Optimal Tree Labeling

Mohamed Babana

January 4, 2019

1 Problem description

We have a tree $T = (V, E)$ and S a finite set of labels, where V is the set of vertices and E is the set of edges. Every vertex $v \in V$ has a label $L(v)$, which is a subset of S. For $e = (u, v) \in E$, we define the weight $w(e)$ as the Hamming distance between $L(v)$ and $L(v)$. In *the optimal tree labeling problem*, you are given an unrooted tree with the labels of all the leaf vertices. The goal is to find some way to label all the non-leaf vertices, such that the total weight of all the edges in the tree is minimized under this labeling.

2 Solution

2.1 Solution with $|S| = 1$

Since $|S| = 1$, we can assume that the labeling is binary

$$L(v) = \begin{cases} 1 & \text{if the vertex } v \text{ has } S \text{ as label} \\ 0 & \text{otherwise} \end{cases}$$

We will call **admissible binary labeling** every binary vertex labeling whose restriction on the set of leaves coincides with a given binary labeling of leaves.

If all vertices of T are leaves, then there is nothing to solve, so we assume that there exists a vertex r of T which is not a leaf. Fix it as a root of T. For each vertex v of T, let T_v be the subtree of T rooted at v. Given an admissible labeling L of T and a vertex v of T, let $w(v, L)$ be the total weight of edges of T_v. For each vertex v and $i = 0, 1$, we define $w_i(v)$ as the minimum total weight of edges of T_v taken over all admissible binary labeling of T such that v is labeled by i, i.e.

$$w_i(v) = \min_{\substack{\text{admissible binary labeling } L \text{ of } T \\ L(v)=i}} w(v, L).$$

$$\boxed{\text{The minimum total edge weight which we are looking for is } \min\{w_0(r), w_1(r)\}}.$$

Let L be an admissible labeling of T and v a non-leaf vertex. Since $\{T_u : u \text{ is a child of } v\}$ are disjoints,

$$\begin{aligned} w(v, L) &= \sum_{\substack{u \text{ is a child of } v \\ e=(u,v)}} w(e) + w(u, L) \\ &= \sum_{\substack{u \text{ is a child of } v \\ L(u)=L(v)}} w(u, L) + \sum_{\substack{u \text{ is a child of } v \\ L(u)=1-L(v)}} 1 + w(u, L) \end{aligned}$$

In addition $w(u,L) \geq w_{L(u)}(u)$, so

$$\begin{aligned} w(v,L) &\geq \sum_{\substack{u \text{ is a child of } v \\ L(u)=L(v)}} w_{L(v)}(u) + \sum_{\substack{u \text{ is a child of } v \\ L(u)=1-L(v)}} 1 + w_{1-L(v)}(u) \\ &\geq \sum_{u \text{ is a child of } v} \min\{w_{L(v)}(u), 1 + w_{1-L(v)}(u)\}. \end{aligned}.$$

Hence, for each $v \in V$ and $i = 0, 1$ we have

$$w_i(v) \geq \sum_{u \text{ is a child of } v} \min\{w_i(u), 1 + w_{1-i}(u)\}.$$

On the other hand, for each vertex $v \in V$ and $i = 0, 1$ let $L_{v,i}$ be an admissible binary labeling such that $L_{v,i}(v) = i$ and $w_i(v) = w(v, L_{v,i})$.

We label v by i and for every child u of v, if $w_i(u) < 1 + w_{1-i}(u)$ we label u by i and the subtree T_u with respect to $L_{u,i}$, otherwise we label u by $1 - i$ and the subtree T_u with respect to $L_{u,1-i}$. We just construct an admissible labeling L' such that $L'(v) = i$ and for every child u of v $w(u, L') = \min\{w_i(u), 1 + w_{1-i}(u)\}$, then

$$w(v, L') = \sum_{u \text{ is a child of } v} \min\{w_i(u), 1 + w_{1-i}(u)\}.$$

Since $w_i(v) \geq w(v, L')$, we prove that

$$w_i(v) \leq \sum_{u \text{ is a child of } v} \min\{w_i(u), 1 + w_{1-i}(u)\}.$$

Finally for each $v \in V$ and $i = 0, 1$ we have the following recursive formula to compute $w_i(v)$,

$$w_i(v) = \sum_{u \text{ is a child of } v} \min\{w_i(u), 1 + w_{1-i}(u)\}.$$

2.1.1 Algorithm

Using the above results we can write the following pseudocode :

Input : T unrooted tree, L a binary labeling of leaves
Output : total minimum weight over all admissible binary labeling of T
find r a non-leaf vertex and root the tree T on r;
return $\min\{w_0(r, T, L), w_1(r, T, L)\}$;

