Article

A Note on Ultrametric Spaces, Minimum Spanning Trees and the Topological Distance Algorithm

Jörg Schäfer

Department of Computer Science and Engineering, Frankfurt University of Applied Sciences, Nibelungenplatz 1, D-60318 Frankfurt am Main, Germany; jschaefer@fb2.fra-uas.de

Received: 9 July 2020; Accepted: 26 August 2020; Published: 28 August 2020

Abstract: We relate the definition of an ultrametric space to the topological distance algorithm—an algorithm defined in the context of peer-to-peer network applications. Although (greedy) algorithms for constructing minimum spanning trees such as Prim’s or Kruskal’s algorithm have been known for a long time, they require the complete graph to be specified and the weights of all edges to be known upfront in order to construct a minimum spanning tree. However, if the weights of the underlying graph stem from an ultrametric, the minimum spanning tree can be constructed incrementally and it is not necessary to know the full graph in advance. This is possible, because the join algorithm responsible for joining new nodes on behalf of the topological distance algorithm is independent of the order in which the nodes are added due to the property of an ultrametric. Apart from the mathematical elegance which some readers might find interesting in itself, this provides not only proofs (and clearer ones in the opinion of the author) for optimality theorems (i.e., proof of the minimum spanning tree construction) but a simple proof for the optimality of the reconstruction algorithm omitted in previous publications too. Furthermore, we define a new algorithm by extending the join algorithm to minimize the topological distance and (network) latency together and provide a correctness proof.

Keywords: ultrametric spaces; greedy algorithms; matroids; greedoids; graphs; trees; minimum spanning trees; networks

MSC: 05C12; 68R10; 68M10

1. Introduction

Our motivation stems from algorithms for peer-to-peer network applications, as considered in [1,2]. In these papers a simple distance called topological distance based on IP-addresses has been introduced along with a greedy topological distance algorithm to compute (spanning) trees with minimum weight measured by this distance. It is important to note that—differently to, e.g., Prim’s algorithm—the topological distance algorithm computes spanning trees incrementally (see Remark 2 below) which is important for the applications considered. In this short note we relate the topological distance and the greedy algorithm to the theory of greedy algorithms in the context of matroids and greedoids, where the objective function stems from an ultrametric instead of a usual metric. Our contributions and motivations are threefold. First, our approach generalizes results of [1] to arbitrary metrics satisfying the axioms of ultrametric spaces. To the best of our knowledge, the connection between the topological distance algorithm, ultrametric spaces, and the abstract, discrete mathematics of matroids and greedoids has not been published before and hopefully some practitioners might find our remarks useful or appealing at the least. Secondly, our approach yields simpler and more elegant proofs of some of the results proven in [1] and enables extensions of the algorithms presented therein. And thirdly, even if the metric involved does not possess the properties
of an ultrametric, our results can nonetheless be useful for practical applications, as often the relevant metrics are close to an ultrametric. For example, in networking applications in which nodes are connected to each other via a combination of local and global area networks, the network parameters of distance and latency of the associated routing paths differ primarily due to the effects of the (few) long-distance routes involved, and henceforth can be approximated quite well by an ultrametric; see remarks in Section 3.3.

2. Greedy Algorithms and Optimality Results

2.1. Introduction and Notation

For the reader’s convenience and to make this note self-contained, we first fix our notation and recall the definitions of ultrametric spaces, matroids, and greedoids from [3–6] respectively.

