Max Wisniewski
Paul Podlech

Dozent : Panos Giannopoulos

Exercise 1 : *Multicut Problem in Trees*

Problem and Definitions

Problem:
In this exercise we will solve the follwing problem.
Given a tree $T(V, E)$ and k pairs of vertices s_i, t_i and costs $c_e \geq 0$ for each edge $e \in E$. The task is to find a minimum-cost set of Edges $F \subset E$ such that for each pair s_i, t_i s_i is in a different component than t_i in the graph $G(V, E \setminus F)$.

To reformulate the last criterion, this means, there exist no path from s_i to t_i, or even better, from every path from s_i, t_i in T there has to be at least one edge in F.

With this we can formulate the IP

$$\begin{array}{rrcll} \min & \sum\limits_{e \in E} c_e x_e & & & \\ \text{subject to} & \sum\limits_{e \in P_i} x_e & \geq & 1, & 1 \leq i \leq k \\ & x_e & \in & \{0,1\} & e \in E. \end{array}$$

This matches exactly our definition, as formulated above.
In this IP P_i is the set of edge on the way from s_i to t_i (In a tree there exists exactly one way from one node to another) and x_e is the variable that denotes, whether e is in F or not.

Next we will formulate the Primal and the Dual LP with which we than will work.

Primal LP

$$\begin{array}{rrcll} \min & \sum\limits_{e \in E} c_e x_e & & & \\ \text{subject to} & \sum\limits_{e \in P_i} x_e & \geq & 1, & 1 \leq i \leq k \\ & x_e & \geq & 0 & e \in E. \end{array}$$

Dual LP

$$\begin{array}{rrcll} \max & \sum\limits_{i=1}^{k} y_i & & & \\ \text{subject to} & \sum\limits_{i: e \in P_i} y_i & \leq & c_e, & e \in E \\ & y_i & \geq & 0 & e \in E. \end{array}$$

As one can see and assume from the simple Cut Problem, the dual is a Multiflow, such that on no edge there is more flow, than the capacity of the edge.

Root, Height and Ancestor
In the following we assume, that the tree is rooted at a arbitrary vertex r. From this

root node we can define the $depth(v)$ of a node $v \in E$ by the number of edges from r to v. The depth can be computed for every node by a simple iteration over the tree with the runtime of $O(n)$.

Next we will donate the *Lowest-Common-Ancestor* $lca(s_i, t_i)$ as the vertex $v = \operatorname{argmin}_{u \in P_i}\{depth(u)\}$. This is namly the vertex on the path that is nearest to the root. This nodes can be found will computing the depth without increasing the runtime.

The Algorithm

We will solve this problem by applying the Primal-Dual-Methode.

The algorithm will work as follows:
We start with an empty set of edge. Next we will iterate over all $lca(s_i, t_i)$ for every $1 \leq i \leq k$ from the one with the biggest depth to the smallest one. Next we try to increment the flow y_i of the given vertex $lca(s_i, ti)$ we took, until some constraint is met with equality. This edge we will than take into the solution set. In the last step we will remove this time from the root to the bottom, every edge, that will result in a still feasable solution.

Let $\mathcal{A}$ be the following algorithm

```
F ← ∅
I ← {1,..,k}
y_i ← 0   ∀i ∈ I
k ← 0
WHILE ∃i : s_i, t_i connected DO
    k ← k + 1
    i ← argmax_{j∈I} lca(s_j, t_j)
    e_k ← argmin_{e_k∈P_i} {c_{e_k} − Σ_{j:e_k∈P_j} y_j}
    Δ ← c_{e_k} − Σ_{j:e_k∈P_j} y_j
    f_i ← f_i + Δ
    I ← I \ {i}
    F ← F ∪ {e_k}

FOR z FROM k DOWNTO 1 DO
    IF F \ {e_z} is feasable THEN
        F ← F \ {e_z}
```

The Proof

In this part we will show, that the follwoing holds

Claim 1. *The algorithm $\mathcal{A}$ has a runtime in P.*

Claim 2. *The algorithm $\mathcal{A}$ is a 2-approximation-algorithm for the multicut problem in trees.*

***Proof* ??:**
tbd

***Proof* 2:**
tbd