Northeastern University
College *of* Engineering

INFO 6205: Program Structure and Algorithms

Final Project
Genetic Algorithm – Knapsack 0-1 Problem

Team 333
Fudi Liu
Harshit Raj
Meghana Srinivasa

Submitted:
04/15/2018

Problem Definition

The Knapsack Problem (KP) is a combinatorial optimization, which seeks for a best solution from a pool of solutions. In this particular KP problem, there are a set of distinct items with randomized weight and value, and the knapsack has a limited weight capacity. A solution is encoded into an array of binaries indicating items selected (1s) and items not selected (0s). A better solution selects items with as high value as possible to put into the knapsack without exceeding the weight capacity. Genetic Algorithm is considered to be a sound methodology in finding one of the optimal solutions for KP problems. By initializing, selecting, and reproducing, the solution pool evolves and gives better results at each generation. This report intends to introduce the implementations, observations, and results of this Genetic Algorithm project.

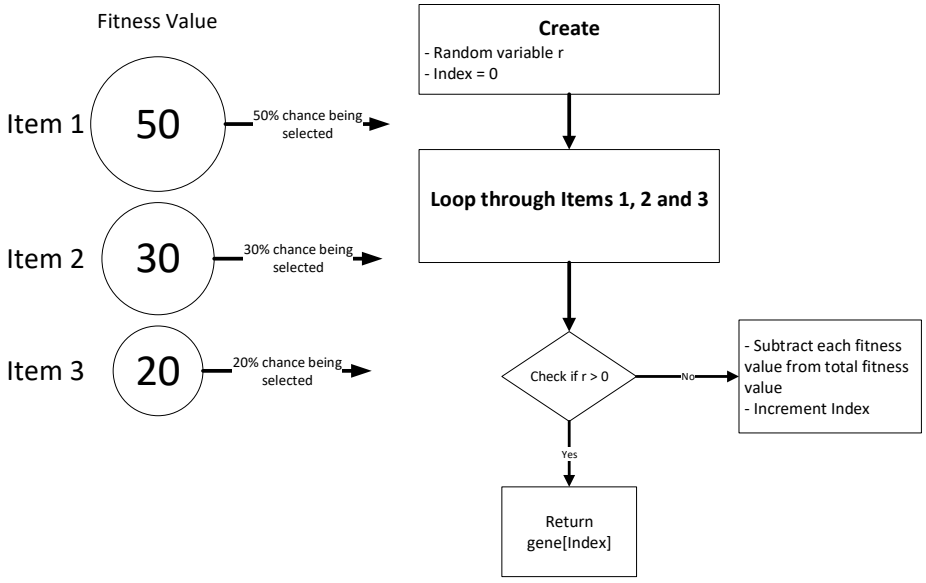
Implementation Design

Genetic Code: The binary encoded gene in our KP problem indicates the selection of items, and is the genetic code, which is also known as the genotype. It is essentially an array of binaries. 0 means the items is not selected, and 1 means the item is selected. Array length is the number of items.

Gene Expression: An item is a gene expression in which class the gene resides. This is also known as the phenotype. The attributes of an item are weight and value.

Fitness Function: The fitness of a gene is evaluated based on the items selected. We first added up all the values of all the items that are selected, then square the sum to give it a larger scale. The equation is $f(x)_{fitness} = (\sum Value_{selectedItem})^2$

Selection Strategy: Instead of sorting the genes by its fitness value, we decided that, to reproduce the next generation, a biased selection strategy should be used. This strategy selects a gene from the population based on its fitness. Genes with higher fitness values will have higher chance being selected. Below is an example explaining the details of the implementation.



Crossover: After selecting two parent genes using above selection strategy, we performed crossover on the two parents, and reproduce a child gene. We randomly select a midpoint where the first half of the child will be from parent A, and the second half will be from parent B. The implementation is as illustrated below.

Parent A: 1 | 0 | 0 | 1 | 1 | 0 | 1 Parent B: 0 | 1 | 1 | 0 | 0 | 1 | 1

Randomly selected midpoint is 3

Child: 1 | 0 | 0 | 0 | 0 | 1 | 1

Mutation: Mutation happens at a given mutation rate on the each child after crossover.

Before: 1 | 0 | 0 | 0 | 0 | 1 | 1

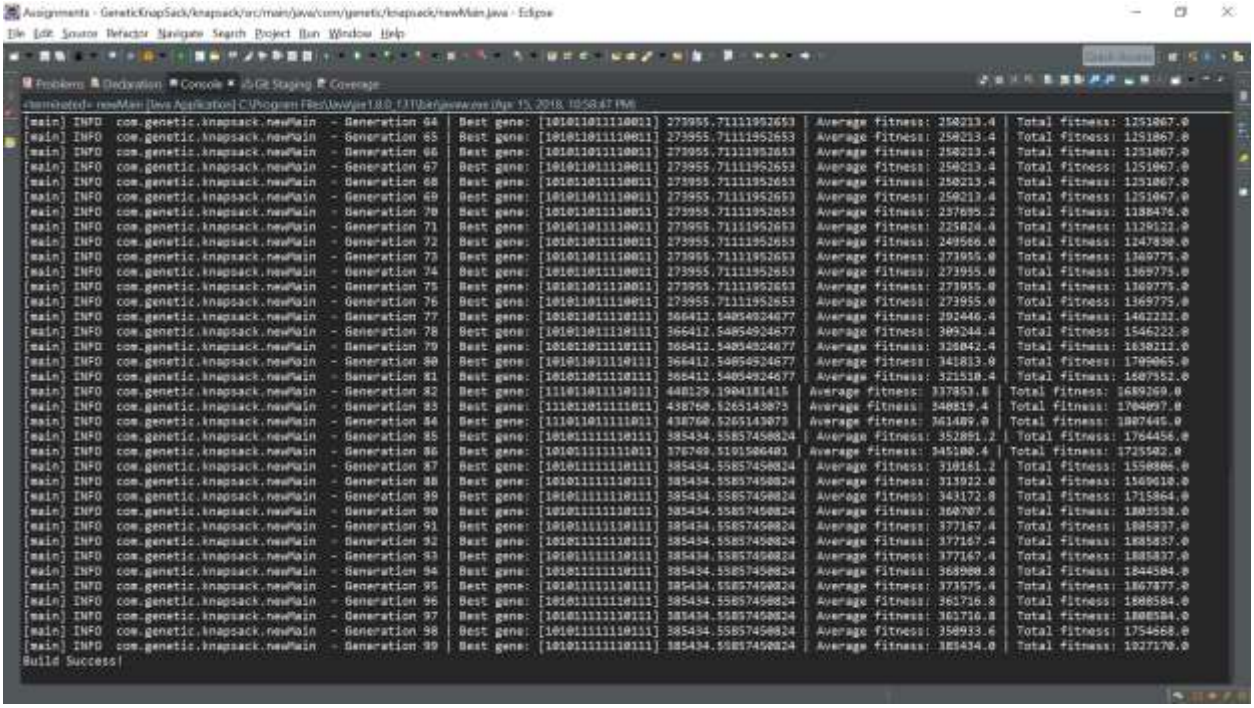
Mutation happens randomly at the give mutation rate

