Algorithmic Game Theory
Exercise Sheet 11

In exercise sheet 9 we studied connection costs in rooted trees. We showed that the cost function is submodular. I reprint the exercise.

Exercise 1 (connection costs in rooted trees are submodular) *Let $T = (V, E)$ be a tree with root r and let w be a non-negative weight function on the edges. For a subset S of the vertices, let $c(S)$ be the cost of the smallest tree connecting the vertices in $S \cup \{r\}$. This tree is the union of the paths from r to v for $v \in S$.*

- *Show that c is submodular.*
- *Let $S \subseteq V$. Characterize the cost shares $\xi(i, S)$, $i \in S$ computed by algorithm Submodular-CostShares?*

Today, we want to show that the submodular cost shares can be computed in polynomial time. Recall the algorithm for computing the cost shares $\xi(i, S)$.

set $\alpha_i = 0$ for all $i \in S$. Set $F = \emptyset$. Set $\alpha = 0$.
while $F \neq S$ **do**
  $\{F$ is the maximum tight set; let α be the common value of all unfrozen cost shares.$\}$
  increase α until a new set goes tight;
  let A be a maximal tight set;
  freeze all unfrozen elements in A;
  $\{$the old F is a subset of A; the new F is equal to A.$\}$
end while

1. At what value α does the while-loop stop for the first time?
2. For a node v of T let T_v be the subtree rooted at v. For v and integer i, let $c_v(i)$ be the minimum cost of connecting i elements of $S \cap T_v$ to v. If T_v contains less than i elements of S, the cost is $+\infty$. Then $c_v(0) = 0$ for all v.

   Show: If v is a leaf and $v \in S$, $c_v(1) = 0$ and $c_v(i) = \infty$ for $i \geq 2$. If v is a leaf and $v \notin S$, $c_v(i) = \infty$ for $i \geq 1$.

   If v is an internal node with children x and y and $v \in S$, then

   $$c_v(i) = \min_{0 \leq j \leq i-1} \left((j \geq 1) \cdot (w_{vx} + c_x(j)) + (i - 1 - j \geq 1) \cdot (w_{vy} + c_y(i - 1 - j)) \right).$$

   Here $(j \geq 1)$ has value 1 if $j \geq 1$ and value 0 if $j < 1$.

   If v is an internal node with children x and y and $v \notin S$, then ...
3. What are the rules for computing $c_v(i)$ if v has ℓ children?
4. Assume that T is binary. Argue that all $c_v(i)$ be computed in polynomial time.

5. Show how to obtain the value under item 1) from the $c_r(i)$'s where r is the root of the tree.
6. How can you find the largest tight set?
7. What is the second value of α at which the while-loop stops?
8. Show that you can determine the second value of α from the $c_r(i)$.
9. Generalize and conclude that the cost shares can be computed in polynomial time.
10. What can you do if the arity of T is not bounded? KM does not know the answer to this item.