**Definition 1.** A metric space \((X, d)\) is called ultrametric iff its metric \(d\) (distance function) satisfies the strong triangle inequality:

\[
d(x, z) \leq \max(d(x, y), d(y, z)) \quad \forall x, y, z \in X.
\] (1)

**Corollary 1.** Let \((X, d)\) be an ultrametric space; then any triangle is an acute, isosceles triangle; i.e., for any three points \(x, y,\) and \(z\) we have

\[
d(x, z) > d(x, y) \implies d(z, y) = d(x, z).
\] (2)

**Proof.** This follows immediately from the definition. \(\square\)

**Definition 2.** A matroid \(M\) on a finite ground set \(X\) is a pair \((X, \mathcal{F})\) with \(\mathcal{F} \subseteq 2^X\) satisfying:

- **M1** \(\emptyset \in \mathcal{F}\) (trivial axiom).
- **M2** \(A \subseteq B \in \mathcal{F}\) implies \(A \in \mathcal{F}\) (hereditary axiom).
- **M3** If \(A, B \in \mathcal{F}\) and \(|A| \geq |B|\), then there exists \(x \in A \setminus B\) such that \(B \cup \{x\} \in \mathcal{F}\) (exchange axiom).

A greedoid \(M\) on a finite ground set \(X\) is a pair \((X, \mathcal{F})\) with \(\mathcal{F} \subseteq 2^X\) satisfying **M1** and **M3**.

If, e.g., for an undirected graph \(G = (X, E)\) we define \(\mathcal{F} := \{T \subseteq E | T\) does not contain a cycle\}, then it is well known that \((X, \mathcal{F})\) becomes a matroid, and henceforth a greedoid; see [4].

Any undirected graph \(G\) equipped with a metric \(d\) can be considered a weighted, undirected graph by assigning weights \(d(x, x')\) to each of the edges \(\{x, x'\}\). For any finite, such as graph \(G\), we denote by \(d(G)\) the total distance (or total weight) of the graph defined as

\[
d(G) := \sum_{x_i, x_j \in X \atop \{x_i, x_j\} \in E} d(x_i, x_j).\] (3)

In the sequel, let \(X = \{x_1, \ldots, x_n\}\) be a finite set equipped with a metric \(d\). For any such set, one can naturally define a complete, graph \(K = (X, E)\), i.e., an undirected and connected graph in which every pair of distinct vertices is connected by a unique edge by taking the set \(X\) as the set of vertices and \(E := \{\{x_i, x_j\} \mid x_i, x_j \in X \text{ and } x_i \neq x_j\}\) as the edge set. Using the metric \(d\) we turn \(K\) into a weighted, complete graph and define the total distance as above.

Now for any complete graph \(K\) and a given ordering of the vertices, i.e., a sequence \(S = (x_1, \ldots, x_n)\) of vertices of \(K\), we consider constructing trees \(T\) sequentially from \(K\) by applying the following greedy algorithm:

**Definition 3.** Let \((K, d)\) be a complete graph equipped with a metric \(d\). Then for any ordered sequence of the vertices \(S = (x_1, \ldots, x_n)\) of \(K\) we define the join algorithm by the following steps:
1. Set $T = \{x_1\}$.
2. For $i = 2$ to $n$, repeat step (3):
3. Find the nearest neighbor $x_j$ with respect to $d$ of $x_i$ in $T$ and add $x_i$ together with the connecting edge $\{x_i, x_j\}$ to $T$.

**Remark 1.** While the join algorithm looks similar to the well-known Prim’s algorithm [4] applied to a complete graph $K$, it is important to note that in Prim’s algorithm the optimum candidate vertex $x_i$ to be added to the tree is chosen from the list of all available vertices by choosing the one with the minimum distance, whereas in the join algorithm the optimum candidate is simply the next one dictated by the ordering of the vertices, and only the best edge is chosen by taking the one with the minimum distance.

Obviously—as can be seen by trivial examples—in general the trees constructed depend on the order the vertices are added. We prove in the next subsection (see Proposition 1) that the total distance does not depend, however, on the order. Furthermore (see Theorem 1), the join algorithm guarantees constructing a minimally spanning tree for any given ultrametric $d$. The notion of an ultrametric is essential, as simple examples show that the join algorithm fails to construct a minimum spanning tree for arbitrary metrics. Indeed, see Proposition 2; the ultrametric is also necessary; i.e., if the join algorithm always constructs a minimum spanning tree, the metric must be an ultrametric.

