Gathering with Minimum Delay in Sensor Networks

Jean-Claude Bermond, Luisa Gargano Adele Rescigno

▶ **To cite this version:**

Jean-Claude Bermond, Luisa Gargano Adele Rescigno. Gathering with Minimum Delay in Sensor Networks. 2008. ⟨inria-00256896⟩

HAL Id: inria-00256896

https://inria.hal.science/inria-00256896v1

Preprint submitted on 18 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Gathering with Minimum Delay in Sensor Networks

Jean–Claude Bermond, Luisa Gargano, Adele A. Rescigno, ????

February 8, 2008

Abstract

Data gathering is a fundamental operation in wireless sensor networks in which data packets generated at sensor nodes are to be collected at a base station. We investigate the delay of the gathering process and give an optimal data gathering schedule for tree networks.

1 Introduction

A wireless sensor network is a multi-hop wireless network formed by a large number of low-cost sensor nodes, each equipped with a sensor, a processor, a radio, and a battery. Due to the many advantages they offer – e.g. low cost, small size, and wireless data transfer – wireless sensor networks become attractive to a vast variety of applications like space exploration, battlefield surveillance, environment observation, and health monitoring.

A basic activity in a sensor network is the systematic gathering of the sensed data at a base station for further processing. A key challenge in such operation is due to the physical limits of the sensor nodes, which have limited power and un–replenishable batteries. It is then important to bound the energy consumption of data dissemination [6, 15, 27]. However, an other important factor to consider in data gathering applications is the *latency* of the information dissemination process. Indeed, the data collected by a node can frequently change thus making essential that they are received by the base station as soon as it is possible without being delayed by collisions [29]. In this paper, we will study optimal–time data gathering algorithms in tree networks.

1.1 Network model

We adopt the sensor network model considered in [8]. In this model each sensor is equipped with an half–duplex interface, hence,

i) *a node cannot receive and transmit at the same time.*

Moreover, each node is equipped with omnidirectional antennas allowing the transmission over a distance R. This implies that for any given node in the network, we can individuate its neighbors as those sensors within distance R from it, that is, within its transmission/interference range. In this model,

ii) *a* collision *happens at a node x if two or more of its neighbors try to transmit at the same time.*

However, simultaneous transmissions among pair of nodes can successfully occur whenever conditions i) and ii) of the above interference model are respected. The time is slotted so that one–hop transmission of one data item consumes one time slot; the network is assumed to be synchronous. Moreover, it is assumed that the only traffic in the network is due to sensor data, thus data transmissions can be completely scheduled.

Summarizing, the network can be represented by means of a direct graph $G = (V, A)$ where V represents the sensor nodes and an arc $(u, v) \in A$ if v is in the transmission/interference range of u. Throughout this paper we assume that all nodes have the same transmission range, hence the graph G is a directed symmetric graph, e.g., $(u, v) \in A$ if and only if $(v, u) \in A$.

The collision–free data gathering problem can be then stated as follows [29].

> **Data Gathering.** Given a graph $G = (V, A)$ and a base station s, for each $v \in V - \{s\}$, schedule the multi-hop transmission of the data items sensed at v to s so that the whole process is collision–free, and the time when the last data is received by s is minimized.

1.2 Our results

We will give optimal gathering schedules in case the graph modeling the sensor network is a tree. We will actually study the data gathering problem by studying the related one–to-all personalized broadcast problem in which the base station wants to communicate different data items to each other node in the network.

> **One–to-all personalized broadcast:** Given a graph $G = (V, A)$ and a base station s, for each $v \in V - \{s\}$, schedule the multi-hop transmission from s to v of the data items destined to v so that the whole process is collision–free, and the time when the last data item is received at the corresponding destination node is minimized.

Solving the above dissemination problem is equivalent to solve the data gathering in sensor networks. Indeed, let T denote the largest time–slot used by a dissemination algorithm, a gathering schedule with delay T consists in scheduling a transmission from node y to x during slot t if and only if the broadcasting algorithm schedules a transmission from node x to y during slot $T - t + 1$, for any t with $1 \leq t \leq T$.

1.3 Related work

Much effort has been devoted to the study of efficient data gathering algorithms taking into consideration various aspects of sensor networks [1, 4].

The problem of minimizing the delay of the gathering process has been recently recognized and studied. The authors of [8] first afford such a problem; they use the same model

for sensor networks adopted in this paper. The main difference with our work is that [8] mainly deals with the case when nodes are equipped with directional antennas, that is, only the designed neighbor of a transmitting node receives the signal while its other neighbors can safely receive from different nodes. Under this assumption, [8] gives optimal gathering schedules for trees. An optimal algorithm for general networks has been presented in [18] under the hypothesis that each node has one packet of sensed data to deliver.
The work in [29] also deals with the latency of data gathering under the assumption of unidirectional antennas; the difference with the model in [8] is the assumption of the possibility to have multiple channels between adjacent nodes. By adopting this model an approximation algorithm with performance ratio 2 is obtained.

Fast gathering with omnidirectional antennas is considered in [2] and [3], under the assumption of possibly different transmission and interference ranges, that is, when a node transmits, all the nodes within a fixed distance d_T can receive while nodes within distance d_I ($d_I \geq d_T$) cannot listen to other transmissions due to interference (in our paper $d_I = d_T$). Lower bounds on the time to gather are given and NP-hardness is proved in [2]; an approximation algorithm with approximation factor 4 is also presented. Paper [3] presents an on–line gathering algorithm under the described model.

Several papers deal with the problem of maximize the lifetime of the network through topology aware placement [6, 9], data aggregation [11, 19, 17, 21], or efficient data flow [7, 15, 25]. Papers [22, 28, 5] consider the minimization of the gathering delay in conjunction with the energy spent to complete the process.

Finally, we notice that several papers deal with broadcasting in wireless networks [24], however this problem is not the reverse of data gathering whereby different data packets are gathered to the sink.

Paper Overview. The rest of the paper is organized as follows: In Section 2 we formally describe the problem. Sections 3 and 4 present the proposed optimal solutions in case of lines and trees, respectively.

2 Mathematical Formulation

We now formulate the one-to-all personalized broadcast problem. Consider a directed symmetric graph $G = (V, A)$ and let $s \in V$ be a special node that will be called the *source*.

Each node $v \in V - \{s\}$ is associated with an integer weight $w(v)$ that represents the number of data items destined to node v. The set $\mathbf{w} = \{w(v) \mid v \in V - \{s\}\}$ will represent the set of weights of the nodes in V.

We need to schedule (time–label) the transmissions in order to create $w(v)$ collision–free routes from s to node v, for each $v \in V - \{s\}$.

Definition 1 *Let* $\mathbf{p} = (u_0, \cdots, u_h)$ *be a path in* G. *An* increasing labeling L of $\mathbf{p}$ *is an assignment of integers,* $L_{\mathbf{p}}(u_0, u_1) \ldots, L_{\mathbf{p}}(u_{h-1}, u_h)$, *to the arcs of* $\mathbf{p}$ *such that*

$$L_{\mathbf{p}}(u_j, u_{j+1}) = L_{\mathbf{p}}(u_{j-1}, u_j) + 1$$

for $j = 1, \ldots, h-1$.
The labeling is called t-increasing*, for some integer* $t \geq 1$*, if it is increasing and* $L_{\mathbf{p}}(u_0, u_1) = t$.

Consider any set $\mathcal{P}$ of paths in G from s to (not necessarily pairwise distinct) nodes in $V - \{s\}$ together with their labelings $L_{\mathbf{p}}$, for $\mathbf{p} \in \mathcal{P}$. Notice that any arc $a \in A$ can belong to either zero, or one or more paths in $\mathcal{P}$.

Definition 2 *The labeling induced by* $\mathcal{P}$ *on the arcs of* G *consists of the multisets*

$$L(u, v) = \{L_{\mathbf{p}}(u, v) \mid \mathbf{p} \in \mathcal{P}\},$$

for each $(u, v) \in A$.

Let $N(u)$ be the set of neighbors of u in G, that is, $N(u) = \{x \mid (u, x) \in A\} = \{x \mid (x, u) \in A\}$.

Definition 3 *The labeling* L *induced by* $\mathcal{P}$ *on the arcs of* G *is called* strictly collision–free (SCF) *if* L *is increasing and, for each* $(u, v) \in A$ *it holds:*

- $L(u, v)$ *is a set (e.g, any integer has at most one occurrence in* $L(u, v)$*),*
- $L(u, v) \cap L(w, u) = \emptyset$*, for each* $w \in N(u)$*,*
- $L(u, v) \cap L(w, z) = \emptyset$*, for each* $w \in N(v) \cup \{v\}$*,* $z \in N(w)$.

Definition 4 *An* instance of SCF labeling *is a triple* $\langle G, \mathbf{w}, s\rangle$ *where* G *is the graph,* s *is the source, and* $\mathbf{w}$ *is the set of weights of the nodes in* G.
A feasible solution *for* $\langle G, \mathbf{w}, s\rangle$ *is a pair* $(\mathcal{P}, L)$ *where:*

- $\mathcal{P}$ *is a set of* $w(v)$ *paths (not necessarily distinct) from* s *to* v *in* G*, for each* $v \in V - \{s\}$*;*
- L *is a SCF–labeling induced by* $\mathcal{P}$.

An optimal solution $(\mathcal{P}^*, L^*)$ *is a feasible solution that minimizes the largest label assigned to any arc of* G.
The value attained by the optimal solution $(\mathcal{P}^*, L^*)$ *for* $\langle G, \mathbf{w}, s\rangle$ *is denoted by* $\mathcal{T}^*(\langle G, \mathbf{w}, s\rangle)$ *(or simply by* $\mathcal{T}^*(G)$ *when* $\mathbf{w}$ *and* s *are clear from the contex).*

Fig. 1: A tree T.

Example. Let $T = (V, E)$ be the tree given in Fig. 1, and let s be the source and $w(u) = 1$ for each $u \neq s$. A solution for $(T, s, \mathbf{w})$ is the pair $(\mathcal{P}, L)$ where

$$\mathcal{P} = \{\mathbf{p}_1 = (s, a, d, f), \mathbf{p}_2 = (s, b, e), \mathbf{p}_3 = (s, a, c), \mathbf{p}_4 = (s, b), \mathbf{p}_5 = (s, a, d), \mathbf{p}_6 = (s, a)\}$$

and L is such that

$$L_{\mathbf{p}_1}(s, a) = 1,\ L_{\mathbf{p}_1}(a, d) = 2,\ L_{\mathbf{p}_1}(d, f) = 3,\ L_{\mathbf{p}_2}(s, b) = 2,\ L_{\mathbf{p}_2}(b, e) = 3,\ L_{\mathbf{p}_3}(s, a) = 4,$$

$$L_{\mathbf{p}_3}(a, c) = 5,\ L_{\mathbf{p}_4}(s, b) = 5,\ L_{\mathbf{p}_5}(s, a) = 6,\ L_{\mathbf{p}_5}(a, d) = 7,\ L_{\mathbf{p}_6}(s, a) = 8$$

and

$$L(s, a) = \{1, 4, 6, 8\},\ L(a, c) = \{5\},\ L(a, d) = \{2, 7\},$$
$$L(d, f) = \{3\},\ L(s, b) = \{2, 5\},\ L(b, e) = \{3\}.$$

Notice that minimizing the largest label is equivalent to minimize the time needed by the algorithm. Indeed, one can just consider solutions where all labels in $\{1, \cdots, T\}$ are used: If some integer c is never used, then we can decrease by 1 the value of each label $c' \geq c + 1$ in the considered feasible solution.

