Optimal mappings with minimun number of connected components in tree-to-tree comparison problems

Pascal Ferraro, Christophe Godin

An algorithm for comparing unordered tree graphs based on a minimum cost mapping with a minimal connectivity

Pascal Ferraro*and Christophe Godin†

March 19, 2001

Abstract

This paper considers the problem of finding a minimum cost mapping between two unordered trees with minimal connectivity. Based on the generalization of an algorithm for computing an edit distance between trees, the proposed algorithm solves this problem in sequential time $O(|T_1| \times |T_2| \times (\deg T_1 + \deg T_2) \times \log_2(\deg T_1 + \deg T_2))$.

1 Introduction

In this paper, we consider rooted and labeled trees (trees whose vertices are labeled). Unordered trees are trees in which the ancestor relationship is the only significant order relationship between vertices. In the following, we consider the problem of comparing unordered trees.

To compute distances between trees, edit distance metrics, initially introduced for string to string comparison problem, were first extended to compare ordered trees [14, 13] and [15, 6] for a review. Zhang recently proposed an algorithm for comparing unordered trees based on a similar framework [18, 19]. Zhang's distance is computed as the minimum cost of the mappings that maps one tree onto the other. In general, this distance is achieved by more than one mapping. Zhang's algorithm computes this distance using an arbitrary optimal mapping.

However, to apply Zhang's algorithm in the context of plant comparison [2, 3, 5], we have been led to look for particular types of optimal mappings. In this paper, we describe an extension of Zhang's algorithm that computes the optimal mapping which induces the minimal number of mapped vertex groups (connected components) over the two compared trees. The algorithm does not change the complexity of Zhang's original algorithm.

2 Definitions and notations

A *directed graph* $G = (V, E)$ consists of a set V of vertices, a set of edges E, each edge being represented by an ordered pair of vertices. The number of vertices of G is denoted by $|G|$. Let

*Department of Computer Science - University of Calgary - 2500 University Drive N.W., Calgary, Alberta, T2N 1N4 Canada. Phone: (011) (403) 220 5114. e-mail ferraro@cpsc.ucalgary.ca

†Plant Modeling Program - CIRAD, TA/40E - 34398 Montpellier Cedex 5, France. Phone: (33) 4 67 59 38 62; Fax (33) 4 67 59 38 58; e-mail godin@cirad.fr

$e = (v, w)$ be an edge in E, vertices v and w are called extremities of e, the vertex v is called the *father* of w and the vertex w is called the *son* of v. The set of sons of v is represented by $son[v]$ and the size of $son[v]$ is denoted by n_v. For any k in $\{1..n_v\}$, v_k represents a son of v. A vertex v is called an *ancestor* of an other vertex w and w is called a *descendant* of v if there exists a sequence of vertices $(x_1, x_2, ..., x_n)$, called a *path*, such that $x_1 = v$ and $x_n = w$, and for each consecutive pair of vertex (x_i, x_{i+1}), x_i is the father of x_{i+1}. The ancestor relationship is a partial order relation which is denoted by $v \leq w$. The least common ancestor of two vertices v and w, denoted by $lca(v, w)$, is a common ancestor of v and w such that every common ancestor x of v and w satisfies $x \leq lca(v, w)$. The set of descendant of v is denoted by $V[v]$ and contains v itself. A *sub-graph* $H = (W, F)$ of a directed graph $G = (V, E)$ is a directed graph such that $W \subseteq V$ and $F \subseteq E$. Let $G = (V, E)$ be a graph and W be a subset of V. Let E_W be the subset of E made of edges having both extremities in W. We define $\mathcal{G}(W) = (W, E_W)$ as *the sub-graph generated by the set of vertices* W. A directed graph is called *connected* if every pair of its vertices are connected by a path. A *connected component* of a directed graph is a connected sub-graph containing a maximum number of edges.

A labeled graph is a graph associated with a labeling function α which affects a label from a finite or infinite set $\Sigma = \{a, b, c, ...\}$ to each vertex. A distance can be defined on vertices of a labeled graph by using a distance on their labels. Let λ be a unique symbol not in Σ and let d be a distance metric which assigns a non-negative real number $d(a, b)$ to each pair of labels of $\Sigma \cup \{\lambda\}$ (a, b).

d is extended to assign to any pair of vertices a non negative real number : $d(x, y) = d(\alpha(x), \alpha(y))$. The distance between the label of x and the label λ is denoted : $d(x, \lambda) = d(\alpha(x), \lambda) = d(\lambda, \alpha(x))$.

A *rooted tree graph*, is a graph in which every vertex except one, called the *root*, has only one father vertex. The root has no father vertex. By extension, the particular graph $\theta = (\emptyset, \emptyset)$ is a tree and is called the null tree. A *sub-tree* is a connected sub-graph of a tree. An *unordered rooted* tree is a rooted tree in which the set of sons of each vertex is not ordered. An unordered rooted tree is just a rooted tree. We use the term unordered to distinguish it from ordered rooted trees. Unless otherwise stated, all trees, in this paper, are unordered labeled rooted trees. A *forest* is a directed graph whose connected components are tree graphs. In this paper, $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ represent trees, respectively rooted in v and$w$). A sub-tree of T_i $(i \in \{1, 2\})$ rooted in x is denoted by $T_i[x] = (V_i[x], E_i[x])$ and the sub-forest of T_i rooted in x is denoted by $F_i[x] = \mathcal{G}(V_i[x] \backslash \{x\})$ (The set of vertices of $F_i[x]$ is denoted by $FV_i[x]$). Particularly, $T_1[v]$ and $T_2[w]$ represent the whole trees T_1 and T_2.

3 Zhang's algorithm

A considerable amount of works has been done on ordered tree comparison. Among various tree metrics, Tai [14] and Selkow [13] proposed an edit distance metric between trees based on the generalization of string comparison defined by Wagner and Fisher [17]. In these works, trees are represented by ordered rooted tree, Zhang [18] extended these works in order to define a metric between unordered rooted trees. We have shown [6, 7] that these algorithms differs only on the constraints given in the definition of mappings and there exists a unified notation to present these methods.

The tree-to-tree correction problem consists in determining the distance between two trees measured by the minimum cost sequence of edit operations needed to transform one tree into

the other. Following Wagner and Fisher original definitions on sequences, three edit operations are used: *changing* a vertex x into a vertex y means changing the label of x, *deleting* a vertex x means making the sons of x become the sons of the father of x and then removing x, *inserting* a vertex x means that x becomes the son of a vertex y and be a subset of the sons of y becomes the set of sons of x. Following [12], we call $C(x, y)$ a changing operation, $D(x)$ an deleting operation and $I(x)$ an inserting operation.

Let s be an edit operation, a cost γ is assigned to each edit operation by letting : if s changes x into y then $\gamma(s) = \gamma(C(x, y)) = d(x, y)$, if s deletes x then $\gamma(s) = \gamma(D(x)) = d(x, \lambda)$ and if s inserts the vertex x then $\gamma(s) = \gamma(I(x)) = d(\lambda, x)$. We extend γ to a sequence of edit operation $S = (s_1, s_2, ..., s_n)$ by letting $\gamma(S) = \sum_{i=1}^{n} \gamma(s_i)$. This makes it possible to define a dissimilarity measure $D(T_1, T_2)$ from tree T_1 to tree T_2 is measured as the minimum cost of all sequences of edit operations which transforms T_1 into T_2, i.e:

$$D(T_1[v], T_2[w]) = \min\{\gamma(S);\ S \text{ is a sequence of edit operations which transforms } T_1[v] \text{ into } T_2[w]\}$$

In order to characterize the effect of an edit operation sequence on a tree, Tai [14] introduced the structure called *mapping* between trees similar to the notion of *trace* between sequences of Wagner and Fisher [17]. A mapping is intuitively a description of how a sequence of edit operations transforms T_1 into T_2, ignoring the order in which edit operations are applied. Let $T_1[v] = (V_1[v], E_1[v])$ and $T_2[w] = (V_2[w], E_2[w])$ be two trees, a *mapping* M is a set of ordered pairs of vertices (x, y) of $T_1[v]$ and $T_2[w]$.

Let M be a mapping between two trees $T_1[v]$ and $T_2[w]$, we define :

$$\begin{aligned} M_v &= \{x_1 \in V_1[v]; \exists x_2 \in V_2[w], (x_1, x_2) \in M\} \\ M_w &= \{x_2 \in V_2[w]; \exists x_1 \in V_1[v], (x_1, x_2) \in M\} \end{aligned}$$

For any vertex x in $M_v \cup M_w$, we will say that x has an image by M.

Similarly let M be a mapping from $T_1[v]$ to $T_2[w]$, $\overline{M}$ represents the set of vertices which are not mapped in M:

$$\begin{aligned} \overline{M_v} &= V_1[v] \backslash M_v \\ \overline{M_w} &= V_2[w] \backslash M_w \end{aligned}$$

$\overline{M}$ represents the union $\overline{M_v} \cup \overline{M_w}$.

Then a cost of M can be define as follows:

$$\begin{aligned} \gamma(M) &= \sum_{(x,y)\in M} d(x, y) + \sum_{x\in\overline{M_v}} d(x, \lambda) + \sum_{y\in\overline{M_w}} d(\lambda, y) \\ &= \sum_{(x,y)\in M} d(x, y) + \sum_{x\in\overline{M}} d(x, \lambda) \end{aligned}$$

Remark that this last equation is true if and only if d is a distance metric.

