

A new characterization of tree medians with applications to distributed algorithms

O. GERSTEL* S. ZAKS*,[†]

September 1991

Abstract

A new characterization of tree medians is presented: we show that a vertex m is a median of a tree T with n vertices iff there exists a partition of the vertex set into $\lfloor n/2 \rfloor$ disjoint pairs (excluding m when n is odd), such that all the paths connecting the two vertices in any of the pairs pass through m . We show that in this case this sum is the largest possible among all such partitions, and we use this fact to discuss lower bounds on the message complexity of the distributed sorting problem. This lower bound implies that, given a network of a tree topology, choosing a median and then route all the information through it is the best possible strategy, in terms of worst-case number of messages sent during any execution of any distributed sorting algorithm. We also discuss the implications for networks of a general topology and for the distributed ranking problem.

1 Introduction

Many problems have been studied regarding location of specific points in networks, optimizing certain measures that have to do with traffic in the

*Department of Computer Science, Technion, Haifa, Israel.

[†]The work of this author was supported in part by the ESPRIT II Basic Research Actions Program of the EC under contract no. 3075 (project ALCOM), while visiting the Department of Computer Science, Aarhus University, DK-8000 Aarhus C, Denmark.

network. Among them, tree networks have been quite popular, and the notion of median is often used. A *median* is a vertex that minimizes the *average* distance from any other vertex (to complement, a *center* is a vertex that minimizes the *maximal* distance from any other vertex). For examples of location problems that concern these notions, the reader is referred to [5, 7, 8]; specifically, in [8] the complexity of determining p -medians in networks is studied, with 1-median being the regular median. The study of these notions goes back to [6]. In [13, 14] the notions of center and median are shown to be extreme cases of general optimality criteria. In [16] it is shown that a vertex is a median iff neither of the subtrees attached to it contains more than one half of the vertices of the tree.

We present a new characterization of tree medians: we show that a vertex m is a median of a tree T with n vertices iff there exists a partition of the vertex set into $\lfloor n/2 \rfloor$ disjoint pairs (excluding m when n is odd), such that all the paths connecting the two vertices in any of the pairs pass through m . We show that in this case this sum is the largest possible among all such partitions, and we use this fact to discuss lower bounds in distributed computations.

A distributed network is composed of processors, that are connected by communication lines. The processors can communicate only by exchanging messages along these lines (we use the commonly used message-passing model, as the one in [2]). Messages arrive without errors, loss or duplication, in an *asynchronous* fashion, that is, within a finite but otherwise arbitrary delay. A protocol has to be designed, that includes operations of sending messages, receiving messages and doing local computations, to solve a given problem. Due to the asynchrony in the network, more than one execution is usually possible for a given starting configuration of the network. For a protocol to solve a given problem it is required that each of its possible executions will result in a correct answer. The complexity of a protocol for a given configuration is the maximum number of messages sent during an execution that starts with this configuration.

We denote the processors as $1, 2, \dots, n$, and assume that processor i has a unique *identity* $id(i)$ for every i . For the sorting problem we also assume the existence of an *initial value* $init(i)$. We want to assign the processors *final values*, $final(i)$ being the final value of processor i . All these values are assumed to be integral.

The *distributed sorting* problem is the task of redistributing the initial values among the processors according to their identities. Formally, it is required that

$$\{\mathit{init}(i) \mid i = 1, 2, \dots, n\} = \{\mathit{final}(i) \mid i = 1, 2, \dots, n\}$$

(in case of repetitions they will be equal as multisets), and that

$$\forall i, j : \mathit{id}(i) < \mathit{id}(j) \Rightarrow \mathit{final}(i) \leq \mathit{final}(j).$$

In the related *distributed ranking* problem it is desired to assign to each processor its rank among the identities; namely, it is required to assign processor i a rank $\mathit{rank}(i) \in \{1, 2, \dots, n\}$ such that

$$\forall i, j : \mathit{id}(i) < \mathit{id}(j) \Rightarrow \mathit{rank}(i) \leq \mathit{rank}(j).$$

The distributed sorting and ranking problems are studied in [10, 15]. Various works also consider the bit complexity of these problems (*e.g.*, [3, 11]). This paper is a revision of that part of [4] that deals with message complexity.