2.1 Notation

The following notation will be used in the sequel.

- *Set a path (resp. a t–path) to node v*: establish a path from s to v together with its increasing labeling (resp. t–increasing labeling);
- A node $v \neq s$ *is completed*: if $w(v)$ paths from s to v have been set.

3 Lines

In this section we present an optimal algorithm to solve the SCF–labeling problem for an instance $\langle G, \mathbf{w}, s\rangle$, where G is a line, s is one of its end points, and node weights are arbitrary non negative integers, that is, $w(v) \geq 0$ for each $v \neq s$.

Let G be the line of length n with nodes $0, 1, \cdots, n$ and let $(i, i+1)$, for $i = 0, \cdots, n-1$, be the connection between subsequent nodes. Assume that the source node is $s = 0$ and $w(n) > 0$ (otherwise delete the end vertices of the line with weight 0).

Lemma 1 *A solution $(\mathcal{P}, L)$ of $\langle G, \mathbf{w}, s\rangle$ is feasible iff*

1) The labeling L induced by $\mathcal{P}$ is increasing,

2) for each $\mathbf{p}, \mathbf{q} \in \mathcal{P}$ with $L_{\mathbf{q}}(s, 1) \geq L_{\mathbf{p}}(s, 1)$: if $\mathbf{p}$ leads from s to node h, with $1 \leq h \leq n$, then

$$L_{\mathbf{q}}(s, 1) \geq L_{\mathbf{p}}(s, 1) + \min\{3, h\}. \tag{1}$$

Proof. Conditions of Definition 3 become in the case of the line:

$$L_{\mathbf{q}}(i,i+1) \neq L_{\mathbf{p}}(i,i+1); \quad L_{\mathbf{q}}(i,i+1) \neq L_{\mathbf{p}}(i+1,i+2); \quad L_{\mathbf{q}}(i,i+1) \neq L_{\mathbf{p}}(i+2,i+3).$$

As $L_{\mathbf{p}}(i,i+1) = L_{\mathbf{p}}(s,1)+i$ and $L_{\mathbf{q}}(i,i+1) = L_{\mathbf{q}}(s,1)+i$ they are equivalent to $L_{\mathbf{q}}(s,1) \neq L_{\mathbf{p}}(s,1)$; $L_{\mathbf{q}}(s,1) \neq L_{\mathbf{p}}(1,2)$; $L_{\mathbf{q}}(s,1) \neq L_{\mathbf{p}}(2,3)$. Thus

- If $h \geq 1$ then $L_{\mathbf{q}}(s,1) \neq L_{\mathbf{p}}(s,1)$ and $L_{\mathbf{q}}(s,1) \geq L_{\mathbf{p}}(s,1)$ trivially imply $L_{\mathbf{q}}(s,1) \geq L_{\mathbf{p}}(s,1)+1$.
- If $h \geq 2$ then $L_{\mathbf{q}}(s,1) \neq L_{\mathbf{p}}(1,2) = L_{\mathbf{p}}(s,1)+1$; this together with $L_{\mathbf{q}}(s,1) \geq L_{\mathbf{p}}(s,1)+1$ implies $L_{\mathbf{q}}(s,1) \geq L_{\mathbf{p}}(s,1)+2$.
- Finally, if $h \geq 3$ we have $L_{\mathbf{q}}(s,1) \neq L_{\mathbf{p}}(2,3) = L_{\mathbf{p}}(s,1)+2$; knowing that $L_{\mathbf{q}}(s,1) \geq L_{\mathbf{p}}(s,1)+2$, we get $L_{\mathbf{q}}(s,1) \geq L_{\mathbf{p}}(s,1)+3$.

□

For each node $i = 1, \cdots, n$, let $w(i)$ be the weight of node i, and define $\beta_i = \sum_{j \geq i} w(j)$, and let $M_1 = w(1) + 2w(2) + 3\beta_3$; $M_2 = 2w(2) + 3\beta_3$ and $M_i = i + 3(\beta_i - 1)$ if $i \geq 3$.

Lemma 2 *For a line G with nodes $s = 0, 1, \ldots, n$, it holds*

$$\mathcal{T}^*(G) \geq \max_{1 \leq i \leq n} M_i.$$

Proof. By the preceding lemma, all the labels in $L(s,1), L(1,2), L(2,3)$ are different. We have $w(1) + w(2) + \beta_3$ labels in $L(s,1)$ as there are $w(1) + w(2) + \beta_3$ paths using the arc $(s,1)$; similarly we have $w(2) + \beta_3$ labels in $L(1,2)$ and β_3 labels in $L(2,3)$. So all together we have at least $M_1 = w(1) + 2w(2) + 3\beta_3$ different labels and so $\mathcal{T}^*(G) \geq M_1$. Note that $M_2 \leq M_1$.

If $i \geq 3$ then the starting labels of any pair of the β_i paths to $j \geq i$ differ of at least 3; besides, since any feasible labeling is an increasing labeling we have that the largest label of arc $(i-1, i)$ is $\geq i + 3(\beta_i - 1)$. □

We first give in Fig. 2 an SCF-labeling on the line G with the source at node 0 and then we will prove that it is optimal.

LINE-labeling $(G, \mathbf{w}, s)$

- Set $\mathcal{P} = \emptyset$, $k = 1$.
- **while** node 1 is not completed **do**
  - Let i be the largest node which is not completed (e.g $i = \max\{j \mid 1 \leq j \leq n, w(j) > 0\}$.
  - Set a k–path to i in G, call it $\mathbf{p}_i$.
  - Let $\mathcal{P} = \mathcal{P} \cup \{\mathbf{p}_i\}$.
  - Let $w(i) = w(i) - 1$.
  - Set $k = k + \min\{3, i\}$.
- **return** $(\mathcal{P}, L)$, where L is the labeling induced by $\mathcal{P}$.

Fig. 2: The SCF-labeling on lines with source at node 0.

Theorem 1 *The algorithm LINE-labeling on G returns an optimal solution $(\mathcal{P}, L)$ of value*

$$\mathcal{T}^*(G) = \max_{1 \le i \le n} M_i.$$

Proof. The solution $(\mathcal{P}, L)$ returned by algorithm LINE-labeling is feasible for $\langle G, \mathbf{w}, s \rangle$. Indeed $\mathcal{P}$ contains $w(i)$ paths from s to each $i \neq s$ and the induced labeling L can be easily proved to be a SCF-labeling of $\mathcal{P}$ by noticing that it satisfies 1) and 2) of Lemma 1.

Let $L_{max}(a)$ be the largest label assigned to the arc a, for $a \in A$. We shall show that $L_{max}(i-1, i) \le M_i$ for each $i = 1, \ldots, n$.
The algorithm LINE-labeling first sets a path to the node n. Hence, the smallest label of any arc $(i-1, i)$, with $i \le h$, is i. Furthermore, paths are set every 3 steps while there are not completed nodes at distance at least 3 from s, and paths to node 2 are set every 2 steps until it is completed. This and Definition 1 say that:
- If $i \ge 3$ then $L(i-1, i) = \{i, i+3, i+6, \cdots, i+3(\beta_i - 1)\}$. Hence $L_{max}(i-1, i) = i + 3(\beta_i - 1) = M_i$.
- If $i = 2$ then by the algorithm first paths to the nodes at distance larger than 2 and then paths to the node at distance 2 are set. This implies that the algorithm labels arc $(1, 2)$ with the integers in the set $X = \{2, 5, 8, \cdots, 2 + 3(\beta_3 - 1)\}$ when paths to the β_3 nodes at distance at least 3 are set, and, if $w(2) > 0$, with the integers in the set $Y = \{2 + 3(\beta_3 - 1) + 3, \cdots, 2 + 3\beta_3 + 2(w(2) - 1)\}$ when paths to node 2 are set for $w(2)$ times. Hence, we have $L(1, 2) = X \cup Y$ and

$$L_{max}(1, 2) = 2 + 3(\beta_3 - 1) \text{ if } w(2) = 0, \text{ and } L_{max}(1, 2) = 2w(2) + 3\beta_3 \text{ if } w(2) > 0.$$

Then the largest label assigned to the arc $(1, 2)$ is at most M_2.
- If $i = 1$ then the algorithm labels arc $(s, 1)$ with the integers in the set $X = \{1, 4, 7, \cdots, 1 + 3(\beta_3 - 1)\}$ when paths to the β_3 nodes at distance at least 3 are set. If $w(2) = 0$ and $w(1) > 0$ the algorithm labels with the integers in $Z = \{1 + 3\beta_3, \cdots 3\beta_3 + w(1)\}$ when paths to 1 are set.
If $w(2) > 0$ the algorithm labels the arc $(s, 1)$ with the integers in $Y = \{1 + 3(\beta_3 - 1) + 3, \cdots 1 + 3\beta_3 + 2(w(2) - 1)\}$ when paths to 2 are set and furthermore, if $w(1) > 0$, arc $(s, 1)$ is labeled with the integers in $Z = \{1 + 3\beta_3 + 2(w(2) - 1) + 2, \cdots 3\beta_3 + 2w(2) + w(1)\}$ when paths to 1 are set.
Hence, we have $L_{max}(s, 1) = 3\beta_3 - 2$ if $w(1) = w(2) = 0$; $L_{max}(s, 1) = 2w(2) + 3\beta_3 - 1$ if $w(1) = 0$ and $w(2) > 0$, and $L_{max}(s, 1) = w(1) + 2w(2) + 3\beta_3$ if $w(1) > 0$. Then the largest label assigned to the arc $(s, 1)$ is at most M_1. □

The above theorem provides a simpler form of the optimal label (i.e. time) when each sensor node has at least one request to be completed.

Corollary 1 *If G is a line and $w(i) \ge 1$, for $1 \le i \le n$, then*

$$\mathcal{T}^*(G) = M_1 = w(1) + 2w(2) + 3\sum_{j=3}^{n} w(j).$$

In the particular case where $w(i) = 1$ for each node then we have that $\mathcal{T}^*(G) = 3n - 3$.

4 Trees

Let $T = (V, E)$ be any tree and s be a fixed node in T. We assume here that each node has exactly one path to be set, that is, $w(v) = 1$ for each $v \in V - \{s\}$ (recall that the source has weight $w(s) = 0$). We will show how to obtain an optimal labeling for $\langle T, \mathbf{w}, s\rangle$.

