

### Assignment One: Problem Solving with an Evolutionary Algorithm

Hand-out date: **17<sup>th</sup> October 2022**  
Hand-in date: **16<sup>th</sup> November 2022**

This CA is worth **40%** of the overall module mark

This is an **individual assessment** and you are reminded of the University's Regulations on Collaboration and Plagiarism. You are required to cite the work of others used in your solution and include a list of references, and must avoid plagiarism, collusion and any academic misconduct behaviours. Further details about Academic Honesty and Plagiarism can be found at

<https://vle.exeter.ac.uk/course/view.php?id=1957>

#### Task

What you will do in this assignment is to implement an evolutionary algorithm and apply it to the problem shown below. You will do a variety of experiments to help find out what parameters for the algorithm are best for this problem. **Implementation of the algorithm can be in the programming language of your choice.** In the following sections, details of the problem are provided, then basic details of the algorithm, and finally a description of the experiments you should carry out. 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 algorithm based 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 Weight = 9.4Kg, Value = £57

Bag 2 Weight = 7.4Kg, Value = £94

.

.

.

Bag N Weight = iKg, Value = £x,

The security van can carry only a limited weight, 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 should read in the 100 bag values from the file "BankProblem.txt" which is provided along with this document.
- 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

The evolutionary algorithm should be implemented as follows:

1. Generate an initial population of  $p$  randomly generated solutions (where  $p$  is a reasonable population size discussed in lectures and in the module reading), and evaluate the fitness of everything in the population.
2. Use the binary tournament selection **twice** (with replacement) 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 which is 10,000 (see Implementation and Experimentation below)

**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 to create two new children.

**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.

## Implementation and Experimentation

Implement the described EA in such a way that you can address the above problem and then run the experiments described below and answer the subsequent questions. 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**.

You should devise your own set of experiments to determine what effect (if any) the following parameters have on the performance of the optimisation:

1. Tournament size( $t$ )
2. Population size ( $p$ )
3. Mutation rate (i.e. the parameter identified in the mutation operator above) ( $m$ )

Your experiments should assess the performance of the algorithm over a number of randomly seeded trials for each setting of  $t$ ,  $p$ ,  $m$ , to provide robust results.

## Analysis

Record the best fitness reached at the end of each trial and any other variables during the run that you think will be important in answering the following questions.

***Hint:** You should think carefully about how best to present your results to show the behaviour of the algorithm during your trials*

Question 1: Which combination of parameters produces the best results?

Question 2: Why do you think this is the case?

Question 3: What was the effect when you removed mutation? What about crossover?

Question 4: If you were to extend your EA to work with a multi-objective version of this problem, which functions in your program would you change and why?

In your answers, describe your observations of the results, and describe any tentative explanations or conclusions you feel like making, in addition to any further experiments you felt it interesting or useful to do to answer the above questions or to further your understanding of the algorithm parameters.

## Submission

Submit **both your report and clearly commented code** as a zip file using E-BART (<https://bart.exeter.ac.uk/>) by **12noon** on the hand-in date shown above. The report should have a **maximum of 4 pages (4 sides of A4, references do not count towards the limit)** which should include a description of your experiments where tables and/or graphs of results should take up no more than 2-3 pages, and your answers to the questions/descriptions in the remaining space.

## Marking Scheme

Correct and efficient implementation of the algorithm	15%
Documentation of code	5%
Correct results from the EA runs	20%
Quality (e.g., readability & usefulness) of tables and graphs	20%
Answers to Questions	20%
Tentative Conclusions & Further Experiments	20%
Overlength submissions (per page)	-10%

---