### 2.2. Optimality Results

**Proposition 1** (Permutation independence of the join algorithm). Assume that $K$ is endowed with a metric $d$ satisfying the strong triangle inequality, i.e., assume $(K, d)$ is an ultrametric space. Let $\pi$ denote a permutation of $1, \ldots, n$. For $n$ vertices $x_1, \ldots, x_n$, by $T_{(1,\ldots,n)}$ we denote the tree obtained by joining $x_1, \ldots, x_n$ in this order, and henceforth, by $T_{x(1,\ldots,n)}$ the tree obtained by joining $x_{\pi(1)}, \ldots, x_{\pi(n)}$ in the permuted order while applying the greedy algorithm join.

Then the total distance of the tree is independent of the order, i.e.,

$$d(T_{\pi(1,\ldots,n)}) = d(T_{(1,\ldots,n)}).$$

**Proof.** It is well known that any permutation can be written as a product of adjacent transpositions (This follows, e.g., immediately from the correctness of sorting algorithms, such as bubble or insertion sort). Henceforth, it is enough to prove the proposition for adjacent transpositions of two newly added vertices $x$ and $y$. We consider several cases (omitting the trivial case $n = 1$):

1. Case $n = 2$: By symmetry of $d$ for any two vertices $x, y \in K$ we have $d(T(x,y)) = d(T(y,x))$.
2. Case $n > 2$: Denote by $T_{(Z,x,y)}$ the tree obtained by joining $x$ and $y$ in this order to a given tree $Z$ applying the join algorithm.

   (a) Assume that $x$ and $y$ are connected by an edge, i.e., that the tree $T_{(Z,x,y)}$ looks as depicted in Figure 1a.

   We consider three cases 2(a)i–2(a)iii:

   i. Let $d(x, z) > d(x, y)$; then by Corollary 1 we conclude that $d(z, y) = d(x, z)$, and henceforth if we define $T'$ as in Figure 1b, we conclude that $d(T_{(Z,x,y)}) = d(T')$. Thus $T'$ is equivalent to $T_{(Z,x,y)}$ with regard to its total distance. However, as $T'$ can be greedily constructed by first joining $y$ and then $x$, we conclude

   $$d(T_{(Z,y,x)}) \leq d(T') = d(T_{(Z,x,y)}).$$

   ii. Let $d(x, z) < d(x, y)$; then by Corollary 1 we conclude that $d(z, y) = d(x, y)$, and henceforth if we define $T''$ as in Figure 1c, we get $d(T_{(Z,x,y)}) = d(T'')$. Again as
The $T''$ can be greedily constructed by first (Note, that the tree $T''$ is symmetric with respect to the order $x$ and $y$ that are added) joining $y$ and then $x$, we conclude

$$d(T_{(Z,y,x)}) \leq d(T'') = d(T_{(Z,x,y)}).$$

iii. Finally, let $d(x,z) = d(x,y)$; then $d(y,z) \leq d(x,y)$. Henceforth for $T''$ defined as in Figure 2b,c or vice versa (Figure 2d,e):

Assume $\land$ for case (c) consider the triangle $K$ of the tree produced by the join algorithm (Definition 3) is the same for any permutation of the vertices $S = (x_1, \ldots, x_n)$ of $K$, then the metric is an ultrametric.

Proposition 2. Let $K = (X, E)$ be an arbitrary complete graph equipped with a metric $d$. If the total distance of the tree produced by the join algorithm (Definition 3) is the same for any permutation of the vertices $S = (x_1, \ldots, x_n)$ of $K$, then the metric is an ultrametric.