4.1 Defintions and Notation

Definition 5 *Given a tree T. We shall denote by $|T|$ the* size of T *in terms of* the weights *of the nodes in T, that is*

$$|T| = \sum_{v \in V(T)} w(v).$$

Notice that $|T|$ represents the number of paths to be set in T. Since we assume that $w(v) = 1$ for each $v \in V - \{s\}$ then we start with $|T| = |V| - 1$.

Root T at s and let $T_1, T_2, \cdots, T_m$ denote the subtrees of T rooted at the sons of s. For each $i = 1, \ldots, m$, we denote by:

– s_i the son of s which is the root of T_i,
– α_i the number nodes at level 2 in T_i,
– β_i the number nodes at level 3 or more in T_i.

Definition 6 *Define the* shade of subtree T_i, *for $1 \leq i \leq m$, as*

$$\tau_i = 1 + 2\alpha_i + 3\beta_i.$$

Consider the case $m = 1$, then T consists of a root of degree 1 and T_1 as the only subtree. A one-to-all personalized optimal broadcasting in T is obtained by applying the optimal algorithm LINE-labeling to the line L obtained from T by replacing the $w(j)$ vertices at distance j in T by a vertex j with weight $w(j)$ in L. Then by Corollary 1 the number of steps is $\mathcal{T}^*(L) = 1 + 2\alpha_1 + 3\beta_1$.

One main idea of the algorithm consists in setting, if that is possible, a path to a node in the subtree T_i having the biggest value τ_i. So we define the following order.

Definition 7 *Given $i, j = 1, \ldots, m$ with $i \neq j$, we say that*

- $T_i \prec T_j$ *if either $\tau_i > \tau_j$ or $\tau_i = \tau_j$ and $|T_i| > |T_j|$,*
- $T_i = T_j$ *if $\tau_i = \tau_j$ and $|T_i| = |T_j|$.*

Unless otherwise stated, in the following we assume that the subtrees are ranked according their shade that is $T_1, \cdots, T_m$ is a reordering of the subtrees of T such that $T_1 \preceq \cdots \preceq T_m$.
In order to describe the SCF labeling algorithm, we introduce the following terminology.

- *One step:* one time–slot.

- A node $v \neq s$ *is completed* if a path from s to v has been set.
- *Set a path (resp. a t-path) to* T_i: set a path (resp. a t-path) to a node v in T_i which is the furthest from s among all nodes in T_i which are not yet completed.

  When we set a path to some T_i the corresponding value $|T_i|$ of the remaining weights in T_i will be decreased by one and also α_i and β_i if they are non zero.
- T_i *is completed*: if a path has been set to each node in T_i, that is $|T_i| = 0$.
- Step t is called *idle* if no t-path is set.
- T_i is *available* at step t (e.g. a t–path to T_i can be set) only if no path was set to a node v in T_i at some step t' s.t. $t' < t < t' + \min\{3, \ell(v)\}$, where $\ell(v)$ is the level of v in T. Said otherwise, if at some step t' we set a path to a node v in T_i, then T_i is not available at step $t' + j$ where $1 \leq j < \min\{3, \ell(v)\}$. in particular if v is at a level at least 3, then T_i is not available at steps $t' + 1$ and $t' + 2$.

4.2 The algorithm

The SCF labeling algorithm on a tree T is given in Fig. 3. We first give an informal description of the behavior of the algorithm during a generic step $t \geq 1$: Let T_i be an available subtree that precedes all the other available subtrees of T according to the order relation $\preceq$; set a t–path to T_i; update the shade of T_i.

TREE-labeling $(T, \mathbf{w}, s)$ *[T has non empty subtrees $T_1, \ldots, T_m$ and root s]*

1. Set $\mathcal{P} = \emptyset$ and $t = 1$
Let $t_i = 1$ for $i = 1, \ldots, m$ *[t_i is the minimum step at which a path to T_i can be set]*
Set $M = \{1, \ldots, m\}$ *[M represents the set of subtrees not yet completed]*

2. **while** $M \neq \emptyset$

2.1 Rename the indices in M so that for the permuted subtrees it holds $T_1 \preceq T_2 \preceq \ldots \preceq T_{|M|}$.

2.2 **if** there exists $i \leq |M|$ with $t_i \leq t$ **then**
Let i be the smallest such index (e.g. $t_1, \ldots, t_{i-1} > t$ and $T_i \preceq T_{i+1} \preceq \ldots \preceq T_{|M|}$).
if NOT ($|M| = 2$, $i = 1$, $\beta_1 = 1, \alpha_2 > \beta_2 = 0$, $t_2 \leq t + 1$) **then**
[Execute the generic step of the algorithm]
- Set a t-path to T_i and call it $\mathbf{p}$
- $\mathcal{P} = \mathcal{P} \cup \{\mathbf{p}\}$.
- If T_i is completed then $M = M - \{i\}$.
- $t_i = t + \min\{3, \ell\}$, where ℓ is the length of $\mathbf{p}$,
- Update T_i, eg.: $\tau_i = \tau_i - \min\{3, \ell\}$,
$w(s_i) = w(s_i) - \begin{cases} 1 & \text{if } \ell = 1 \\ 0 & \text{oth.} \end{cases}$, $\alpha_i = \alpha_i - \begin{cases} 1 & \text{if } \ell = 2 \\ 0 & \text{oth.} \end{cases}$, $\beta_i = \beta_i - \begin{cases} 1 & \text{if } \ell \geq 3 \\ 0 & \text{oth.} \end{cases}$.

2.3 **else** *[Here is the special case: $|M| = 2$, $i = 1$, $\beta_1 = 1, \alpha_2 > \beta_2 = 0$]*
- Set a t-path to T_1 and call it $\mathbf{p}$
- Set a $t + 1$-path to s_2 and call it $\mathbf{q}_1$
- Set a $t + 2$-path to T_2 and call it $\mathbf{q}_2$
- $\mathcal{P} = \mathcal{P} \cup \{\mathbf{p}, \mathbf{q}_1, \mathbf{q}_2\}$.
- $t_1 = t + 3$ and $t_2 = t + 4$.
- Update T_1 and T_2 (e.g. $\tau_1 = \tau_1 - 3$, $\beta_1 = 0$, $\tau_2 = \tau_2 - 3$, $w(s_2) = 0$, $\alpha_2 = \alpha_2 - 1$).
- If $\alpha_2 = 0$ then $M = \{1\}$.
- $t = t + 2$.

2.4 $t = t + 1$.
endwhile

3. **return** $(\mathcal{P}, L)$

Fig. 3: The SCF labeling algorithm on trees.

The TREE-labeling algorithm sets, at step t, a t-path to T_i only if T_i is available. As in Lemma 1, we can then conclude that

Lemma 3 *The solution $(\mathcal{P}, L)$ returned by algorithm TREE-labeling on $\langle T, \mathbf{w}, s \rangle$ is feasible.*

The rest of the paper is devoted to prove the optimality of the algorithm.

4.3 Preliminary Results

In this section we establish some facts that will be used in order to prove the optimality of the proposed algorithm.

Fact 1 *For any subtree T_i with $|T_i| > 1$ it holds*

$$2|T_i| - 1 \leq \tau_i \leq 3|T_i| - 3.$$

Proof. By definition $|T_i| = \beta_i + \alpha_i + 1$ and $\tau_i = 3\beta_i + 2\alpha_i + 1$. Hence

$$2|T_i| - 1 = 2\beta_i + 2\alpha_i + 1 \leq 3\beta_i + 2\alpha_i + 1 = \tau_i \leq 3\beta_i + 3\alpha_i = 3|T_i| - 3,$$

where the last inequality follows noticing that $|T_i| > 1$ implies $\alpha_i \geq 1$. □

Fact 2 *Let $T_i \preceq T_j$.*

- *If $\tau_i = \tau_j$ and $T_i \prec T_j$ then $\alpha_i > \alpha_j$ and $\beta_i < \beta_j$.*
- *$T_i = T_j$ (e.g., $\tau_i = \tau_j$ and $|T_i| = |T_j|$) iff $\alpha_i = \alpha_j$ and $\beta_i = \beta_j$.*

Fact 3 *If $T_i \preceq T_j$ then $\beta_j \leq \begin{cases} |T_i| - 2 & \text{if } |T_i| \geq 2 \\ 0 & \text{otherwise} \end{cases}$.*

Proof. Trivially, if $|T_i| = 1$ then $\beta_j = 0$. Let then $|T_i| > 1$. If $T_i \preceq T_j$ then $\tau_i \geq \tau_j$. This implies that $3\beta_i \geq 3\beta_j + 2\alpha_j - 2\alpha_i$. From this we get

$$|T_i| = 1 + \alpha_i + \beta_i \geq 1 + \alpha_i + \frac{3\beta_j + 2\alpha_j - 2\alpha_i}{3} \geq \beta_j + \frac{2}{3}\alpha_j + \frac{1}{3}\alpha_i + 1.$$

Hence, noticing that $\alpha_i \geq 1$, we get $|T_i| \geq \beta_j + 2$. □

Definition 8 *Define, for each $i, j = 1, \ldots, m$, with $i \neq j$,*

$$\Delta_{i,j} = |T_i| + |T_j| + \beta_i - 1, \qquad \text{and} \qquad \epsilon_T = \begin{cases} 1 & \text{if } T_1 = T_2 \\ 0 & \text{otherwise} \end{cases}.$$

Fact 4 *For any i, j it holds $\Delta_{i,j} - \tau_i = |T_j| - |T_i|$*

Proof. Recalling that $\tau_i = 3\beta_i + 2\alpha_i + 1$ and by Definition 8 we have

$$\Delta_{i,j} - \tau_i = |T_i| + |T_j| + \beta_i - 1 - (1 + 2\alpha_i + 3\beta_i) = |T_i| + |T_j| - 2|T_i| = |T_j| - |T_i|.$$

□

Fact 5 *$\Delta_{i,j} \geq \max\{|T|, \tau_1 + \epsilon_T\}$ only if either $i = 1$ and $j = 2, 3$ or $i = 2$ and $j = 1$.*

Proof. Assume first either $i \geq 3$ or $i = 2$ and $j \geq 3$. We have $|T| - |T_i| - |T_j| \geq |T_1|$ or $|T| - |T_i| - |T_j| \geq |T_2|$. By Fact 3 we know that $\beta_i < \min\{|T_1|, |T_2|\} - 1$. Hence, in any case we get

$$|T| - \Delta_{i,j} = |T| - |T_i| - |T_j| - \beta_i + 1 > 2,$$

which implies $\Delta_{i,j} < |T| \leq \max\{|T|, \tau_1 + \epsilon_T\}$.