A *valid* mapping is a mapping that must respect additional constraints. In most tree to tree correction problem, at least two constraints must be satified:

- $x \in V_1[v]$ and $y \in V_2[w]$

- For any pair (x_1, x_2), (y_1, y_2) in M:

$$x_1 = y_1 \Leftrightarrow x_2 = y_2 \tag{C1}$$

$$x_1 \leq y_1 \Leftrightarrow x_2 \leq y_2 \tag{C2}$$

The relation between a trace and a sequence of edit operations has been shown by Wagner and Fisher [17]; Tai [14] generalized this result for mappings between trees. Given S, a sequence of edit operations from $T_1[v]$ to $T_2[w]$, there exists a mapping M from $T_1[v]$ to $T_2[w]$ such that $\gamma(M) \leq \gamma(S)$. Conversely, for any valid mapping M from $T_1[v]$ to $T_2[w]$, there exists a sequence of edit operations such that $\gamma(S) = \gamma(M)$. Based on this results it can be shown that :

$$D(T_1[v], T_2[w]) = \min\{\gamma(M);\ M \text{ is a mapping from } T_1[v] \text{ to } T_2[w]\} \tag{1}$$

Taï [14] proved these results for ordered rooted tree graphs and mapping defined by constraints (C1) and (C2). Based on the results of Kilpelainen and Mannila [11], Zhang [20] showed that finding $D(T_1[v], T_2[w])$ in case of unordered rooted tree graphs and mapping defined from constraints (C1) and (C2) is a problem MAX SNP-hard. He proved [18, 19] that the Tai's results can be generalized to unordered trees by adding a constraint (C3) to modify the definition of a valid mapping.

A *valid mapping* M is a mapping satisfying C1, C2 and C3 such that for any triple (x_1, x_2), (y_1, y_2), (z_1, z_2) in M

$$lca(x_1, y_1) < z_1 \Leftrightarrow lca(x_2, y_2) < z_2 \tag{C3}$$

The dissimilarity measure D thus defined is shown to be a distance metric [18].

Zhang proposed a dynamic programming algorithm to compute the distance $D(T_1[v], T_2[w])$:

$$\begin{array}{l} D(\theta, \theta) = 0 \\ D(F_1[v], \theta) = \sum\limits_{v_k \in son(v)} D(T_1[v_k], \theta) \quad D(T_1[v], \theta) = D(F_1[v], \theta) + d(v, \lambda) \\ D(\theta, F_2[w]) = \sum\limits_{w_k \in son(w)} D(\theta, T_2[w_k] \quad D(\theta, T_2[w]) = D(\theta, F_2[w]) + d(\lambda, w) \end{array}$$

$$D(T_1[v], T_2[w]) = \min \begin{cases} D(F_1[v], F_2[w]) + d(v, w) \\ D(\theta, T_2[w])]) + \min\limits_{w_k \in son(w)} \{D(T_1[v], T_2[w]) - D(\theta, T_2[w_k])\} \\ D(T_1[v], \theta) + \min\limits_{v_k \in son(v)} \{D(T_1[v], T_2[w]) - D(T_1[v_k], \theta)\} \end{cases}$$

$$D(F_1[v], F_2[w]) = \min \begin{cases} \min\limits_{R} \{d(R)\} \\ D(\theta, F_2[w])]) + \min\limits_{w_k \in son(w)} \{D(F_1[v], F_2[w]) - D(\theta, F_2[w_k])\} \\ D(F_1[v], \theta) + \min\limits_{v_k \in son(v)} \{D(F_1[v], F_2[w]) - D(F_1[v_k], \theta)\} \end{cases}$$

where R is a restricted mapping defining the optimum mapping between trees of two forests [19].

The distance to the null tree is introduced for following lemmas. If the valid mapping between a tree $T_1[v] = (V_1[v], E_1[v])$ and the null tree θ is represented by M, this valid mapping is obviously empty : $M = \emptyset$ and $\overline{M} = V_1[v]$. Then :

$$D(F_1[v], \theta) = \sum_{x \in V_1[v] \setminus \{v\}} d(x, \lambda) \quad D(T_1[v], \theta) = \sum_{x \in V_1[v]} d(x, \lambda)$$
$$D(\theta, F_2[w]) = \sum_{x \in V_2[w] \setminus \{w\}} d(\lambda, x) \quad D(\theta, T_2[w]) = \sum_{x \in V_2[w]} d(\lambda, x)$$

The problem of finding a restricted mapping is solved as a problem of finding a minimum cost maximum flow in a network [19]. The complexity of Zhang's algorithm is due to the minimum cost maximum flow computation. In Zhang's algorithm, this problem is solved by the Edmons and Karp's [1] algorithm improved by Tarjan [16] whose complexity is $O(m|f^*| \log_2 n)$, where m, n and $|f^*|$ represent respectively the number of edges, the number of vertices and the value of a maximum flow on the network. Finally, the complexity is bounded by $O(|T_1| \times |T_2| \times (\deg(T_1) + \deg(T_2)) \times \log_2(\deg(T_1) + \deg(T_2)))$ where $\deg(T_i)$, for any i in $\{1, 2\}$, represents the number maximum of sons for any vertex in T_i.

4 Extension of Zhang's algorithm

In general, there is not a unique optimal valid mapping corresponding to the distance $D(T_1, T_2)$ defined by equation 1. In its original form, Zhang's algorithm enables us to compute this distance by exhibiting one arbitrary optimal mapping.

To apply Zhang's algorithm in the context of comparing plants [4, 5], we have been led to look for the optimal solution for which the mapped vertices are as much grouped as possible. Figure 1 shows two optimal solutions (i.e. corresponding to the same $D(T_1, T_2)$) inducing different numbers of groups of mapped vertices on T_1 and T_2. In this paper, we describe an extension of Zhang's algorithm that computes the optimal solution which induces the minimal number of connected components over the two compared trees.

4.1 Problem position

In order to illustrate the property that there exists different optimal valid mappings, consider two trees T_1 and T_2 represented in figure 1. We compare both trees using Zhang's algorithm with a local distance defined as follows:

- insertion and deletion cost = 1
- matching cost = 0

Let us notice that the use of such a local distance relies on matching as many vertices as possible by respecting the constraints of mapping. The distance is then the number of vertices which could not be matched. We have represented on figure 1 two different valid mappings M_1 and M_2 with same cost.

A mapping M from T_1 to T_2 induces different number of groups of mapped vertices. The different groups corresponding to our example are represented by a dotted line on the figure 1. On each tree structure T_1 and T_2, an induced graph is thus defined in the following way:

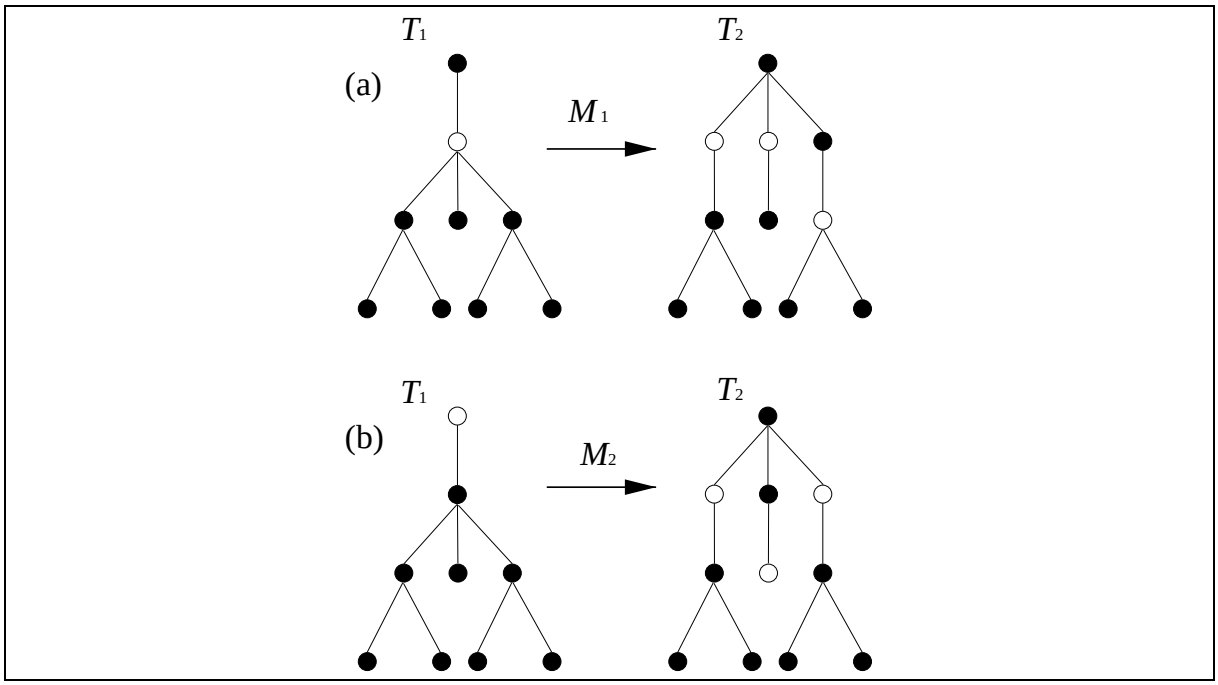

Figure 1: Two different mappings having the same cost. The mapped vertices are represented in black. (a) optimal valid mapping (cost = 4) with 9 connected components. (b) optimal valid mapping (cost = 4) with 4 connected components.

- a vertex v belongs to the induced graph if and only if v has an image by M;
- an edge (x, y) belongs to the induced graph if and only if x and y have an image by M.

The number of connected components of the induced graph on T_1 (resp. T_2) is called *the number of connected components of* M on T_1 (resp. on T_2). *The number of connected components* of M is the sum of the connected components on T_1 and T_2. The number of connected components of M is then necessarily equal to or higher than 2. In this paper, we are interested by optimal valid mappings having a minimum number of connected components. We have presented in the previous section a recursive algorithm which calculates the cost of an optimum mapping by comparing various sub-tree structures in an ascending way (from leafs to the root). We will show in the following that it is possible to follow the same recursive scheme to determine the number of connected components of optimal mapping and to calculate that which has the minimum number of connected components.

