## Exercises on Evolutionary Computation

## Important requirement. Please provide URL link to a (GitHub) repository containing code for Exercises 4, 6, and 8, and clear instructions on how to reproduce your results.

1. (Schemata) [0.5 point] Consider the two schemata $A1 = \#1\#100\#\#\#$, $A2 = \#\#010\#011$. 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 $f_1(x) = f(x) + 20$.

   (a) For each of the two fitness functions: 1) provide a table with fitness function value and selection probability for each individual; 2) plot the pie chart (roulette wheel) with selection probability of the three individuals.

   (b) Which fitness function yields a lower selection pressure?

   (c) 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) Perform 10 runs (again, 1500 iterations). In a single Figure, plot best fitness against elapsed number of iterations of each run. How many times the algorithm finds the optimum?

(c) Now replace (c) in the above algorithm with (c' ): replace $x$ with $x_m$. Perform 10 runs (again, 1500 iterations). In a single Figure, plot the best fitness against the elapsed number of iterations. 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.

   (a) How does this differ from the (1+1) ES in how the search space is explored when optimizing a function?

   (b) How does the (1+$\lambda$) ES strategy behave with respect to the value of $\lambda$ 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 (MAs). Use the 2-opt algorithm as local search technique in the memetic algorithm. The 2-opt algorithm tries to swap all pairs of cities to see if this improves the length of the tour (see, e.g. `https://en.wikipedia.org/wiki/2-opt`).

   (b) Consider the TSP problem instance given in the file 'file-tsp'. The file 'file-tsp' contains a $50 \times 2$ matrix with the coordinates $(x_i, y_i)$ for city $i = 1, \ldots, 50$. Also, select a 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`. Run the MA and EA 10 times on these two problem instances.

   (c) For each algorithm (EA and MA), and for each problem instance, provide a figure containing plots average and best fitness against the elapsed number of iterations (1500 iterations) for all 10 runs.

   (d) Compare the plots of the two algorithms. Which method (EA or MA) works best?

   (e) Do you consider the above comparison of the MA and EA algorithms (based on their results over the same number of iterations) fair? Justify your answer. If your answer is no, then explain what a fair comparison could be.

   (f) In general, on the TSP problem, are memetic algorithms more effective than simple EAs? (To answer this question, please rely on (recent) results from the literature, and include references to the papers used to answer this question).

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.

Provide the following two figures:

(a) a plot of the fitness (y-axis) of the best individual in each generation (x-axis);

(b) a plot with the size (y-axis), that is, number of nodes, of the individuals with the highest fitness in their generation (x-axis), i.e. the individuals corresponding to the fitnesses shown in (a)); a plot of average sizes (y-axis) for each generation (x-axis). Plot also the spread (for instance as shadowed area) between the minimum and maximum size at each generation.

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 - and briefly describe - a specific paper from the literature)