Lab 4: 26th March 2012

Exercise 1: Evolutionary algorithms

1. Found a problem where EAs would certainly perform very poorly compared to alternative approaches. Explain why.

Solution
Suppose that we want to find the maximum of a differentiable function $f(x)$ that we know. In this case we can simply use an analytical method: calculate the values $x_1, \ldots, x_n$ for which $f'(x)$ is zero and then select the $x_i$ for which $f(x_i)$ is maximum. An evolutionary algorithm would most probably also find the solution, but it would take much longer.

2. How big is the phenotype space in the 8-queen problem?

Solution
The phenotype space contains all the possible configurations.

In the most general version of the 8-queen problem, a configuration is valid if each queen is put on a different square, with no other restrictions. Therefore, the size of the phenotype space is

$$size(PS) = \binom{8^2}{8}$$

that is, the number of combinations of 8 squares taken from the 64 squares in the chessboard. In general, if the number of queens is $n$ and the chess board has $n*n$ squares the size of the phenotype space is

$$size(PS) = \binom{n^2}{n}$$

In the version of the problem presented in the slides there are some additional constraints on the configurations:

- There is exactly one queen per row
- There is exactly one queen per column

So $size(PS)$ is much smaller, and in case of 8 queens is

$$size(PS) = 8!$$

because there are 8 possible positions of first queen on the first row, 8-1 possible positions for the second queen in the second row, and so on. In the general case ($n$ queens), the size of the phenotype space is

$$size(PS) = n!$$
Exercise 2: Genetic algorithms

1. Discuss whether there is the survival of the fittest in generational GAs.

Solution
In generational GAs the entire population is replaced by the offspring at each generation. Therefore, the fittest individual at generation $t$ survives at generation $t+1$ if

- It is selected for mating
- It is not “destroyed” by crossover and mutation

Assuming fitness proportionate selection, the number of copies in the mating pool of the fittest is as higher as the fittest is outstanding in the population (its fitness is much higher than all other finesses). At least one of these copies should not be “destroyed” by crossover and mutation. This depends on the probabilities of crossover and mutation.

2. Calculate the probability that a binary chromosome of length $L$ will not be changed by applying the usual bit-flip mutation with probability $p_m=1/L$.

Solution
The chromosome does not change if none of its bits is changed. The probability that no bits are changed is

$$p_{nc} = (1 - p_m)^L$$

3. Given the fitness function $f(x)=x^2$, calculate the probability of selecting the individuals $x=1$, $x=2$, and $x=3$, using roulette wheel selection. Calculate the probability of selecting the same individuals when the fitness function is $f_1(x)=f(x)+10$.

Solution
Roulette wheel selection is done by giving to each individual $x=i$ a probability of being selected $p(x=i)$

$$p(x = i) = \frac{f(x = i)}{\sum_{i=1}^{3} f(x = i)}$$

Therefore

$$p(x = 1) = \frac{f(x = 1)}{\sum_{i=1}^{3} f(x = i)} = \frac{1}{1 + 4 + 9} = \frac{1}{14}$$

$$p(x = 2) = \frac{f(x = 2)}{\sum_{i=1}^{3} f(x = i)} = \frac{4}{1 + 4 + 9} = \frac{4}{14}$$

$$p(x = 3) = \frac{f(x = 3)}{\sum_{i=1}^{3} f(x = i)} = \frac{9}{1 + 4 + 9} = \frac{9}{14}$$

If we use the transposed fitness function $f_1$, then
\[ p(x = i) = \frac{f_i(x = i)}{\sum_{i=1}^{3} f_i(x = i)} \]

Therefore

\[ p(x = 1) = \frac{f_1(x = 1)}{\sum_{i=1}^{3} f_i(x = i)} = \frac{11}{11 + 14 + 19} = \frac{11}{44} \]

\[ p(x = 2) = \frac{f_1(x = 2)}{\sum_{i=1}^{3} f_i(x = i)} = \frac{14}{11 + 14 + 19} = \frac{14}{44} \]

\[ p(x = 3) = \frac{f_1(x = 3)}{\sum_{i=1}^{3} f_i(x = i)} = \frac{19}{11 + 14 + 19} = \frac{19}{44} \]

The selection pressure is lower because the probability values are closer.