Algorithm 1: *OptimalTreeBinaryLabeling*

Input : v a vertex, T a rooted tree, L a binary labeling of leaves
Output : minimum weight of the set of edges T_v over all admissible labeling such that v has i as label
if *v is a leaf* **then**
    if $L(v) = i$ **then**
        return *0*;
    else
        return ∞;
    end
else
    return $\sum_{u \textit{ is a child of } v} \min\{w_i(u), 1 + w_{1-i}(u)\}$;
end

Algorithm 2: w_i

2.1.2 Analysis of the algorithm

- The termination of the algorithm is clear since we have a initial condition on leaf vertices.
- Using a **depth first search**, we can solve the problem in $O(|E|) = O(|V|)$. The algorithm has a linear time complexity.

2.2 Solution for the general case

To solve the problem with an arbitrary S, we sum the results over all $s \in S$. We have the following pseudocode :

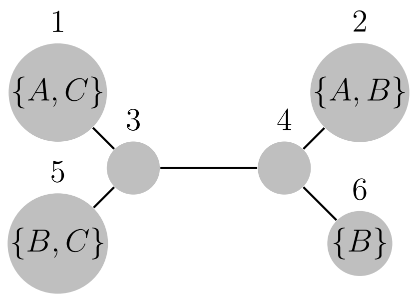
```
Input : T unrooted tree, L a labeling of leaves, S set of labels
Output : total minimum weight over all admissible labeling
L' a binary labeling;
w = 0;
for s ∈ S do
    for v ∈ V do
        if v is a leaf then
            if s ∈ L(v) then
                L'(v) = 1
            else
                L'(v) = 0
            end
        else

        end
    end
    w = w + OptimalTreeBinaryLabeling(T, L')
end
return w;
```

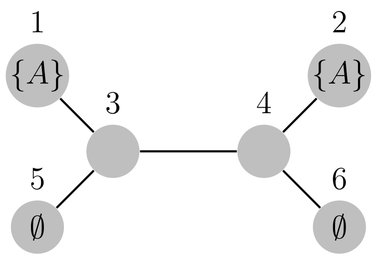
Algorithm 3: optimal tree labeling

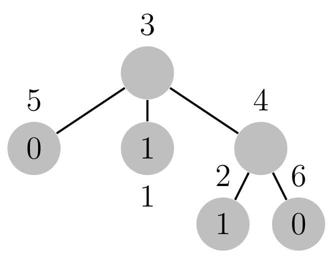
2.3 Example

In this section we apply the algorithm to the following example :



We start by appliying the *OptimalTreeBinaryLabeling* for the 'A'





We have $w_0(1) = w_1(5) = w_0(2) = w_1(6) = \infty$ and $w_1(1) = w_0(5) = w_1(2) = w_0(6) = 0$ then

$$w_0(4) = \min\{w_0(2), 1 + w_1(2)\} + \min\{w_0(6), 1 + w_1(6)\} = \min\{\infty, 1 + 0\} + \min\{0, 1 + \infty\} = 1$$

and

$$w_1(4) = \min\{w_1(2), 1 + w_0(2)\} + \min\{w_1(6), 1 + w_0(6)\} = \min\{0, 1 + \infty\} + \min\{\infty, 1 + 0\} = 1$$

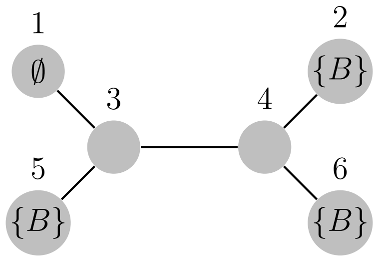
Finally

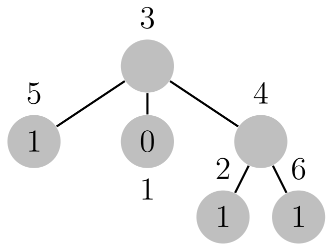
$$\begin{aligned} w_0(3) &= \min\{w_0(1), 1 + w_1(1)\} + \min\{w_0(5), 1 + w_1(5)\} + \min\{w_0(4), 1 + w_1(4)\} \\ &= \min\{\infty, 1 + 0\} + \min\{0, 1 + \infty\} + \min\{1, 1 + 1\} \\ w_0(3) &= 2 \end{aligned}$$

and

$$\begin{aligned} w_1(3) &= \min\{w_1(1), 1 + w_0(1)\} + \min\{w_1(5), 1 + w_0(5)\} + \min\{w_1(4), 1 + w_0(4)\} \\ &= \min\{0, 1 + \infty\} + \min\{\infty, 1 + 0\} + \min\{1, 1 + 1\} \\ w_1(3) &= 2 \end{aligned}$$

