuo Gen, Runwei Cheng, *Genetic Algorithms and Engineering Optimization*, Wiley, 2000.