A very brief introduction to genetic algorithms

Radoslav Harman

Design of experiments seminar

FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

COMENIUS UNIVERSITY IN BRATISLAVA

25.2.2013

Optimization problems:

1. Set of permissible solutions (called *chromosomes* in GAs).
2. Utility function (*fitness function*).

The utility function can be an explicit mathematical function, or it can be evaluated by computation (simulation, experiment). The optimization problems solved by GAs are typically non-convex, with many local optima and a medium to high level of interaction of components of permissible solutions (*epistasis*).

Population-based optimization algorithms.

The actual state of the algorithm is represented by a *set* (*population*) of solutions. In each iteration a new set of solutions (new *generation*) is created based on the previous set of solutions (old generation). The specifics of the transition from the old generation to the new one define particular population-based method.

General scheme of GAs:

1) Choose an *initial population* of chromosomes.

2) Stop if the actual population satisfies some *stopping rule*, otherwise continue.

3) Choose pairs of parent chromosomes to produce offspring chromosomes in a process called *recombination*, or *mating* (usually the more fit chromosomes are chosen with higher probability). Most or all offspring undergo a random change called *mutation*.

4) Select a subset of the population in a process imitating *survival of the fittest* and *natural selection*, such that only the best (i.e., most fit) chromosomes survive. Continue by step 2).

More about individual phases of GAs:

1. **Initial population.** The size N_pop of the population is usually selected to be tens or hundreds of chromosomes. The size of the population is often kept constant for all generations.

2. **Survival of the fittest.** Usually the GA algorithms use *proportional selection* that selects the best (prop x N_pop) of the chromosomes. The rest of the chromosomes is discarded. Some algorithms use the so-called *threshold selection.* The principle of *aging* of chromosomes sometimes improves performance of GAs.

3. **Selection for mating.** To add new chromosomes to the population, we need to choose a couple of chromosomes, called *parents*. Methods of selection of parents include: a) uniform random selection, b) rank-weighted selection or cost-weighted selection (both known under the name *roulette wheel selection*), c) *tournament selection*. All methods, except for the

first one, select the chromosomes with high fitness more frequently than the chromosomes with low fitness.

4. **Recombination.** Two parents usually produce only one or two *offspring* (but each chromosome can be repeatedly selected for mating). The total number of offspring generated is often chosen in a way that keeps the size of population constant from generation to generation. Recombination strongly depends on the set of permissible solutions (on their coding and constraints).

5. **Mutation.** Some GAs decrease the magnitude of mutations for successive generations, and some do not mutate the best chromosomes at all (*elite* approach). Generally, it is advisable to use relatively small magnitudes of mutation for relatively large populations and vice-versa. Optimal mutation also strongly depends on the set of permissible solutions.

6. **Stopping rule.** The GA algorithm is stopped after a fixed number of generations, or if no improvement takes place during a given number of generations, or if the population is „homogeneous“.

Variants of GAs are characterized by different sets of all permissible solutions (possible chromosomes), which require different operations of recombination and mutation. Not only coding of chromosomes, but also (linear or nonlinear) constraints play a crucial role.

Classical binary GAs. They represent chromosomes by strings of bits of fixed length, i.e., the set of all permissible solutions is the set of vertices of the n-dimensional unit cube.

Recombination. The standard recombination in binary GAs is the *single crossover* recombination, whereby two parents produce two offspring. Sometimes *double crossover*, or *uniform crossover* is used.

Mutation. In binary GAs, the standard mutation is a *point mutation,* i.e., each bit of the chromosome is reversed with probability p_mut called the *mutation rate*.

Continuous GAs. The basic set of permissible solutions is a „continuous" subset of R^n, usually a convex set with non-empty interior, particularly the n-dimensional unit cube. The problem can be converted to a binary GA (straightforward binary coding of variables, or *Gray coding*). One can also use the so-called real-valued GAs, with chromosomes coded as vectors with real-valued components, which often performs better. Real-valued GAs can also be *hybridized* with a standard local-search optimization procedure.

Recombination: In real-valued GAs one can use *blending* (convex combination) of some or all components of the parents, or even extrapolation of some or all components of the parents.

Mutation: Mutation can alter only selected components or all components of chromosomes using a continuous random variates generator.

Permutation GAs. The basic set of permissible solutions is the set of all permutations of 1,...,n, which can represent feasible solutions of a travelling salesman problem (and other interesting problems of combinatorial optimization).

Recombination. There are specific nontrivial recombination methods for this problem, e.g., *order crossover* or *cycle crossover.*

Mutation can be a simple exchange of two coordinates of the permutation. There are also other techniques analogous to the methods of generation of candidate solutions in simulated annealing.

GAs are becoming popular in design of experiments (Park et al. 2006 and other papers).

Today I will present my **GA for computing optimal exact designs** of experiments with the standard size constraints and without replications. Permissible solutions are the binary sequences with exactly N bits (N is the required size) representing design points selected for the experiment.

Recombination: „Uniform size-constrained recombination": The (single) offspring has the same chromosomes at points where the components of the parental chromosomes match. In the non-matching part an appropriate number of ones and zeros is randomly generated.

Mutation: Poisson random number of exchanges of design points (i.e., ones with zeroes).

Demonstration. D-optimal block design with b=15 size-two blocks and t=10 treatments.