Using our characterization for tree medians, we show that, given a network of a tree topology, choosing a median and then route all the information through it is the best possible strategy, in terms of worst-case execution of any distributed sorting algorithm. We also discuss the implication of this characterization for general networks and for the ranking problem.

In Section 2 we present preliminary notions concerning medians in graphs, and discuss their characterization for tree networks. In Section 3 we present our new characterization for medians in trees, as well as few graph theoretic results, used in Section 4, where we discuss the implication of this new characterization for the distributed sorting and ranking problems. Open problems are mentioned in Section 5.

2 Preliminaries

Let $G = G(V, E)$ be an undirected graph, with a vertex set V and an edge set E (we follow [1] for basic terminology). A vertex $u \in V$ such that $(u, v) \in E$

is called a *neighbor* of v . A *median* of a graph G is a vertex $m \in V$ that minimizes the average (shortest) distance to all vertices. Formally, for a vertex $v \in V$ let its *valence* $\Delta_G(v)$ be

$$\Delta_G(v) = \sum_{w \in V} d_G(v, w),$$

where $d_G(a, b)$ is the length (number of edges) of a shortest path connecting the vertices a and b in G . We use $\Delta(v)$ and $d(a, b)$ for $\Delta_G(v)$ and $d_G(a, b)$, respectively, when the graph G is clear from the text. Define $\Delta(G)$ as the minimum valence of a vertex in G ; namely,

$$\Delta(G) = \min_{v \in V} \Delta_G(v).$$

A vertex attaining this minimum is called a *median*; that is, a median is a vertex m satisfying

$$\Delta_G(m) = \Delta(G).$$

In [6] it is shown that in a tree there are either one or two adjacent medians.

The *size* $|T|$ of a tree T is the size of its vertex set V . For two vertices $v, w \in V$, denote by $T_{v,w}$ the largest subtree of T that includes w but does not include v . The following theorem characterizes tree medians in terms of the sizes of these subtrees:

Theorem Z (Zelinka [16]): Let T be a tree with a vertex set V , $|V| = n$, and let $v \in V$. The following two conditions are equivalent:

1. v is a median of T .
2. $|T_{v,w}| \leq n/2$ for every neighbor w of v .

□

This characterization is often used when dealing with tree medians. For example, it implies a trivial linear time sequential algorithm for identifying tree medians; this algorithm was used for a data-base application in [12]. The characterization was also used in [9] to obtain a distributed algorithm for finding medians in tree networks. In all these examples, the process of determining the median(s) starts at the leaves, and progresses towards the median(s), keeping track of the sizes of the subtrees.

3 The new characterization

Let G be a graph with vertex set V . A *pairing* P in G is a partition of the vertex set V into disjoint pairs, leaving at most one vertex unmatched (we term this vertex *free*(P)) when $|V|$ is odd. Formally,

$$P = \{ \{a_i, b_i\} \mid i = 1, \dots, \lfloor n/2 \rfloor, \forall i : a_i, b_i \in V, \forall i, j : a_i \neq b_j, \\ \text{and } \forall i \neq j : a_i \neq a_j \text{ and } b_i \neq b_j \}.$$

For such a pairing P , define

$$\Gamma_G(P) = \sum_{i=1}^{\lfloor n/2 \rfloor} d(a_i, b_i).$$

We use $\Gamma(P)$ for $\Gamma_G(P)$ when the graph G is clear from the text.

Define $\Gamma(G)$ as

$$\Gamma(G) = \max\{\Gamma_G(P) \mid P \text{ a pairing in } G\}.$$

A pairing P satisfying $\Gamma(G) = \Gamma_G(P)$ is called *maximal*.

Lemma 1: Let G be a graph. Then

$$\Gamma_G(P) \leq \Delta_G(v)$$

for every $v \in V$ and every pairing P in G .

