Genetic Algorithm

Basic Principles

- Coding or Representation
  - String with all parameters
- Fitness function
  - Parent selection
- Reproduction
  - Crossover
  - Mutation
- Convergence
  - When to stop

Coding - Representation

- Parameters of the solution (genes) are concatenated to form a string (chromosome)
- All kind of alphabets can be used for a chromosome (numbers, characters), but generally a binary alphabet is used
- Order of genes on chromosome can be important
- Generally many different codings for the parameters of a solution are possible
- Good coding is probably the most important factor for the performance of a GA
- In many cases many possible chromosomes do not code for feasible solutions

Genetic Algorithm

Selection:

Chance to be selected as parent proportional to fitness

- Roulette wheel

To avoid problems with fitness function

- Tournament

Tournament

- Binary tournament
  - Two individuals are randomly chosen; the fitter of the two is selected as a parent
- Probabilistic binary tournament
  - Two individuals are randomly chosen; with a chance p, $0.5 < p < 1$, the fitter of the two is selected as a parent
- Larger tournaments
  - n individuals are randomly chosen; the fittest one is selected as a parent

- By changing n and/or p, the GA can be adjusted dynamically

Genetic Algorithm

Reproduction

- Crossover
  - Two parents produce two offspring
  - There is a chance that the chromosomes of the two parents are copied unmodified as offspring
  - There is a chance that the chromosomes of the two parents are randomly recombined (crossover) to form offspring
  - Generally the chance of crossover is between 0.6 and 1.0
- Mutation
  - There is a chance that a gene of a child is changed randomly
  - Generally the chance of mutation is low (e.g. 0.001)

Genetic Algorithm

Mutation: Random mutation (binary)

Random point	0	0	1	0	0	0	0	0	0	0

Offspring	0	0	0	0	0	0	0	0	0	0
Offspring	0	0	1	0	0	0	0	0	0	0

Random mutation point: 3

Genetic Algorithm

Mutation: Random mutation (Permutation)

Random point	0	0	1	0	1	0	0	0	0	0

Offspring	1	2	5	6	4	3	8	7	10	9
Offspring	1	2	4	6	5	3	8	7	10	9

Random mutation points: 3 and 5

Genetic Algorithm

GA: Non-Uniform Mutation

Michalewichz's Non-Uniform Mutation is one of the widely used mutation operators in real coded GAs.

From the point $x^t = (x_1^t, x_2^t, \ldots . x_n^t)$ the muted point $x^{t+1} = (x_1^{t+1}, x_2^{t+1}, \ldots . x_n^{t+1})$ is created as follows:

$$x_i{}^{t+1} = \begin{cases} x_i^t + \Delta\left(t, x_i^u - x_i^t\right) & if \quad r \leq \frac{1}{2} \\ x_i^t - \Delta\left(t, x_i^t - x_i^l\right) & if \quad r > \frac{1}{2} \end{cases}$$

GA: Non-Uniform Mutation

where t is the current generation number and r is a uniformly distributed random number in interval [0,1]

x_i^l and x_i^u are lower an duper bounds of the ith component of the decision vector respectively.

The function $\Delta(t, y) = y\left(1 - u^{\left(1-\frac{t}{T}\right)}\right)^b$ where u is a uniformly distributed random number in the interval [0,1], T is the maximum number of generations and b is a parameter, determining the strength of the mutation operator.