Assume now $i = 1$ and $j \geq 4$ and suppose, by contradiction, that $\Delta_{1,j} \geq |T|$ and $\Delta_{1,j} \geq \tau_1 + \epsilon_T$. We have

$$|T_2| + |T_3| \leq |T| - |T_1| - |T_j| = |T| - \Delta_{1,j} + \beta_1 - 1 \leq \beta_1 - 1 \leq |T_1| - 3. \tag{2}$$

From the assumption that $\Delta_{1,j} \geq \tau_1 + \epsilon_T$ and by Fact 4 we get

$$|T_1| \leq |T_j|.$$

From this, (2), and Fact 1, we have

$$\tau_j = 3\beta_j + 2\alpha_j + 1 \geq 2|T_j| - 1 \geq 2|T_1| - 1 \geq 2(|T_2| + |T_3|) + 5 > \frac{2}{3}(\tau_2 + \tau_3) + 5 \geq \frac{4}{3}\tau_3 + 5 > \tau_3$$

thus contradicting the assumption $T_3 \preceq T_j$ for any $j \geq 4$. □

4.4 A lower bound

We establish a lower bound on the optimal labeling of any instance $\langle T, \mathbf{w}, s \rangle$.

Let T be such that $T_1 \preceq T_2 \preceq \ldots \preceq T_m$. Define

$$Max(T) = \max\{|T| = N - 1,\ \tau_1 + \epsilon_T,\ \Delta_{1,2},\ \Delta_{2,1},\ \Delta_{1,3}\}$$

Theorem 2 *Assuming that $T_1 \preceq T_2 \preceq \ldots \preceq T_m$, we have $\mathcal{T}^*(T) \geq Max(T)$.*

Proof. Any algorithm needs to set a path to each node, hence $\mathcal{T}^*(T) \geq |T|$.
By Definition 6 and Corollary 1, the shade τ_i of T_i is the minimum label that can be assigned when only paths to the nodes in T_i are set. Since paths must be set to all nodes in each T_i, for $i = 1, \cdots, m$, and $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_m$ we have that $\mathcal{T}^*(T) \geq \tau_1$.
Furthermore, if $T_1 = T_2$ then at least $\tau_1 + 1$ labels are necessary.
Consider now $\Delta_{i,j}$. For each path to a node at level at least 3 in T_i no path to some other node in T_i can be set in the following 2 steps. Moreover, at most one of the following two steps can be used to set a path to T_j, except for the eventual step in which a path to the root of T_j is set and immediately after a path to some other node in T_j is set. The remaining step can be used to set a path to some T_ℓ with $\ell \neq i, j$. Hence, for any algorithm there are at least $\beta_i - 1 - \sum_{\ell \neq i,j} |T_\ell|$ idle steps, which implies $\mathcal{T}^*(T) \geq |T| + \beta_i - 1 - \sum_{\ell \neq i,j} |T_\ell| = \Delta_{i,j}$.
By Fact 5, we obtain that $\mathcal{T}^*(T)$ is lower bounded by $Max(T)$. □

4.5 Optimality

We show now that the SFC–labeling algorithm for trees is optimal, that is, the maximum label assigned to any arc of T is $\mathcal{T}(T) \leq Max(T)$ thus matching the lower bound of Theorem 2.

We first recall that we are in the hypothesis that the weight of each node is 1. The order in which nodes are chosen as end–points of the paths set by the algorithm implies that the largest label assigned to an arc of T is always to be searched among those assigned to the arcs outgoing the root s of T. Therefore, it coincides with the largest t for which a t–path is set in T.

Lemma 4 *Let t denote the largest integer such that a t–path is set in T during the execution of the SFC–labeling algorithm. The largest label assigned by the algorithm to any arc of T is $\mathcal{T}(T) = t$.*

By the above Lemma, we need to show that the largest t such that a t–path is set in T is upper bounded by $Max(T)$.

The proof will proceed by induction. We will consider the first steps of the algorithm mainly those which send to different subtrees (before the step where we send again to a subtree to which we already sent) and we will apply the induction on the tree T' obtained by deleting the vertices completed in these first steps. For that we introduce the following definition.

Definition 9 *We denote the fact that the algorithm on T starts by setting k paths to pairwise different subtrees of T (that is, it sets a t–path to some node v_i in T_i, for $i = 1, \ldots, k$) by*

$$\langle T_1 \ldots T_k \rangle$$

We denote by T' the updated tree, resulting from $\langle T_1 \ldots T_k \rangle$, that is, T' has subtrees $T'_1 \ldots, T'_k, T'_{k+1} \ldots, T'_m$, where

- *T'_i denotes the updated subtree T_i after the i–path to v_i has been set (that is, $w'(v_i) = 0$ and $|T'_i| = |T_i| - 1$, for $i = 1, \ldots, k$*
- *$T'_{k+1} = T_{k+1}, \ldots, T'_m = T_m$.*

Notice that the subtrees $T'_1 \ldots, T'_m$, are not necessarily ordered according to the relation $\preceq$. Let $i_1, i_2, \cdots, i_m$ be a permutation of $1, \ldots, m$ such that $T'_{i_1} \preceq \ldots \preceq T'_{i_m}$; we will always consider permutations that maintain the original order on equal subtrees, that is

$$\text{if } \; T'_{i_j} = T'_{i_\ell} \;\; \text{ then } \;\; i_j < i_\ell. \tag{3}$$

We denote by $\alpha'_i, \beta'_i, \tau'_i$ the parameters of T'_i. In particular (unless the special case $k = 2$, $\beta_1 = 1, \alpha_2 > \beta_2 = 0$, $T_3 = \emptyset$, and T_2 is available) we have for $i = 1, \ldots, k$:

$$\alpha'_i = \alpha_i - \begin{cases} 1 & \text{if } \beta_i = 0,\ \alpha_i \geq 1 \\ 0 & \text{otherwise} \end{cases}, \quad \beta'_i = \beta_i - \begin{cases} 1 & \text{if } \beta_i \geq 1 \\ 0 & \text{otherwise} \end{cases}, \quad \tau'_i = \tau_i - \begin{cases} 3 & \text{if } \beta_i \geq 1 \\ 2 & \text{if } \beta_i = 0,\ \alpha_i \geq 1 \\ 1 & \text{if } |T_i| = 1 \\ 0 & \text{if } T_i = \emptyset \end{cases}.$$

The following properties hold for T'.

Fact 6 *Assume $\langle T_1 \ldots T_k \rangle$ and that NOT ($k = 2$, $\beta_1 = 1, \alpha_2 > \beta_2 = 0$ and $T_3 = \emptyset$). For any $1 \leq i < j \leq k$.*

1) If $\tau_i > \tau_j$ then $\tau'_i \geq \tau'_j$;

2) if $T_i \prec T_j$ and $T'_j \prec T'_i$ then $\beta_j = 0$, $\beta_i \geq 2$, $|T_i| < |T_j|$, and $\tau_i = \tau_j + 1$.

Proof. If $|T_j| = 1$, then $\tau'_j = 0$; otherwise $\tau'_i \geq \tau_i - 3$ and $\tau'_j \leq \tau_j - 2$; so $\tau'_i \geq \tau'_j$.
Assume now that $T_i \prec T_j$ and $T'_j \prec T'_i$. By Definition 7 we can have four cases:

- $\tau_i = \tau_j$ with $|T_i| > |T_j|$, and $\tau'_j = \tau'_i$ with $|T'_j| > |T'_i|$. This is impossible since it should be both $|T_i| > |T_j|$ and $|T_j| = |T'_j| + 1 > |T'_i| + 1 = |T_i|$.

- $\tau_i = \tau_j$ with $|T_i| > |T_j|$, and $\tau'_j > \tau'_i$. By the algorithm this case can occur only if $\beta_j = 0$ and $\beta_i \geq 1$. By Fact 2 this is impossible.

- $\tau_i > \tau_j$ and $\tau'_j > \tau'_i$. It is impossible by 1).

- $\tau_i > \tau_j$ and $\tau'_j = \tau'_i$ with $|T'_j| > |T'_i|$. We can have both $\tau_i > \tau_j$ and $\tau'_j = \tau'_i$ only if $\tau'_i = \tau_i - 3 = \tau_j - 2 = \tau'_j$. Hence we have $\tau_i = \tau_j + 1$ and $\beta_j = 0 < \beta_i$. Furthermore, $|T'_j| > |T'_i|$ implies $|T_j| > |T_i|$ and $\beta_i \geq 2$ (otherwise, if $\beta_i = 1$ we would get $T'_i = T'_j$). □

Fact 7 *Assume $\langle T_1 \dots T_k \rangle$ with either $k \geq 4$ or $k = 3$ and $T_3 \preceq T'_1, T'_2$:*

i) $|T| \geq \tau_1 + k - 2$;

ii) $|T_i| \geq \beta_1 + 1$, for each $i = 2, \dots, k$;

iii) $|T_i| \geq \beta_2 + 1$, for each $i = 3, \dots, k$.

Proof.
Let T' be the tree resulting after $\langle T_1 \dots T_k \rangle$. If $|T_1| = 1$ then $|T_i| = 1$ for each $i = 1, \cdots, k$; hence, i), ii), and iii) hold. We assume then that $|T_1| \geq 2$ which implies $\alpha_1 \geq 1$.
We first prove ii). If $k \geq 4$ then at steps $4, \dots, k$ we did not choose T'_1 which was available and so $T_2 \preceq T_3 \preceq \dots \preceq T_k \preceq T'_1$. If $k = 3$ $T_3 \preceq T'_1$ by hypothesis. So we have

$$T_i \preceq T'_1 \quad \text{for each} \quad i = 2, \dots, k.$$

Let $\Delta = 1$ if $\beta_1 \geq 1$ and 0 otherwise .
It can occur either $\tau_i > \tau'_1 = \tau_1 - 2 - \Delta$ or $\tau_i = \tau'_1 = \tau_1 - 2 - \Delta$ with $|T_i| \geq |T'_1| = |T_1| - 1$, Hence, w.l.o.g. let $1 \leq \ell \leq k$ be such that

$$\tau_2 \geq \cdots \geq \tau_\ell > \tau_1 - 2 - \Delta$$

and

$$\tau_{\ell+1} = \cdots = \tau_k = \tau_1 - 2 - \Delta \qquad \text{with} \qquad |T_{\ell+1}|, \dots |T_k| \geq |T_1| - 1. \tag{4}$$

Recalling that $\tau_i > \tau'_1 \geq 1$ we get

$$\alpha_i \geq 1, \quad i = 1, \dots, k. \tag{5}$$

For any $i = 2, \ldots, \ell$ we have $\tau_i = 3\beta_i + 2\alpha_i + 1 \geq 3\beta_1 + 2\alpha_1 - \Delta$. We can then deduce that,

$$\beta_i \geq \beta_1 + \frac{2}{3}\alpha_1 - \frac{2}{3}\alpha_i - \frac{1+\Delta}{3}$$

and

$$|T_i| = \beta_i + \alpha_i + 1 \geq \beta_1 + \frac{2}{3}\alpha_1 - \frac{2}{3}\alpha_i - \frac{1+\Delta}{3} + \alpha_i + 1 = \beta_1 + \frac{2}{3}\alpha_1 + \frac{\alpha_i}{3} + \frac{2-\Delta}{3} \geq \beta_1 + \frac{2}{3}\alpha_1 + \frac{1}{3}, \tag{6}$$

where the last inequality holds since $\alpha_i \geq 0$ and $\Delta \leq 1$.
From this, recalling that $\alpha_1 \geq 1$, we get that $|T_i| \geq \beta_1 + 1$ and ii) holds for any $i \leq \ell$.
For $i = \ell + 1, \ldots, k$, we have $|T_i| \geq |T_1| - 1 = \beta_1 + \alpha_1 \geq \beta_1 + 1$. Hence ii) holds for each $i = 2, \ldots, k$.