Let us illustrate the recursive calculation of the number of connected components of a valid mapping on a simple example. Consider the comparison between T_1 and T_2. Let us assume an optimal mapping M is known at some stage of the recursion for a sub-tree T rooted in v. Its number of related components is denoted by c. Let us determine the number of connected components of the mapping at the following stage, i.e. when we consider the sub-tree $T^{'}$ of T_1 rooted in $fat[v]$. $c^{'}$ denotes the number of connected components the new valid mapping $M^{'}$. Two distinct cases must be considered according to whether v has or not an image by M (figure 2) :

1. The root of T has no image :

(a) the inserted vertex has no image. Then the number of connected components does not change: $c^{'} = c$;

(b) the inserted vertex has an image then the number of connected components is increased: $c^{'} = c + 1$.

2. The root of T has an image:

(a) the inserted vertex has no image and then the number of connected components does not change: $c^{'} = c$;

(b) the inserted vertex has an image. This new vertex can be aggregate with the connected component of v, thus the number of connected components is does not change: $c^{'} = c$.

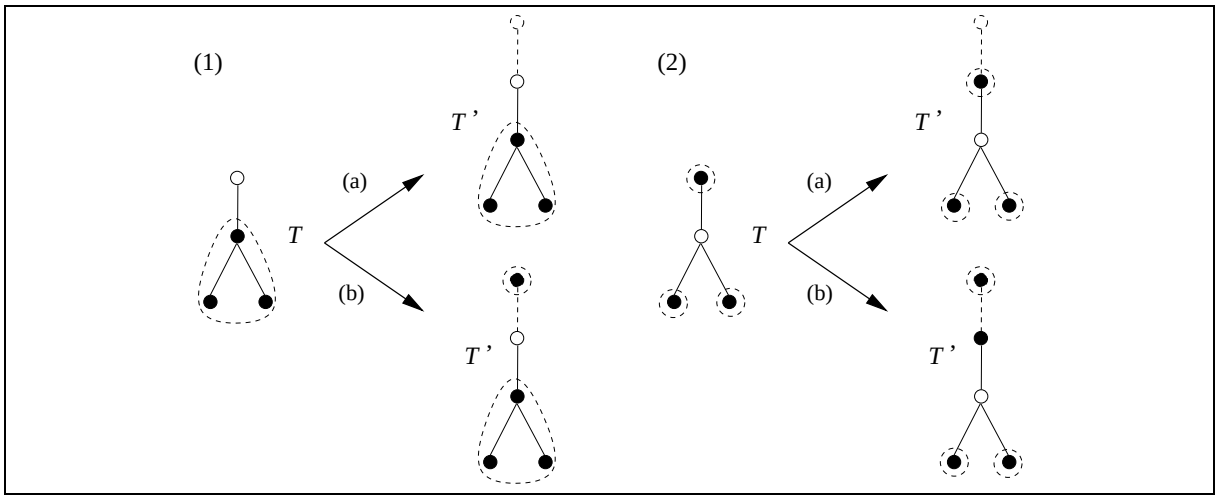

Figure 2: Recursive determination of the number of connected components : case of a tree. Vertices which have an image are represented in black. To study the connected components resulting from the addition of a root vertex to an original tree, two cases need be considered depending on whether the root of the original tree has an image or not (cases 1 and 2). Then, from these original trees, two new different cases need be considered, depending on whether the added vertex has or not an image (cases a and b). In case 1.a, 2.a and 2.b the number of connected components remains the same. In cases 1.b, the number of connected components increases.

In general, the problem can be stated as follows: knowing the mapping of a forest F of T_1, we have to compute the number of connected components. If the vertex $v^{'}$ is inserted as root of the tree structures of the forest. Then, we can again consider two cases according to whether "the roots" of F have or not an image by mapping (figure 3) :

There exists r tree graphs respectively rooted in $v_1, v_2, \ldots, v_r$ of F such that $v_1, v_2, \ldots, v_r$ have an image by the mapping:

1. the inserted vertex has no image and then the number of connected components does not change: $c^{'} = c$;

2. the inserted vertex has an image. This new vertex can then be aggregate with vertices $v_1, v_2, \ldots, v_r$ and $v^{'}$ in a unique connected component, thus the number of connected of T is : $c^{'} = c - r + 1$.

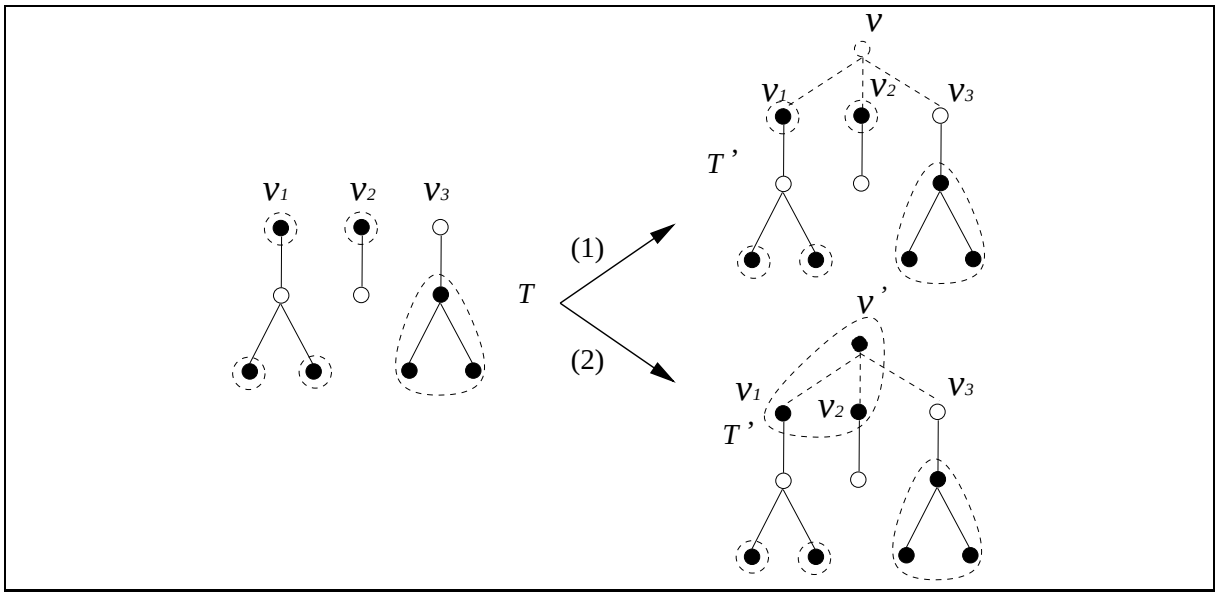

Figure 3: An example of the recursive determination of connected component number : case of a forest. Vertices which have an image are represented in black. To study the connected components resulting from the addition of a root vertex to an original forest, two cases need be considered, depending on whether the added vertex has or not an image (cases a and b). In case a, the number of connected components is not affected by the addition of v'. In case b, v_1, v_2 and $v^{'}$ can be grouped in the same connected component, which changes the overall number of connected component in the resulting tree.

In the following, based on a similar reasoning, we compute the number of connected components for any valid mapping.

4.2 Formalization

4.2.1 Notations and definitions

Let $F_1[v]$ (resp. $F_2[w]$) be a forest rooted in v (resp. w), $M \subset V_1[v] \times V_2[w]$ is a set of ordered pairs of vertices $(x, y) \in V_1[v]\backslash\{v\} \times V_2[w]\backslash\{w\}$ satisfying constraints (C1), (C2) and (C3), where $V_1[v]\backslash\{v\}$ (resp. $V_2[w]\backslash\{w\}$) represents the set $V_1[v]$ (resp. $V_2[w]$) minus $\{v\}$ (resp. $\{w\}$).

The set of valid mappings from $T_1[v]$ to $T_2[w]$ (resp. from $F_1[v]$ to $F_2[w]$) is denoted by $\mathcal{T}(v, w)$ (resp. $\mathcal{F}(v, w)$).

Let M be a valid mapping in $\mathcal{T}(v, w) \cup \mathcal{F}(v, w)$. $\mathcal{G}(M_v)$ and $\mathcal{G}(M_w)$ are the graph respectively induced by the set of vertices M_v and M_w. We denote by $c(M_v)$ (resp. $c(M_w)$) *the number of connected components of* $\mathcal{G}(M_v)$ (resp. $\mathcal{G}(M_w)$). $c(M) = c(M_v) + c(M_w)$ is called *the connected component number of* M.

Let $S[v]$ be a tree or a forest rooted at a vertex v, the set of roots of $S[v]$ is denoted by $root[v]$. Thus, for example, if $S[v]$ is a forest $F[v]$ and if $v_1, v_2, \ldots, v_p$ are the p sons of v, then $root[v] = \{v_1, v_2, \ldots, v_p\}$, and if $S[v]$ is a tree $T[v]$ then $root[v] = \{v\}$. The number of sons of v

(resp. w) which have an image by M are respectively denoted by $r(M_v)$ and $r(M_w)$:

$$\begin{aligned} r(M_v) &= |root[v] \cap M_v| \\ r(M_w) &= |root[w] \cap M_w| \end{aligned}$$

$r(M)$ represents the number of sons of v and w which have an image by M:

$$r(M) = r(M_v) + r(M_w)$$

4.3 Properties of valid mapping

To determine recursive relations that will enable us to compute an optimal valid mapping with a minimum number of connected components, we now introduce several useful properties of valid mappings.

4.3.1 Case of trees

Let $T_1[v]$ and $T_2[w]$ be two rooted trees and let M be a valid mapping from $T_1[v]$ to $T_2[w]$.

