

PROGRAMMING ASSIGNMENT 2 - REPORT

Name : Harini Gowdagere Tulasidas

Email: gharini@pdx.edu

In this assignment the knapsack problem was implemented using the genetic algorithm. The crossover used is single point cross over and the cross over point is chosen in random. It generates two children genomes and one of them is mutated on a probability of 50%. The population size , weights ,values and capacity of the knapsack is accepted from user and the optimal value is returned.

The dataset used for the experiment : https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/knapsack_01.html

The program was executed with different population sizes(between 10 to 30)and different number of iterations(between 200 - 400). The results were optimal or very close to the optimal solution specified for the particular data set. In some rare cases an irregular result where the final result was (0 ,1 ,1, 1, 1, 1, 1, 1, 1]) (all items are in the knapsack and fitness is 0) was also seen where the optimal solution couldn't be obtained within the number of iterations. It was also hugely dependent on the initial population. But in most cases the program produced optimal result.

Some of the results are shown below.

Test 1:

Enter the number of elements: 4

Enter the weights: 45 40 50 90

Enter the Values: 3 5 8 10

Enter the capacity of the Knapsack: 100

Enter population Size: 10

Enter the number of Iterations: 250

Initial population :

(0, [0, 1, 0, 1])

(0, [1, 0, 0, 1])

(0, [0, 0, 0, 0])

(0, [1, 0, 0, 1])

(3, [1, 0, 0, 0])

(0, [0, 0, 0, 0])

(0, [0, 0, 1, 1])

(0, [1, 0, 1, 1])

(0, [0, 1, 1, 1])

(5, [0, 1, 0, 0])

Final value : (13, [0, 1, 1, 0])

The total value/fitness of the Knapsack is : 13

The total weight in the knapsack is 90

Test 2:

Enter the number of elements: 10

Enter the weights: 23 31 29 44 53 38 63 85 89 82

Enter the Values: 92 57 49 68 60 43 67 84 87 72

Enter the capacity of the Knapsack: 165

Enter population Size: 20

Enter the number of Iterations: 200

Initial population :

(0, [0, 0, 1, 1, 1, 1, 0, 1, 0, 1])
(0, [1, 1, 0, 1, 1, 1, 1, 1, 1, 0])
(0, [0, 1, 0, 1, 1, 1, 1, 0, 1, 0])
(0, [1, 0, 0, 0, 0, 1, 1, 0, 0, 1])
(228, [1, 0, 1, 0, 0, 0, 0, 0, 1, 0])
(0, [0, 1, 1, 1, 1, 1, 1, 0, 1, 1])
(0, [0, 0, 0, 1, 0, 1, 0, 1, 0, 1])
(222, [1, 0, 0, 0, 0, 1, 0, 0, 1, 0])
(0, [1, 1, 1, 1, 1, 1, 1, 0, 0, 1])
(0, [0, 1, 1, 0, 0, 1, 1, 1, 1, 0])
(0, [1, 1, 0, 0, 0, 1, 1, 1, 1, 0])
(0, [1, 1, 1, 1, 1, 0, 0, 1, 1, 1])
(0, [0, 1, 0, 0, 0, 1, 0, 1, 1, 0])
(0, [0, 1, 1, 1, 0, 0, 1, 1, 1, 1])
(0, [0, 0, 1, 0, 0, 0, 0, 1, 0, 1])
(0, [1, 1, 0, 0, 1, 1, 0, 1, 0, 1])
(0, [1, 1, 0, 1, 1, 0, 1, 0, 1, 1])
(0, [1, 0, 0, 1, 1, 1, 0, 0, 0, 1])
(0, [0, 0, 0, 0, 0, 1, 0, 1, 0, 1])
(0, [0, 0, 0, 0, 1, 1, 1, 1, 1, 0])

Final value : (309, [1, 1, 1, 1, 0, 1, 0, 0, 0, 0])

The total value/fitness of the Knapsack is : 309

The total weight in the knapsack is 165

Test 3:

Enter the number of elements: 7

Enter the weights: 41 50 49 59 55 57 60

Enter the Values: 442 525 511 593 546 564 617

Enter the capacity of the Knapsack: 170

Enter population Size: 20

Enter the number of Iterations: 300

Initial population :

(0, [0, 1, 1, 0, 0, 1, 1])
(0, [0, 1, 1, 1, 0, 0, 1])
(0, [1, 0, 1, 1, 0, 0, 1])
(1581, [1, 0, 0, 1, 1, 0, 0])
(0, [0, 0, 0, 0, 0, 0, 0])
(953, [1, 0, 1, 0, 0, 0, 0])
(564, [0, 0, 0, 0, 0, 1, 0])
(593, [0, 0, 0, 1, 0, 0, 0])
(0, [1, 0, 0, 0, 1, 1, 1])
(0, [0, 1, 0, 0, 1, 1, 1])
(1629, [0, 1, 1, 1, 0, 0, 0])
(0, [1, 1, 0, 1, 0, 1, 1])
(0, [1, 1, 1, 0, 0, 1, 1])
(1621, [0, 0, 1, 0, 1, 1, 0])
(0, [0, 0, 0, 1, 1, 0, 1])
(0, [1, 1, 0, 0, 1, 1, 0])
(0, [0, 1, 1, 1, 1, 0, 1])
(0, [1, 0, 1, 1, 1, 1, 0])
(967, [1, 1, 0, 0, 0, 0, 0])
(0, [1, 0, 1, 1, 1, 0, 1])

Final value : (1735, [0, 1, 0, 1, 0, 0, 1])

The total value/fitness of the Knapsack is : 1735

The total weight in the knapsack is 169

Test 4:

Enter the number of elements: 15

Enter the weights: 70 73 77 80 82 87 90 94 98 106 110 113 115 118 120

Enter the Values: 135 139 149 150 156 163 173 184 192 201 210 214 221
229 240

Enter the capacity of the Knapsack: 750

Enter population Size: 20

Enter the number of Iterations: 200

Initial population :

(0, [1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0])
(1116, [0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1])
(0, [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0])
(0, [0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0])
(0, [1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1])
(0, [1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1])
(1069, [1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0])
(0, [0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1])
(0, [0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1])
(0, [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1])
(0, [0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0])
(1374, [1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0])
(1155, [0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0])
(1062, [1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1])
(0, [1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0])
(0, [0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0])
(0, [0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0])
(0, [1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1])
(0, [1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1])
(0, [1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1])

Final value : (1450, [0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1])

The total value/fitness of the Knapsack is : 1450

The total weight in the knapsack is 749

Test 5:

Enter the number of elements: 24

Enter the weights: 382745 799601 909247 729069 467902 44328 34610
698150 823460 903959 853665 551830 610856 670702 488960 951111 323046
446298 931161 31385 496951 264724 224961 169684

Enter the Values: 825594 1677009 1676628 1523970 943972 97426 69666
1296457 1679693 1902996 1844992 1049289 1252836 1319836 953277 2067538
675367 8533655 1826027 65731 901489 577243 466257 369261

Enter the capacity of the Knapsack: 6404180

Enter population Size: 20

Enter the number of Iterations: 200

Initial population :

(19282318, [0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0,
1, 0, 1, 0, 0])
(0, [1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1,
0, 1])
(0, [1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1,
1, 1])
(0, [1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1,
1, 0])
(9924954, [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,
1, 1, 1, 0])
(0, [1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
1, 0])
(0, [1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1,
0, 1])
(0, [0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0,
1, 0])
(17450082, [1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0,
1, 0, 0, 1, 0])
(19900323, [1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1,
0, 0, 1, 0, 0])
(0, [0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0,
1, 1])
(12835258, [0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
1, 1, 1, 0, 0])
(10752220, [0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0,
0, 1, 1, 1, 0])
(0, [0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
1, 0])

```
(0, [1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1])
(0, [1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1])
(0, [1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0])
(0, [0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1])
(12343238, [0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0])
(0, [0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0])
```

```
Final value : (21113701, [1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1])
```

The total value/fitness of the Knapsack is : 21113701

The total weight in the knapsack is 6395142