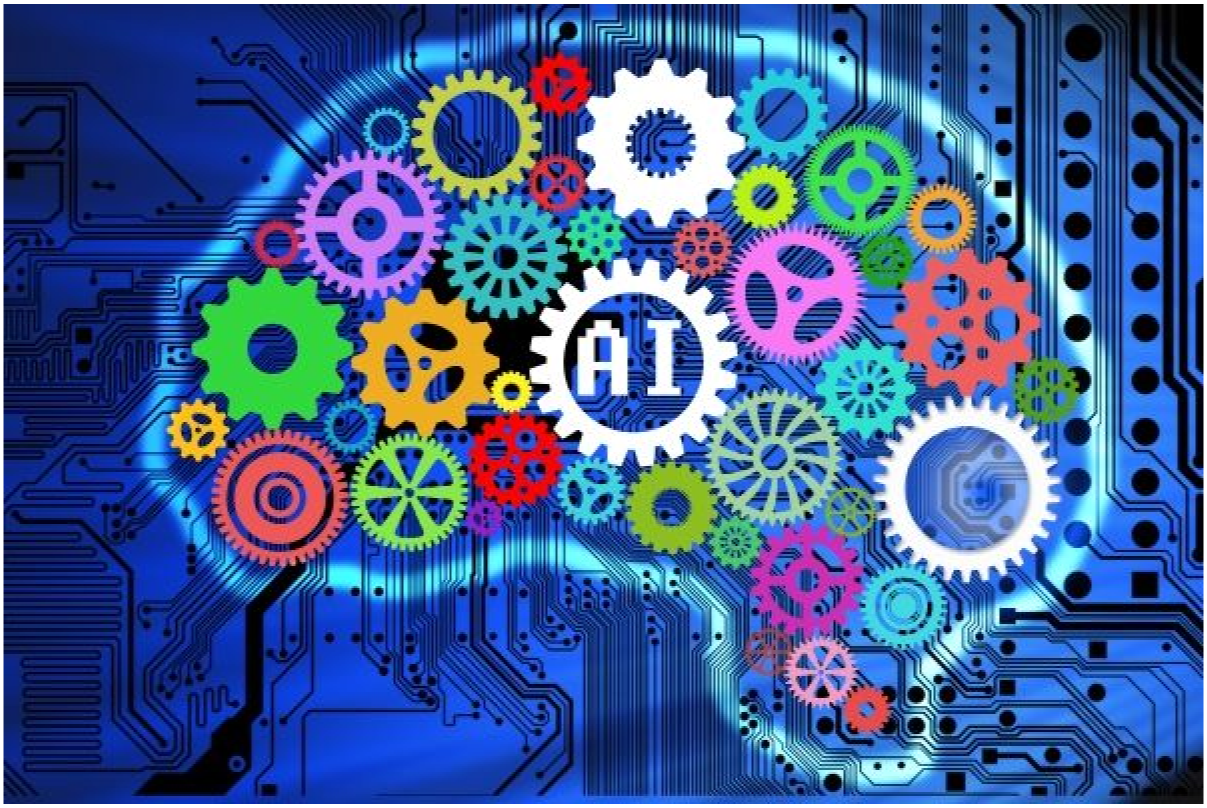

Artificial Intelligence Report

A delivery optimization problem solved using a Genetic Algorithm

Nadia Naidoo
217032715
21/09/2019

Introduction

This report documents the application of a Genetic Algorithm on a delivery optimization problem. A Delivery company needs to allocate a number of boxes to each delivery vehicle which has a weight capacity that cannot be exceeded. Each box is also assigned a value and the delivery van should reach a minimum predefined quota. The programmer's task is to read a specifically formatted text file and use genetic algorithm principles to try to optimize each vans cargo (maximum value and keep weight under or at capacity).

A description of a chromosome in the initial population and the population size used

From the input, three arrays were created; two integer arrays storing weights and values and one character array storing the names of each box. The arrays lengths were equated exactly to the number of objects (types of boxes). Each chromosome was created as a binary string stored in an array of the same length. The 1s represent if that box would be included and 0s represent if they are not included.

An example of a chromosome representing 6 types of boxes:

1	0	1	1	0	1

For each delivery van 1000 chromosomes were created however the initial population only included those whose total weights were less than the predefined capacity.

A definition of the fitness function used

For the optimization of the delivery van the the fitness function used was the maximum total value, since the weight limitations were adhered to in other areas of the code. The total value was calculated by the sum of the products of each value and their corresponding value in the chromosome.

$$f = \sum_{i=1}^{n} (v_i * c_i)$$

where :
f - fitness
v_i - i^{th} element of the value array
c_i - i^{th} element of the chromosome array

And for optimal results

$$f \geq quota$$

A description of the selection method used

A tournament selection method was used to find the parents of the next generation as follows:

1. Firstly a population of 'K' elements were selected from the existing population
2. The fittest chromosome would become a parent in the next generation
3. Step 1 and 2 were conducted to find each parent until a suitable next generation population was produced

A description of the mutation operator and the mutation rate used

The mutation operation randomly selected two nodes on the chromosome and swapped them if and only if a randomly generated number between 0 and 1 was less than or equal to the mutation rate which was chosen to be 0.001.

A description of the crossover operator and the crossover rate used

The crossover operation randomly selected two parent chromosomes by tournament selection. It then randomly chose a node and swapped over the same portion up until that node on the corresponding parent chromosome. Cross over only occurred if and only if a randomly generated number between 0 and 1 was less than or equal to the crossover rate which was chosen to be 0.9.

A description of the termination criterion

Termination occurred either when an optimal solution (weight<=capacity and value>=quota) was found or else when a user defined number of generations occurred. For instance after 1000 generations if there is an existing population the fittest of that generation will be output. If both of the previous conditions were not able to find a solution then a “No solution” response was returned.