Exam Evolutionary Computing
11.12.2007

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.

QUESTIONS

1. We are to solve a financial optimization problem with evolutionary computing. The problem is to create a good investment portfolio by selecting n ($1 \leq n \leq 10$) companies $(c_1, \ldots, c_n)$ whose shares to buy and to determine how many shares to buy for each c_i. Note that n itself is a decision variable, we may go for 1, 2, 3, ... or 10 companies. The total amount of money to be invested is given: M. Hence, the (present) value of all shares together must not exceed M. Here you can assume that all necessary data, e.g., the price per share, are available. The utility of a given portfolio p is its expected value $F(p)$ after one year. This can be determined by a prediction system, which we assume to have. That is, for any p we can have $F(p)$ without knowing how it is calculated; F may be separable or not, linear or not, ...

   Your task is to define an EA suitable[1] for solving this problem. In particular, specify

   (a) **(5p)** what "dialect" of EAs to use for this problem and why, (EP, ES, GA, GP?)
   (b) **(5p)** a representation (the syntax of the chromosomes and a mapping between chromosomes and portfolios,
   (c) **(5p)** an appropriate crossover operator,
   (d) **(5p)** an appropriate mutation operator,
   (e) **(3p)** an appropriate selection mechanism,
   (f) **(3p)** an initialization method,
   (g) **(3p)** a stop condition,
   (h) **(3p)** a way to handle constraints (if applicable/necessary within your EA).

2. (a) **(3p)** Explain what order-based representation is.
   (b) **(5p)** Sketch a problem that can be well treated by order-based representation. Explain why order-based representation is a good idea for this problem.
   (c) **(4p)** Give the description of an order-based crossover operator.
   (d) **(3p)** Give the description of an order-based mutation operator.

[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.

3. (a) **(3p)** What is a constraint satisfaction problem (CSP)?
   (b) **(3p)** Why is it not straightforward to solve CSPs with EAs?
   (c) **(6p)** Outline one possible approach to solve a CSP with an EA. Explain the way constraints are handled in detail.

4. (a) **(6p)** Explain self-adaptation of parameters in evolutionary algorithms. Hints: discuss *what* is self-adaptation supposed to do, *why* do we expect that it will achieve this, and *how* does the mechanics of self-adaptation work in general.
   (b) **(6p)** Describe in detail how self-adaptation 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?

5. (a) **(3p)** What is anytime (algorithm) behaviour?
   (b) **(3p)** Given an anytime algorithm, is it worth to spend efforts on heuristic initialisation? Give arguments.
   (c) **(3p)** Given an anytime algorithm, is it worth to spend efforts on "long" runs? Give arguments.

6. **(4p)** Consider the following statement:

   > 'An EAs with a high average solution quality is always better than an EA with a low average solution quality.'

   Is this statement correct or not? Give arguments.