COMP04 ASSIGNMENT 2

Divyan Hirasen
215018696

Chromosome in initial population

A chromosome is represented by an object of the Chromosome.java class. A chromosome has class attributes such as its weight and its value depending on the gene sequence which it contains. The gene sequence length of a chromosome is equal to the number of objects which are being used in the current problem. With regards to the gene sequence, a '1' and '0' simply indicate whether the item which is related to that gene index is included in the calculation of its weight and value.

As an example, if we have 5 items, a possible representation would be

Chromosome	1	0	1	1	1
Object	A	B	C	D	E

Therefore, the chromosome represents a combination of objects A, C, D, E.

Population size used

The population size used in the program is equal to 50. This allows a large enough initial population for the genetic algorithm to converge on the most probable solution chromosome based on its fitness. This also reduces the chances of premature convergence. Moreover, it is small enough to allow the genetic algorithm time to perform crossovers and mutations, producing a solution set with a more refined fitness.

Fitness function used

The fitness value calculated for a chromosome takes into consideration not only its gene sequence, but also its total weight, total value, the minimum value required and the maximum weight allowed. This allows a very specific fitness value, the small the fitness value, the better. Furthermore, no chromosome in any population will exist without a fitness value. Even if the characteristics of the chromosome, such as its weight and value, are not in the range specified, a fitness value is provided to it. This allows that chromosome a chance to get a 'better' fitness with more appropriate characteristics through crossover and mutation. With a fitness of 0.0 indicating a solution is found. The fitness is calculated as a percentage which indicates how 'far' it is away from a solution.

There are 4 ways to dispense a fitness value to a chromosome:

Case 1: If the weight of the chromosome exceeds the capacity of the van and the value of the chromosome exceeds the minimum value specified.

Unsatisfied Condition 1 Satisfied Condition 1

ChromosomeWeight > vanCapacity && ChromosomeValue > minQuota

fitnessValue = (ChromosomeWeight – vanCapacity) / vanCapacity

Case 2: If the weight of the chromosome is less than the capacity of the van and the value of the chromosome is less than the minimum value specified.

Satisfied Condition 1 Unsatisfied Condition 2

ChromosomeWeight <= vanCapacity && ChromosomeValue < minQuota

fitnessValue = (minQuota – ChromosomeValue) / minQuota

Case 3: If the weight of the chromosome exceeds the capacity of the van and the value of the chromosome is less than the minimum value specified.

Unsatisfied Condition 1 Unsatisfied Condition 2

ChromosomeWeight > vanCapacity && ChromosomeValue < minQuota

fitnessValue = (ChromosomeWeight – vanCapacity) / vanCapacity + = (minQuota – ChromosomeValue) / minQuota

Case 4:

Satisfied Condition 1 Satisfied Condition 2

ChromosomeWeight <= vanCapacity && ChromosomeValue >= minQuota

fitnessValue = 0.0

As you can see, a larger fitness value indicates how 'far' off this specific chromosome is from being a possible solution. This enables us to keep chromosomes, even if they currently are not useful, because there is a possibility that through crossover and mutation it may become a new path of convergence which would have otherwise been lost.

Selection method used

The selection method used to select 2 parents prior to performing a crossover is the tournament selection. This entails a tournament to take place, size of which is 10% that of the population. Thereafter, random individuals of the current population are selected to participate in the tournament, and only the most fit is taken as the winner, and the parent of the next generation. Along with this, elitism is included to make sure a chromosome with the best traits is always sent to the next generation.

Population size of 50

Tournament size of 10

So 10 chromosomes are randomly selected from the population and added to the tournament, and the best is selected as parent one, this is repeated for parent two.

Mutation operator

The mutation method used is uniform mutation. This type of mutation goes through each gene in the chromosome and checks the probability of mutating that specific gene.

	None	None	Mutate	Mutate	Mutate	None
Chromosome	1	0	1	1	1	0
Result	1	0	0	0	0	0

The probability of each gene mutating is set to 10%.

Crossover operator

The crossover method used is uniform crossover where after selecting 2 parents, the genes of both parents are iterated over and if the probability of that gene being involved in a crossover is true, then the genes are switched from each parent.

	None	None	Cross	Cross	Cross	None
Parent One	1	0	1	1	1	0
Parent Two	1	0	1	0	0	1
Child One	1	0	1	0	0	1
Child Two	1	0	1	1	1	0

The probability of a crossover occurring is set to 80%

Termination Criterion

The criteria used for termination are as follows:

- Max generation found (10 000 Generations)
    - This allows substantial attempts for the genetic algorithm to converge to a solution before termination.
- Solution found
    - Once a chromosome of fitness 0.0 is found, the generation is tracked and the program terminated.