In this case the minimal weight is 2.
Applying the *OptimalTreeBinaryLabeling* for the 'B'





We have $w_1(1) = w_0(5) = w_0(2) = w_0(6) = \infty$ and $w_0(1) = w_1(5) = w_1(2) = w_1(6) = 0$ then

$$w_0(4) = \min\{w_0(2), 1 + w_1(2)\} + \min\{w_0(6), 1 + w_1(6)\} = \min\{\infty, 1 + 0\} + \min\{\infty, 1 + 0\} = 2$$

and

$$w_1(4) = \min\{w_1(2), 1 + w_0(2)\} + \min\{w_1(6), 1 + w_0(6)\} = \min\{0, 1 + \infty\} + \min\{0, 1 + \infty\} = 0$$

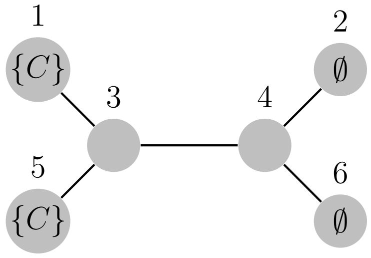
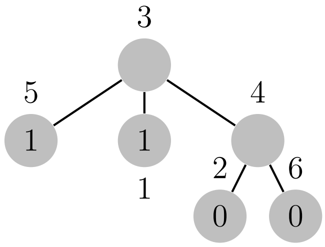
Finally

$$\begin{aligned} w_0(3) &= \min\{w_0(1), 1 + w_1(1)\} + \min\{w_0(5), 1 + w_1(5)\} + \min\{w_0(4), 1 + w_1(4)\} \\ &= \min\{0, 1 + \infty\} + \min\{\infty, 1 + 0\} + \min\{2, 1 + 0\} \\ w_0(3) &= 2 \end{aligned}$$

and

$$\begin{aligned} w_1(3) &= \min\{w_1(1), 1 + w_0(1)\} + \min\{w_1(5), 1 + w_0(5)\} + \min\{w_1(4), 1 + w_0(4)\} \\ &= \min\{\infty, 1 + 0\} + \min\{0, 1 + \infty\} + \min\{0, 2 + 1\} \\ w_1(3) &= 1 \end{aligned}$$

In this case the minimal weight is 1.
Applying the *OptimalTreeBinaryLabeling* for the 'C'

We have $w_0(1) = w_0(5) = w_1(2) = w_1(6) = \infty$ and $w_1(1) = w_1(5) = w_0(2) = w_0(6) = 0$ then

$w_0(4) = \min\{w_0(2), 1 + w_1(2)\} + \min\{w_0(6), 1 + w_1(6)\} = \min\{0, 1 + \infty\} + \min\{0, 1 + \infty\} = 0$

and

$w_1(4) = \min\{w_1(2), 1 + w_0(2)\} + \min\{w_1(6), 1 + w_0(6)\} = \min\{\infty, 1 + 0\} + \min\{\infty, 1 + 0\} = 2$

Finally

$$\begin{aligned}
w_0(3) &= \min\{w_0(1), 1 + w_1(1)\} + \min\{w_0(5), 1 + w_1(5)\} + \min\{w_0(4), 1 + w_1(4)\} \\
&= \min\{\infty, 1 + 0\} + \min\{\infty, 1 + 0\} + \min\{0, 1 + 2\} \\
w_0(3) &= 2
\end{aligned}$$

and

$$\begin{aligned}
w_1(3) &= \min\{w_1(1), 1 + w_0(1)\} + \min\{w_1(5), 1 + w_0(5)\} + \min\{w_1(4), 1 + w_0(4)\} \\
&= \min\{0, 1 + \infty\} + \min\{0, 1 + \infty\} + \min\{2, 1 + 0\} \\
w_1(3) &= 1
\end{aligned}$$

In this case the minimal weight is 1. Then the total minimum weight is $1 + 1 + 2 = 4$.

3 Tests

- Machine : youssouf-GP62-6QE .
- CPU : Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
- Limit of 30 seconds per test.

input file	weight	CPU
labeling.10	214502	1.51
labeling.1	24	0.17
labeling.2	1682	0.26
labeling.3	6936	0.43
labeling.4	12927	0.56
labeling.5	3360	0.42
labeling.6	24971	0.66
labeling.7	29938	0.65
labeling.8	43447	0.78
labeling.9	128297	1.07