Let us define a partition of the set of valid mapping as follows: $\mathcal{T}(v, w)$:

1. $M = \emptyset$:
   - $\mathcal{T}(v, w)_{\emptyset,\emptyset} = \{M \in \mathcal{T}(v, w) \,|\, M = \emptyset\}$
2. $M \neq \emptyset$ then :
   - $\mathcal{T}(v, w)_{\subset,=} = \{M \in \mathcal{T}(v, w) \,|\, v \in M_v \text{ and } w \notin M_w\}$
   - $\mathcal{T}(v, w)_{=,\subset} = \{M \in \mathcal{T}(v, w) \,|\, v \notin M_v \text{ and } w \in M_w\}$
   - $\mathcal{T}(v, w)_{=,=} = \{M \in \mathcal{T}(v, w) \,|\, v \in M_v \text{ and } w \in M_w\}$
   - $\mathcal{T}(v, w)_{=,=} = \{M \in \mathcal{T}(v, w) \,|\, v \notin M_v \text{ and } w \notin M_w\}$

Subsequent lemmas and propositions show properties of this partition that will enable us to designthe final algorithm.

Proposition 1 *Let M be a valid mapping in $\mathcal{T}(v, w)$:*

1. *M is in $\mathcal{T}(v, w)_{\subset,=}$ if and only if there exist $w_k \in son[w]$ and $M' \in \mathcal{T}(v, w_k)$ such that $M = M'$ and:*
$$\gamma(M) = \gamma(M') + D(\theta, T_2[v]) - D(\theta, T_2[v_k])$$
2. *M is in $\mathcal{T}(v, w)_{=,\subset}$ if and only if there exist $v_k \in son[v]$ and $M' \in \mathcal{T}(v_k, w)$ such that $M = M'$ and:*
$$\gamma(M) = \gamma(M') + D(T_1[v], \theta) - D(T_1[v_k], \theta)$$
3. *M is in $\mathcal{T}(v, w)_{\subset,\subset}$ if and only if there exists $M' \in \mathcal{F}(v, w)$ such that $M = M'$ and:*
$$\gamma(M) = \gamma(M') + d(v, \lambda) + d(\lambda, w)$$

4. M is in $\mathcal{T}(v,w)_{=,=}$ if and only if $M^* = M \setminus \{(v,w)\} \in \mathcal{F}(v,w)$ and :

$$\gamma(M) = \gamma(M^*) + d(v,w)$$

Proof : The proof of this proposition is given in [19]. □

Results of this proposition are represented in figure 4.

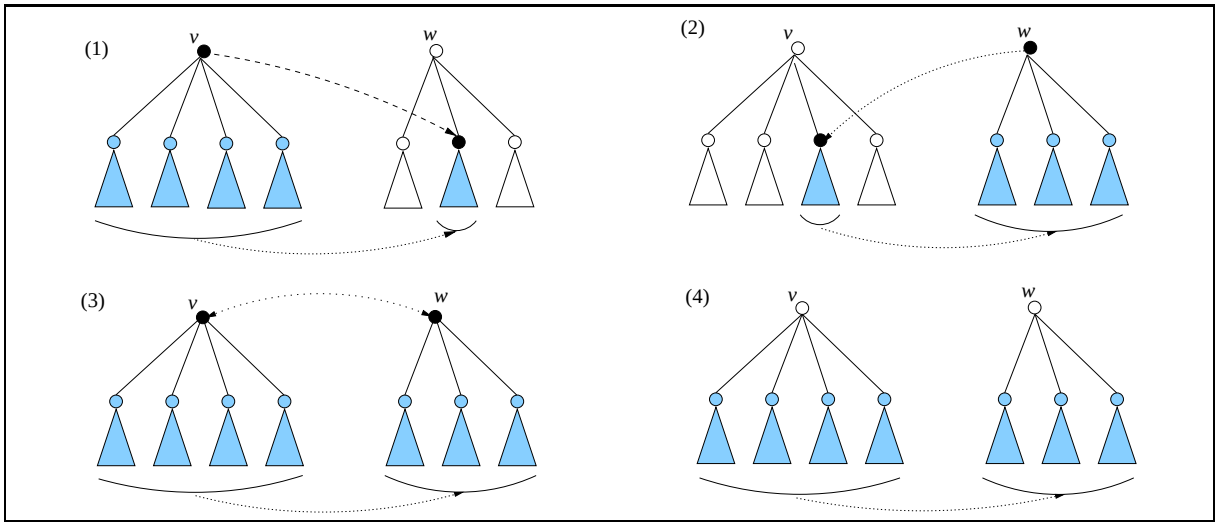

Figure 4: Recursive relations between EDMs following proposition 1. Vertices which have an image are represented in black,vertices which have no image are represented in white, and vertices in grey may have an image or not. A typical mapping in (1) $\mathcal{T}(v,w)_{\subset,=}$, (2) $\mathcal{T}(v,w)_{=,\subset}$, (3) $\mathcal{T}(v,w)_{=,=}$, (4)$\mathcal{T}(v,w)_{\subset,\subset}$

The following lemmas give recursive relations for computing $\mathcal{T}(v,w)$ and its associated quantities $r(M)$ and $c(M)$.

Lemma 1 *Let $T_1[v] = (V_1[v], E_1[v])$ and $T_2[v] = (V_2[v], E_2[v])$ be two trees, and let M be a valid mapping of $\mathcal{T}(v,w)_{\subset,=}$:*

$$\exists w_k \in son[w] \text{ and } M' \in \mathcal{T}(v,w_k) | \, M = M' \ :$$
$$\begin{aligned} c(M) &= c(M') \\ r(M_v) &= 1 \\ r(M_w) &= 0 \end{aligned}$$

Proof : According to proposition 1, for any valid mapping M in $\mathcal{T}(v,w)_{\subset,=}$, there exists w_k, a son of w and a mapping M' in $\mathcal{T}(v,w_k)$ such that $M = M'$. Since the root of T2[w] has no image, $r(M_w) = 0$. The number of connected components of M is then equal to the number of connected components from M_v onto $T_1[v]$ plus the number of connected components from M_{w_k} onto $T_2[w_k]$:

$$c(M) = c(M')$$

By definition of $\mathcal{T}(v,w)_{\subset,=}$, the vertex v has an image , then : $r(M_v) = 1$ (see figure 2).

□

Lemma 2 *Let $T_1[v] = (V_1[v], E_1[v])$ and $T_2[v] = (V_2[v], E_2[v])$ be two trees, and let M be a valid mapping in $\mathcal{T}(v,w)_{=,\subset}$:*

$$\exists w_k \in son[w] \text{ and } M' \in \mathcal{T}(v_k, w) \,|\, M = M' \; : \quad \begin{aligned} c(M) &= c(M') \\ r(M_v) &= 0 \\ r(M_w) &= 1 \end{aligned}$$

Proof : This is the symmetric case of lemma 1 □

Lemma 3 *Let $T_1[v] = (V_1[v], E_1[v])$ and $T_2[v] = (V_2[v], E_2[v])$ be two trees, and let M be a valid mapping in $\mathcal{T}(v,w)_{=,=}$:*

$$\exists M^* \in \mathcal{F}(v,w) \,|\, M = M^* \cup \{(v,w)\} : \quad \begin{aligned} c(M) &= c(M^*) - r(M^*) + 2 \\ r(M_v) &= 1 \\ r(M_w) &= 1 \end{aligned}$$

Proof : According to the definition of $\mathcal{T}(v,w)_{=,=}$, both v and w have an image by M, then:

$$\begin{aligned} r(M_v) &= 1 \\ r(M_w) &= 1 \end{aligned}$$

According to proposition 1, for any valid mapping M in $\mathcal{T}(v,w)_{=,=}$, there exists a valid mapping M^* in $\mathcal{F}(v,w)$ such that $M = M^* \cup \{(v,w)\}$. Since v has an image, v can be grouped with the $r(M_v^*)$ roots of the trees of $F_1[v]$ which have an image by M^* in a single connected component (see figure 3), therefore:

$$c(M_v) = c(M_v^*) - r(M_v^*) + 1$$

We can establish the same result for $c(M_w)$, thus:

$$\begin{aligned} c(M) &= c(M_v) + c(M_w) \\ &= c(M_v^*) - r(M_v^*) + 1 + c(M_w^*) - r(M_w^*) + 1 \\ &= c(M^*) - r(M^*) + 2 \end{aligned}$$

□

Let M be a mapping of $\mathcal{T}(v,w)_{\subset,\subset}$, and let M' be the maping : $M \cup \{(v,w)\}$ (M' is obviously a valid mapping of $\mathcal{T}(v,w)_{=,=}$). Furthermore $\gamma(M) - \gamma(M') = d(v,\lambda) + d(\lambda,w) - d(v,w)$, and then using the triangular inequality of d, $\gamma(M) \geq \gamma(M')$. The cost of a mapping M of $\mathcal{T}(v,w)_{\subset,\subset}$ is always higher than the cost of mapping $M \cup \{(v,w)\}$ of $\mathcal{T}(v,w)_{=,=}$. It is thus not necessary to compute the number of connected component of these sub-optimal mapping.

In the previous lemma, the computation of the number of connected components of a mapping between forests appears as $c(M^*)$. This computation is studied in the next section.

4.3.2 Case of forests

Let $F_1[v]$ and $F_2[w]$ be two rooted forests and let M be a valid mapping from $F_1[v]$ to $F_2[w]$. We define a partition of the set of valid mapping in three sets:

- $\mathcal{F}(v,w)_{\subset,=} = \{M \in \mathcal{F}(v,w) \,|\, \forall (x,y) \in M y < w\}$
- $\mathcal{F}(v,w)_{=,\subset} = \{M \in \mathcal{F}(v,w) \,|\, \forall (x,y) \in M y < w\}$
- $\mathcal{F}(v,w)_{=,=} = \mathcal{F}(v,w) \cap (\mathcal{F}(v,w)_{\subset,=} \cup \mathcal{F}(v,w)_{=,\subset})$

We can easily remark that these subsets form a partition of $\mathcal{F}(v,w)$.