In the same way we can have iii) by noting that if we have $k \geq 4$ then either $T_1' \preceq T_2'$ and so $T_4 \preceq T_1' \preceq T_2'$, or $T_2' \preceq T_1'$ and so by Fact 6, $\beta_2 = 0$ and so T_2' was available at step 4 and so $T_4 \preceq T_2'$

Consider now inequality i). By (6) we have that for each $i = 2, \ldots, \ell$

$$|T_i| \geq \beta_1 + \frac{2}{3}\alpha_1 + \frac{1}{3} = |T_1| - \frac{\alpha_1}{3} - \frac{2}{3} \tag{7}$$

Hence, by (4) and (7) we have

$$\begin{aligned} |T| &= \sum_{i=1}^{m} |T_i| \geq \sum_{i=1}^{k} |T_i| = |T_1| + \sum_{i=2}^{\ell} |T_i| + \sum_{i=\ell+1}^{k} |T_i| \\ &\geq |T_1| + (\ell - 1)(|T_1| - \frac{\alpha_1}{3} - \frac{2}{3}) + (k - \ell)(|T_1| - 1) \\ &= k|T_1| - k + 1 - (\ell - 1)\frac{\alpha_1 - 1}{3} \\ &= \tau_1 + (k-3)\beta_1 + (k-2)\alpha_1 - (\ell - 1)\frac{\alpha_1 - 1}{3}. \end{aligned} \tag{8}$$

The function in (8) is decreasing in ℓ and its minimum, for $\ell = k$, is $\tau_1 + (k-3)\beta_1 + (k-2)\alpha_1 - (k-1)\frac{\alpha_1 - 1}{3}$. Recalling that $\alpha_1 \geq 1$ and $k \geq 3$, we get the desired bound $|T| \geq \tau_1 + k - 2$. □

Theorem 3 *Assume that $T_1 \preceq \ldots \preceq T_m$. Denote by $(\mathcal{P}, L)$ the solution returned by algorithm TREE-labeling. It holds*

$$\mathcal{T}(T) \leq Max(T) \tag{9}$$

Proof. At any step of the algorithm the tree can have any number $m \geq 1$ of subtrees of positive weight. When we say that the algorithm sets a t–path to a subtree T_i and $|T_i| = 0$ at step t, this means that no t–path is actually set (e.g. t is an idle step).

We first analyze the special case of the algorithm in which $m = 2$, $\beta_1 = 1$, $\beta_2 = 0$ and T_2 is available. So $\tau_1 > \tau_2$ and $\alpha_1 \geq \alpha_2 - 1$. The first two steps of the algorithm are $\langle T_1 T_2 \rangle$,

where the path set to T_2 is a path to s_2 (the root of T_2). Let T' be the tree resulting after $\langle T_1 T_2\rangle$, at the third step a path to T_2' is set. Hence, the first three steps of the algorithm are

$$\langle T_1 T_2\rangle\langle T_2'\rangle$$

Let T^2 be the tree resulting after $\langle T_1 T_2\rangle\langle T_2'\rangle$. Next the algorithm on T proceeds as follows

$$\langle T_1^2 T_2^2\rangle\langle T_1^3 T_2^3\rangle \ldots \langle T_1^\ell T_2^\ell\rangle \ldots \langle T_1^{\alpha_1+1} T_2^{\alpha_1+1}\rangle\langle T_1^{\alpha_1+2}\rangle.$$

where T^ℓ is the tree resulting from $T^{\ell-1}$ after the 2 steps $\langle T_1^{\ell-1} T_2^{\ell-1}\rangle$. To see this, we notice that in each T^ℓ it holds $T_1^\ell \prec T_2^\ell$, since $\tau_1^2 = \tau_1 - 3 > \tau_2 - 3 = \tau_2^2$ and $\tau_1^\ell = \tau_1^{\ell-1} - 2 > \tau_2^\ell = \max\{\tau_2^{\ell-1} - 2, 0\}$, for $\ell > 2$. Moreover, in the hypothesis of this case $\alpha_1 \geq \alpha_2 - 1$, which implies that $T_2^\ell = \emptyset$ for $\ell > \alpha_2$. Finally, by the hypothesis we have

$$\epsilon_T = 0, \quad |T| = 3 + \alpha_1 + \alpha_2 \leq 3 + 2\alpha_1 + 1 = \tau_1, \quad \text{and } \Delta_{1,2}, \Delta_{2,1} \leq |T|.$$

Hence, $Max(T) = \tau_1$; but

$$\mathcal{T}(T) = 3 + 2\alpha_1 + 1 = \tau_1 = Max(T).$$

The rest of the proof is devoted to show that $\mathcal{T}(T) \leq Max(T)$ for each tree. The proof is by induction on the shade of T_1, (recall that $T_1 \preceq T_2 \preceq \ldots \preceq T_m$). As a base consider the trees of the special case above and trees T such that $\tau_1 = 1$; in the latter case, we have $|T_i| = 1$ for each $i = 1, \ldots, m$ and $\mathcal{T}(T) = |T| = Max(T)$.

Suppose now that (9) holds for any tree in which the shade of the first subtree (according to the relation $\preceq$) is at most $\tau_1 - 1$; we prove that (9) holds for T.

Notice that we are assuming that T does not belong to the special case (e.g., $m = 2$, $\beta_1 = 1$, $\beta_2 = 0$, and T_2 is available) and that $|T_1| \geq 2$.

We separate four cases according to the value attaining $Max(T)$.

Case 1: $Max(T) = \Delta_{1,2} > \max\{\tau_1 + \epsilon_T, |T|\}$.

In such a case we know that $\beta_1 > 1$, otherwise $\Delta_{1,2} = |T_1| + |T_2| + \beta - 1 \leq |T|$; hence, the first tree steps of the algorithm are (including the case $|T_3| = 0$)

$$\langle T_1 T_2 T_3\rangle.$$

Let T' be the tree resulting after $\langle T_1 T_2 T_3\rangle$. We will show that after the first 3 steps $\langle T_1 T_2 T_3\rangle$, the algorithm on T proceeds as on input T' and

$$Max(T') \leq Max(T) - 3. \tag{10}$$

This implies the desired inequality

$$\mathcal{T}(T) = 3 + \mathcal{T}(T') \leq 3 + Max(T') = Max(T).$$

By definition of $\Delta_{1,2}$ and using $\Delta_{1,2} > |T|$, we get

$$|T| - |T_1| - |T_2| < \beta_1 - 1. \tag{11}$$

By Fact 4 and using $\Delta_{1,2} > \tau_1$, we get

$$|T_1| < |T_2|. \tag{12}$$

By (11) and Fact 1, we get

$$\tau_3 < 3|T_3| \leq 3(|T| - |T_1| - |T_2|) < 3(\beta_1 - 1) = (3\beta_1 + 2\alpha_1 + 1) - (2\alpha_1 + 4),$$

from which, since $\alpha_1 \geq 1$, it follows

$$\tau_3 < \tau_1 - 6 = \tau_1' - 3. \tag{13}$$

Moreover, by (11) and (12) we have

$$|T_2| \geq |T_1| + 1 \geq \beta_1 + \alpha_1 + 2 \geq \beta_1 + 3 > (|T| - |T_1| - |T_2|) + 4 \geq |T_3| + |T_4| + 4; \tag{14}$$

which, using Fact 1, implies

$$\tau_2 \geq 2|T_2| - 1 > 2(|T_3| + |T_4|) + 7 \geq 4\min\{|T_3|, |T_4|\} + 7.$$

Noticing that Fact 1 implies $4|T_4| > \tau_4$ and $4|T_3| > \tau_3 \geq \tau_4$, we get

$$\tau_2 \geq \tau_4 + 8. \tag{15}$$

From (13) and (15) and recalling that $\tau_1 \geq \tau_2$, we obtain that in the tree T', resulting after $\langle T_1 T_2 T_3 \rangle$:

$$T_1' \prec T_3', \quad T_1' \prec T_4' = T_4, \quad T_2' \prec T_4' = T_4.$$

Moreover, we have

$$T_2' \prec T_3';$$

indeed, if we assume $T_2' \succeq T_3'$ we either have $|T_2| = |T_3|$ or, by Fact 6, we have $|T_3| > |T_2|$ contradicting (14).

We notice now that $T_1' \neq T_2'$, since by (12) they have different weights. Hence, by the definition of $\prec$ (cfr. Definition 7), we get that the only possible orderings on the the subtrees of T' are:

$$T_1' \prec T_2' \prec T_3', \quad T_1' \prec T_2' \prec T_4', \quad T_2' \prec T_1' \prec T_3', \quad T_2' \prec T_1' \prec T_4'.$$

Moreover, both sequences of steps $\langle T_1 T_2 T_3 \rangle \langle T_1' T_2' \rangle$ and $\langle T_1 T_2 T_3 \rangle \langle T_2' T_1' \rangle$ are possible during the execution of the algorithm on T; in particular if $T_2' \prec T_1'$ we know by Fact 6 that $\beta_2 = 0$.

Hence, after the first 3 steps, the algorithm on T proceeds as on input T'. For T' we have:

$$|T'| = |T| - \begin{cases} 3 & \text{if } |T_3| > 0 \\ 2 & \text{otherwise} \end{cases}, \qquad \epsilon_{T'} = \epsilon_T = 0 \text{ (since } |T_1| < |T_2|), \qquad \tau_1' = \tau_1 - 3 \text{ (since } \beta_1 > 1).$$

In case $T_1' \prec T_2'$, it holds

$$\Delta_{1,2}' = \Delta_{1,2} - 3, \qquad \Delta_{2,1}' = \begin{cases} \Delta_{2,1} - 3 & \text{if } \beta_2 > 0 \\ |T_2'| + |T_1'| - 1 < |T'| & \text{if } \beta_2 = 0 \end{cases}, \qquad \Delta_{1,3}', \Delta_{1,4}' < \Delta_{1,2} - 3,$$

where the last inequality follows from (14).

In case $T'_2 \prec T'_1$, by Fact 6 we have $\beta_2 = 0$, $\beta_1 \geq 1$ and $\tau_1 > \tau_2$; hence $\tau'_2 = \tau_2 - 2 = \tau_1 - 3$ and

$$\Delta'_{1,2} = \Delta_{1,2} - 3, \qquad \Delta'_{2,i} = |T'_2| + |T'_i| + \beta'_2 - 1 = |T'_2| + |T'_i| - 1 < |T'| \ (i = 1, 3, 4).$$

Summarizing, in both cases $T'_1 \prec T'_2$ and $T'_2 \prec T'_1$, inequality (10) holds.