Proof: Let

$$P = \{ \{a_i, b_i\} \mid i = 1, \dots, \lfloor n/2 \rfloor \}$$

be a pairing. Then

$$\begin{aligned} \Gamma_G(P) &= \sum_{i=1}^{\lfloor n/2 \rfloor} d(a_i, b_i) \leq \\ &\leq \sum_{i=1}^{\lfloor n/2 \rfloor} [d(a_i, v) + d(v, b_i)] \leq \sum_{u \in V} d(u, v) = \Delta_G(v). \end{aligned}$$

□

Lemma 2: For every graph G with vertex set V

$$\Gamma(G) \leq \Delta(G);$$

Moreover, if $\Gamma_G(P_0) = \Delta_G(v_0)$ for some vertex $v_0 \in V$ and some pairing P_0 in G , then P_0 is maximal and v_0 is a median; namely,

$$\Gamma_G(P_0) = \Gamma(G) \text{ and } \Delta_G(v_0) = \Delta(G).$$

Proof: Let v_1 be a median and P_1 a maximal pairing. Then by the definitions and Lemma 1 we get

$$\Gamma(G) = \Gamma_G(P_1) \leq \Delta_G(v_1) = \Delta(G).$$

If v_0 and P_0 satisfy $\Gamma_G(P_0) = \Delta_G(v_0)$, then for every pairing P Lemma 1 implies

$$\Gamma_G(P) \leq \Delta_G(v_0) = \Gamma_G(P_0),$$

hence

$$\Gamma_G(P_0) = \Gamma(G).$$

Similarly, for every vertex v we get

$$\Delta_G(v_0) = \Gamma_G(P_0) \leq \Delta_G(v),$$

hence

$$\Delta_G(v_0) = \Delta(G).$$

□

Lemma 3: Let T be a tree with a vertex set V and let $m \in V$ a median in T . There exists a pairing P in T such that the path between every pair $\{a, b\} \in P$ passes through m , and such that $free(P) = m$ when $|V|$ is odd.

Proof: Assume, by contradiction, that for every pairing P there exist a pair of vertices $\{u, w\} \in P$ such that the path connecting them does not pass through m . Let P' be a maximal pairing in G . Let $\{u, w\} \in P'$ such that the path connecting them does not pass through m . There exists a neighbor k of m such that $u, w \in T_{m,k}$. There also exists another pair $\{x, y\} \in P'$ such that $x, y \notin T_{m,k}$, since otherwise $T_{m,k}$ will contain more than $n/2$ vertices, contradicting Theorem Z.

Now construct P'' from P' by exchanging x and w ; that is,

$$P'' = P' \cup \{\{x, u\}, \{y, w\}\} - \{\{w, y\}, \{u, w\}\}.$$

Clearly

$$\Gamma(P'') = \Gamma(P') + [d(x, u) + d(y, w)] - [d(x, y) + d(u, w)].$$

Since both the path connecting x and u and the one connecting y and w pass through m , and since the path connecting u and w does not, we get

$$\begin{aligned} d(x, u) + d(y, w) &= [d(x, m) + d(m, u)] + [d(y, m) + d(m, w)] = \\ &= [d(x, m) + d(m, y)] + [d(u, m) + d(m, w)] > d(x, y) + d(u, w), \end{aligned}$$

so we have

$$\Gamma(P'') > \Gamma(P'),$$

a contradiction.

It remains to show that $free(P') = m$ in the case when $|V|$ is odd. If $free(P') \neq m$ then there exist two vertices $v, w \neq m$, such that $\{m, v\} \in P'$ and $free(P') = w$. Let

$$P'' = P' \cup \{\{v, w\}\} - \{\{m, v\}\}.$$

Clearly $free(P'') = m$.

Case 1: v, w are not in the same subtree $T_{m,x}$ for any neighbor x of m .