In the following, I and J denotes respectively the set of indexes of the sons of v and w : $I = \{1 \dots n\}$ and $J = \{1 \dots m\}$

Proposition 2 *Let M be a valid mapping in $\mathcal{F}(v,w)$:*

1. *M is $\mathcal{F}_{\subset,=}(v,w)$ if and only if there exist $w_k \in son[w]$ and $M' \in \mathcal{F}(v,w_k)$ such that $M = M'$ and:*
$$\gamma(M) = \gamma(M') + D(\theta, F_2[v]) - D(\theta, F_2[v_k])$$
2. *M is $\mathcal{F}_{=,\subset}(v,w)$ if and only if there exist $v_k \in son[v]$ and $M' \in \mathcal{F}(v_k,w)$ such that $M = M'$ and:*
$$\gamma(M) = \gamma(M') + D(F_1[v], \theta) - D(F_1[v_k], \theta)$$
3. *M is $\mathcal{F}_{=,=}(v,w)$ if and only if there exists a mapping K in $I \times J$ and a partition $(M^{k,l})_{(k,l)\in K}$ of M where $(M^{k,l}) \in \mathcal{T}(v_k, w_l)$ such that : $M = \bigcup_{(k,l)\in K} \left(M^{k,l}\right)$, and:*
$$\gamma(M) = \sum_{(k,l)\in K} \gamma(M^{k,l})$$
4. *$M = \emptyset$ and:*
$$\gamma(M) = D(F_1[v], \theta) + D(\theta, F_2[v])$$

Proof: This result is a direct consequence of the definition of valid mapping between forests and the previous partition [19, 6]. These results are represented by the figure 5. □

The following lemma gives recursive relations for computing $\mathcal{F}(v,w)$ and its associated quantities $r(M)$ and $c(M)$.

Lemma 4 *Let $F_1[v] = (V_1[v], E_1[v])$ and $F_2[v] = (V_2[v], E_2[v])$ be two forests, and let M be a valid mapping of $\mathcal{F}(v,w)_{\subset,=}$:*

$$\exists w_k \in son[w] \text{ and } M' \in \mathcal{F}(v,w_k) \,|\, M = M'$$
$$\begin{aligned} c(M) &= c(M') \\ r(M_w) &= 0 \\ r(M_v) &= r(M'_v) \end{aligned}$$

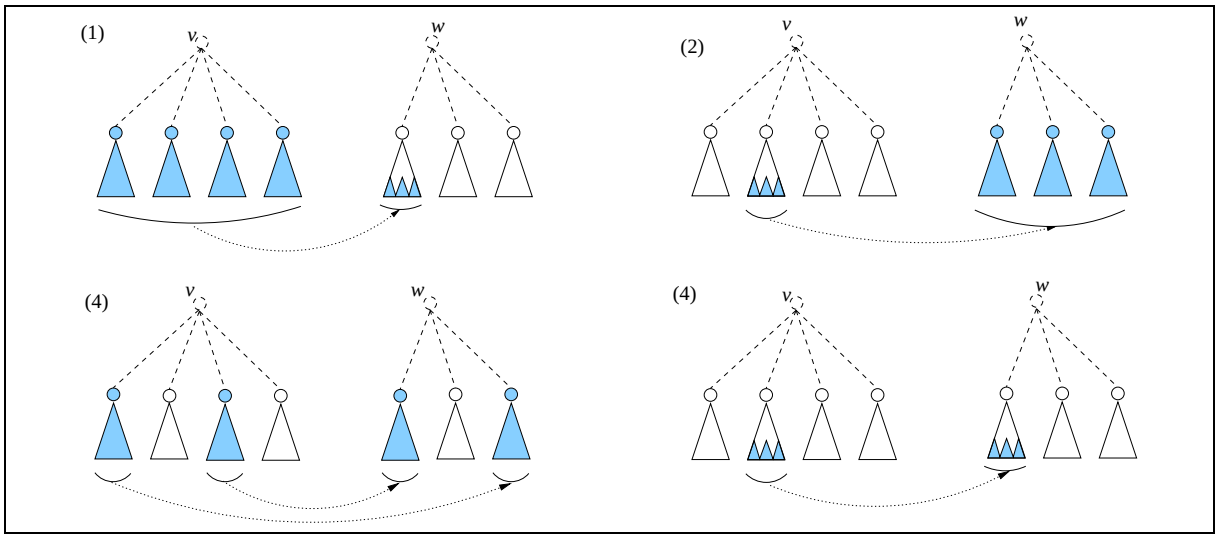

Figure 5: Recursive relations between EDMs following proposition 2. A typical mapping in (1) $\mathcal{T}(v,w)_{\subset,=}$, (2) $\mathcal{T}(v,w)_{=,\subset}$, (3) $\mathcal{T}(v,w)_{=,=}$, (4)$\mathcal{T}(v,w)_{\subset,\subset}$. Vertices which have no image are represented in white color. Vertices which can have or not an image by the mapping are represented in grey color.

Proof : According to proposition 2, for any valid mapping M in $\mathcal{F}(v,w)_{\subset,=}$, there exists w_k son of w and $M^{'}$ a mapping in $\mathcal{F}(v,w_k)$ such that $M = M^{'}$. In $\mathcal{F}(v,w)_{\subset,=}$, neither w nor any of its sons can have an image and thus: $r(M_w) = 0$. Similarly the number of roots of $F_1[v]$ which have an image by M is necessarily equal to $r(M^{'}_v)$. Furthermore, the number of connected components of M is equal to the number of connected components from $M^{'}_v$ in $F_1[v]$ plus the number of connected components from $M^{'}_{w_k}$ in $F_2[w_k]$:

$$c(M) \quad = \quad c(M^{'})$$

□

Lemma 5 *Let $F_1[v] = (V_1[v], E_1[v])$ and $F_2[v] = (V_2[v], E_2[v])$ be two trees, and let M be a valid mapping in $\mathcal{F}(v,w)_{=,\subset}$:*

$$\exists v_k \in son[v] \,|\, \exists M^{'} \in \mathcal{F}(v_k, w) \,|\, M = M^{'}$$
$$\begin{aligned} c(M) &= c(M^{'}) \\ r(M_v) &= 0 \\ r(M_w) &= r(M^{'}_w) \end{aligned}$$

Proof : This is the symmetric case of lemma 4 □

Lemma 6 *Let $F_1[v] = (V_1[v], E_1[v])$ and $F_2[v] = (V_2[v], E_2[v])$ be two forests, and let M be a valid mapping in $\mathcal{F}_{=,=}(v,w)$, then there exists a mapping K in $I \times J$ and a partition $(M^{k,l})_{(k,l)\in K}$*

of M where $(M^{k,l}) \in \mathcal{T}(v_k, w_l)$ such that :

$$\begin{aligned} c(M) &= \sum_{(k,l)\in K} c(M^{k,l}) \\ r(M_v) &= \sum_{(k,l)\in K} r(M^{k,l}_{v_k}) \\ r(M_w) &= \sum_{(k,l)\in K} r(M^{k,l}_{w_l}) \end{aligned}$$

Proof : According to Zhang [19], if M is in $\mathcal{F}_{=,=}(v,w)$ there exists a mapping K in $I \times J$ and a partition $(M^{k,l})_{(k,l)\in K}$ of M where $(M^{k,l}) \in \mathcal{T}(v_k, w_l)$ such that : $M = \bigcup_{(k,l)\in K} \left(M^{k,l}\right)$. Then the number of connected components of M is obviously equal to $\sum_{(k,l)\in K} c(M^{k,l})$. Furthermore if the root v_k of $T_1[v_k]$ has an image by $M^{k,l}$ (this means that $v_k \in M^{k,l}_{v_k}$), then v_k has an image by M (this means that $v_k \in M_v$) and finally:

$$r(M_v) = \sum_{(k,l)\in K} r(M^{k,l}_{v_k})$$

The computation of $r(M_w)$ is done similarly. □

5 Recursive computation of the optimal valid mapping

We propose in this section an algorithm that determines the optimal valid mapping with a minimum number of connected components.

In order to determine such a particular optimal valid mapping, we need to introduce for any valid mapping M, the following quantity: $\overrightarrow{\Gamma}(M) = \begin{bmatrix} \Gamma(M) \\ c(M) \end{bmatrix}$. The vector $\overrightarrow{\Gamma}(M)$ is called *cost vector of the mapping M*. The set of pairs of $\mathbb{R}^2$ is ordered in a lexicographic order:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} < \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \Leftrightarrow (x_1 < x_2) \text{ or } (x_1 = x_2 \text{ and } y_1 < y_2)$$

In this way, a pair X with lower cost than Y will always be considered as lower than Y in $\mathbb{R}^2$, whatever its number of connected components defined in its second coordinate. However, if pairs X and Y have identical cost, the lowest pair is the one with minimum number of connected components.

Using this order, we are looking for the valid mapping M verifying:

$$\overrightarrow{D}(T_1[v], T_2[w]) = \min_{M\in\mathcal{T}(v,w)} \left\{ \overrightarrow{\Gamma}(M) \right\}$$

In order to give a recursive relation to compute $\overrightarrow{D}(T_1[v], T_2[w])$ in the following lemma, we need to introduce the intermediate quantity: $\overrightarrow{\Gamma'}(M) = \begin{bmatrix} \Gamma(M) \\ c(M) - r(M) \end{bmatrix}$.