Proof. Assume $|X| \geq 3$, as otherwise the statement is trivially true. We first consider $|X| = 3$; i.e., we consider the triangle $K = (A, B, C)$ (Figure 2a). We join $B$ and $C$ to $A$ with $B$ first and afterwards $C$ (Figure 2b,c) or vice versa (Figure 2d,e):

$$d(T_{(Z,y,x)}) \leq d(T_{(Z,x,y)}).$$

Therefore putting (2a) and (2b) together we conclude that in all cases we have $d(T_{(Z,y,x)}) \leq d(T_{(Z,x,y)}).$ By exchanging the roles of $x$ and $y$, i.e., a symmetrization argument, we conclude that only equality can hold. $\Box$

![Figure 1](image-url)

(a) $T_{(Z,y,x)}$ (b) $T'$ (c) $T''$

Figure 1. (a) Tree $T_{(Z,y,x)}$ with $x$ and $y$ connected; (b) equivalent trees $T'$ and (c) $T''$ with same distance.

While the ultrametric property is sufficient to ensure order independence, it is also necessary in the following sense.

Proposition 2. Let $K = (X, E)$ be an arbitrary complete graph equipped with a metric $d$. If the total distance of the tree produced by the join algorithm (Definition 3) is the same for any permutation of the vertices $S = (x_1, \ldots, x_n)$ of $K$, then the metric is an ultrametric.

Proof. Assume $|X| \geq 3$, as otherwise the statement is trivially true. We first consider $|X| = 3$; i.e., we consider the triangle $K = (A, B, C)$ (Figure 2a). We join $B$ and $C$ to $A$ with $B$ first and afterwards $C$ (Figure 2b,c) or vice versa (Figure 2d,e):

$$d(T_{(Z,y,x)}) \leq d(T_{(Z,x,y)}).$$

Henceforth, $T_{(A,B,C)} = T_{(A,C,B)}$ implies for case (b) $\land$ (d): $a \leq c = b$, for case (b) $\land$ (e): $c \leq a = b$, for case (c) $\land$ (d): $b \leq c = a$, and for case (c) $\land$ (e): $b \leq a \land c \leq a$. Now the first three cases already
imply the ultrametric property. To prove the ultrametric property for the fourth case also, we consider the permutation of joining \( A \) and \( B \) to \( C \) or vice versa. Then we get \( T_{(C,A,B)} = T_{(C,B,A)} \) and repeat the above argument. This yields the ultrametric property for the first three cases directly again. For the fourth case we get \( b \leq c \land a \leq c \), which altogether yields \( b = a = c \) (and symmetrizing by considering \( T_{(B,B,C)} = T_{(B,C,B)} \) eventually yields \( a = b = c \)) which proves the ultrametric property in this case as well.

If \(|X| > 3\), we consider an arbitrary triangle \( D := (A,B,C) \subset K \). The join algorithm produces the same total distance for any permutation and henceforth without loss of generality we can consider \( D \) to be the first three vertices joined. By definition of the join algorithm, if the total distance of the first three joined vertices would be different, the total distance for \( K \) would be different, which—by assumption—is not the case. Thus, we can apply the former argument for \(|X| = 3\) to conclude that the triangle \( D \) satisfies the ultrametric property. As this applies to any triangle \( D \), the metric is an ultrametric on \( K \).

**Theorem 1.** Let \((K,d)\) be a complete graph equipped with an ultrametric \( d \). Then the join algorithm constructs a minimum spanning tree of \( K \) for the ultrametric \( d \).

**Proof.** Prim’s algorithm applied to the ultrametric \( d \) as an objective function (i.e., a real-valued function \( d : E \rightarrow \mathbb{R} \)) to optimize, is a greedy algorithm that is optimal, as proven in [4]. According to Proposition 1 we can re-order the vertices to match the order of Prim’s algorithm without changing the topological distance.

**Corollary 2.** Let \((K,d)\) be a complete graph equipped with a metric \( d \). If the join algorithm always constructs a minimum spanning tree of \( K \) for the metric \( d \); then \( d \) is an ultrametric.