Case 2: $Max(T) = \Delta_{2,1} > \max\{\tau_1 + \epsilon_T, |T|\}$**.**

We first notice that by definition of $\Delta_{2,1}$ and using $\Delta_{2,1} > |T|$, we get

$$2 \leq |T| - |T_1| - |T_2| + 2 \leq \beta_2, \tag{16}$$

Using Fact 4 and $\Delta_{2,1} > \tau_1 \geq \tau_2$ we get

$$|T_2| < |T_1|. \tag{17}$$

Moreover, since $\Delta_{2,1} > \tau_1 + \epsilon_T$, we get

$$|T_1| + \beta_1 < |T_2| + \beta_2, \tag{18}$$

which also implies $T_1 \neq T_2$ and

$$\epsilon_T = 0. \tag{19}$$

Finally, from (16) we have $|T_3| \leq |T| - |T_1| - |T_2| \leq \beta_2 - 2$; from this and Fact 1 it follows

$$\tau_3 \leq 3|T_3| - 3 \leq 3\beta_2 - 9 \leq \tau_2 - 9. \tag{20}$$

We distinguish now two subcases on the value of β_1.

- $\beta_1 \geq 1$.
  The first tree steps of the algorithm are therefore $\langle T_1 T_2 T_3 \rangle$. Let T' be the tree resulting after $\langle T_1 T_2 T_3 \rangle$. From (16) and (20) and recalling that $\tau'_1 = \tau_1 - 3 \geq \tau_2 - 3 = \tau'_2$, we obtain that in the tree T':

  $$T'_1 \prec T'_2, \qquad T'_2 \prec T'_3, T'_4.$$

  Hence, after the first 3 steps the algorithm on T proceeds as on input T'. For T' we have:

  $$T'_1 \prec T'_2 \prec T'_3 \qquad \text{or} \qquad T'_1 \prec T'_2 \prec T'_4 = T_4 \prec T'_3.$$

  Moreover,

  $$|T'| = |T| - \begin{cases} 3 & \text{if } |T_3| > 0 \\ 2 & \text{otherwise} \end{cases}, \qquad \epsilon_{T'} = \epsilon_T = 0 \text{ (since } |T_1| \neq |T_2|), \qquad \tau'_1 = \tau_1 - 3 \text{ (since } \beta_1 > 0).$$

  $$\Delta'_{1,2} = \Delta_{1,2} - 3, \qquad \Delta'_{2,1} = \Delta_{2,1} - 3, \qquad \Delta'_{1,3} = \begin{cases} \Delta_{1,3} - 3 & \text{if } |T_3| > 0 \\ \Delta_{1,3} - 2 \leq \Delta_{1,2} - 3 & \text{otherwise} \end{cases};$$

  similarly, if $T_4 \prec T'_3$, from Fact 5 one has $\Delta'_{1,4} \leq \Delta_{2,1} - 3$.

  Summarizing, it holds $Max(T') = Max(T) - 3$. Therefore,

  $$\mathcal{T}(T) = 3 + \mathcal{T}(T') \leq 3 + Max(T') = Max(T).$$

- $\beta_1 = 0$.
  The first two steps of the algorithm are $\langle T_1 T_2 \rangle$. Let T' be the tree resulting after $\langle T_1 T_2 \rangle$. Recalling that $\beta_2 \geq 2$ and that $\tau_1 - 2 > \tau_3$ (cfr. (16) and (20)), we obtain that the first 4 steps of the algorithm are

  $$\langle T_1 T_2 \rangle \langle T_1' T_3' = T_3 \rangle.$$

  Let T'' be the tree resulting after $\langle T_1 T_2 \rangle \langle T_1' T_3' \rangle$. Using (20), we have

  $$\tau_4'' = \tau_4 \leq \tau_3 \leq \tau_2 - 9 = \tau_2' - 6 = \tau_2'' - 6.$$

  Hence, after the first 4 steps the algorithm on T proceeds as on input T'' where the subtrees with largest shade are T_1'' and T_2'' with

  $$T_1'' \prec T_2'' \qquad \text{or} \qquad T_2'' \prec T_1''$$

  followed by T_3'' and T_4'' in some order. Moreover,

  $$|T''| = |T| - \begin{cases} 4 & \text{if } |T_3| > 0 \\ 3 & \text{otherwise} \end{cases}, \quad \epsilon_{T'} = \epsilon_T = 0 \text{ (since } \beta_2 \geq 2, \beta_1 = 0), \quad \tau_1'' = \tau_1 - 4 \text{ (since } \beta_1 = 0),$$

  Therefore $|T''| \leq \Delta_{2,1} - 4 = Max(T) - 4$ and $\tau_1'' < Max(T) - 4$
  Moreover, if $T_2'' \prec T_1''$

  $$\tau_2'' = \tau_2 - 3 \leq \tau_1 - 3 \leq \Delta_{2,1} - 4 = Max(T) - 4.$$

  Finally,

  $$\Delta_{1,2}^{''}, \Delta_{1,3}^{''}, \Delta_{1,4}^{''} < |T''| \text{ (since } \beta_1 = 0), \qquad \Delta_{2,1}^{''} = \Delta_{2,1} - 4,$$

  and, by Fact 3,

  $$\Delta_{2,3}'', \Delta_{2,4}^{''} \leq |T_2''| + |T_3''| + |T_4''| + \beta_2'' - 1 \leq |T_2''| + |T_3''| + |T_4''| + |T_1''| - 3 < |T''|.$$

  Summarizing, it holds $Max(T'') \leq Max(T) - 4$. Therefore,

  $$\mathcal{T}(T) = 4 + \mathcal{T}(T') \leq 4 + Max(T') \leq Max(T).$$

Case 3: $Max(T) = \Delta_{1,3} > \max\{\tau_1 + \epsilon_T, |T|\}$**.**
In this case we have $\beta_1 > 1$, otherwise $\Delta_{1,3} \leq |T|$; hence, the first tree steps of the algorithm are

$$\langle T_1 T_2 T_3 \rangle.$$

By definition of $\Delta_{1,3}$ and using Fact 4, we get like in case 1

$$|T| - |T_1| - |T_3| < \beta_1 - 1, \tag{21}$$

$$|T_1| < |T_3|. \tag{22}$$

By (21) and Fact 1, we get

$$\tau_3 \leq \tau_2 \leq 3|T_2| - 3 \leq 3(|T| - |T_1| - |T_3|) - 3 < 3\beta_1 - 6 \leq (\tau_1 - 3) - 6 = \tau_1 - 9. \quad (23)$$

from which it follows

$$\tau_2' \leq \tau_2 \leq \tau_1 - 9 \leq \tau_1' - 6, \qquad \tau_3' \leq \tau_3 \leq \tau_2 \leq \tau_1' - 6. \quad (24)$$

Moreover, by (21) and (22) we have

$$|T_3| \geq |T_1| + 1 \geq \beta_1 + \alpha_1 + 2 \geq \beta_1 + 3 \geq (|T| - |T_1| - |T_3|) + 4 \geq |T_2| + |T_4| + 4; \quad (25)$$

which, by Fact 1, implies

$$\tau_3 \geq 2|T_3| - 1 \geq 2(|T_2| + |T_4|) + 7 \geq 4\min\{|T_2|, |T_4|\} + 7.$$

Noticing that Fact 1 implies $4|T_4| \geq \tau_4$ and $4|T_2| > \tau_2 \geq \tau_4$, we get

$$\tau_2 \geq \tau_3 \geq \tau_4 + 7. \quad (26)$$

Let T' be the tree resulting after $\langle T_1 T_2 T_3 \rangle$. Recalling that $\beta_1 \geq 2$ and using (24) and (26), we have

$$\tau_1' = \tau_1 - 3, \qquad \tau_4 \leq \tau_2' \leq \tau_2 \leq \tau_1' - 6, \qquad \tau_4 \leq \tau_3' \leq \tau_3 \leq \tau_2 \leq \tau_1' - 6. \quad (27)$$

From this we obtain that the only possible orderings of the first subtrees of T' are:

$$T_1' \prec T_2' \preceq T_3', \qquad T_1' \prec T_3' \prec T_2'.$$

We notice that both sequences of steps $\langle T_1 T_2 T_3 \rangle \langle T_1' T_2' T_3' \rangle$ and $\langle T_1 T_2 T_3 \rangle \langle T_1' T_3' T_2' \rangle$ are possible during the execution of the algorithm on T; in particular if $T_3' \prec T_2'$ we know by Fact 6 that $\beta_3 = 0$. Hence, after the first 3 steps, the algorithm on T proceeds as on input T'.
For T' we have:

$$|T'| = |T| - 3, \quad \epsilon_{T'} = \epsilon_T = 0 \text{ (by (24) and (27))}, \quad \tau_1' = \tau_1 - 3 \text{ (by (27))},$$

$$\Delta_{1,2}' \leq \Delta_{1,2} - 3, \quad \Delta_{1,3}' \leq \Delta_{1,3} - 3, \quad \Delta_{2,1}' \leq \begin{cases} \Delta_{2,1} - 3 & \text{if } \beta_2 > 0 \\ |T'| & \text{otherwise} \end{cases}$$

and, if $T_3' \prec T_2'$, $\beta_3 = 0$ and

$$\Delta_{3,1}' = |T_3'| + |T_1'| - 1 < |T'|.$$

Hence $Max(T') \leq Max(T) - 3$ and

$$\mathcal{T}(T) = 3 + \mathcal{T}(T') \leq 3 + Max(T') \leq Max(T).$$

==========

Case 4: $Max(T) = \max\{\tau_1 + \epsilon_T, |T|\}$**.**
Let k be the largest integer such that the first k steps of the algorithm are

$$\langle T_1 T_2 \dots T_k \rangle,$$

and, letting T' be the tree resulting after $\langle T_1 T_2 \dots T_k \rangle$, it holds $T_{k+1} \not\prec T'_i$, for each $i = 1, \dots, k$.

• First assume that either $k \geq 4$ or $k = 3$ and $T_3 \preceq T'_1, T'_2$. Let the ordering on T' be such that

$$T'_i \preceq T'_j \preceq T'_\ell$$

i.e., T'_i has the largest shade among all the subtrees of T', followed by T'_j and by some T'_ℓ.
Moreover,

a) during the execution of the algorithm on T, any sequences of steps $\langle T_1 T_2 \dots T_k \rangle \langle T'_i \rangle$ is possible, for any $1 \leq i \leq k-1$.
Indeed, any of the subtrees $T'_1, T'_2, \dots, T'_{k-2}$ is available at step $k+1$ and i can assume any value among $1, \dots, k-2$. Moreover, if $T'_{k-1} \prec T'_1$ then Fact 6 implies $\beta_{k-1} = 0$; hence T'_{k-1} is available at step $k+1$ and $i = k-1$ can hold.

b) if either $T'_k \prec T'_1$ or $T'_k \prec T'_2$ then by Fact 6 we have $\beta_k = 0$, and this implies that T'_k is available at step $k+2$ and j can assume value k.

We now distinguish two cases according to the value of i.

