CSE 6140 - Homework 4

Alexander Winkles

Problem 1

1. We will first design a branch-and-bound algorithm for minimum set cover. Consider a universe of elements U and a collection of sets S such that $\bigcup_{x \in S} x = U$.

   (a) Our subproblem will be a subset of S that we check to see if $\bigcup_{x \in S} x = U$.

   (b) To chose a subproblem to expand, we pick an element of S (possibly based on size or arbitrarily) and remove it.

   (c) We expand our subproblem by removing one set $y \in S$ and checking if $\bigcup_{x \in S-\{y\}} x = U$.

   (d) An appropriate lower bound will be the size of S each step.

   With all of this in mind, our algorithm is this:

---

procedure BnB_SetCover(U, S)
    $P \leftarrow \{S\}$ ▷ Assigns S to active subproblems
    Sol $\leftarrow 0$ ▷ Assigns initial solution
    bestval $\leftarrow \infty$ ▷ Assigns initial bound
    while $P \neq \emptyset$ **do**
        choose $T \in P$ ▷ Chooses a subproblem
        $P \leftarrow P - \{T\}$ ▷ Removes subproblem from set
        for $i \in T$ **do** ▷ Expands current subproblem into smaller ones
            $T_i = T - \{i\}$
        for T_i **do**
            $K \leftarrow \bigcup_{x \in T_i} x$
            if $K = U$ & $|T_i| <$ bestval **then** ▷ Checks if subproblem is a solution
                Sol $\leftarrow T_i$
                bestval $\leftarrow |T_i|$
            if $lowerbound(T_i) \leq$ bestval & $|T_i| \neq \emptyset$ **then** ▷ Adds back to subproblem set
                $P \leftarrow P \cup T_i$
    return Sol

   I believe this algorithm will work well for minimum set cover problems, as it does a good job of pruning solutions that are not minimal.

2. Now we will design a greedy heuristic.

Problem 2

1. In this greedy strategy, we take the containers in order and add them to trucks. Consider $W = \{0.5, 1.0, 0.5\}$. By the algorithm provided, this will require 3 trucks, as the first truck will take 0.5, then since 1.0 will overload

it, 1.0 will go to truck 2, which cannot fit any more, so the last 0.5 will go to truck three. However, the optimal number of trucks for this is 2, one that holds both 0.5 and the other that holds the 1.0.

2. Let $W = \sum_{i=1}^{n} w_i$ be the total weight of all the containers and let t^* be the minimum number of trucks required. Let t be the solution given by the greedy algorithm. We know that $W \leq t^*$, because otherwise we would need additional trucks to carry all of the containers. Consider a worst case scenario where the algorithm puts each container on its own truck; in this case we will have $t = n$, thus we know $Wt = Wn$. Additionally, we can write the previous inequality as $Wn \leq nt^*$. Combining these, we find that $Wt \leq nt^*$, or $t \leq (n/W)t^*$. Note that since each $w_i \leq 1$ by definition, $n/W \geq 1$.

3. Now suppose that the algorithm uses an even number of trucks, so $t = 2z$.