**Proof.** Proposition 2.

**Corollary 3.** Every sub tree produced by the join algorithm is a minimum spanning tree for the ultrametric \( d \).

**Proof.** Trivial.

**Remark 2.** From Corollary 3 it follows that the join algorithm is incremental in the sense that the join algorithm can construct a minimum spanning tree for \( K_i \), for \( i \leq n \) without any information about \( x_j, j > i \). It is also stable; i.e., a vertex added to the tree is never rearranged later. This is important for the network applications considered in [1,2,7] as it allows for dynamically constructing an optimal overlay network for, e.g., video conference participants with minimal modifications of the overlay network when new participants join.

**Corollary 4.** The join algorithm has a run-time complexity of \( O(n^2) \).

**Proof.** For each \( x_i \) added to the tree, we have to calculate the minimum of \( d(x_i, x_j) \) for \( j < i \), which is in \( O(i) \). Henceforth, the whole algorithm is in \( O(\sum_{i=2}^{n} i) = O(n^2) \).

**Remark 3.** Prim’s algorithm can be implemented in \( O(|E| + n \lg n) \) using a Fibonacci heap; see [8]. For a complete graph, we have \(|E| = n(n−1)/2\). As calculating the distances \( d(x_i, x_j) \) for all edges is in \( O(n^2) \), Prim’s algorithm has the same complexity as the join algorithm.

An alternative to Prim’s algorithm for constructing minimum weighted spanning trees is Kruskal’s algorithm [4]. It considers (sub-) forests of trees in \( X \) and greedily adds links of minimum weight until all forests are connected. The reconstruction algorithm as defined in [1] to “repair” trees after removing one vertex (see Section 3) is based on Kruskal’s idea and attaches whole subtrees by a modification of step (3) of Definition 3:
Definition 4. Let $T$ be a minimum spanning tree in $(K,d)$ obtained by applying the join algorithm and deleting a vertex $x \in K$. The reconstruction algorithm is then defined by the following steps:

1. Consider $x$ as the root of the subtree denoted by $T_x$ and find all children $x_i$ of $x$ in $T_x$.
2. For each child $x_i$ find the nearest neighbor $x_j$ with respect to $d$ of $x_i$ in $T \setminus T_x$ and add $x_i$ together with the connecting edge $\{x_i, x_j\}$ to $T \setminus T_x$.

Theorem 2. If $(K,d)$ is an ultrametric space, the reconstruction algorithm constructs a minimum spanning tree for the ultrametric $d$.

Proof. If one considers forests of trees, one can apply Kruskal’s algorithm to the greedoid defined by the matroid above and the subtrees defined by each neighbor $x_i$ as a root of a subtree. For details, see [4]. In Kruskal’s algorithm one selects the minimum link connecting all vertices of the subtree connecting to the existing tree. By the construction of the join algorithm and the fact that $(K,d)$ is an ultrametric space, the minimum is obtained at the root $x_i$. □

Remark 4. A subtlety might be worthwhile/important to note: It is claimed in [9] that the general proof of the optimality of the greedy algorithm for general greedoids, i.e., Theorem XI.1.3 of [4] contains a subtle error. However, our proofs are not affected by this argument, as Kruskal’s algorithm applies to matroids, as pointed out therein, and Prim’s algorithm—known to be valid for a long time anyway—is covered by Theorem XI.2.2 of [4] (see Theorem 9 of [9]).

3. Applications

3.1. Topological Distance

In [1] the so-called topological distance was introduced as a tool to optimize peer-to-peer networks. For practical usage, the incremental nature of the join algorithm (as opposed to Prim’s algorithm) is crucial—see [7]—as this algorithm is used as a basis for a peer-to-peer application protocol. For the reader’s convenience we recall its definition (It is similar to a metric based on a taxonomy—known to satisfy the strong triangle inequality):

Definition 5. Let $x$ and $y$ be two vertices representing communication nodes in an IP-network; then their topological distance $d(x,y)$ is defined by counting the number of different static network coordinates $C_i(x)$. Formally, if $h$ denotes the number of coordinates, and $C(x) = (C_0(x), \ldots, C_i(x), \ldots, C_{h-1}(x))$, then:

$$d(x,y) = h - 1 - m,$$

where $m := \max_{1 \leq i \leq h-1} \{i \mid C_i(x) = C_i(y) \forall k \leq i\}$.