- Let $i \leq k-1$.
By a) and b) , we have that after the first k steps, the algorithm on T proceeds as on input T'. For T' we have:

$$|T'| = |T| - k, \qquad \tau'_i \leq \tau_i - 2 \leq \begin{cases} \tau_1 - 3 & \text{if } i \geq 2 \text{ (by Fact 6)} \\ \tau_1 - 2 & \text{if } i = 1 \end{cases} ;$$

furthermore,

$$\begin{array}{ll} \beta_1, \beta_2 < |T_3|, \dots, |T_k| & \text{(by Fact 7)} \\ \beta_i = \beta_j = 0 & \text{for } 3 \leq i \leq k-1 \text{ and } 3 \leq j \leq k \text{ (by Fact 6 since } T'_i \preceq T'_j \prec T'_1 \text{ or } T'_2) \\ \beta_j \leq |T_1| - 2, |T_2| - 2 & \text{for } j > k \text{ (by Fact 3 since } T_1, T_2 \preceq T_j) \end{array}$$

this and Fact 3 imply that

$$\Delta'_{i,j}, \Delta'_{j,i}, \Delta'_{i,\ell} \leq |T'|.$$

Hence,

$$\begin{array}{rcl} Max(T') & = & \max\{|T'|, \tau'_i + \epsilon_{T'}\} \\ & \leq & \begin{cases} \max\{|T| - k, \tau_1 - 3 + \epsilon_{T'}\} & \text{if } i \geq 2 \text{ or } i = 1 \text{ and } \beta_1 \geq 1 \\ \max\{|T| - k, \tau_1 - 2 + \epsilon_{T'}\} & \text{if } i = 1 \text{ and } \beta_1 = 0 \end{cases} \end{array}$$

Since $\epsilon_{T'} \leq 1$ and by i) of Fact 7, we have that if either $i \geq 2$ or $i = 1$ and $\beta_1 \geq 1$ then $\max\{|T| - k, \tau_1 - 3 + \epsilon_{T'}\} = |T| - k$.

Let us consider now $i = 1$ and $\beta_1 = 0$.
Obviously if $\epsilon_{T'} = 0$ then by Fact 7 we have $\max\{|T| - k, \tau_1 - 2 + \epsilon_{T'}\} = |T| - k$.
Assume then $\epsilon_{T'} = 1$. In this case $T'_1 = T'_j$ for some $j \geq 2$. This implies that

$$|T| \geq |T_1| + |T_j| + k - 2 = 2(\alpha_1 + 1) + k - 2 = 2\alpha_1 + k = \tau_1 + k - 1$$

and $\max\{|T| - k, \tau_1 - 2 + \epsilon_{T'}\} = \max\{|T| - k, \tau_1 - 1\} = |T| - k$. Hence, we have

$$Max(T') \leq |T| - k,$$

and

$$\mathcal{T}(T) = k + \mathcal{T}(T') \leq k + Max(T') \leq |T| \leq Max(T).$$

- Let $i = k$.
In this case we have $T'_k \prec T'_1, T'_2, \ldots, T'_{k-1}$. Hence, by Fact 6 we get $\beta_1, \ldots, \beta_{k-1} \geq 1$ and $\beta_k = 0$ that imply $T'_1 \preceq \ldots \preceq T'_{k-1}$. Since a path to T'_k cannot be set at step $k + 1$, we obtain that the first $k + 1$ steps of the algorithm are

$$\langle T_1 T_2 \ldots T_k \rangle \langle T'_1 \rangle.$$

Let T'' be the tree resulting after $\langle T_1 T_2 \ldots T_k \rangle \langle T'_1 \rangle$. We have that, after the first $k+1$ steps, the algorithm on T proceeds as on input T'' where the subtree with largest shade is T''_k followed by T''_2, T''_3 in this order; i.e.,

$$T''_k \prec T''_2 \preceq T''_3.$$

Moreover,

$|T''| = |T| - k - 1$, $\quad \tau''_k = \tau'_k = \tau_k - 2 = \tau_1 - 3$ (by Fact 6) , $\quad \epsilon_{T''} = 0$ (since $T''_k \prec T''_2$)

$\Delta''_{k,2}, \Delta''_{k,3} < |T''|$ (since $\beta_k = 0$), $\quad \Delta''_{2,k} \leq |T''_k| + |T''_2| + |T''_1| - 3 \leq |T''| - 3$ (by Fact 3)

Hence, by using Fact 7 we get $Max(T'') = \max\{|T''|, \tau''_k + \epsilon_{T''}, \Delta''_{k,2}, \Delta''_{2,k}, \Delta''_{k,3}\} \leq \max\{|T| - k - 1, \tau_1 - 3\} = |T| - k - 1 = |T''|$ and

$$\mathcal{T}(T) = k + 1 + \mathcal{T}(T'') \leq k + 1 + Max(T'') \leq k + 1 + |T''| = |T| \leq Max(T).$$

• Assume now that $k = 3$ and either $T'_1 \prec T_3$ or $T'_2 \prec T_3$ (including also the case $T_3 = \emptyset$). Suppose that $\beta_1 = 0$: since $k = 3$, we can deduce that $T'_1 \succ T_3$. Therefore, $T'_2 \prec T_3 \prec T'_1$. By Fact 6, we get to the contradiction $\beta_1 \geq 2$. Hence, throughout this case we can assume

$$\beta_1 \geq 1 \text{ and } \tau'_1 = \tau_1 - 3. \tag{28}$$

Furthermore,

$$|T'| = |T| - 3 \qquad \Delta'_{1,2} = \Delta_{1,2} - 3, \qquad \Delta'_{2,1} \leq \begin{cases} \Delta_{2,1} - 3 & \text{if } \beta_2 \geq 1 \\ |T| - 3 & \text{if } \beta_2 = 0 \end{cases}, \qquad \Delta'_{1,3} = \Delta_{1,3} - 3, \tag{29}$$

and by Fact 7

$$\Delta'_{1,4} = |T_1| + |T_4| + \beta_1 - 1 - 2 \leq |T_1| + |T_4| + |T_2| - 5 < |T| - 5. \tag{30}$$

Moreover, by Fact 6 we get

$$\text{if } T'_2 \prec T'_1 \quad \text{then} \quad \beta_2 = 0, \ \ \tau'_2 = \tau_2 - 2 = \tau_1 - 3 \text{ and } \Delta'_{2,3}, \Delta'_{2,4} < |T| - 3 \tag{31}$$

$$\text{if } T'_3 \prec T'_2, T'_1 \quad \text{then} \quad \beta_3 = 0, \text{ and } \Delta'_{3,1}, \Delta'_{3,2} < |T| - 3, \tag{32}$$

$$\text{if } T'_4 \prec T'_2, T'_1 \quad \text{then} \quad \beta_4 = 0, \text{ and } \Delta'_{4,1}, \Delta'_{4,2} < |T| - 3. \tag{33}$$

We show now that in each of the possible orderings on the subtrees of T', it holds

$$Max(T') \leq \max\{|T| - 3, \tau_1 + \epsilon_T - 3\} = Max(T) - 3; \tag{34}$$

indeed:

- If either $T'_1 \preceq T'_2 \preceq T'_3$ or $T'_1 \preceq T'_2 \preceq T_4 \prec T'_3$ then we have (34) by using (28), (29), (30) and considering that $\epsilon_{T'} = \epsilon_T$;
- if either $T'_1 \preceq T'_3 \prec T'_2$ or $T'_1 \preceq T'_3 \preceq T_4 \prec T'_2$ then we have (34) by using (28), (29), (32), (30) and considering that $\epsilon_T \leq 1$ and that $\epsilon_{T'} = 0$ (since $\beta_1 \geq 1$ and $\beta_3 = 0$);
- if either $T'_1 \preceq T_4 \prec T'_2$ or $T'_1 \preceq T_4 \prec T'_3 \prec T'_2$ then we have (34) by using (28), (29), (33), (30) and considering that $\epsilon_T \leq 1$ and that $\epsilon_{T'} = 0$ (since $\beta_1 \geq 1$ and $\beta_4 = 0$);
- if either $T'_2 \prec T'_1 \preceq T'_3$ or $T'_2 \prec T'_1 \preceq T_4 \prec T'_3$ then we have (34) by using (31), (29) and considering that $\epsilon_T = 0$ (since $\beta_1 \geq 1$ and $\beta_2 = 0$) and that $\epsilon_{T'} = 0$ (since $T'_2 \prec T'_1$);
- if either $T'_2 \preceq T'_3 \prec T'_1$ or $T'_2 \prec T'_3 \preceq T'_4 \prec T'_1$ then we have (34) by using (31), (29), (32) and considering that $\epsilon_T = 0$ (since $\beta_1 \geq 1$ and $\beta_2 = 0$), that $\tau'_1 = \tau'_2$ (by Fact 6) and that if $\epsilon_{T'} = 1$ then $\beta_2 = \beta_3 = 0$, $\alpha_2 = \alpha_3$ and $|T| > |T_1| + |T_2| + |T_3| = |T_1| + 2\alpha_2 + 2 \geq \tau'_2 + 4 = \tau'_1 + 4$;
- if either $T'_2 \preceq T_4 \prec T'_1$ or $T'_2 \preceq T_4 \prec T'_3 \prec T'_1$ then we have (34) by using (31), (29), (33) and reasoning as in the previous case.

Hence, by (34) we have

$$\mathcal{T}(T) = 3 + \mathcal{T}(T') \leq 3 + Max(T') \leq Max(T).$$

• Finally, consider the last possible case: $k = 2$, $T'_1 \preceq T_3$ and $\beta_1 = 0$.
This means that the first two steps of the algorithm are $\langle T_1 T_2 \rangle$ and at the third step a path to T'_1 can be set. Hence, we have

$$\tau'_1 = \tau_1 - 2.$$

Noticing that, by Fact 6, if $\beta_1 = 0$ then $T'_2 \not\prec T'_1$, we distinguish 4 cases according to the relation between T'_2 and T_3.

- If $T'_1 \preceq T'_2 \preceq T_3$ and $\beta_2 = 0$ then

  $$|T'| = |T| - 2, \qquad \epsilon_{T'} = \epsilon_T \qquad \Delta'_{1,2} = \Delta'_{2,1} = |T_1| + |T_2| - 3 \leq |T| - 3, \qquad \Delta'_{1,3} < |T| - 2.$$

  Hence, $Max(T') = \max\{|T| - 2, \tau_1 + \epsilon_T - 2\} = Max(T) - 2$ and

  $$\mathcal{T}(T) = 2 + \mathcal{T}(T') \leq 2 + Max(T') = 2 + Max(T) - 2 = Max(T).$$

- If $T'_1 \preceq T'_2 \preceq T_3$ and $\beta_2 > 0$ (except for $\beta_2 = 1$ and $T_3 = \emptyset$) then the first 4 steps of the algorithm are

  $$\langle T_1 T_2 \rangle \langle T'_1 T'_3 \rangle$$

  where $T'_3 = T_3$. Let T'' be the tree resulting after $\langle T_1 T_2 \rangle \langle T'_1 T'_3 \rangle$. Hence, after the first 4 steps the algorithm on T proceeds as on input T''.