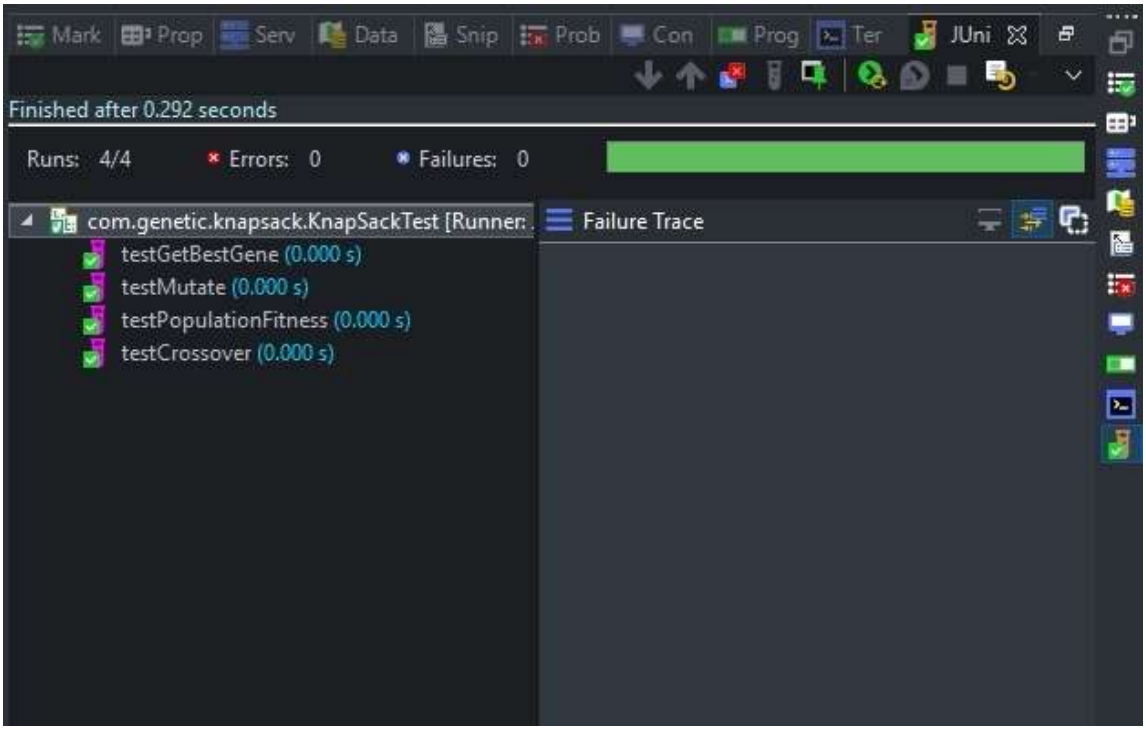
After: 1 | 0 | 1 | 0 | 0 | 1 | 1

Evolution: Genes are stored as a list in the Population class, and each Population is stored as a list in the Generation class. From each population, the selection Strategy will select two parents according to their fitness value, and by calling crossover, these two parents will reproduce a child, which inherits half of the gene from each parents. By repeating this process, the child gradually evolves, and has a higher fitness value.

