Alexis Montes

CSCI 381

Karp-21 Project

Problem Assigned: Knapsack Problem

**Data for knapsack problem:**

Small:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Weight | 2 | 2 | 3 | 7 | 1 | 6 | 4 | 9 | 2 | 1 |
| Value | 8 | 9 | 5 | 5 | 1 | 4 | 5 | 3 | 6 | 9 |

Medium:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Weight | 2 | 2 | 3 | 7 | 1 | 6 | 4 | 9 | 2 | 1 | 6 | 5 | 1 | 8 | 3 |
| Value | 8 | 9 | 5 | 5 | 1 | 4 | 5 | 3 | 6 | 9 | 7 | 2 | 4 | 4 | 9 |

Large:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Weight | 2 | 2 | 3 | 7 | 1 | 6 | 4 | 9 | 2 | 1 | 6 | 5 | 1 | 8 | 3 | 8 | 4 | 7 | 6 | 3 |
| Value | 8 | 9 | 5 | 5 | 1 | 4 | 5 | 3 | 6 | 9 | 7 | 2 | 4 | 4 | 9 | 1 | 3 | 8 | 5 | 3 |

**How fitness function was determined:**

In the case of the knapsack problem the solution is the set of items that can fit within a designated knapsack weight without being more than it can hold. So we can consider the value of the objects as its fitness unless that weight is more then the knapsack can hold.

In Pseudocode:

if(Objects.weight < Knapsack.max\_weight){

Return objects.value;

} else {

Return 0;

}

**Stopping criteria:**

While for some approaches a program can stop looking for solutions if it finds a solution that meets a certain threshold, but in my approach it only stops after a certain number of data generations are found.

**Solutions generated:**

**Small Set:**

Brute Force:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | true | true | true | true | true | false | true | true |

Fitness Score: 52

Random:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | true | true | false | true | true | false | true | true |

Fitness Score: 51

Genetic Algorithm:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | true | true | true | true | true | false | true | true |

Fitness Score: 52

**Medium Set:**

Brute Force:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | true | true | true | true | true | false | false | true | false | false | true | false | true |

Fitness: 59

Random:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | true | true | true | true | true | false | false | true | false | false | true | false | true |

Fitness: 59

Genetic Algorithm:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | true | true | false | false | true | false | true | true | true | false | false | false | true |

Fitness: 59

**Large Set:**

Brute Force:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | false | false | true | false | false | false | true | true | true | false | true | false | true | false | true | true | false | false |

Fitness 64

Random:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| false | true | true | true | false | false | true | false | false | true | true | false | true | false | true | false | false | false | false | true |

Fitness: 56

Genetic Algorithm:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| true | true | false | false | true | false | false | false | true | true | true | false | true | false | true | false | true | true | false | false |

Fitness 64

**Statistics**

**Small Set**

Random Statistics

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Fitness | Average Fitness | Max Fitness |
| Generation 0 | 0 | 17 | 24 |
| Generation 50 | 25 | 25 | 34 |
| Generation 100 | 39 | 42 | 56 |

Genetic Algorithm

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Fitness | Average Fitness | Max Fitness |
| Generation 0 | 32 | 40 | 45 |
| Generation 50 | 49 | 51 | 64 |
| Generation 100 | 63 | 64 | 64 |

**Medium Set**

Random

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Fitness | Average Fitness | Max Fitness |
| Generation 0 | 0 | 12 | 17 |
| Generation 50 | 16 | 17 | 19 |
| Generation 100 | 30 | 45 | 59 |

Genetic Algorithm

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Fitness | Average Fitness | Max Fitness |
| Generation 0 | 34 | 34 | 52 |
| Generation 50 | 54 | 56 | 63 |
| Generation 100 | 63 | 64 | 64 |

**Large Set**

Random

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Fitness | Average Fitness | Max Fitness |
| Generation 0 | 12 | 15 | 18 |
| Generation 50 | 20 | 20 | 23 |
| Generation 100 | 43 | 46 | 56 |

Genetic Algorithm

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Fitness | Average Fitness | Max Fitness |
| Generation 0 | 0 | 43 | 45 |
| Generation 50 | 45 | 56 | 64 |
| Generation 100 | 62 | 64 | 64 |