$\overrightarrow{D'}(F_1[v], F_2[w])$ denotes the minimum of this quantity on $\mathcal{F}(v,w)$:

$$\overrightarrow{D'}(F_1[v], F_2[w]) = \min_{M\in\mathcal{F}(v,w)} \{\overrightarrow{\Gamma'}(M)\}$$

Proposition 3 $\overrightarrow{D}(T_1[v], T_2[w])$ *is recursively computed using the following relation:*

$$\overrightarrow{D}(T_1[v], T_2[w]) = \min \begin{cases} \overrightarrow{D'}(F_1[v], F_2[w]) + \begin{bmatrix} d(v,w) \\ 2 \end{bmatrix} \\ \overrightarrow{D}(\theta, T_2[w]) + \min_{w_k \in son(w)} \left\{ \overrightarrow{D}(T_1[v], T_2[w_k]) - \overrightarrow{D}(\theta, T_2[w_k]) \right\} \\ \overrightarrow{D}(T_1[v], \theta) + \min_{v_k \in son(v)} \left\{ \overrightarrow{D}(T_1[v_k], T_2[w]) - \overrightarrow{D}(T_1[v_k], \theta \right\} \end{cases}$$

Proof : Using lemmas 1, 2, 3 and proposition 1, we deduce that if M is in $\mathcal{T}(v,w)$ then M verifies one and only one of the following assertions:

1. $\exists w_k \in son[w]$ and $M^{'} \in \mathcal{T}(v, w_k)$ such that $M = M^{'}$:

$$\begin{aligned} \overrightarrow{\Gamma}(M) &= \begin{bmatrix} \gamma(M^{'}) + D(\theta, T_2[w]) - D(\theta, T_2[w_k]) \\ c(M^{'}) \end{bmatrix} \\ &= \begin{bmatrix} \gamma(M^{'}) \\ c(M^{'}) \end{bmatrix} + \begin{bmatrix} D(\theta, T_2[w]) \\ 0 \end{bmatrix} - \begin{bmatrix} D(\theta, T_2[w_k]) \\ 0 \end{bmatrix} \\ &= \overrightarrow{\Gamma}(M^{'}) + \overrightarrow{D}(\theta, T_2[w]) - \overrightarrow{D}(\theta, T_2[w_k]) \end{aligned}$$

2. $\exists v_k \in son[v]$ and $M^{'} \in \mathcal{T}(v_k, w)$ such that $M = M^{'}$:

$$\begin{aligned} \overrightarrow{\Gamma}(M) &= \begin{bmatrix} \gamma(M^{'}) + D(T_1[v], \theta) - D(T_1[v_k], \theta) \\ c(M^{'}) \end{bmatrix} \\ &= \overrightarrow{\Gamma}(M^{'}) + \overrightarrow{D}(T_1[v], \theta) - \overrightarrow{D}(T_1[v_k], \theta) \end{aligned}$$

3. $(v,w) \in M$ and $M^* \in \mathcal{F}(v,w)$

$$\begin{aligned} \overrightarrow{\Gamma}(M) &= \begin{bmatrix} \gamma(M^*) + d(v,w) \\ c(M^*) - r(M^*) + 2 \end{bmatrix} \\ &= \overrightarrow{\Gamma'}(M^*) + \begin{bmatrix} d(v,w) \\ 2 \end{bmatrix} \end{aligned}$$

4. $M \in \mathcal{F}(v,w)$ and M is necessarily the cost of M is sub-optimal.

This is the results of our proposition. □

>From this last result, in order to determine $\overrightarrow{D}(T_1[v], T_2(w))$, we need to compute

$$\overrightarrow{D'}(F_1[v], F_2[w]) = \min_{M \in \mathcal{F}(v,w)} \{\overrightarrow{\Gamma'}(M)\}$$

For any valid mapping M, the following quantities are introduced:

$$\overrightarrow{\Gamma^v}(M) = \begin{bmatrix} \gamma(M) \\ c(M) - r(M_v) \end{bmatrix} \text{ and } \overrightarrow{\Gamma^w}(M) = \begin{bmatrix} \gamma(M) \\ c(M) - r(M_w) \end{bmatrix}$$

The minimum of these quantities on $\mathcal{T}(v,w)$ and $\mathcal{F}(v,w)$ are denoted as follow:

$$\begin{aligned}
\overrightarrow{D^{\prime}}(T_1[v],T_2[w]) &= \min_{M\in\mathcal{T}(v,w)}\{\overrightarrow{\Gamma^{\prime}}(M)\}\\
\overrightarrow{D^{w}}(T_1[v],T_2[w]) &= \min_{M\in\mathcal{T}(v,w)}\{\overrightarrow{\Gamma^{w}}(M)\}\\
\overrightarrow{D^{v}}(F_1[v],F_2[w]) &= \min_{M\in\mathcal{F}(v,w)}\{\overrightarrow{\Gamma^{v}}(M)\}\\
\overrightarrow{D^{w}}(F_1[v],F_2[w]) &= \min_{M\in\mathcal{F}(v,w)}\{\overrightarrow{\Gamma^{w}}(M)\}
\end{aligned}$$

Proposition 4 *Quantities* $\overrightarrow{D^{\prime}}(F_1[v],F_2[w])$, $\overrightarrow{D^{v}}(F_1[v],F_2[w])$, $\overrightarrow{D^{w}}(F_1[v],F_2[w])$ *and* $\overrightarrow{D}(F_1[v],F_2[w])$ *are recursively computed using the following relations:*

$$\overrightarrow{D^{\prime}}(F_1[v],F_2[w]) = \min\left\{\begin{array}{l}
\overrightarrow{D}(F_1[v],\theta)+\min_{v_k\in son(v)}\left\{\overrightarrow{D^{w}}(F_1[v_k],F_2[w])-\overrightarrow{D}(F_1[v_k],\theta)\right\}\\
\overrightarrow{D}(\theta,F_2[w])+\min_{w_k\in son(w)}\left\{\overrightarrow{D^{v}}(F_1[v],F_2[w_k])-\overrightarrow{D}(\theta,F_2[w_k])\right\}\\
\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^{\prime}}(M)\right\}
\end{array}\right.$$

$$\overrightarrow{D^{v}}(F_1[v],F_2[w]) = \min\left\{\begin{array}{l}
\overrightarrow{D}(F_1[v],\theta)+\min_{v_k\in son(v)}\left\{\overrightarrow{D}(F_1[v_k],F_2[w])-\overrightarrow{D}(F_1[v_k],\theta)\right\}\\
\overrightarrow{D}(\theta,F_2[w])+\min_{w_k\in son(w)}\left\{\overrightarrow{D^{v}}(F_1[v],F_2[w_k])-\overrightarrow{D}(\theta,F_2[w_k])\right\}\\
\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^{v}}(M)\right\}
\end{array}\right.$$

$$\overrightarrow{D^{w}}(F_1[v],F_2[w]) = \min\left\{\begin{array}{l}
\overrightarrow{D}(F_1[v],\theta)+\min_{v_k\in son(v)}\left\{\overrightarrow{D^{w}}(F_1[v_k],F_2[w])-\overrightarrow{D}(F_1[v_k],\theta)\right\}\\
\overrightarrow{D}(\theta,F_2[w])+\min_{w_k\in son(w)}\left\{\overrightarrow{D}(F_1[v],F_2[w_k])-\overrightarrow{D}(\theta,F_2[w_k])\right\}\\
\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^{w}}(M)\right\}
\end{array}\right.$$

$$\overrightarrow{D}(F_1[v],F_2[w]) = \min\left\{\begin{array}{l}
\overrightarrow{D}(F_1[v],\theta)+\min_{v_k\in son(v)}\left\{\overrightarrow{D}(F_1[v_k],F_2[w])-\overrightarrow{D}(F_1[v_k],\theta)\right\}\\
\overrightarrow{D}(\theta,F_2[w])+\min_{w_k\in son(w)}\left\{\overrightarrow{D}(F_1[v],F_2[w_k])-\overrightarrow{D}(\theta,F_2[w_k])\right\}\\
\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma}(M)\right\}
\end{array}\right.$$

Proof : Using lemmas 4, 5 and 6 and proposition 2, we deduce that if M is in $\mathcal{F}(v,w)$ then M verifies one and only one of the following assertions:

1. $\exists w_k \in son[w]$ and $M^{'} \in \mathcal{F}(v,w_k)$ such that $M = M^{'}$. In this case, $r(M_v) = r(M_v^{'})$ and $r(M_w) = 0$:

$$\begin{aligned}
\overrightarrow{\Gamma}(M) &= \left[\begin{array}{c}\gamma(M^{'})+D(\theta,F_2[w])-D(\theta,F_2[w_k])\\ c(M^{'}-r(M_v^{'}))\end{array}\right]\\
&= \left[\begin{array}{c}\gamma(M^{'})\\ c(M^{'}-r(M_v^{'}))\end{array}\right]+\left[\begin{array}{c}D(\theta,F_2[w])\\ 0\end{array}\right]-\left[\begin{array}{c}D(\theta,F_2[w_k])\\ 0\end{array}\right]\\
&= \overrightarrow{\Gamma^{v}}(M^{'})+\overrightarrow{D}(\theta,F_2[w])-\overrightarrow{D}(\theta,F_2[w_k])
\end{aligned}$$

2. $\exists v_k \in son[v]$ and $M^{'} \in \mathcal{F}(v_k, w)$ such that $M = M^{'}$. In this case, $r(M_w) = r(M^{'}_w)$ and $r(M_v) = 0$:

$$\begin{aligned}\overrightarrow{\Gamma}(M) &= \begin{bmatrix} \gamma(M^{'}) + D(F_1[v],\theta) - D(F_1[v_k],\theta) \\ c(M^{'} - r(M^{'}_w)) \end{bmatrix} \\ &= \overrightarrow{\Gamma^w}(M^{'}) + \overrightarrow{D}(F_1[v],\theta) - \overrightarrow{D}(F_1[v_k],\theta)\end{aligned}$$

The computation of the minimum on $\mathcal{F}(v,w)$ gives the results of our proposition. The intermediate quantities $\overrightarrow{D^{'}}(F_1[v],F_2[w])$, $\overrightarrow{D^{t}}(F_1[v],F_2[w])$, $\overrightarrow{D^{w}}(F_1[v],F_2[w])$ and $\overrightarrow{D}(F_1[v],F_2[w])$ calculated in a similar fashion. □

The method for computing quantities $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma}(M)\right\}$, $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^t}(M)\right\}$, $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^w}(M)\right\}$ and $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^{'}}(M)\right\}$ is given in the following section. These computation is based on the computation of $\overrightarrow{D^{'}}(T_1[v],T_2[w])$, $\overrightarrow{D^{t}}(T_1[v],T_2[w])$, $\overrightarrow{D^{w}}(T_1[v],T_2[w])$ and $\overrightarrow{D}(T_1[v],T_2[w])$ calculated as follow.

