ECMM409 Nature-Inspired Computation
Assignment One: Problem Solving with an Evolutionary Algorithm

This is an INDIVIDUAL assignment and is worth 20% of the module **Handout:** w/b 20th October 2008 **Handin:** 12.00noon Thursday 20th November – On Paper, To the Education Office **Back to you:** w/b 1st December

You are reminded of the University's Regulations on Collaboration and Plagiarism, details of which are available on the School web page

What you will do in this assignment is implement an evolutionary algorithm. You will apply it to the problem below. You will do a variety of experiments to help find out what parameters for the EA are best in this case. Implementation can be in the programming language of your choice. In the following sections, basic details of the problem are provided, then basic details of the algorithm, and finally a step by step guide of what you are expected o do. The final section indicates what should be in your report to be handed in.

The Problem

Working for a bank, you have been asked to develop an evolutionary system which will find the largest amount of money that can be packed into a security van. The money is separated into 100 bags of different denominations and the weight and value of the money of each bag is shown on the outside of the bag.

e.g.
Bag 1 Value = £94, Weight = 5.7Kg
Bag 2 Value = £74, Weight = 9.4Kg
.
.
.
Bag N Value = £x, Weight = iKg

The security van has a weight limit to what it can carry, so your system must try and decide which bags to put on the van, and which ones to leave behind. The best solution will be the one which packs the most money (in terms of value) into the van without overloading it.

- Your system will have to read in the 100 bag values from this file on the study resources: http://www.secamlocal.ex.ac.uk/studyres/ECM3412/BankProblem.txt
- The file contains the weight limit for the security van and the values and weights for each bag of money. Weights are all in kilos and the values are all in pounds sterling.
- You must decide how to represent this problem to the evolutionary algorithm, you must also decide what the fitness function should be.

The Evolutionary Algorithm

Implement the EA like this:

1. Generate an initial population of *p* randomly generated solutions, and evaluate the fitness of everything in the population.
2. Use the binary tournament selection twice to select two parents *a* and *b*.
3. Run crossover on these parents to give 2 children, *c* and *d*.

4. Run mutation on *c* and *d* to give two new solutions *e* and *f*. Evaluate the fitness of *e* and *f*.
5. Run weakest replacement, first using *e*, then *f*.
6. If a termination criterion has been reached, then stop. Otherwise return to step 2.

Termination Criterion: Will simply be having reached a maximum number of fitness evaluations. The result is then the fitness of the best chromosome in the population at the end.

Binary Tournament Selection: Randomly choose a chromosome from the population; call it *a*. Randomly choose another chromosome from the population; call this *b*. The fittest of these two (breaking ties randomly) becomes the selected parent.

Single-Point Crossover: Randomly select a 'crossover point' which should be smaller than the total length of the chromosome. Take the two parents, and swap the gene values between them ONLY for those genes which appear AFTER the crossover point.

Mutation: This is dependent on your representation, look at the lecture slides for some ideas on which mutation to implement given your representation. **Your mutation function must take a single integer parameter which will determine how many times it is repeated on a solution (e.g. M(1) – one mutation per chromosome, M(3) – 3 mutations).**

Weakest Replacement: If the new solution is fitter than the worst in the population, then overwrite the worst (breaking ties randomly) with the new solution.

WHAT YOU HAVE TO DO

Implement the described EA in such a way that you can address the above problem, and then run experiments which address the following questions. But first note that, in all of the below, a single trial means that you run the algorithm once and stop it when 10,000 fitness evaluations have been reached. Different trials of the same algorithm should be seeded with different random number seeds.

Run five trials of the EA with crossover & operator M1 and population size 20.
Run five trials of the EA with crossover & operator M1 and population size 100.
Run five trials of the EA with crossover & operator M3 and population size 20.
Run five trials of the EA with crossover & operator M3 and population size 100.
Run five trials of the EA with only operator M3 and population size 100.
Run five trials of the EA with only crossover and population size 100.

What to hand in

Hand in a report of a maximum of 6 pages (6 sides of A4). Tables and/or graphs of results should take up no more than 3 pages. In the remaining space, I want to see the fitness evaluation function (clearly commented) from your source code, and the answers to the following three questions.

Question 1: Which representation did you use and why?
Question 2: Which fitness function did you use and why?
Question 3: Which operator/popsize combination is best for the problem?
Question 4: What happens when you run the algorithm with only the crossover/mutation?
Question 5: What do you think the reasons are for your answers to Question 3&4?

In your answers, describe your observations of the results, and describe any tentative conclusions you feel like making, and describe any further experiments you felt it interesting or useful to do.

I would also like you to answer the following question:

Question 6: How would you modify your system to make this a multi-objective problem? In your answer you should refer to every aspect of the system that would have to be modified.

Markscheme

Correct/plausible and efficient implementation of fitness function	20%
Correct/plausible representation scheme	20%
Correct results from the EA runs	20%
Quality (e.g. readability & usefulness) of tables and graphs	10%
Answers to Questions 1-5	15%
Answer to Question 6	15%