Exam Evolutionary Computing
14.12.2006

NOTES:

1. Your name must be written on each sheet in CAPITALS.
2. You can answer the questions in English or in Dutch.
3. Points to be collected: 90, free gift: 10 points, maximum total: 100 points.
4. Grade: total number of points divided by 10, rounded up to the first decimal.

QUESTIONS

1. We are to schedule a number of experiments with evolutionary computing. That is, we have a number of experiments $E = \{e_1, \ldots, e_n\}$ and a number of experimental devices $D = \{d_1, \ldots, d_m\}$. For each experiment $e \in E$ there is one and only one device that can be used to perform it. If needed, we can use the notation $dev(e)$ to designate this device. (Note that devices can be used to perform different experiments.) The experiments cannot be performed in an arbitrary order, some expemeriments can only start after having finished others. This relationship is given by the binary constraint $pred(x, y)$, where $pred(e, f)$ means that e must be finished before we can start f $(e, f \in E)$. Furthermore, we know the time $t(e) = t(e, d(e))$ it requires to perform an experiment e on device $d(e)$. In this problem context a schedule is an assignment of starting times to experiments, the length of a schedule is time between 0 and the finishing time of the last experiment. Our task is to make a schedule that is

   (a) complete: every experiment e is assigned a time to start it on $d(e)$,

   (b) correct: no experiment starts before its predecessors are finished,

   (c) optimal: as short as possible.

   Specify an EA suitable[1] for solving this problem. In particular, give

   (a) **(5p)** a representation (the syntax of the chromosomes and a mapping between chromosomes and schedules),

   (b) **(5p)** a fitness function,

   (c) **(5p)** an appropriate crossover operator,

   (d) **(3p)** an appropriate mutation operator,

   (e) **(3p)** an appropriate selection mechanism,

   (f) **(3p)** an initialization method,

   (g) **(3p)** a stop condition.

[1] The EA does not have to be "smart" (efficient). But the representation and the operators should be such that a solution can be found.

2. (a) **(6p)** Explain self-adaptation (SA) of parameters in evolutionary algorithms. *What* is SA supposed to do, *why* do we expect that SA will achieve this, and *how* does the mechanics of SA work in general?

   (b) **(6p)** Describe in detail how SA of mutation stepsizes works in evolution strategies.

   (c) **(6p)** Is the order in which the $\vec{x}$ part and the $\vec{\sigma}$ part are mutated important? Why?

3. (a) **(3p)** What is a memetic algorithm?

   (b) **(3p)** Compare the performance and the range of applicability of a usual EA and a memetic algorithm.

   (c) **(3p)** One of the crossovers on the following list is inherently memetic. Which one and why? LIST = partially mapped crossover, edge crossover, order crossover, cycle crossover.

4. **(5p)** Consider the following statement:

   > 'The No Free Lunch Theorem implies that is is useless to compare different evolutionary algorithms.'

   Is this statement correct or not? Give arguments.

5. (a) **(4p)** What is the (methodological) problem with comparing an EA with other style search algorithms, for instance with a constructive method?

   (b) **(6p)** Give a possible solution to this problem.

   (c) **(6p)** For what kind of problems is *peak performance* a correct indicator of algorithm quality? For what kind of problems is it better to use *average performance*?

6. (a) **(6p)** Explain fitness proportional selection in genetic algorithms.

   (b) **(6p)** Name three problems with this selection mechanism.

   (c) **(3p)** Describe a possible alternative that would solve (some) of the problems with fitness proportional selection.