Proposition 5 $\overrightarrow{D^{'}}(T_1[v],T_2[w])$ *is recursively computed using the following relation:*

$$\overrightarrow{D^{'}}(T_1[v],T_2[w]) = \min\begin{cases} \overrightarrow{D^{'}}(F_1[v],F_2[w]) + \begin{bmatrix} d(v,w) \\ 0 \end{bmatrix} \\ \overrightarrow{D}(\theta,T_2[w]) + \min_{w_k\in son(w)}\left\{\overrightarrow{D}(T_1[v],T_2[w_k]) - \overrightarrow{D}(\theta,T_2[w_k])\right\} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \overrightarrow{D}(T_1[v],\theta) + \min_{v_k\in son(v)}\left\{\overrightarrow{D}(T_1[v_k],T_2[w]) - \overrightarrow{D}(T_1[v_k],\theta\right\} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{cases}$$

$$\overrightarrow{D^{t}}(T_1[v],T_2[w]) = \min\begin{cases} \overrightarrow{D^{'}}(F_1[v],F_2[w]) + \begin{bmatrix} d(v,w) \\ 1 \end{bmatrix} \\ \overrightarrow{D}(\theta,T_2[w]) + \min_{w_k\in son(w)}\left\{\overrightarrow{D}(T_1[v],T_2[w_k]) - \overrightarrow{D}(\theta,T_2[w_k])\right\} \\ \overrightarrow{D}(T_1[v],\theta) + \min_{v_k\in son(v)}\left\{\overrightarrow{D}(T_1[v_k],T_2[w]) - \overrightarrow{D}(T_1[v_k],\theta\right\} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{cases}$$

$$\overrightarrow{D^{w}}(T_1[v],T_2[w]) = \min\begin{cases} \overrightarrow{D^{'}}(F_1[v],F_2[w]) + \begin{bmatrix} d(v,w) \\ 1 \end{bmatrix} \\ \overrightarrow{D}(\theta,T_2[w]) + \min_{w_k\in son(w)}\left\{\overrightarrow{D}(T_1[v],T_2[w_k]) - \overrightarrow{D}(\theta,T_2[w_k])\right\} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \overrightarrow{D}(T_1[v],\theta) + \min_{v_k\in son(v)}\left\{\overrightarrow{D}(T_1[v_k],T_2[w]) - \overrightarrow{D}(T_1[v_k],\theta\right\} \end{cases}$$

Proof : Similar to proposition 3 □

6 Restricted Mapping

In prop 3 and 4, all the quantities cab be recursively evaluated except for $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^{'}}(M)\right\}$, $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^t}(M)\right\}$, $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma^w}(M)\right\}$ et $\min_{M\in\mathcal{F}_{=,=}(v,w)}\left\{\overrightarrow{\Gamma}(M)\right\}$. These quantities need to be computed separetly, using a special scheme.

A solution for computing $\min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma'}(M) \right\}$. Other similar quantities can be computed likewise. As shown by Zhang [18], this optimization problem can be modeled as a minimum cost maximum bipartite matching problemand solved using aweighted maximum matching algorithm [1]. However the cost used in our case (section 5.) is not a real non-negative number but is defined on $\mathbb{R}^2$. To extend the minimum cost maximum bipartite matching problem to costs in $\mathbb{R}^2$ we need to define a total order on $\mathbb{R}^2$. As in the previous section, we use for this purpose a lexicographic order, and each cost is greater than $(0, 0)$.

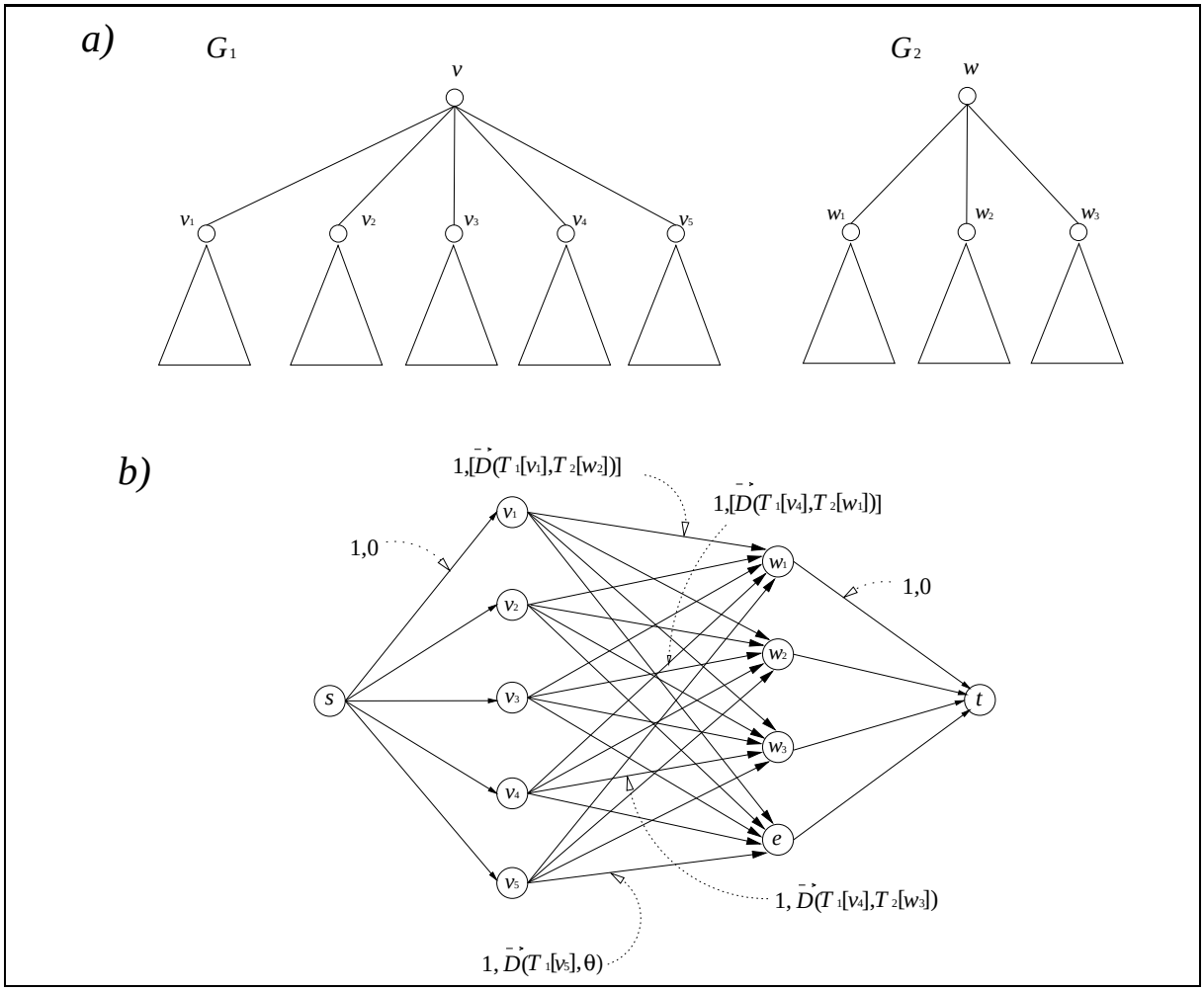

Figure 6: Restricted mapping and modelisation as a network flow

Given $F_1[v]$ and $F_2[w]$, we construct a graph $G(v, w) = (V, E)$ as follows (figure 6:

1. *vertex set* : $V = \{s, t, e_v, e_w\} \cup son[v] \cup son[w]$, where s is the source, t is the sink, and e_v and e_w represent two empty trees;

2. *edge set* : (s, v_k), (s, e_v), (e_w, t), (w_l, t) with a cost $(0, 0)$, (v_k, w_l) with cost $\overrightarrow{D}(T_1[v_k], T_2[w_l])$, (v_k, e_v) with cost $\overrightarrow{D}(T_1(v_k), \theta)$, (e_w, w_l) with cost $\overrightarrow{D}(\theta, T_2(w_l))$, and (e_v, e_w) with cost $(0, 0)$. All the edges have capacity one except (s, e_v), (e_v, e_w) and (e_w, t) which capacities are n_v, $\max\{n_v, n_w\} - \min\{n_v, n_w\}$, and n_w, respectively.

G is a graph whose edges are labeled with integer capacities, non-negative costs in R^2, and the maximum flow $f^* = n_v + n_w$ [19].

When the cost is a real, Zhang [19] showed that the cost of the minimum cost maximum flow the is exactly $\min_{M \in \mathcal{F}_{=,=}(v,w)} \{\overrightarrow{\gamma}(M)\}$. In our case, using a similar scheme we have shown [5] that the cost of the minimum cost maximum flow the is exactly $\min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{\overrightarrow{\Gamma'}(M)\right\}$. The computation of the other quantities is similar and used the same modeling by the minimum cost maximum bipartite matching problem.