4. A generational GA has a population size of 100 individuals; uses fitness proportional selection without elitisms; and after \( t \) generation has a mean population fitness of 76.0. There is one copy of the current best member, which has fitness 157.0.

- What is the expectation for the number of copies of the best individual in the mating pool?
- What is the probability that there will be no copies of the best individual in the mating pool?

**Solution**

By simple calculation we can rewrite the formula for fitness proportional selection as

\[ p(x_i) = \frac{f(x_i)}{n\bar{f}} \]

where \( \bar{f} \) is the average fitness of the population. So the expected number of copies of the best individual is

\[ E(\text{best}) = p(\text{best}) * n = \frac{f(\text{best})}{n\bar{f}} * n = \frac{f(\text{best})}{\bar{f}} = \frac{157}{76} \approx 2 \]

The probability that there will be no copies of the best individual in the mating pool is the probability that all of the 100 individuals in the mating pool are different from the best.

\[ p(\text{not best}) = \left(1 - \frac{f(\text{best})}{n\bar{f}}\right)^n = \left(1 - \frac{157}{100 \cdot 76}\right)^{100} \approx 0.12 \]

5. Write a computer program to implement a generational GA for the One-Max problem:

\[ f(x) = \sum_{i=1}^{L} x_i \]

Use the following parameters:

- Representation: binary strings of length \( L=25 \);
- Initialization: random
- Parent selection: fitness proportionate, implemented using roulette wheel;
- Recombination: one-point crossover with probability $p_c = 0.7$;
- Mutation: bit-flip with probability $p_m = 1/L$;
- Population size = 100;
- Termination condition: 100 generation or optimum found.

After every generation find the best, worst, and mean fitness in the population and (possibly) plot them on a graph with time (generation) on the x axis.
Do 10 run and find the mean and standard deviation of the time (generations) taken to find the optimum.

**Solution**

Given the specification of the problem, it is rather simple to write a Java program.
Exercise 2: Genetic programming

1. Give a suitable function and terminal set that allows the following expression is syntactically correct

\[(x \land \text{true}) \rightarrow ((x \lor y) \lor (z \leftrightarrow (x \land y)))\]

Solution
F = \{\land, \rightarrow, \lor, \leftrightarrow\}, T = \{x, y, z, \text{true}\}

The s-expression corresponding to the formula is

\[(\rightarrow (\land x \text{true}) (\lor (x \land y) \leftrightarrow (z (\land x y))))\]

2. Implement a genetic programming that solves the symbolic regression problem presented in slide page 18, down of lecture 5.

Solution
Given the specification of the problem, it is rather simple to write a Java program.

3. Design a GA for the credit-scoring problem (slides pages 9-10 of lecture 5). Discuss advantages and disadvantages of this GA versus a GP.

Solution
A possible design of a GA for the credit-scoring problem is as follows:

- The individuals of the population bit strings of fixed length L.
  - The first \(L_1\) bits represent different number of children. E.g., the 1\(^{st}\) bit is 1 if the applicant does not have children, the 2\(^{nd}\) bit is 1 if the application has 1 child, the 3\(^{rd}\) bit is 1 if (s)he has 2 children, etc.
  - The following \(L_2\) bits represent ranges of salary. E.g., the \(L_1+1\)^{st} bit is 1 is the salary is below 5000, the \(L_1+2\)^{st} is 1 if the salary ranges from 5000 to 10000, etc.
  - The following \(L_3\) bits are used to represent the marital status. E.g., the \(L_3+1\)^{st} bit is 1 if the applicant is married, etc.

It has to be noted that in well-formed chromosomes one and only one bit in each of the three parts has value 1.

- The fitness of an individual is the percentage of well-classified cases.
- The selection of the mating-pool is fitness proportional
- The recombination operators are mutation and one-point crossover. These operators should avoid creating bad-formed individuals.

The main disadvantage of this GA versus GP is that the representation of the individuals is “forced” as we need to use fixed length bit strings.