Project Proposal – CSCE686

Chad Willis

Updated June 20th

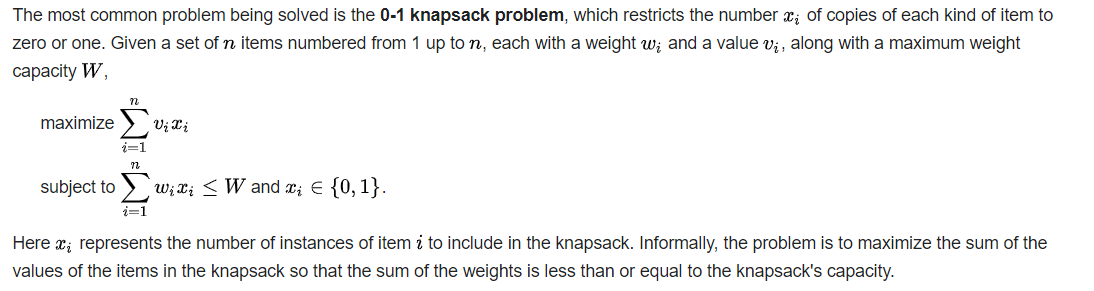
Problem Selection / Problem Domain Details

The real world problem combining two NP-Complete problems involves a single delivery van bringing groceries to local grocery stores and then going back to the starting point. The delivery driver wants to know the minimum miles needed to travel to go to every store and back. To determine this minimum amount of miles needed to travel, the traveling salesman problem can be used. At the same time, the other NP-Complete problem will be the knapsack problem. The van can only fit so many groceries into the van at max capacity. All the foods have a different weights and values. The delivery driver wants to find the maximum value of the groceries that can initially be put into the van such that the weights of the items put into the van are smaller than or equal to the max capacity.  
  
The traveling salesman problem can be defined as:  
Given a list of nodes and the weights between each pair of nodes, what is the least weight possible route that visits each node and returns to the origin node. This is different than the  [Hamiltonian Cycle](https://www.geeksforgeeks.org/backtracking-set-7-hamiltonian-cycle/). The Hamiltonian cycle problem is to find if there exist a tour that visits every city exactly once. Here we know that Hamiltonian Tour exists (because the graph is complete) and in fact many such tours exist, the problem is to find a minimum weight Hamiltonian Cycle.

One implementation of the TSP problem is as follows:

1. Consider city 1 as the starting and ending point. Since route is cyclic, we can consider any point as starting point.
2. Generate all (n-1)! permutations of cities.
3. Calculate cost of every permutation and keep track of minimum cost permutation.
4. Return the permutation with minimum cost.

The knsapsack problem can be defined as:  
Given weights and values of n items, put the items into a knapsack of capacity W to get the max total value in knapsack. Basically, you want to find the maximum value such that the weights of the items put into the bag are smaller than or equal to W. An item cannot be broken into smaller pieces; this is called the 0-1 property.   
In the case of , the Knapsack is analogous to the amount of supplies that the team can handle. The n items are all of individual supplies counted. The goal is to have the highest value, which has to do with how effective or needed the supplies are, while not going over the maximum weight allowed for the team to carry.



References

<https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/>

<https://en.wikipedia.org/wiki/Knapsack_problem>

<https://www.geeksforgeeks.org/traveling-salesman-problem-tsp-implementation/>

<https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/>

<https://www.geeksforgeeks.org/travelling-salesman-problem-implementation-using-backtracking/>

Phase 2: