Evolutionary Algorithms Winter 2017/2018
PD Dr.-Ing. habil. Christian Borgelt from 2017.11.23

Exercise Sheet 2

Exercise 5 n-Queens Problem

In the lecture we considered solving the n-queens problems (placing n queens onto an $n \times n$ chessboard, such that in no rank, file or diagonal there is more than one queen) with the help of an evolutionary algorithm. As an encoding a chromosome with n genes was chosen, each of which represents one rank. A gene states the file on which the queen in the corresponding rank is placed.

a) Try to make intuitively plausible how the evolutionary algorithm finds a solution! What *intuitive* function have mutation and crossover?

b) Why is the evolutionary algorithm sometimes unsuccessful?

(Hint: On the web page for this lecture, a command line implementation of the algorithm in C is available, which may be helpful to answer this question:
`http://www.borgelt.net/teach/ea_eng.html`)

Exercise 6 n-Queens Problem

The encoding of the n-queens problem that was used in the lecture relies on the knowledge that in each rank of the chessboard there can be only one queen. Solution candidates with more than one queen per rank are already excluded by this encoding.

a) Design an evolutionary algorithm that uses an encoding that only represents whether a square is occupied by a queen or not (that is, an encoding that does not already exclude more than one queen per rank)! (Hint: there is more than one possibility for such an encoding.) What could be a crossover operation in this case?

b) Which advantages and disadvantages does the chosen encoding have, compared to the one used in the lecture? What do you expect w.r.t. the execution time of the algorithm?

Exercise 7 n-Queens Problem

In the encoding of candidate solutions of the n-queens problem that was used in the lecture, two genes of a chromosome may specify the same file. That is, a candidate solution may describe a situation in which two queens occupy the same file. However, it is immediately clear that all queens must be placed in different files. Therefore it suffices to consider *permutations* of the numbers 1 to n (or 0 to $n-1$) as possible encodings of candidate solutions.

a) What problems occur if one tries to restrict the encoding to such permutations as chromosomes?

b) How could these problems be handled?

Exercise 8 n-Queens Problem

In the lecture we discussed the program `qea.c` that is available on the web page accompanying the lecture (see Exercise 5) and that tries to solve the problem with an evolutionary algorithm, as well as the programm `queens.c` that solves it with a backtracking approach. Experiment with these programs and answer the following questions:

a) Is crossover or mutation less harmful to dispense with (for this problem)? How do you explain your observations? (Hints: crossover can be switched off by setting the fraction of individuals to be subjected to this operation to 0 by specifying `-f0`. To switch off mutation, set the mutation probability to 0 with `-m0`.)

b) Is the default value of the mutation probability (0.1 bzw. 10%) optimal? What values lead to better results?

c) Compare the execution time of the two programs for $n = 10$, $n = 20$ and $n = 30$! For the mutation probability use one of the values you found to be appropriate in part b). If needed, increase the (maximal) number of generations to be computed (using `-g`*number*).