For example, for the vertices $x = (RIPE, DE, FRA, 80.x.x.x, 80.70.x.x)$, and $y = (RIPE, DE, BLN, 90.x.x.x, 90.80.x.x)$, we get $d(x,y) = 3$.

Lemma 1. Let $X = \{x_1, \ldots, x_n\}$ and let $d$ denote the topological distance. Then the following is true:

1. $d$ is a metric.
2. $d$ satisfies the strong triangle axiom; i.e., $X$ is an ultrametric space if endowed with $d$.

Proof. Obvious from the definition. □

Theorem 3. The JOIN algorithm as defined in [1] and analyzed in [2] constructs a minimum spanning tree for the topological distance $d$. 

Proof. As the JOIN algorithm is equivalent to the algorithm join defined in Definition 3, this is a corollary of Lemma 1 and Theorem 1. □

Theorem 4. The RECONSTRUCT_TREE algorithm as defined in [1] and analyzed in [2] constructs a minimum spanning tree for the topological distance $d$.

Proof. The RECONSTRUCT_TREE algorithm is equivalent to the algorithm reconstruction defined in Definition 4; thus, this is a corollary of Lemma 1 and Theorem 2. □

3.2. Latency

When constructing applications, in addition to minimizing the topological distance $d$, one often also wants to control the maximum latency of each communication link; i.e., one tries to minimize the maximum latency of each (pair-wise) link. We prove that it is possible to modify the JOIN algorithm of [1,2] in such a way that, among all trees with minimum topological distance $d$, it also minimizes the maximum latency of each (pair-wise) link. We formally define latency as follows:

Definition 6. Let $T$ be a tree equipped with a metric $d$. Any two vertices $x_i$ and $x_j$ of a tree are connected by a unique path of $m$ vertices which we denote by $x_{i_1}, \ldots, x_{i_m}$ such that $x_{i_1} = x_i$ and $x_{i_m} = x_j$. Then the latency $\ell$ between $x_i$ and $x_j$ is defined as

$$\ell(x_i, x_j) := \sum_{k=1}^{m-1} d(x_{i_k}, x_{i_{k+1}}),$$

and the tree latency $L$ of the tree is defined as

$$L(T) := \max_{x_i, x_j \in T} \ell(x_i, x_j).$$

Definition 7. The snowflake (The name stems from the snowflake forms it tends to generate) algorithm JOIN-S algorithm is defined by joining each node as in the JOIN algorithm with the following modification. If multiple minimal candidates with the same topological distance exist, then the one is chosen that implies the lower tree latency for the resulting tree.

Theorem 5. The JOIN-S algorithm creates a tree with minimum topological distance and minimum latency among all those trees.

Proof. Define $d' := d + \lambda L$; then $d'$ is an objective function, i.e., a real-valued function $d' : E \rightarrow \mathbb{R}$, that we can minimize within the framework of matroids and greedoids; see [4]. If we set $\lambda > 0$ small enough to ensure that $\lambda L < 1$, then to obtain the minimum, we first have to find the minimum of $d$ henceforth for the trees with minimum topological distance, and then afterwards—among all trees with the same minimum—the one with the minimum tree latency. (The condition $\lambda L < 1$ effectively subordinates the latency costs $L$ to the topological costs $d$). It is also clear that a greedy choice corresponds to the JOIN-S algorithm. Henceforth the proof follows from general matroid/greedoid theory, as defined in [4] if we can ensure that the analogy of Proposition 1 is true. By carefully examining the proof of Proposition 1 we conclude that the equivalent trees depicted in Figure 1a,b not only have the same $d$ but also the same latency $L$; thus case 2(a)i passes through without modification. On the other hand, for case 2(a)ii and 2(a)iii—as the “symmetric” case depicted in Figure 1c has the same $d$ but lower latency $L$—it would therefore be selected by algorithm JOIN-S. As the corresponding configuration depicted in Figure 1c is independent of the order, we can conclude that in any case JOIN-S is order-independent which completes the proof. □