  First notice that in this case $T''_1 \preceq T''_3$ since otherwise by Fact 6 it should be $\beta_1 \geq 2$; furthermore, $T''_2 = T'_2 \prec T'_3$ since $T'_2 \preceq T_3$. Hence, the possible orderings of the subtrees of the tree T'' are:

  $$T''_1 \preceq T''_2 \prec T''_3, \qquad T''_1 \preceq T''_2 \prec T''_4, \qquad T''_2 \prec T''_1 \preceq T''_3, \qquad T''_2 \prec T''_1 \preceq T''_4, \qquad T''_2 \prec T''_4 \prec T''_1.$$

  For T'' we have

  $$|T''| = |T| - 4, \qquad \Delta''_{1,2}, \Delta''_{1,3}, \Delta''_{1,4} < |T| - 4 = |T''| \text{ (since } \beta_1 = 0)$$

  $$\Delta''_{2,1} = \Delta_{2,1} - 4$$

  $$\Delta''_{2,3}, \Delta''_{2,4} < |T| - 4 = |T''| \text{ (since } \beta_2 \leq T_1 - 2 \text{ by Fact 3)}$$

  $$\Delta''_{4,2} < |T| - 4 = |T''| \text{ (since } \beta_4 \leq T_1 - 2 \text{ by Fact 3)}$$

  To bound $Max(T')$, we distinguish two cases according to the relation between T''_2 and T''_1. First notice that $\epsilon_T = 0$ since $\beta_1 = 0$ and $\beta_2 > 0$.

  Let $T''_1 \preceq T''_2$. Since

  $$\epsilon_{T''} = 1 \text{ iff } \beta_2 = 1 \text{ and } \alpha_2 = \alpha_1 - 2$$

  we have

  $$\text{if } \epsilon_{T''} = 1 \text{ then } |T| \geq |T_1| + |T_2| + |T_3| > |T_1| + |T_2| = 2\alpha_1 + 1 = \tau_1$$

  and

  $$\tau''_1 + \epsilon_{T''} = \begin{cases} \tau_1 - 3 \leq |T| - 4 & \text{if } \epsilon_{T''} = 1 \\ \tau_1 - 4 & \text{if } \epsilon_{T''} = 0 \end{cases}$$

Hence,

$$\begin{aligned} Max(T'') &= \max\{|T''|, \tau_1'' + \epsilon_{T''}, \Delta_{2,1}''\} \\ &\leq \begin{cases} \max\{|T|-4, \Delta_{2,1}-4\} \leq Max(T)-4 & \text{if } \epsilon_{T''}=1 \\ \max\{|T|-4, \tau_1-4, \Delta_{2,1}-4\} \leq Max(T)-4 & \text{if } \epsilon_{T''}=0 \end{cases} \end{aligned} \quad (35)$$

Let $T_2'' \prec T_1''$. Since $T_2'' \neq T_1''$ we have $\epsilon_{T''} = 0$. Furthermore, since $T_1 \preceq T_2$, $\beta_1 = 0$ and $\beta_2 > 0$ we have $|T_1| > |T_2|$. Hence,

$$\Delta_{2,1}'' = |T_1| + |T_2| + \beta_2 - 5 \geq 2|T_2| + \beta_2 - 4 = 3\beta_2 + 2\alpha_2 + 2 - 4 = \tau_2 - 3 = \tau_2''$$

and

$$Max(T'') = \max\{|T''|, \Delta_{2,1}''\} = \max\{|T|-4, \Delta_{2,1}-4\} \leq Max(T)-4. \quad (36)$$

By (35) and (36) we have

$$\mathcal{T}(T) = 4 + \mathcal{T}(T') \leq 4 + Max(T') = Max(T).$$

- If $T_1' \preceq T_2'$ and $T_3 \prec T_2'$ then the only possible orderings of the subtrees of the tree T' are:

$$T_1' \prec T_3 \preceq T_2', \qquad T_1' \preceq T_3 \preceq T_4.$$

For T' we have

$$|T'| = |T| - 2, \qquad \Delta_{1,2}', \Delta_{1,3}', \Delta_{1,4}' < |T| - 2 \text{ (since } \beta_1 = 0)$$

$$\Delta_{3,1}' \leq |T_3| + |T_1| + |T_2| - 4 < |T| - 4 \text{ (since } \beta_3 \leq T_2 - 2 \text{ by Fact 3).}$$

Hence, if $\epsilon_{T'} \leq \epsilon_T$ then we have $Max(T') \leq Max(T) - 2$.

Suppose now that $\epsilon_{T'} = 1$ and $\epsilon_T = 0$. We have $T_1' = T_3$, $\alpha_3 = \alpha_1 - 1$, $\beta_1 = 0 = \beta_3$ and $|T| \geq |T_1| + |T_2| + |T_3| = |T_2| + 2\alpha_1 + 1 \geq \tau_1 + 1$; hence,
$Max(T') = \max\{|T'|, \tau_1' + \epsilon_{T'}\} = \max\{|T|-2, \tau_1 - 1\} = |T| - 2 \leq Max(T) - 2$.

In any case we get

$$\mathcal{T}(T) = 2 + \mathcal{T}(T') \leq 2 + Max(T') \leq Max(T).$$

- If $T_1' \preceq T_2'$, $\beta_2 = 1$ and $T_3 = \emptyset$ then after the first step $\langle T_1 \rangle$ on T, we get that the algorithm continues as having in input a tree $\overline{T}$ corresponding to the special case considered in the base. For $\overline{T}$ we have: $|\overline{T}| = |T| - 1$, $\overline{\tau}_1 = \tau_2$, and $Max(\overline{T}) = \tau_2$. We get that $Max(\overline{T}) = \tau_2 \leq \tau_1 - 1$. Hence

$$\mathcal{T}(T) = 1 + \mathcal{T}(\overline{T}) \leq 1 + Max(\overline{T}) \leq \tau_1 \leq Max(T).$$

□

References

[1] Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. : Wireless sensor networks: a survey. Computer Networks, **38** (2002) 393–422.

[2] Bermond J.-C., Galtier J., Klasing R., Morales N., Perennes S.: Hardness and approximation of gathering in static radio networks. Proceedings FAWN06 (2006).

[3] V. Bonifaci, P. Korteweg, A. Marchetti-Spaccamela, L. Stougie, An Approximation Algorithm for the Wireless Gathering Problem, Proc. of SWAT 06, 2006.

[4] Chong C-Y, Kumar S.P.: Sensor networks: Evolution, opportunities, and challenges. Proceedings of the IEEE ,**91 (8)** (2003) 1247-1256.

[5] S. Coleri, P. Varaiya, Energy Efficient Routing with Delay Guarantee for Sensor Networks, Wireless Networks, to appear

[6] Dasgupta K., Kukreja M., Kalpakis K.: Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks. Proceedings IEEE ISCC'03 (2003) 341-348.

[7] Falck E., Floreen P., Kaski P., Kohonen J., Orponen P.: Balanced data gathering in Energy-constrained sensor networks. Proceedings of ALGOSENSORS 2004, LNCS 3121, 2004, 59-70.

[8] Florens C., Franceschetti M., McEliece R.J. Lower Bounds on Data Collection Time in Sensory Networks. IEEE Journal on Selected Areas in Communications, **22 (6)** (2004) 1110–1120.

[9] Ganesan D., Cristescu R., Beferull-Lozano B.: Power-efficient sensor placement and transmission structure for data gathering under distortion constraints. IPSN 2004, (2004) 142-150.

[10] Gupta P., Kumar P.R.: The Capacity of Wireless Networks. IEEE Transactions on Information Theory **46 (2)** (2000) 388–404.

[11] Gupta H., Navda V., Das S.R., Chowdhary V.: Efficient gathering of correlated data in sensor networks. Proceedings of ACM MobiHoc'05, 2005, 402-413.

[12] Gandhi R., Parthasarathy S., Mishra A.: Minimizing broadcast latency and redundancy in ad hoc networks. Proceedings of Int. Symposium on Mobile Ad Hoc Networking and Computing 2003, 2003.

[13] nostro

[14] Heinzelman W.R., Kulik J., Balakrishnan H.: Adaptive protocols for information dissemination in wireless sensor networks. Proceedings of ACM MobiCom 99, (1999) 174-185.

[15] Ho B., Prasanna V.K.: Constrained flow optimization with application to data gathering in sensor networks. Proceedings of ALGOSENSORS 2004, LNCS **3121** (2004) 187-200.

[16] Intanagonwiwat C., Govindan R., Estrin D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. Proceedings of ACM MobiCom 00 (2000) 56-67.

[17] Intanagonwiwat C., Govindan R., Estrin D., Heidemann J., Silva F.: Directed diffusion for wireless sensor networking, IEEE/ACM Trans. Netw. **11(1)**, (2003) 2-16

[18] L. Gargano, A.A. Rescigno, "Optimally Fast Data Gathering in Sensor Networks", Proc. 31st International Symposium on Mathematical Foundations of Computer Science (MFCS 2006), Lecture Notes in Computer Science , Vol. 4162, pp. 399-411, Springer Verlag.

[19] Krishnamachari B., Estrin D., Wicker S.: Modeling data-centric routing in wireless sensor networks. Proceedings of IEEE INFOCOM 2002, (2002).

[20] Kahn J.M., Katz R.H., Pister K.S.J.: Mobile Networking for Smart Dust. Proceedings of ACM MobiCom 99, (1999).

[21] Lindsey S., Raghavendra C.: Pegasis: Power-efficient gathering in sensor wireless networks. Proceedings of IEEE Aerospace Conference, 2002.

[22] Lindsey S., Raghavendra C., Sivalingam K.M.: Data gathering algorithms in sensor networks using energy metrics. IEEE Transactions on Parallel and Distributed Systems **13 (9)** (2002) 924-935.

[23] Mirkovic J., Venkataramani G.P., Lu S.,Zhang L.: A self-organizing approach to data forwarding in largescale sensor networks. Proceedings of IEEE Int. Conference on Communications ICC'01, (2001).

[24] Pelc A.: Broadcasting in radio networks . Handbook of Wireless Networks and Mobile Computing, I. Stojmenovic, Ed. John Wiley and Sons, Inc.,(2002) 509-528.

[25] Padmanabh K., Roy R.: Multicommodoty flow fased maximum lifetime routing in wireless sensor network. Proceedings of IEEE ICPADS 2006,(2006) 187-194

[26] Sohrabi K., Gao J.i, Ailawadhi V., Pottie G.: Protocols for Self-organization of a Wireless Sensor Network. IEEE Personal Communications, **7** (2000) 16–27.

[27] Shen C.,Srisathapornphat C., Jaikaeo C.: Sensor information networking architecture and applications. IEEE Personal Communications, (2001) 52-59.

[28] Yu Y., Krishnamachari B., Prasanna V.: Energy-latency tradeoffs for data gathering in wireless sensor networks. Proceedings of IEEE INFOCOM 2004, (2004).

[29] Zhu X., Tang B., Gupta H.: Delay efficient data gathering in sensor networks. Proceedings of MSN 2005, LNCS **3794** (2005) 380-389.