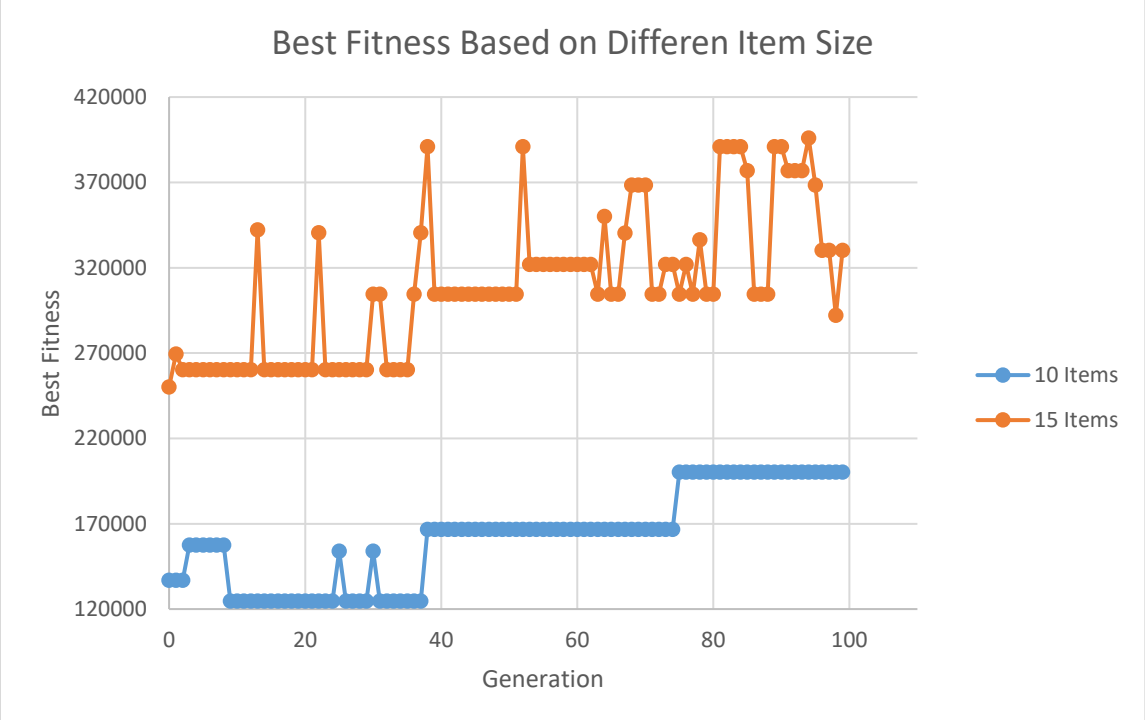
Results

Our project builds and passes all the test cases



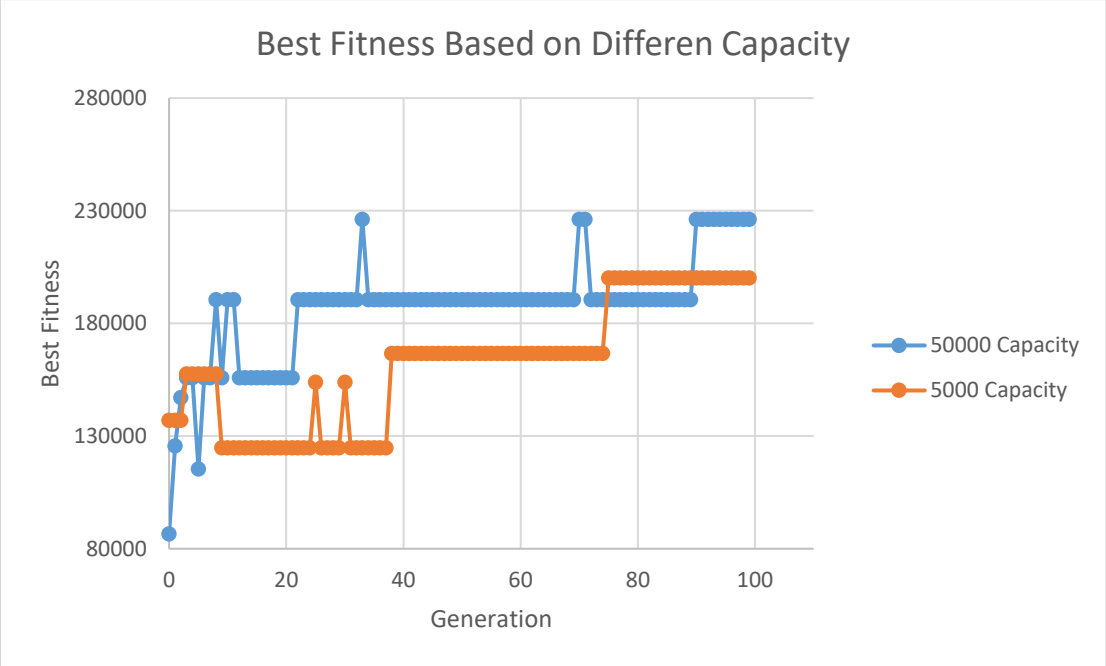


I ran the algorithm for graphs with knapsack capacity 5000, generation 100, 10 items, and population size 5, and then ran again with the same parameters except for 15 items. The output was exported to a csv file, and plotted.



It is clear that by running the Genetic Algorithm, the best fitness is gradually increasing in the next generation. This is because the selection mechanism will select the parents with higher

fitness value to reproduce the child. In addition, it is shown in the graph that the results produced by 15 items have higher fitness value. This is due to the added diversity from additional items. There is a higher chance that the parents will have better fitness value comparing to the ones from 10 items.



Another observation is that, as the weight capacity of the knapsack increases, the more items will be added to the knapsack, and there is a higher chance of selecting genes with higher fitness values. This result is shown in the graph above.