A0M33EOA: Competencies

Petr Pošík, Jiří Kubalík

October 30, 2016

Abstract

List of competencies students shall gain after completing course Evolutionary Optimization Algorithms taught at Czech Technical University in Prague, Dept. of Cybernetics, primarilly for (but not limited to) students of Open Informatics study programme.

Prerequisities. Optimization: problems and methods.

Before entering this course, a student shall be able to

- define an optimization task in mathematical terms; explain the notions of search space, objective function, constraints, etc.; and provide examples of optimization tasks;
- describe various subclasses of optimization tasks and their characteristics;
- define exact methods, heuristics, and their differences;
- explain differences between constructive and generative algorithms and give examples of both.

1 Black-box optimization. Local search, and evolutionary algorithms.

After this lecture, a student shall be able to

- describe and explain what makes real-world search and optimization problems hard;
- describe black-box optimization and the limitations it imposes on optimization algorithms;
- define a neighborhood and explain its importance to local search methods;
- describe a hill-climbing algorithm in the form of pseudocode; and implement it in a chosen programming language;
- explain the difference between best-improving and first-improving strategy; and describe differences in the behaviour of the resulting algorithm;
- enumerate and explain the methods for increasing the chances to find the global optimum;
- explain the main difference between single-state and population-based methods; and name the benefits of using a population;
- describe a simple EA and its main components; and implement it in a chosen programming language.

2 Standard EAs. Schema theorem. Representations.

After this lecture, a student shall be able to

- know and actively use the terminology inspired by biology;
- explain the importance, role and types of representation, selection, and genetic operators;
- distinguish between premature convergence and stagnation of EA, and suggest methods to fight them;
- explain the trade-off between exploration and expoitation;
- implement 1- and 2- point crossover, uniform crossover, bit-flip mutation;
- describe options for encoding the solutions of traveling salesperson problem, and the relevant genetic operators;
- give examples of real-world problems, EAs have been applied to;

- describe what a schema theory is and what it is used for;
- explain the schema theorem, and the influence of its individual parts related to selection, crossover and mutation;
- state the so-called Building Block Hypothesis and comment on its relation to the chosen representation.

3 EAs for real-parameter optimisation. Evolution strategies, CMA-ES.

After this lecture, a student shall be able to

- perform the mapping of chromosomes from binary to real space when using binary encoding for real-parameter optimization;
- describe and exemplify the effects of such a genotype-phenotype mapping on the neighborhood structures induced by mutation and crossover;
- give examples and describe some mutation and crossover operators designed for spaces of real number vectors;
- explain the main features of ES and differences to GAs;
- explain the notation $(\mu/\rho \overset{+}{,} \lambda)$-ES;
- describe the differences between mutation with isotropic, axis-parallel, and general Gaussian distribution, including the relation to the form of the covariance matrix, and the number of parameters that must be set/adapted for each of them;
- explain and use two simple methods of mutation step size adaptation (1/5 rule and self-adaptation);
- write a high-level pseudocode of CMA-ES and describe CMA-ES in the $(\mu/\rho \overset{+}{,} \lambda)$ notation;
- implement DE algorithm;
- explain the basic forms of DE mutation and crossover.

4 Genetic programming. Fundamentals and applications.

5 Epistasis. Estimation-of-Distribution Algorithms.

After this lecture, a student shall be able to

- explain what an epistasis is and show an example of functions with and without epistatic relations;
- demonstrate how epistatic relationships can destroy the efficiency of the search performed by an optimization algorithm, and explain it using schemata;
- describe an Estimation-of-Distribution algorithm and explain its differences from ordinary EA;
- describe in detail and implement a simple UMDA algorithm for binary representations;
- understand, fit to data, and use simple Bayesian networks;
- explain the commonalities and differences among EDAs not able to work with any interactions (PBIL, cGA, UMDA);
- explain the commonalities and differences among EDAs able to work with only pairwise interactions (MIMIC, COMIT, BMDA);
- explain the commonalities and differences among EDAs able to work with multivariate interactions (ECGA, BOA);
- explain the model learning procedures used in ECGA and BOA;
- understand what effect the use of a more complex model has on the efficiency of the algorithm when used on problems with increasingly hard interactions.