7 Algorithm and Complexity

The detailed algorithm is depicted in algorithms 7.1 and 7.2. The general structure of the algorithm is similar to that of Zhang's algorithm except that at each step of the recursion, additional tests are performed which enable us to choose one solution with minimum number of connected components when several equivalent optimal solutions are available.

Algorithm 7.1 Initialization: computation of the distances between trees and empty tree

$\overrightarrow{D}(\theta, \theta) = 0$
$\overrightarrow{D^v}(\theta, \theta) = 0$
$\overrightarrow{D^w}(\theta, \theta) = 0$
$\overrightarrow{D'}(\theta, \theta) = 0$
`For any` $x \in T_1$`, compute`

$$\overrightarrow{D}(F_1[x], \theta) = \sum_{x_k \in son(x)} \overrightarrow{D}(T_1[x_k], \theta) \quad \text{and} \quad \overrightarrow{D}(F_1[x], \theta) = \overrightarrow{D}(F_1[x], \theta) + (d(x, \lambda), 0)$$

`For any` $y \in T_2$`, compute`

$$\overrightarrow{D}(\theta, F_2[y]) = \sum_{y_k \in son(y)} D(\theta, T_2[y_k]) \quad \text{and} \quad \overrightarrow{D}(\theta, F_2[y]) = D(\theta, F_2[y]) + (d(\lambda, y), 0)$$

The final time complexity is thus that of Zhang's algorithm [19]:

$$O(|T_1| \times |T_2| \times (\deg T_1 + \deg T_2) \times \log_2(\deg T_1 + \deg T_2))$$

8 Conclusion

Using the definition of a distance metric between unordered labeled trees proposed by Zhang, we have presented an algorithm for computing an optimal matching between rooted trees minimizing the number of connected components induced by the mapping. This algorithm extends Zhang's algorithm by proposing a characterization of an optimal mapping. This algorithm does not increase Zhang's algorithm complexity and computes a mapping with the minimum cost and having a minimum connectivity.

Algorithm 7.2 Optimal valid mapping with a minimum number of connected components

`Initialization algorithm.`

`For any` $v \in T_1$ `do and For any` $w \in T_2$ `do:`

`(A) Computation of restricted mappings:`

$$\min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma}(M) \right\} \quad - \quad \min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma^v}(M) \right\}$$

$$\min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma^w}(M) \right\} \quad - \quad \min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma'}(M) \right\}$$

`(B) First optimization step`

$$\overrightarrow{D}(F_1[v], F_2[w]) = \min \begin{cases} \overrightarrow{D}(\theta, F_2(w)) + \min_{w_k \in son(w)} \left\{ \overrightarrow{D}(F_1[v], F_2[w_k]) - \overrightarrow{D}(\theta, F_2(w_k)) \right\} \\ \overrightarrow{D}(F_1(v), \theta) + \min_{v_k \in son(v)} \left\{ \overrightarrow{D}(F_1[v_k], F_2[w]) - \overrightarrow{D}(F_1(v_k), \theta) \right\} \\ \min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma}(M) \right\} \end{cases}$$

`(C) Intermediate optimization step using step (B)`

$$\overrightarrow{D^v}(F_1[v], F_2[w]) = \min \begin{cases} \overrightarrow{D}(\theta, F_2(w)) + \min_{w_k \in son(w)} \left\{ \overrightarrow{D^v}(F_1[v], F_2[w_k]) - \overrightarrow{D}(\theta, F_2(w_k)) \right\} \\ \overrightarrow{D}(F_1(v), \theta) + \min_{v_k \in son(v)} \left\{ \overrightarrow{D}(F_1[v_k], F_2[w]) - \overrightarrow{D}(F_1(v_k), \theta) \right\} \\ \min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma v}(M) \right\} \end{cases}$$

$$\overrightarrow{D^w}(F_1[v], F_2[w]) = \min \begin{cases} \overrightarrow{D}(\theta, F_2(w)) + \min_{w_k \in son(w)} \left\{ \overrightarrow{D}(F_1[v], F_2[w_k]) - \overrightarrow{D}(\theta, F_2(w_k)) \right\} \\ \overrightarrow{D}(F_1(v), \theta) + \min_{v_k \in son(v)} \left\{ \overrightarrow{D^w}(F_1[v_k], F_2[w]) - \overrightarrow{D}(F_1(v_k), \theta) \right\} \\ \min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma^w}(M) \right\} \end{cases}$$

`(D) Intermediate optimization step using step (C)`

$$\overrightarrow{D'}(F_1[v], F_2[w]) = \min \begin{cases} \overrightarrow{D}(\theta, F_2(w)) + \min_{w_k \in son(w)} \left\{ \overrightarrow{D^v}(F_1[v], F_2[w_k]) - \overrightarrow{D}(\theta, F_2(w_k)) \right\} \\ \overrightarrow{D}(F_1(v), \theta) + \min_{v_k \in son(v)} \left\{ \overrightarrow{D^w}(F_1[v_k], F_2[w]) - \overrightarrow{D}(F_1(v_k), \theta) \right\} \\ \min_{M \in \mathcal{F}_{=,=}(v,w)} \left\{ \overrightarrow{\Gamma'}(M) \right\} \end{cases}$$

`(E) Main optimization`

$$\overrightarrow{D}(T_1[v], T_2[w]) = \min \begin{cases} \overrightarrow{D'}(F_1[v], F_2[w]) + (d(v,w), 2) \\ \overrightarrow{D}(\theta, T_2[w])]) + \min_{w_k \in son(w)} \left\{ \overrightarrow{D}(T_1[v], T_2[w]) - \overrightarrow{D}(\theta, T_2[w_k]) \right\} \\ \overrightarrow{D}(T_1[v], \theta) + \min_{v_k \in son(v)} \left\{ \overrightarrow{D}(T_1[v], T_2[w]) - \overrightarrow{D}(T_1[v_k], \theta \right\} \end{cases}$$

The work presented here is part of project to develop a set of tools for analyzing plants which are modeled by rooted tree graphs [8]. The proposed algorithms and their implementation are currently integrated into this tool set [9, 10, 4].

References

[1] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the Association for Computing Machinery*, 19:248–264, 1972.

[2] P. Ferraro and C. Godin. A distance measure between plant architectures based on the comparison of their topological structures. In *Second International Workshop on Functional-Structural Trees Models*, Clermont-Ferrand (France), October 12-15 1998.

[3] P. Ferraro and C. Godin. Un algorithme de comparaison d'arborescences non ordonnées appliqué à la comparaison de la structure topologique des plantes. In *SFC'98, Recueil Des Actes*, pages 77–81, Montpellier (France), September 21-23 1998.

[4] P. Ferraro and C. Godin. An algorithm for comparing unordered tree graphs based on a minimum cost mapping with a minimal connectivity. In *Third International Conference on Orders, Algorithms and Applications*, Montpellier (France), August 24-27 1999.

[5] P. Ferraro and C. Godin. A distance measure between plant architectures. *Annals of Forest Science*, june 2000.

[6] Pascal Ferraro. *Méthodes algorithmiques de comparaison d'arborescences. Applications à la comparaison de l'architecture des plantes.* PhD thesis, Institut National Polytechnique de Toulouse (France), 2000.

[7] Pascal Ferraro and Christophe Godin. An unified point of view on tree graph comparison. *???*, Submitted in 2001.

[8] Christophe Godin and Yves Caraglio. A multiscale model of plant topological structures. *Journal of theoretical biology*, 191:1–46, 1998.

[9] Christophe Godin, Evelyne Costes, and Yves Caraglio. Exploring plant topological structure with the amapmod software: an outline. *Silva Fennica*, 31:355–366, 1997.

[10] Christophe Godin, Yann Guédon, Evelyne Costes, and Yves Caraglio. Measuring and analyzing plants with the amapmod software. In M. Michalewicz, editor, *Advances in computational life sciences, Vol I : Plants to ecosystems*, volume January, pages 63–94. Csiro, Australia, 1997. chapitre 4.

[11] Pekka Kilpellainen and H. Mannila. The tree inclusion problem. In *Proc. Internat. Joint Conf. on the Theory and Practice of Software*, volume 1, pages 202–214, 1991.

[12] Andrew S. Noetzel and Stanley M. Selkow. An analysis of the general tree-editing problem. In David Sankoff and Joseph B. Kruskal, editors, *Time Wraps, Strings Edits, and Macromolecules: the theory and practice of sequence comparison*, chapter 8, pages 237–252 ,. Addison-Wesley Publishing Company Inc, University of Montreal, Montreal, Quebec, Canada, 1983.

[13] Stanley M. Selkow. The tree-to-tree editing problem. *Information processing letters*, pages 184–186, 1977.

[14] Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the Association for Computing Machinery*, pages 422–433, 1979.

[15] E. Tanaka and K. Tanaka. The tree-to-tree editing problem. *International journal Pattern Recognition And Atificial Intelligency*, 2(2):221–240, 1988.

[16] Robert Endre Tarjan. *Data Structures and Network Algorithms.* CBMS-NFS - Regional Conference Series In Applied Mathematics, 1983.

[17] Robert A. Wagner and Michael J. Fisher. The string-to-string correction problem. *Journal of the association for computing machinery*, 21:168–173, 1974.

[18] Kaizhong Zhang. A new editing-based distance between unordered trees. In *Combinatorial Pattern Matching, 4th Ann. Symp.*, pages 254–265, Padala (Italy), 1993. CPM'93.

[19] Kaizhong Zhang. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15:205–222, 1996.

[20] Kaizhong Zhang and Tao Jiang. Some max snp-hard results concerning unordered labeled trees. *Information Processing Letters*, 49:249–254, 1994.