Evolutionary Algorithms Winter 2017/2018
PD Dr.-Ing. habil. Christian Borgelt from 2017.12.07

Exercise Sheet 4

Exercise 13 Roulette Wheel Selection

As mentioned in the lecture, it is one of the disadvantages of fitness proportional selection (roulette wheel selection) that it can lead to a substantial variance when selecting the individuals of the next generation. In this exercise we consider this disadvantage in more detail by computing the probabilities with which an individual is selected for the next generation.

a) Compute the probability that an individual with the relative fitness p is represented k times in the next generation if the individuals are chosen by roulette wheel selection!
(Hint: Note that this probability depends on the population size *popsize*.)

b) How does the probability in a) change if there are already m copies of the individual in the current population?

c) Determine the expected value and the variance of the number of descendants of an individual with the relative fitness p!
(Hint: It may not be the most convenient way to start with the formula determined in a), even though one may reach the result on this path as well.)

d) Compute numeric values for the quantities considered in a) to c) for a relative fitness $p = 0.02$, a population size $\mathit{popsize} = 100$, and $k = 0, 1, 2$ descendants!

Exercise 14 Scaling the Fitness Function

Let the following chromosomes for the 8-queens problem be given (encoding as discussed in the lecture, that is, a list of file positions, numbering starting with zero):

$$\begin{array}{lcl@{\qquad\qquad}lcl}
s_1 &=& (1,0,7,6,1,6,7,2), & s_6 &=& (2,3,4,7,6,6,2,2), \\
s_2 &=& (5,1,3,0,2,0,6,0), & s_7 &=& (6,1,7,4,3,0,1,2), \\
s_3 &=& (7,1,6,6,4,4,5,2), & s_8 &=& (5,6,5,3,0,0,7,5), \\
s_4 &=& (7,2,1,1,1,4,1,2), & s_9 &=& (6,6,2,0,0,0,4,1), \\
s_5 &=& (4,1,0,2,6,7,4,2), & s_{10} &=& (1,7,2,4,1,3,0,6)
\end{array}$$

Determine the relative fitness of these chromosomes using

a) linear dynamic scaling with $\alpha = 1.2$

b) σ-scaling with $\beta = 2$!

Start from the fitness function that was also considered in the lecture to evaluate chromosomes for the n-queens problem (negated number of collisions, that is, negated number of pairs of queens on the same file or diagonal).

Exercise 15 Stochastic Universal Sampling

In the lecture the method of *stochastic universal sampling* was discussed as a variant of the expected value model. This method uses a roulette wheel with *popsize* (population size) markers. How can this selection method be implemented efficiently? (Give a code fragment in a programming language of your choice or in pseudo-code.)

Exercise 16 Tournament Selection

As an alternative to roulette wheel selection we considered in the initial example of an evolutionary algorithm a solution of the n-queens problem that uses tournament selection.

a) Compute the probability that the *best* individual in the current population is represented k times in the next generation if the individuals of the next generation are determined by tournament selection! (The tournament participants are to be selected *without* replacement.)

b) In analogy, compute the probability for the *worst* individual!

c) How do the probabilities in a) and b) change if the tournament participants are chosen *with* replacement?

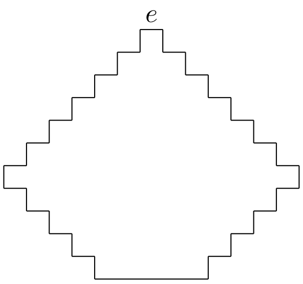
d) Compute the expected value and the variance of the number of descendants of the *best* individual!

e) Compute numeric values for the quantities in a), b) and d) for a population size $popsize = 100$, a tournament size $tmsize = 2$ (duel selection) and $k = 0, 1, 2$ descendants!

(Hint: Note in a) to d) that the quantities to determine depend on the population size *popsize* and the tournament size *tmsize*. You may make the simplifying assumption that all individuals have a different fitness.)

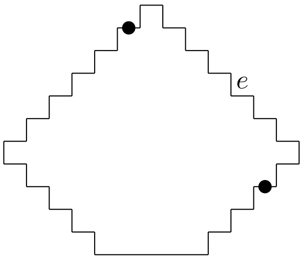
Additional Exercise Timelapse™: Double Pyramid and Lizard

In the computer game Timelapse™ (adventure game by Barracuda Inc.) one has to solve, among others, the following puzzle: on the tip of a double step pyramid sits a lizard (see picture on the right, the lizard is symbolized by an e). Below the double pyramid 6 switches are located, three to the left (switches l_1, l_2 and l_3) and three to the right (switches r_1, r_2 and r_3). With the help of these switches the lizard can be ordered to move across the double pyramid. If a switch on the left is pressed, the lizard moves counterclockwise, if a switch on the right is pressed, the lizard moves clockwise. Each switch is marked with two numbers (see picture on the right). The upper number indicates the number of steps the lizard moves if the switch is pressed. The lower number indicates how often a switch may be pressed. (That is, one may order the lizard only three times to move four steps counterclockwise etc.) Each step the lizars stops on gets marked (in the game: lighted), but if the lizard stops on a step for the second time, the marking is removed again. (As an example, the bottom picture on the right shows the state after the switches r_3, l_1, l_2, r_1 have been pressed (in this order). Note the reduced counters on the switches.) The puzzle is solved if every step of the double pyramid is marked. That is, we are looking for an order in which to press the switches, so that the lizard stops on each step once.



l_1	l_2	l_3
4	5	8
3	5	4

r_1	r_2	r_3
5	6	8
4	3	3



l_1	l_2	l_3
4	5	8
2	4	4

r_1	r_2	r_3
5	6	8
3	3	2

In the game Timelapse™ one can find a hidden scroll, on which the solution to the puzzle is stated. However, one may also consider to solve it with a computer program (e.g. by backtracking, simulated annealing, or an evolutionary algorithm). How? If possible, provide a solution to the puzzle. Additional question: How large is the search space (number of solution candidates)?