Exercises on Evolutionary Computation

1. (Schemata) [0.5 point] Consider the two schemata $A1 = \#0\#101\#\#\#$, $A2 = \#\#010\#111$. Which of the two schemata has the highest chance to survive mutation, for a mutation rate $p_m = 0.01$? (Justify your answer).

2. (Building Block Hypothesis) [0.5 point] Describe a problem where the Building Block Hypothesis does not hold. Explain why.

3. (Selection Pressure) [1 point] Given the fitness function $f(x) = x^2$, calculate the probability of selecting the individuals $x = 2$, $x = 3$, and $x = 4$, using roulette wheel selection. Calculate the probability of selecting the same individuals when the fitness function is scaled as follows $f_1(x) = f(x) + 20$. Which fitness function yields a lower selection pressure? What can you conclude about the effect of fitness scaling on selection pressure?

4. (Role of selection in GA' s) [2 points] A simple $(1+1)$-GA for binary problems works as follows.

   (a) Randomly generate a bit sequence x.

   (b) Create a copy of x and invert each of its bits with probability p. Let x_m be the result.

   (c) If x_m is closer to the goal sequence than x then replace x with x_m.

   (d) Repeat the process from step (b) with the new x until the goal sequence is reached.

   The Counting Ones problem amounts to find a bit string whose sum of its entries is maximum. Implement a simple $(1 + 1)$-GA for solving the Counting Ones problem.

   (a) Use bit strings of length $l = 100$ and a mutation rate $p = 1/l$. For a run of 1500 iterations, plot the best fitness against the elapsed number of iterations.

   (b) Now do 10 runs. How many times the algorithm finds the optimum?

   (c) Now replace (c) in the above algorithm with (c'): replace x with x_m. Is there a difference in performance when using this modification? Justify your answer.

5. (Evolutionary strategies vs local search) [1 point] Consider a (1+5) ES. How does this differ from the (1+1) ES in how the search space is explored when optimizing a function? How does the (1+λ) ES strategy behave with respect to the value of λ when compared to greedy algorithms? (Recall that greedy algorithms perform a sequence of locally optimal steps in order to search for an optimal solution.)

6. (Memetic algorithms vs simple EAs) [2.5 point] Implement the simple EA for the TSP described in our first lecture (see slides).

(a) Implement a variant of this algorithm based on memetic algorithms. Compare the performance of the two algorithms in a fair way on the TSP problem instance given in the file 'file-tsp' and on one small instance at your choice from the 'Symmetric Traveling Salesman Problem' benchmark instances available at `http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html`. The file 'file-tsp' contains a 50×2 matrix with the coordinates (x_i, y_i) for city $i = 1, \ldots, 50$. Please provide URL link to your source code and results containing clear instructions on how to reproduce your results.

(b) On the TSP problem are memetic algorithms more effective than the simple EA' s? (To answer this question, use the results of your investigation as well as recent results from the literature).

7. (Genetic Programming representation) [0.5 point] Give a suitable function, terminal set and s-expression for the following logical and mathematical formulas:

   (a) $(y \wedge true) \rightarrow ((x \vee y) \vee (z \leftrightarrow (x \wedge y)))$,

   (b) $0.234 * z + x$–0.789.

8. (Genetic Programming behaviour) [2 points] Implement a GP program for finding a symbolic expression that fits the following data:

$(Input) Dependent Variable$	$Y(Output)$
−1.0	0.0000
−0.9	−0.1629
−0.8	−0.2624
−0.7	−0.3129
−0.6	−0.3264
−0.5	−0.3125
−0.4	−0.2784
−0.3	−0.2289
−0.2	−0.1664
−0.1	−0.0909
0	0.0
0.1	0.1111
0.2	0.2496
0.3	0.4251
0.4	0.6496
0.5	0.9375
0.6	1.3056
0.7	1.7731
0.8	2.3616
0.9	3.0951
1.0	4.0000

with the following parameter setting: population size: 1000,
function set: $\{+, -, *, log, exp, sin, cos, div\}$,
terminal set: x,
number of generations 50,
crossover probability 0.7,
mutation probability: 0,
fitness: - sum of absolute errors.

You can use an existing GP framework: see for instance list of implementation frameworks mentioned in the syllabus.

Plot the following:

(a) best of generation fitness (y-axis) versus generation (x-axis).

(b) best of generation size (y-axis) versus generation (x-axis).

Can you observe any undesirable phenomenon from these plots? In case of positive answer, how would you try to overcome the related problem (you can refer to the literature).

Please provide URL link to your source code and results containing clear instructions on how to reproduce your results.