Name: Hani Mustafa Yaqbeh Project1 Dr.Majdi Mafarjeh

Student ID: 1160155 Artificial Intelligence

***Introduction***

Knapsack problem

is an optimization problem where you have a sack with a given capacity and a set of items each with a weight and a profit. The goal is to find the best set of items to fit in the sack where the total weight of the items selected doesn’t exceed the capacity of the sack and the total profit of item is as large as possible.

Genetic Algorithm

is a set of operation is repeated to solve a problem , which is follow the idea of stay for the best profit (fittest) from generation to anther to reach an optimal solution. On the other hand genetic it's a iterative procedure which is will represent a nominee solutions to be like a string of genes which is called a chromosome. Genetic are used to improve the performance other AI methods.

Genetic Terms

1)Chromosome: that mean a nominee solutions to be like a string of genes that represented or encoded to a string of bits.

2)Fittness: The probability that the sates will live to reproduced

3)CrossOver: which that two chromosomes exchange chunks between of them of genes info ,and the result called offspring.

4)Mutation: In this term, offspring is subject to mutation that’s for in which primary bits of DNA are changed from parent to offspring.

5)Selection: select which to select items instead of profit then sent this items to make crossover on it .

**How I Use Genetic Algorithm to Solve Knapsack Problem**

1)generate initial population:

Generate random population of *n* chromosomes which each chromosome contain a string of genes which randomly represented zero’s and one’s and see if weight for the chromosome is less or equal maximum Weight that can I take it from My Knapsack I will add it to my population if not I will ignore it .

2)Selection:

After I make A fitness function which is get the best profit for each chromosomes and sort them from high to low then I take a selection After I gave a crossover ration I multiply it with size of population then do sublist to population arraylist and put it in the population for crossover and what I have still in the old population put it in the new population .

3)crossover :

which I take a randomly index for first chromosome and another one for another chromosome then I make a crossover randomly and do a swip for there genes that was token .After that check if the weight still less than or equal maximum Weight that can I take it from My Knapsack I will add it to the new population if not I will ignore it .

4)Mutation:

After what I do in the above I take the mutation ratio and then I multiply with the new population size which( have the offspring and that what was stilled in the old population put it in the new population) then but number for crossing point then make randomly changing for bits if 1 the 0 and if 0 the 1 . After that if the profit after mutation for the chromosome is bigger than before add it to the population and remove the old one .

***Results***

Num of Items=100

Maxweight=500

Crossover=0.7

mutationRate=3

populationSize=100

mutationRatio=0.05

Generaation=10

Max profit =6209, max weight = 453 at generation=1 time=40ms

Another runtime Max profit =6198, max weight = 490 at generation=3

Another runtime Max profit =6180, max weight = 468 at generation=9 time=43ms

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Num of Items=1000

Maxweight=5000

Crossover=0.7

mutationRate=3

populationSize=120

mutationRatio=0.05

Generaation=10

Max profit =559940, max weight = 4440 at generation=2 time=4260ms

Another runtime Max profit =551167, max weight = 4492 at generation=2 ,time=204ms

Another runtime Max profit =540698, max weight = 4450 at generation=4 time=198ms

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Num of Items=100

Maxweight=500

Crossover=0.7

mutationRate=10

populationSize=100

mutationRatio=0.03

Generaation=10

Max profit =6958, max weight = 494 at generation=3 time=34ms

Another runtime Max profit =6890, max weight = 496 at generation=4 ,time=71ms

Another runtime Max profit =7115, max weight = 496 at generation=6 time=90ms

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////