In this case

$$\Gamma(P'') = \Gamma(P') + d(v, w) - d(m, v)$$

and

$$d(v, w) = d(v, m) + d(m, w) > d(v, m).$$

Therefore $\Gamma(P'') > \Gamma(P')$, a contradiction.

Case 2: v, w are in the subtree $T_{m,x}$ for some x .

Choose some pair $\{a, b\} \in P'$ such that a and b are both not in $T_{m,x}$ (there exists such a pair, otherwise $T_{m,x}$ will contain more than $n/2$ vertices, contradicting Theorem Z). Let

$$P'' = P' \cup \{\{a, v\}, \{b, w\}\} - \{\{m, v\}, \{a, b\}\}.$$

We have

$$\begin{aligned} \Gamma(P'') &= \Gamma(P') + [d(a, v) + d(b, w)] - [d(m, v) + d(a, b)] = \\ &= \Gamma(P') + [d(a, m) + d(m, v)] + [d(b, m) + d(m, w)] - [d(m, v) + d(a, b)] = \\ &= \Gamma(P') + [d(a, m) + d(m, b)] - d(a, b) + d(m, w) \geq \Gamma(P') + d(w, m) > \Gamma(P'), \end{aligned}$$

a contradiction.

□

We now present our characterization for tree medians:

Theorem 4: Let T be a tree with a vertex set V , $|V| = n$, and let $v \in V$. The following conditions are equivalent:

1. v is a median of T .
2. There exists a pairing P in T (with v being the unmatched vertex in case n is odd), such that all the paths between its pairs pass through v .
3. There exists a pairing P in T such that $\Gamma_T(P) = \Delta_T(v)$.

Proof: 1 \Rightarrow 2 : By Lemma 3.

2 \Rightarrow 3 : By the assumption we have

$$d(a, b) = d(a, v) + d(v, b)$$

for every pair $\{a, b\} \in P$.

Therefore,

$$\begin{aligned} \Gamma_T(P) &= \sum_{\{a,b\} \in P} d(a, b) = \\ &= \sum_{\{a,b\} \in P} [d(a, v) + d(v, b)] = \\ &= \sum_{u \in V} d(u, v) = \Delta_T(v). \end{aligned}$$

3 \Rightarrow 1 : By Lemma 2.

□

In the next two lemmata we study the relation between $\Gamma(G)$ and $\Delta(G)$, needed for our discussion of lower bounds in the next section.

Lemma 5: $\Delta(T) = \Gamma(T)$ for every tree T .

Proof: Let v be a median of the tree T . By Theorem 4 there exists a pairing P in T for which $\Gamma_T(P) = \Delta_T(v)$. The lemma follows from Lemma 2.

□

Lemma 6: For every graph G ,

$$\Gamma(G) \leq \Delta(G) \leq 2 \cdot \Gamma(G),$$

and these bounds are best possible.

Proof: By Lemma 2 we have $\Gamma(G) \leq \Delta(G)$.

For the other part of the inequality, define a set \mathcal{P} of pairings in G as *complete* if

$$\forall a, b \in V \quad \exists P \in \mathcal{P} : \{a, b\} \in P.$$

Two pairings P_1 and P_2 are *distinct* if $P_1 \cap P_2 = \emptyset$.

There exists a complete set of distinct pairings in G . To see this, observe that this problem is equivalent to the 1-factorization of the complete graph K_{2n} (for the case of $|V|$ odd, one can use the 1-factorization of the complete graph K_{2n+2}). There are many such complete sets (see, e.g. [1]).

