Course: Natural Computing

*7. Feedback on interim reports

J. Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

Question 1: Optimal population size

Goal: Understand cooperation in population-based MHO.

You are not required to obtain "nice" results, but if you aim to:

- Use well-understood algorithm
- Start at just one particle/ant/individual
- Problem should be not too hard (or the #FE not too small), because then the algorithm will mainly do random search, and more particles will usually lead to better results
- Problem should be not too easy (or the #FE not too large), because then good results will always be produced.
- Parameters setting: "exploitation" (or critical) range possible.
- Run several times and average to get a smooth curve

If your results happen to be not so "nice":

- Try to explain the observed behaviour

Question 1b: Dependence on problem difficulty

Goal: Understand that the Q1a) result is not unchangeable, but results from a compromise between cooperation and redundancy

- Two cases are perfect, if the second one is well-chosen
- If you (do not) observe a clear difference of the optimal numbers, try to give an explanation.

General remarks on question 1:

- Comparison between different populations size needs to be fair (E.g. keep #FE constant).
- Under certain conditions, the optimal population size may be ∞, but this means that the algorithm does not work well.
- I'm not aware of of any cases where there is more than one (local) optima of the population size, but could be possible for hierarchically structured problems.

Question 2a) Run

- 2_0) Introduction: Please don't forget this part!
- For a fair comparison, use the same parameters for a) and b) (could be more than one parameter set)
- 2a) Distinguish between GA, Fitness, and Evaluation regarding used information
- Present your results as a table or graph:
  - time needed for perfect solution vs. size
  - fraction of correct solutions for a constant setup vs. size
  - combination of both may be less clear

Question 2b) Suitable Fitness function:

- Make your new fitness function explict
- How do the suboptimal fitness values compare for different k?
- If you have less than 100% correct for the 3x3, it may be useful to analyse the failures of the algorithm.

Question 2c) Discuss parameters

- Did you work with more than one set of parameters?
- Did you expect other parameters to be good, but then found other ones?

If not:

- You could consider to try some odd parameters
- Did you check any other options for fitness functions?
- If you don't have much time: Discuss whether your assumptions in 2_0 were useful.

Question 3a/3b) Run GP

- 3a) Try to provide evidence that you actually solved the problem and that you did it yourself.
- 3b) Show time course of fitness
- Are any solution components found early on?
- Does the sequence contain "exceptional" values (could be the first item or the first two)?

Question 3c)

You can try to answer one of the following questions

- Typical solution: Does the algorithm work similarly well on a similar problem?
- Was the choice of parameters, operators, number of fitness cases, or initialisation critical for success?
- Would the solution have been found, if it wasn't known during algorithm design? (i.e. what about other alphabets?)

Negative results?

- Algorithms are slow, not easy to average, or get stuck
- Several problems here are search problems and require a perfect solution (which is not typical of MHO), what can be inferred from suboptimal solutions?
- For a perfect positive result often just two cases need to be presented, for a negative result, this gets more complicated.

General hints

- Give any numbers in a reasonable precision.
- Logathmic scales are very good, but try to use non-logarithmic values at the axis tics, or explain what logarithm you are using.
- Mention your termination criteria.
- Code listings and screenshots can be included in the appendix.
- What about any extra tasks? Discouraged. Creativity? Appreciated.