Genetic Algorithms / Evolutionary Computing

Idea: Apply the concepts and techniques of biological evolution (e.g. selection, crossover, and mutation) to computer science problems.

Persons: The original ideas go back to research that was performed by J.H. Holland in the 70s; the popularity of genetic algorithms increased recently — many researchers (most notably Goldberg, De Jong, and Grefenstette) are currently working in this field.

Application Areas: optimization problems, heuristic search, computerized learning.

Short Characterization: In essence, genetic algorithms are based on the natural selection principles which guide the generation of new chromosomes in living entities via genetic recombination.

Literature: D.E. Goldberg: "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley, 1989.

Genetic Operators

Crossover: rearrange components of the parents' chromosomes in the offspring — recombination of the genes of the parents in the offspring.

Mutation: random change in one or several components of a chromosome, any change of the chromosome stucture that is not caused by recombination.

Characteristics of Genetic Algorithms

- work on multiple solutions in parallel (a complete population), and not on a single solution.
- new solutions are created by exchanging parts of solutions (crossover, genetic recombination), and not by applying new solution generating operators to single solutions — it always takes at least two solutions to create a new solution.
- do not rely on gradients and do not require continuous (differentiable) fitness functions.
- are **randomized** algorithms — they depend on probabilistic events such as rolling the dice; many things in a genetic algorithm approach are only predictable in a probabilistic sense; such as chosen crossover points, when a mutation occurs, who produces offsprings with whom,...
- they are based on the "survival of the fittest": they use selection mechanisms which guarantee that fitter solution reproduce with a higher probability.
- can be applied to symbolic and integer-valued fitness function.
- can only applied to problems for which a fitness function exists.

Components of Genetic Algorithms

- a chromosomal representation of problem solutions.
- a way to create an initial population of solutions.
- an evaluation function, rating solutions in terms of their "fitness".
- genetic operators that alter the composition of children during reproduction.
- a reproduction plan — e.g. which members of a population will participate to which extend in the breeding of the next generation.
- a termination condition
- values of parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators).

GeAl in a Nutshell

`G(t)` := the population at time t

`compose` := operator that generates a new chromosome (solution) from the chromosomes of its parents.

`mutate` := mutates one or several bits / strings in a chromosome.

`should-have-mutation`: returns true when a mutation should occur (depending on the mutation probability distribution).

```
DO {
 Select org1 from G(t);
 Select org2 from G(t);
 org:= compose(org1,org2);
 If should-have-mutation() then org:= mutate(org);
 Insert org into G(t+1) }
UNTIL population of G(t+1) is complete;
```

How to Select Parents of Future Children

Roulette Wheel Selection: Solutions receive portions on a roulette wheel that is proportional to their fitness. Then the roulette wheel is spinned (several times) to create a mating pool with respect to the next generation.

Seeded Selection: Subdivide the solutions into (four in the example code) groups depending on their quality — the first group contains the best solutions, the second, ... Use the following mating schema:

```
DO {
 Pair a member of the Group1 with a member of Group1;
 Pair a member of the Group1 with a member of Group2;
 Pair a member of the Group1 with a member of Group3;
 Pair a member of the Group2 with a member of Group2;
 Insert the four new offsprings into G(t+1)}
UNTIL population of G(t+1) is complete;
```

Remark: if the sizes of all groups are the same, then the off-spring distribution of the four groups would be 4/3/1/0. However, if the size of the first group is half of the size of the other three groups the distribution would be: 6/3/1/0.

Random Selection: Pair member of G(t) that were randomly selected.

Rule Learning using Genetic Alg.

Two Popular Approaches:

- *The Michigan Approach* — one population represents a single rule set; rules are rewarded if they contribute to a successful solution, and rules are penalized if they don't, using the popular **bucket brigade algorithm**, which is described in ch. 6 of Goldberg's book. Seems to be especially useful for real-time systems.
- *The Pittsburgh Approach* — each member of the population is a rule set; that is, a population consists of sets of rule-sets. That is, the approach tries to learn complete rule-sets and not only single rules.

another name for the reseach field: classifier systems

Diversity

Remark: **Diversity** is a very important, but it is also a controversal, and not too well understood issue in genetic algorithm research.

Diversity in the Population

- ⊕ Prevents premature convergence.
- ⊕ Allows to find multiple good solutions and not only a single good solution. For example, in many applications you are interested to find multiple good solutions (e.g. in design problem such as designing a house or a dress, or in systems that generate explanations).
- ⊖ Coping with diversity introduces a significant overhead into genetic algorithms; e.g. similarity measures have to be introduced.
- ⊖ Might significantly slow down termination.

Diversity of Mates

- ⊕ prevents the creation of offsprings that are very similar to their parents, which rarely results in a signficant fitness increase.
- ⊕ allows [sometimes] to overcome local minima.
- ⊕ tends to lead to diversity in the population
- ⊖ however, breeding similar individuals still seems to be desirable as long the (local) maximum has not been reached.
- ⊖ breeding highly fit individuals with large degrees of diversity rarely results in offsprings that are fitter than their parents.

Remark: In many biological systems we have diversity in the population, but not too much diversity in mating. A population consists of multiple species; however, members of different species, e.g. dogs and cats, are not allow to produce offsprings; offsprigs are created involving quite similar individuals (e.g. dogs are only allowed to produce offspings with other dogs, but not with crocodiles).