Homework 6 Advanced Analytics and Metaheuristics

Group 16: Nicholas Jacob

March 29, 2024

1. Strategies for the problem

   (a) Two differing initialization solutions for the knapsack could be random or all zeros.

      i. Empty: We could also just ask that initially your knapsack is empty. We simply make the initial all zeros. We know this will be in the feasible set initially!

```
for i in range(n):
    x.append(0)
```

      ii. Random: Random has the benefit of being just that but it also has the serious issue of not being in the feasible set (nor any where close to the feasible set). This could be dealt with by re-seeding the random until it was inside the feasible regime.

```
for i in range(n):
    x.append(myPRNG.randint(0,1))  #initial seeding

i = myPRNG.randint(0,n-1) #random index to make zero if needed
while evaluate(x)[1] > maxWeight: #winnowing down randomly
  x[i] = 0
  i=myPRNG.randint(0,n-1)
```

   (b) Neighborhood

      i. Permutation
      ii. +1 Permutation
      iii. Sliding window
      iv. Slide and Flip?

(c) Infeasible

i. Small Value: If the weight is outside of allowed, simply make the value small (negative). While this works, the infeasible will match and end the while loop in the infeasible region.

```
if totalWeight > maxWeight:
    totalValue = -1000
```

This did not work well. Tried random value, that didn't work either. Maybe something with making the Value the opposite of its value so it will continue to decrease when you are out of the feasible region.

ii. Something Else