GAGP 2009: Assignment 1

This is an unmarked exercise. It requires you to write a program and to use it to carry out a number of investigations. You are to write up your investigations in the form of a conference paper with a maximum length of 6 pages. Submit your code and your paper electronically using the submit program:

submit ai3 gagp 1 <filename-of-your-paper> <filename-of-your-code>

Please use ai3 even if you are in another year or degree and see the man page for submit if you wish to submit more than two files. Submit your report as a pdf file and your code as a plain text file or files.

You should submit your assignment by 4pm of Monday 2nd November, 2009.

The travelling salesperson problem poses itself to a salesperson who is supposed to do business in N different cities. In order to reduce travel costs, the salesperson must not visit a city more than once and sequence of the cities should be chosen such that the total length of the journey is minimal. Or in slightly more abstract formulation: Given a list of cities and their pairwise distances, the task is to find a shortest possible tour that visits each city exactly once (wikipedia).

An *admissible solution* is any sequence that contains each city only once. An optimal solution has shortest length. We assume that the total length is the length of a closed tour, i.e. the salesperson returns finally to the city where she started from.

Generate a configuration of N cities as homogeneously distributed random points (x_i,y_i))in the square [0,1]x[0,1]. The fitness function for a particular tour is

$$f\left((x_1, y_1),\ldots,(x_N y_N)\right) = \sum_{i=1}^{N} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$, where x_{N+1} is identical to x_1 .

The solutions can be encoded by a string of length N, each character being a number from 1 to N, each character occurring only once. This corresponds to a tour starting in the city indicated by the first character, moving on to the city denoted by the second character etc. and finally returning to the first city again.

1. How many admissible solutions exist? How many optimal solutions are there (at least)?
2. Read the article on the "travelling salesman problem" on wikipedia. Describe briefly why the problem is difficult.
3. Design a Genetic Algorithm for solving the problem. Start with N=10 cities, but try to reach larger problem sizes, too. Use population size of, e.g., P=3N and the standard values p_m=0.01 and p_c=0.7.
    a. Use standard roulette wheel selection (or the second variant, i.e. "SUS" in Mitchell, p. 167)
    b. Simple 1-point crossover will mostly result in non-admissible strings. One way of dealing with this problem is to "repair" the genes of the offspring: If some character occurs twice in the child genome after crossover then replace one of the two characters by one of the missing characters in order to arrive at a string that encodes each city only once. Other ways of crossover are possible and have been suggested in the literature (check at scholar.google.com). Run the GA with two different types of crossover.

    c. Mutation is less of a problem. Test both swapping genes and inversion of a substring of random length.

4. Plot the fitness of the best individual as a function of the generation number for four different cases (two crossover schemes, two mutation schemes). Discuss criteria for termination of the algorithm. Compare in particular mutation by inversion with no recombination and mutation by swapping with recombination.
5. Test the algorithm with a configuration of cities where an optimal solution is obvious (e.g. placing the cities equidistantly on the circumference of a circle).
6. Try to improve your results based on the experiences in the previous point (5.) for an again random configuration of cities (not too many) by changing the parameters of the algorithm (P, p_m, p_c) or by introducing elitism.
7. Discuss the quality of the results and more generally the applicability of the GA to this problem.
8. Formulate the main points of your results as a brief conclusion.

In all cases you need to choose values of other parameters within the GA, and you should explain and justify how you came to this choice, given that you are aiming to get good performance from your GA. You also need to decide how you are going to present results that give a good overall picture of the behaviour of the GA.

Bear in mind that you will be using random number generators and that, to get a representative performance and results that are statistically significant, you will need to run each experiment several times. See Appendix A of the previous years' lecture notes for information on experiments and statistics. Also look at Mitchell Chapter 5 for issues to consider when designing a GA.

Your paper should describe the experimental design (i.e. what the experiments are and what they are intended to show), the results, and discuss the results in the light of the task. Use tables and figures where appropriate. Make sure that the code contains a concise description of its design.

The following criteria will be used in evaluating your assignment (that it is unmarked does not mean it remains unread):

- Is the design of the algorithm sound?
- How clearly is your design and implementation described?
- How sound are the experiments and how good are the results obtained?
- To what extent did your experiments give the GA "a good workout"?
- Does the discussion of your results shed light on how the GA works as you vary the parameters and undertake the tasks of varying difficulty?

Additional tasks ;-)

- Browse through the collection of problems at http://www.aridolan.com/ga/gaa/gaa.html
- If you have a windows 98/XP running on your computer you may want participate in the Golem@Home experiment (http://www.demo.cs.brandeis.edu/golem/)