Consider a complete set of distinct pairings. Define

$$\Psi(G) = \sum_{(a,b) \in V} d_G(a, b).$$

Clearly

$$\Psi(G) = 2 \cdot \sum_{P \in \mathcal{P}} \Gamma(P).$$

Since $\Gamma(P) \leq \Gamma(G)$ for every $P \in \mathcal{P}$, and since $|P| \leq |V|$, we get

$$\Psi(G) \leq 2 \cdot |V| \cdot \Gamma(G).$$

On the other hand,

$$\Psi(G) = \sum_{a \in V} \sum_{b \in V} d(a, b) = \sum_{a \in V} \Delta(a).$$

But

$$\forall a \in V : \Delta(a) \geq \Delta(G),$$

hence

$$\Psi(G) \geq |V| \cdot \Delta(G),$$

and therefore

$$|V| \cdot \Delta(G) \leq \Psi(G) \leq 2 \cdot |V| \cdot \Gamma(G),$$

which implies

$$\Delta(G) \leq 2 \cdot \Gamma(G).$$

Since in any tree or a ring with an even number of vertices we have $\Gamma(G) = \Delta(G)$, and since in a complete graph with an odd number of vertices we have $\Delta(G) = 2 \cdot \Gamma(G)$, the bounds in this lemma are best possible.

□

4 Applications

We now discuss the implication of our characterization for the distributed sorting and ranking problems. We assume that the identities, initial values or ranks can only be compared with each other; this implies, for example, that in order for the initial value $init(i)$ residing in processor i to get to processor j in a network with topology G , at least $d_G(i, j)$ messages have to be transmitted, regardless of other traffic in the network. The proof of the following theorem is a modification of the one in [15]:

Theorem 7: Given a distributed network with a topology of a graph G , there exists a distribution of initial values for which every sorting (ranking) algorithm requires at least $3 \cdot \Gamma(G)$ ($2 \cdot \Gamma(G)$) messages during every execution.

Proof: Let $P = \{\{a_i, b_i\} \mid i = 1, \dots, \lfloor n/2 \rfloor\}$ be a maximal pairing. For the sorting problem assume that $id(i) = i$ for every processor i , and consider the distribution $init$ defined as follows: $init(a_i) = b_i$ and $init(b_i) = a_i$. If $|V|$ is odd and $free(P) = k$ then $init(k) = k$. Given any distributed sorting algorithm α , then for every pair $\{a_i, b_i\} \in P$, and during every execution of α , the values of a_i and b_i have to be compared, as well as their identities, and the resulting final values have to be transferred appropriately.

For the ranking problem consider the assignment of identities such that $id(a_i) = 2i - 1$ and $id(b_i) = 2i$, for every i . Given any distributed ranking algorithm α , then during every execution of α , the identities of of processors

a_i and b_i have to be compared, and the resulting ranks have to be transferred appropriately.

This amounts to a total of at least $3 \cdot \Gamma_G(P)$ ($2 \cdot \Gamma_G(P)$) messages for the sorting (ranking) problem.

□

Using Lemma 6, one can bound the number of messages in terms of $\Delta(G)$, which seems to be easier to estimate than $\Gamma(G)$. This is stated as follows:

Corollary 8: Given a distributed network with a topology of a graph G , there exists a distribution of initial values for which every sorting (ranking) algorithm requires at least $\frac{3}{2}\Delta(G)$ ($\Delta(G)$) messages during every execution.

□

Using Lemmata 5 and 6, one can obtain better bounds for tree networks:

Corollary 9: Given a distributed network with a topology of a tree T , there exists a distribution of initial values for which every sorting (ranking) algorithm requires at least $3 \cdot \Delta(T)$ ($2 \cdot \Delta(T)$) messages during every execution.

□

The algorithm in [15] assumes the existence of a spanning tree in a network, and then performs sorting and ranking through a center of this tree. It is shown that there are networks of a tree topology for which this algorithm is optimal. Using our characterizations we can state a stronger result; namely, by running the algorithm of [15] on a tree network, with the median as the root of the tree, one gets a distributed sorting (ranking) algorithm that uses (see [15]) at most $3 \cdot \Delta(T) + O(n)$ ($2\Delta(T) + O(n)$) messages during every execution; Corollary 9 implies that there are initial configurations for which this is also a lower bound, hence we get

Corollary 10: Given a tree network, by modifying the algorithm of [15] to use the median as the root of the tree, one gets a distributed sorting (ranking) algorithm that is optimal for every tree.