Definition 8. The snowflake RECONSTRUCT_TREE-S algorithm is defined by replacing the join algorithm in Definition 4 of the reconstruction algorithm with the JOIN-S algorithm of Definition 7.
Corollary 5. The RECONSTRUCT_TREE-S creates a tree with minimum topological distance and minimum latency among all those trees.

Proof. Follows from the equivalence of the JOIN algorithm with Prim’s and Kruskal’s algorithm applied to $d'$. □

3.3. Outlook

For practical applications it would be interesting to investigate whether distance measures can be approximated by distances satisfying the strong triangle inequality—in other words, whether we can use ultrametric spaces as approximation tools. For example, the network configuration depicted in Figure 3a is based on the data measured in [2]. The edge weights depicted denote the actual latencies in milliseconds. As shown in Figure 3b it can be quite well approximated by an ultrametric—the individual weights have to be changed only by small percentages, and the total distances of the trees produced by the join algorithm differ only by less than 2% if calculated with the ultrametric rather than using the empirical numbers. This is primarily a result of the latencies of local and national links respectively being similar to each other while different in scale. As follow-up work to [7], we plan to empirically investigate this more quantitatively using appropriate proximity measures to judge the approximation quality.

Figure 3. (a) Measured latency; (b) Latency approximated by ultrametric.

Funding: This research was funded by Central Innovation Programme SMEs (ZIM) by the German Federal Ministry for Economic Affairs and Energy (BMWi).

Acknowledgments: This work was supported by a research grant from the Central Innovation Programme SMEs (ZIM) by the German Federal Ministry for Economic Affairs and Energy (BMWi) which is gratefully acknowledged. The author also thanks Raul Bertone and Saidar Ramazanov for helpful discussions.

Conflicts of Interest: The author declares no conflict of interest.

References
1. Alekseev, S.; Schäfer, J. A New Algorithm for Construction of a P2P Multicast Hybrid Overlay Tree Based on Topological Distances. In Proceedings of the Seventh International Conference on Networks & Communications, Zürich, Switzerland, 2–3 January 2016; pp. 307–321. [CrossRef]
2. Alekseev, S.; Schäfer, J. Evaluation of a Topological Distance Algorithm for Construction of a P2P Multicast Hybrid Overlay Tree. Int. J. Comput. Netw. Commun. 2016, 8, 1–20. [CrossRef]
3. Krasner, M. Nombres semi-réels et espaces ultramétriques. C. R. Acad. Sci. 1944, 219, 433–435.
4. Korte, B.; Lovasz, L.; Schrader, R. Greedoids; Springer: Berlin/Heidelberg, Germany, 1991.
5. Korte, B.; Lovász, L. Structural properties of greedoids. Combinatorica 1983, 3, 359–374. [CrossRef]
6. Korte, B.; Lovász, L. Greedoids—A Structural Framework for the Greedy Algorithm. In Progress in Combinatorial Optimization; Pulleyblank, W.R., Ed.; Academic Press: Cambridge, MA, USA, 1984; pp. 221–243. [CrossRef]
7. Silva, M.A.; Bertone, R.; Schäfer, J. Topology Distribution for Video-Conferencing Applications. In Proceedings of the 2019 10th International Conference on Networks of the Future (NoF), Rome, Italy, 1–3 October 2019; Volume 10.

8. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. Introduction to Algorithms, 3rd ed.; MIT Press: Cambridge, MA, USA, 2009.

9. Szeszlér, D. Optimality of the Greedy Algorithm in Greedoids. In Proceedings of the 11th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, Tokyo, Japan, 27–30 May 2019; Volume 11, pp. 438–445.

© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).