□

This corollary implies that, given a network of a tree topology, choosing a median and then route all the information through it is the best possible strategy, in terms of worst-case number of messages sent during any execution of any distributed sorting or ranking algorithm; in other words, if we are interested in the worst-case communication complexity, then for every tree we cannot avoid the non-distributed flavor of these problems, in which all the traffic will have to go through the median of the tree.

5 Open problems

- It follows from Lemma 6 that for every graph

$$1 \leq \Delta(G)/\Gamma(G) \leq 2.$$

Characterize those graphs for which

$$\Delta(G)/\Gamma(G) = \alpha,$$

for a constant $1 \leq \alpha \leq 2$, and study their applications for location problems.

- Our results apply only for the analysis of worst case executions. Extend these studies for average case analysis.
- Find additional applications of our new characterization of medians, for either graph theoretic studies or algorithmic ones.
- Our results show that a slight modification of the algorithm in [15] is optimal for every tree network. Design an optimal algorithm for the distributed sorting problem for general networks.
- The lower bound argument in the proof of Theorem 7 uses very strong assumptions. Show (or disprove) that these bounds hold also for weaker assumptions (*e.g.*, where any operations can be performed on the various values)

References

- [1] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, North-Holland, 1976.
- [2] R.G. Gallager, P.A. Humblet and P.M. Spira, *A distributed algorithm for minimum spanning tree*, *ACM Trans. on Programming Languages and Systems*, 5, 1, 1983, pp. 66–77.
- [3] O. Gerstel, Y. Mansour and S. Zaks, *Bit complexity of order statistics on a distributed star network*, *Information Processing Letters*, 30, 1989, pp. 127–132.
- [4] O. Gerstel and S. Zaks, *On the message complexity and bit complexity of distributed sorting*, TR-373, Department of Computer Science, Technion, Haifa, Israel, August 1985.
- [5] S.L. Hakimi, *Optimum locations of switching centers and the absolute centers and medians of a graph*, *Operations Res.*, 12, 1964, pp. 450–459.
- [6] C. Jordan, *Sur les assemblages des lignes*, *J. Reine Angew. Math.*, 70, 1869.
- [7] O. Kariv and S.L. Hakimi, *An algorithmic approach to network location problems. I: the p -centers*, *SIAM J. Appl. Math.*, Vol. 37, No. 3, December 1979, pp. 513–538.
- [8] O. Kariv and S.L. Hakimi, *An algorithmic approach to network location problems. II: the p -medians*, *SIAM J. Appl. Math.*, Vol. 37, No. 3, December 1979, pp. 539–560.
- [9] E. Korach, D. Rotem and N. Santoro, *Distributed algorithms for finding centers and medians in networks*, *ACM Trans. on Programming Languages and Systems*, Vol. 6, No. 3, July 1984, pp. 380–401.
- [10] E. Korach, D. Rotem and N. Santoro, *Distributed algorithms for ranking the vertices of a network*, *Proceedings of the 13th South-eastern Conference on Combinatorics, Graph Theory and Computing*, 1982, pp. 235–246.

- [11] M.C. Loui, *The complexity of Sorting on distributed systems*, *Information and Control*, Vol. 60, 1-3, 1984, Vol. 60, 1-3, 1984, pp. 70–85.
- [12] O. Wolfson and A. Milo, *The multicast policy and its relationship to replicated data placement*, *ACM Trans. on Database Systems*, Vol. 16, 1, 1991, pp. 181–2055.
- [13] K.B. Reid, *Centroids to centers in trees*, *Networks*, Vol. 21, 1991, pp. 11–17.
- [14] P.J. Slater, *Centers to centroids in graphs*, *J. of Graph Theory*, 2, 1978, pp. 209–222.
- [15] S. Zaks, *Optimal distributed algorithms for sorting and ranking*, *IEEE trans. on Computers*, c-34, 4, April 1985, pp. 376–379.
- [16] B. Zelinka, *Medians and peripherians of trees*, *Arch. Math. (Brno)*, 1968, pp. 87–95.