The Sequential Sum Problem and Performance Bounds on the Greedy Algorithm for the On-line Steiner Problem

Zevi Miller, Dan Pritikin
Department of Mathematics and Statistics, Miami University, Oxford, Ohio 45056

Manley Perkel
Department of Mathematics and Statistics, Wright State University, Dayton, Ohio 45435

I. H. Sudborough
Program in Computer Science, University of Texas-Dallas, Richardson, Texas 75083

This article is motivated by versions of the dynamic or "on-line" Steiner tree problem (OST) introduced by Imase and Waxman [4]. In this problem one is given an edge-weighted graph G and a sequence $\sigma = (x_1, \ldots, x_n)$ of distinct vertices of G. The requirement is to construct for each $i \le n$ a tree T_i spanning the first i vertices of σ subject to the condition that $T_{i-1} \subseteq T_i$ for all i, where T_i is constructed without knowledge of the remaining vertices x_j, $j > i$. The goal of the on-line Steiner problem is to minimize the performance ratio; that is, the maximum (over $1 \le i \le n$) of the ratio of the weight of T_i to the weight of the minimum weight tree in G spanning the first i vertices (the latter tree is called the "Steiner tree" for these vertices). In [4] a lower bound of $1 + \frac{1}{2}\lfloor \log_2(n-1) \rfloor$ was proved for this ratio. The authors further made the interesting conjecture that there is some on-line algorithm for the OST whose performance ratio achieves this lower bound. We show that a strong form of the greedy algorithm achieves a ratio that converges to the conjectured $\frac{1}{2}\log_2(k) + O(1)$ as the proportion of degree 2 vertices in the instance graph grows. Our results also imply improvements in certain cases on the known upper bound $\lceil \log_2(n) \rceil$ for the performance ratio of the greedy algorithm. Our approach is to study a related graph parameter. For each sequence σ as above, define the associated cost

$$L(\sigma) = \sum_{i=2}^{n} c(i, \sigma),$$

where $c(i, \sigma) = \min_{1 \le t < i} \operatorname{dist}(x_i, x_t)$. Then let $\operatorname{Opt}(n, G)$ be the maximum of $L(\sigma)$ over all such sequences σ of length n. The problem of, given n and G, determining $\operatorname{Opt}(n, G)$ we call the Sequential Sum Problem (SSP).

In this article we analyze the SSP, obtaining exact values and bounds on $\operatorname{Opt}(n, G)$ and relating these bounds to the greedy algorithm for the OST. For example, we calculate $\operatorname{Opt}(n, P)$ for the path P, and obtain a surprising characterization of all length n sequences σ which realize $\operatorname{Opt}(n, P)$. By analyzing $\operatorname{Opt}(n, P)$ for the "continuous" path, we derive upper bounds on the performance ratio of the greedy algorithm for the OST in arbitrary graphs. On the other hand, generalizing the lower bound argument of [4] we show that there are instances of OST, which can significantly "fool" any on-line algorithm for OST. Specifically, given any tree T normalized to have total edge weight 1, we construct a graph G and a length $k \le |V(T)|$ sequence σ of vertices of G for which the performance ratio of any on-line algorithm for the OST with input σ is lower bounded by $\operatorname{Opt}(k, T)$. Finally, we show that the SSP for arbitrary G is NP-complete. **NETWORKS, Vol. 45(3), 143–164 2005**

Keywords: sequential sum problem; on-line Steiner tree problem; dynamic greedy algorithm; vertex greedy algorithm

1. INTRODUCTION

We consider connected edge-weighted graphs $G = (V, E)$. When all edge weights are 1 we say that G is *simple*. We denote by *weight* (G) the sum of the edge weights in G, and call this the weight of G.

We begin with the *Steiner tree problem* (STP) for graphs, on which there is an extensive literature [5, 8]. In the instance of STP, one is given a connected edge-weighted graph G and a *terminal set* S, that is, a subset S of the vertices of G. The objective of STP is to find the minimum possible weight of any subtree T of G containing the vertices of S, and if possible, to construct this subtree. Such a subtree of least weight is called a *minimum Steiner tree* for S and its weight is denoted by $w(S)$. Given a sequence σ of distinct vertices $x_1, \ldots, x_k$ of G, we let $w(\sigma) = w(S)$ where $S = \{x_1, \ldots, x_k\}$.

Received Month 2005; accepted Month 2005
Correspondence to: Z. Miller; e-mail: millerz@muohio.edu
DOI 10.1002/net.20057
Published online 11 March 2005 in Wiley InterScience (www.interscience.wiley.com).

The well-known "shortest path" heuristic for STP (developed in [7] and [6] independently) builds a spanning tree for S, where say $|S| = k$, as follows. Having inductively built a tree T_i spanning a subset S_i of i vertices in S, this heuristic selects a vertex $x \in S - S_i$ closest to T_i, and defines T_{i+1} to be the union of T_i with a shortest path from x to T_i. The bound

$$\frac{\text{weight}(T_k)}{w(S)} \leq 2\left(1 - \frac{1}{k}\right)$$

for the performance ratio of this heuristic was proved in [7] and [6].

Observe that this heuristic requires a knowledge of all of S. One may consider an on-line version of STP introduced by Imase and Waxman [4], called by them the *dynamic Steiner tree problem* (DST) and by others later the *on-line Steiner tree problem* (OST), in which this knowledge is unavailable. In this problem we are presented with one vertex of S at a time (having no knowledge of the vertices which will come later), and the object is to build a nested sequence of trees $T_1 \subseteq T_2 \subseteq \cdots \subseteq T_k$, where T_i spans the first i vertices presented, keeping the ratio

$$\max_{1<i\leq k} \frac{\text{weight}(T_i)}{w(S_i)}$$

as small as possible.

More precisely, *instance* I of OST as defined in [4] consists of an edge-weighted graph G and a sequence σ of distinct vertices of G, denoted $\sigma = x_1, \ldots, x_k$. Let $S_i, i \leq k$, denote $\{x_1, x_2, \ldots, x_i\}$, the *terminal set* at step i. A *solution* to the instance of OST (not necessarily an optimal solution) is a sequence of trees $T_1, \ldots, T_k$ in G such that for all i we have that T_i contains (i.e., connects) S_i and T_{i+1} contains T_i. An *on-line algorithm* A for the OST is a procedure which for each i constructs the tree T_i without knowledge of the set of points $\{x_t; t > i\}$. The performance of an on-line algorithm A on an instance I of OST was then defined as

$$A(I) = \max_{1<i\leq k} \frac{\text{weight}(T_i)}{w(S_i)},$$

where T_i is the tree produced by A for connecting S_i. The *performance ratio* of algorithm A was defined in [4] by

$$C_A(k) = \sup\{A(I) : I \text{ is an instance of OST, with } |\sigma| = k\}.$$

Theorem 1 of [4] showed that given any on-line algorithm A for OST, there is an instance consisting of a graph G and a sequence $x_1, x_2, \ldots, x_k$ for which the trees T_i produced by A satisfy

$$\frac{\text{weight}(T_i)}{w(S_i)} \geq 1 + \frac{1}{2}\lfloor \log_2(i-1) \rfloor \tag{2.1}$$

for $i = 2, 3, \ldots, k$. The important implication of that theorem is that every on-line algorithm A can be "fooled" to the extent that it has performance ratio $C_A(k)$ bounded below by $1 + \frac{1}{2}\lfloor \log_2(k-1) \rfloor$, a performance ratio that becomes arbitrarily large (i.e., poor) as k approaches ∞.

Perhaps the most natural choice of an algorithm for obtaining a solution to OST is called the *dynamic greedy algorithm* (DGA), which creates T_{i+1} from T_i by connecting x_{i+1} by a shortest path in G to a nearest node (not necessarily a terminal node) in T_i, appending the vertices and edges of that path, and breaking ties at random between rival shortest paths. Theorem 2 in [4] states that DGA has its performance ratio upper bounded according to the inequality.

$$\frac{\text{weight}(T_i)}{w(S_i)} \leq \lceil \log_2(i) \rceil \quad \text{for all } i > 1, \tag{2.2}$$

so that DGA has performance at worst twice optimal.

Research on OST subsequent to [4] refined the results contained therein and considered variations of OST. In [9], a refinement of the bound (2.2) for the OST on general metric spaces was obtained by showing that for a class C of on-line algorithms, which includes DGA, the performance ratio is bounded above by $O(\log(dk/w(S)))$, where d is the maximum distance between any two terminal vertices. In [1], the OST for Euclidean space of arbitrary dimension was considered, where a lower bound of $\Omega(\log k/\log(\log k))$ was shown for the performance ratio of any on-line algorithm and an especially short proof of (2.2) for OST was also given.

A variation of STP relevant to on-line problems is the Generalized Steiner Tree Problem [GSP]. An instance of GSP consists of a collection K of pairs of vertices from a graph G, and the objective is to construct a minimum weight graph H in which every pair in K belong to the same connected component of H (so that H is in general a forest of Steiner trees). In the on-line version, call it OGSP, the pairs in K are revealed to us in some sequence and at the ith step we have no knowledge of the jth pair, $j \geq i + 1$. At each step one must add vertices and edges (if necessary) to the then current graph so that in the resulting new graph each revealed pair will lie in the same connected component of the new graph. The worst-case performance ratio $R(k)$ is defined analogously as the maximum, over all collections K with $|K| = k$, of $w(C)/w(F)$, where C is the final graph produced by the algorithm, and F is the minimum weight forest for which each pair of K lies in the same component of F. In [3], a polynomial time algorithm for OGSP with performance ratio $O(\log k)$ is given, improving on earlier algorithms for this problem with ratios $O(\log^2(k))$ [2] and earlier $O(\sqrt{k} \log k)$ [9]. Also in [9], a lower bound of $\Omega(\log k)$ was proved for the performance ratio of any on-line algorithm for OGSP. In [10], a version of OGSP is considered in which each pair (x, y) of K is required to be joined by some integer number $r_{(x,y)}$ of vertex or edge disjoint paths.

Turning back now to the OST, Imase and Waxman [4] made the interesting conjecture that the lower bound (2.1) for the performance ratio of any on-line algorithm can actually be achieved by some algorithm, perhaps even the greedy algorithm DGA. One goal in this article is to explore this conjecture. We show that a restricted form of the DGA achieves a ratio that converges to the conjectured $\frac{1}{2}\log_2(k) + O(1)$ as the proportion of degree 2 vertices in the instance graph grows. In this restriction, called the *vertex greedy algorithm* (VG) in [1], at the ith step the newly revealed terminal vertex

x_i is joined by any shortest path to whichever of the previously revealed terminal vertices $x_t, t < i$, is closest to x_i in the instance graph. The new output graph is then the old one with all vertices and edges of this path adjoined except for those already present in the old output graph. The motivation for this choice of algorithm lies in the convenience of deriving an upper bound on the performance ratio, in its application to finding a lower bound for this ratio of any on-line algorithm for OST (see the next section), and in the intrinsic interest of the resulting graph parameter, which is discussed below.

Given a weighted graph G and a sequence σ of distinct vertices $x_1, \ldots, x_k$ of G, we assign to each vertex x_i (for $i \geq 2$) in the sequence a cost given by

$$c(i, \sigma) = \min_{1 \leq t < i} \operatorname{dist}(x_i, x_t),$$

the distance between x_i and its nearest predecessor. Because x_1 has no predecessors, we let $c(1, \sigma) = 0$. When the choice of σ is clear from context, we abbreviate $c(i, \sigma)$ by just c_i, and on occasion we refer to the ith vertex of σ by $\sigma(i)$. As a slight abuse of notation, for a vertex $x = \sigma(i)$ we write $c(x, \sigma)$ for $c(i, \sigma)$ when the position of x in σ is either understood or unimportant. We assign to the entire sequence σ the cost

$$L(\sigma) = \sum_{i=1}^{k} c(i, \sigma).$$

For each graph G and integer $k \geq 1$ let

$$\operatorname{Opt}(k, G) = \max\{L(\sigma) : \sigma = (x_1, \ldots, x_k) \text{ is a sequence of distinct vertices in } G\}.$$

We can then introduce the following decision problem.

Sequential Sum Problem (SSP) Given a graph G and integers k and r (with $k \leq |V(G)|$), determine if $\operatorname{Opt}(k, G) \geq r$; that is, determine if there exists a length k sequence σ for which $L(\sigma) \geq r$.

We will show that this problem is NP-complete. Our main effort will be in obtaining upper bounds for $\operatorname{Opt}(k, G)$; in effect, upper bounds for $L(\sigma)$ for any graph G and length k sequence from $V(G)$. Now $L(\sigma)$ is just an upper bound on the weight of the graph produced by the on-line algorithm VG applied to the instance G and σ. Hence, the resulting upper bound we obtain for $L(\sigma)/w(\sigma)$, again valid for any graph G and sequence σ on $V(G)$, is then an upper bound on the performance ratio of VG (and hence of DGA) on length k sequences for the OST.

As a first example of SSP, consider the sequence $\sigma = [\sigma(1), \sigma(2), \ldots, \sigma(10)]$ of vertices in the tree on 10 vertices shown in the second diagram of Figure 1, regarding each of the nine edges as having weight 1 (and ignoring for now the ordered pair notation). Then $L(\sigma) = 0 + 6 + 6 + 2 + 2 + 2 + 1 + 1 + 1 + 1 = 22$. It turns out that σ is optimal, that is, $L(\sigma) = \operatorname{Opt}(10, G)$. We might also check if σ is "greedy," in the sense that for each $i \geq 2$, $\sigma(i)$ is chosen so as to maximize $c(i, \sigma)$ given the initial choices of $\sigma(1), \sigma(2), \ldots, \sigma(i-1)$.

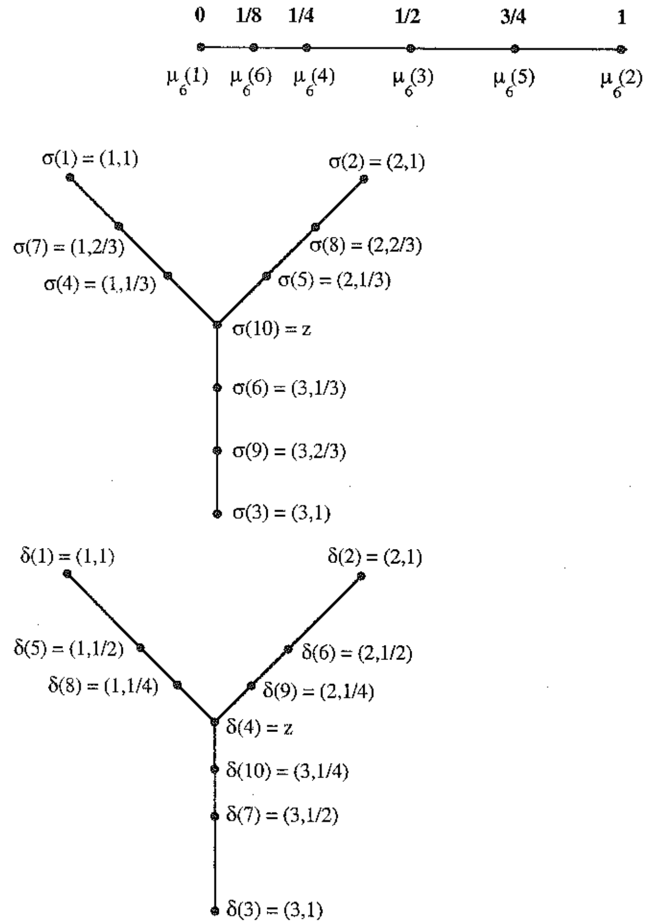


FIG. 1. Two metric spaces; con-K_2 with sequence μ_6, and con-$K_{1,3}$ with sequences σ and δ.

In fact, σ is not greedy, because $\sigma(4)$ was not chosen in this way. But note that had we been greedy in forming a sequence of 10 vertices in G, the resulting sequence would have had cost $0 + 6 + 6 + 3 + 1 + 1 + 1 + 1 + 1 + 1 = 21$ and been nonoptimal. This example illustrates that the greedy approach does not always produce a sequence of maximum cost $\operatorname{Opt}(k, G)$, even in the case where G is a simply weighted tree and $k = |V(G)|$.

The results of this article are organized as follows. In Section 2 we show that for any weighted tree T, suitably normalized to have total edge weight 1, there is a graph G for which the performance ratio of any on-line algorithm for the OST, applied to instances with length k sequences of vertices from G, is lower bounded by $\operatorname{Opt}(k, T)$. This provides a motivation for studying the SSP for trees, while also generalizing the adversary lower bound argument obtained in [4]. In Section 3 we establish basic properties, used throughout the article, of the function $L(\sigma)$. Here we also show the NP-completeness of the SSP. In Section 4 we compute $\operatorname{Opt}(k, P)$ exactly for the path P and any $k \leq |V(P)|$, and also find a characterization of all length $|V(P)|$ sequences of vertices from P that realize $\operatorname{Opt}(|V(P)|, P)$. In Section 5, we solve a continuous version of the SSP for the path, and use this result to obtain upper bounds for the ratio $L(\sigma)/w(\sigma)$, where

σ is any instance of OST, that is, any on-line sequence of k points in a discrete weighted graph G. These bounds imply an upper bound on the performance ratio of VG, and hence, of DGA, which converges to the bound $\frac{1}{2}\log_2(k)+O(1)$ conjectured in [4] as the proportion of degree 2 vertices in the instance graph grows. We include an appendix, which gives constructions demonstrating tightness of our various bounds on performance ratio given in the article.

2. SSP FOR TREES AND A LOWER BOUND FOR THE ON-LINE STEINER TREE PROBLEM

Theorem 1 of [4] giving the lower bound (2.1) was proved by constructing graphs G_n designed so that to each on-line algorithm A for OST there corresponds an instance $I = (G, \sigma)$ with σ of the form $v_0, v_1, \ldots, v_{2^n}$, for which σ "fools" algorithm A so badly as to make $A(I) \geq 1 + \frac{1}{2}n$. To capture the spirit of their construction, see their graph G_3 in Figure 2a, in which each edge has weight 1/8. Given an on-line algorithm A, there is a sequence $v_0, v_1, \ldots, v_8$ comprised of one node at each "level" altitude in the figure, such that A has performance $1 + \frac{1}{2}3$, where v_0, v_1 are as shown. To see this, observe that the tree T_1 selected by A (without advance knowledge of the choice of x_2) must include a (v_1, v_0)-path, which must reside entirely in either the left or right "half" of G_3. Say such a path $P(1)$ is in the left half. In response we choose v_2 to be the mid-"level" node in the right half of G_3. Although T_1 may have already included some right-half edges, T_2 must include a path $P(2)$ from v_2 to either v_0 or v_1, using edges other than those in $P(1)$. Suppose paths $P(1)$ and $P(2)$ are as shaded in Figure 2b. In response, we can choose v_3 and v_4 as shown in Figure 2b so that paths $P(1)$ and $P(2)$ are unhelpful in connecting v_3 and v_4 to nodes in $P(1)$ and $P(2)$. Similarly, T_3 must include a path $P(3)$ from v_3 to v_0 or v_2, and T_4 must include a path $P(4)$ from v_4 to v_1 or v_2. If paths $P(3)$ and $P(4)$ are for instance as in Figure 2c then in response we can choose v_5, v_6, v_7, and v_8 as in that figure. The combined lengths of paths $P(1)$ through $P(8)$ will be

$$\frac{8}{8} + \left(\frac{4}{8}\right) + \left(\frac{2}{8} + \frac{2}{8}\right) + \left(\frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}\right) = 1 + \frac{1}{2}3,$$

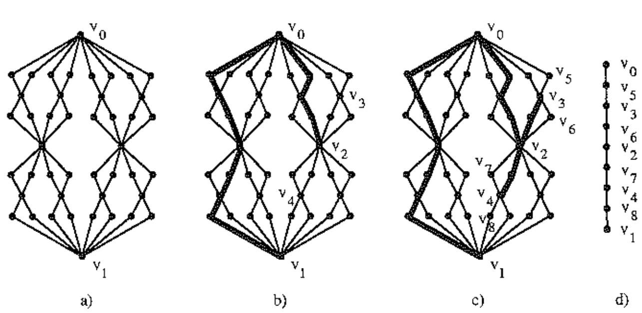


FIG. 2. The graph G_3 of Imase and Waxman.

as in Theorem 1 of [4] in the case $n = 3, k = 2^3$. Not only has the "bait-and-switch" structure of G_3 forced the on-line algorithm A to use paths $P(i)$ whose internal vertices are useless at later stages, but also we have ended up with a sequence $v_0, v_1, \ldots, v_8$ that could have been connected by a much shorter tree (in fact, a path) of weight 1, giving us the result that $\frac{|T_9|}{w(S_9)} \geq 1 + \frac{1}{2}\lfloor\log_2(8)\rfloor$, where $w(S_9) = 1$. From another point of view, observe that

$$\frac{8}{8} + \frac{4}{8} + \frac{2}{8} + \frac{2}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}$$

is precisely $L(\sigma)$ for the path graph G and the sequence $v_0, v_1, \ldots, v_8$ of its vertices as shown in Figure 2d (where each edge has weight 1/8). There is a more general connection that we will see later.

Our Theorem 2.1 below generalizes the construction G_n in Theorem 1 found in [4]. The terminal set useful with the construction G_n happened to be such that an optimal Steiner tree connecting them was a path. But a terminal set in an arbitrary graph might have any tree structure (as opposed to a path) for a minimum Steiner tree. Consequently, here is our plan. Consider any edge-weighted tree G. For a given k, consider a sequence $\sigma = v_1, v_2, \ldots, v_k$ of vertices in G for which $L(\sigma) = \mathrm{Opt}(k, G)$. In a manner to be described, construct a larger graph G' determined by G and σ. This G' will have the property that for any on-line algorithm A for OST on G', there is a sequence $\sigma' = x_1, x_2, \ldots, x_k$ of vertices in G' for which the tree T_k in G' produced by A and connecting $x_1, x_2, \ldots, x_k$ satisfies $|T_k| \geq \mathrm{Opt}(k, G)$, yet for which a minimum Steiner tree connecting $x_1, x_2, \ldots, x_k$ in G' is isomorphic to the minimum Steiner tree connecting $v_1, v_2, \ldots, v_k$ in G. In general, G' is a graph in which any on-line algorithm for OST applied to a well-chosen on-line sequence of k vertices in G' can be "fooled" to produce a tree of cost at least $\mathrm{Opt}(k, G)$. Of course, the weight of this Steiner tree in G is at most weight (G), so we will have the following.

Theorem 2.1. *Given any algorithm A for OST and any tree G and any* $k \leq |V(G)|$, *there is an instance* $I = (G', \sigma')$ *with* $|\sigma'| = k$ *for which algorithm A applied to I satisfies*

$$A(I) \geq \frac{\mathrm{Opt}(k, G)}{\mathrm{weight}(G)},$$

and for which an optimal Steiner tree connecting the terminal set S_k *in* G' *will be isomorphic to a subtree of G. Therefore, given any tree G of weight* 1 *and any k, we have that every algorithm A for OST has performance ratio bounded by the inequality* $C_A(k) \geq \mathrm{Opt}(k, G)$.

Theorem 2.1 is used here for purposes of motivating the usefulness of our study of $\mathrm{Opt}(k, G)$ and SSP. It opens the door for possible improvements in the lower bound (2.1) above given by Theorem 1 of [4], to the extent that improvement can be shown by simply citing an example tree G of weight 1 (where heavier trees can be rescaled) with a

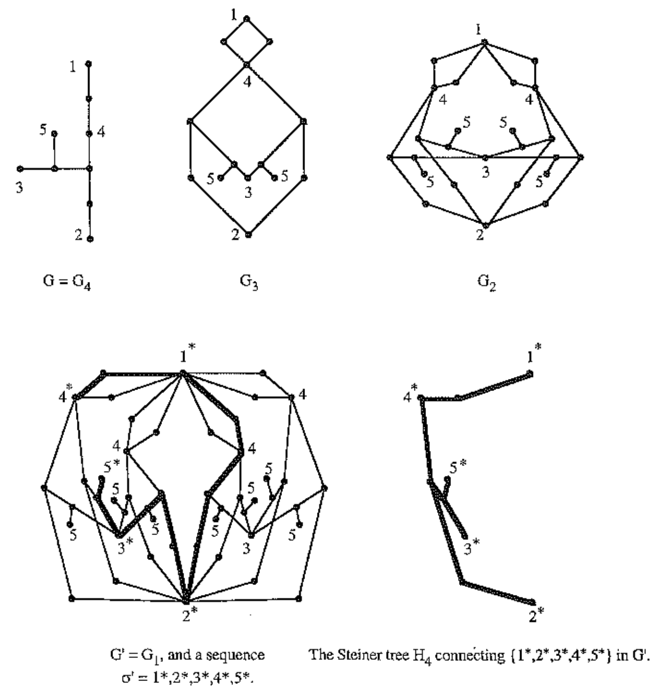


FIG. 3. Constructing G' from G.

sequence $\sigma = v_1, v_2, \ldots, v_k$ of its vertices, and directly computing $L(\sigma)$ [which will be a lower bound for $\mathrm{Opt}(k, G)$, and hence also for $C_A(k)$]. Theorem 2.1 also tells us that instances in which on-line algorithms A for OST perform "poorly" are not confined to instances in which the terminal set is easily connected by a path, but that most any tree structure could do the trick for the choice of Steiner tree, some structures working better than others.

The proof of our Theorem 2.1 is inspired by the proof of Theorem 1 from [4], which we have outlined above, where the role of G_n in the above example, founded on a path structure, is played by a more general construction G' tailored to the structure of an arbitrary tree G.

Proof of Theorem 2.1. Consider any tree G, such as the example shown in Figure 3. Consider any on-line algorithm A for OST, and consider any $k \leq |V(G)|$, where we choose $k = 5$ for our example. Let σ be any sequence of distinct vertices $v_1, v_2, \ldots, v_k$ in G for which $L(\sigma) = \mathrm{Opt}(k, G)$, for example, the sequence $\sigma = 1, 2, 3, 4, 5$ shown in Figure 3. We construct a sequence of graphs $G = G_{k-1}, G_{k-2}, G_{k-3}, \ldots, G_1 = G'$ to arrive at the desired G'. We begin with $G = G_{k-1}$, labeling each vertex of σ using each vertex name as its label. To construct G_{i-1}, we create two copies of the labeled graph G_i, and for $t = 1, 2, 3, \ldots, i$ we identify together the two vertices of label v_t to form a single vertex of label v_t. As usual, identifying together two vertices in a graph means that we replace those two vertices by a new single vertex, which we join by an edge to exactly those vertices to which either or both of the original two vertices were previously joined. In Figure 3, we have made prominent those vertices in each G_{i-1} that are formed from vertex identification, having placed them along the vertical line of symmetry in G_{i-1}, so that the reader can cover up either the left or right half of the figure for G_{i-1} to see a copy of G_i. Observe that this rule for constructing G_{i-1} from G_i does specify a well-defined process for iteratively constructing graph $G_{k-1}, G_{k-2}, G_{k-3}, \ldots, G_1$ in that order, because it is easy to verify by induction that each G_i contains exactly one vertex of each of the labels $v_1, v_2, \ldots, v_{i+1}$ (so that when we make two copies of G_i there are exactly two vertices of label x_t for $t = 1, 2, 3, \ldots, i$, as required).

We apply A to $G' = G_1$, and based upon σ and A we construct a sequence $\sigma' = x_1, x_2, \ldots, x_k$ of vertices in G' for which A performs demonstrably poorly, as follows. We require that each vertex x_i be a vertex of label v_i. Let x_1 (resp. x_2) be the vertex in G' of label v_1 (resp. v_2). The tree T_1 selected by A (without advance knowledge of the choice of x_3) must include an (x_2, x_1)-path $P(1)$, which must reside entirely in just one of the two copies of G_2 within G_1. In our later selection of vertices $x_3, x_4, \ldots, x_k$, we require that they be selected from the *other* copy H_2 of G_2 within G_1, the copy not containing $P(1)$. In particular, this requirement specifies that x_3 be the unique vertex of label v_3 in H_2. The tree T_2 selected by A (without advance knowledge of the choice of x_3) must include an (x_3, x_1)-path or (x_3, x_2)-path $P(2)$, which must reside entirely in H_2, but which furthermore must reside in just one of the two copies of G_3 within H_2. In our later selection of vertices $x_4, \ldots, x_k$, we require that they be selected from the *other* copy H_3 of G_3 within H_2, the copy not containing $P(2)$. In particular, this requirement specifies that x_4 be the unique vertex of label v_4 in H_3. Continuing in this fashion, we have completely specified σ'. For our example in Figure 3, the shaded paths at the left represent a possible collection $P(1), P(2), P(3), P(4)$, where the asterisked vertices show the corresponding choices of x_1, x_2, x_3, x_4, x_5. The paths $P(1), P(2), \ldots, P(k-1)$ are edge-disjoint, because at each stage we have confined our options to a completely unused copy of some G_i. Also, the lengths of paths $P(1)$; $P(2), \ldots, P(k-1)$ are at least $c(2, \sigma), c(3, \sigma), \ldots, c(k, \sigma)$ respectively, because edges in G' have weights equal to the edges from which they arose in the original G. Therefore, the weight of the tree T_k selected by A is at least $L(\sigma)$, that is, $|T_k| \geq L(\sigma)$.

Also, it is easy to see that H_{k-1} is a copy of G (see, e.g., the graph H_{k-1} indicated by the shaded edges at right in Fig. 3) and contains all of the vertices of σ', so it is easy non-dynamically to specify a subtree of G' connecting all vertices of σ', that tree being of weight only *weight* (G). Therefore, for the instance I given by G' and σ' determined by A, we have that

$$A(I) = \max_{1<i\leq k} \frac{\mathrm{weight}(T_i)}{w(S_i)} \geq \frac{\mathrm{weight}(T_k)}{w(S_k)} \geq \frac{L(\sigma)}{\mathrm{weight}(H_{k-1})} = \frac{\mathrm{Opt}(k, G)}{\mathrm{weight}(G)},$$

completing the proof. ∎

3. PRELIMINARIES; NP-COMPLETENESS AND PROPERTIES OF OPTIMAL SEQUENCES FOR THE SSP

Concerning the computational complexity of the SSP, we have the following simple result showing its NP-completeness.

Theorem 3.0. *SSP is NP-complete.*

Proof. The SSP clearly lies in NP because, given a length k sequence σ from $V(G)$ satisfying $L(\sigma) \geq r$, one can verify in polynomial time (as a function of $k+|V(G)|$) that $L(\sigma) \geq r$ using a shortest path algorithm. We will give a polynomial time reduction from the independent set problem (ISP), known to be NP-complete, to SSP. Let $\beta(G)$ stand for the independence number of a graph G, that is, the maximum size of a set of points in G, no two of which are joined by an edge.

Consider an instance of ISP consisting of

Input: Graph G, integer k

Question: Is $\beta(G) \geq k$?

The corresponding instance of SSP has as input a graph H and integers k and $2(k-1)$, where H is defined by; $V(H) = V(G) \cup \{x\}$, and $E(H) = E(G) \cup \{xy : y \in V(G)\}$. That is, H is constructed from G by adding a new vertex x, and joining x by an edge to every vertex y in G.

Clearly, $\beta(G) \geq k$ if and only if $\text{Opt}(k, H) \geq 2(k-1)$, showing that SSP is NP-complete for arbitrary graphs. ∎

Now consider an edge-weighted tree T and a sequence $\sigma = (x_1, x_2, \ldots, x_k)$ of k of its vertices for which $L(\sigma) = \text{Opt}(k, T)$; that is, an optimal sequence σ for the SSP. In the remainder of this section we will show how to reduce the search space for such optimal sequences. In fact, we will show that there is always an optimal sequence with the following properties: (a) x_1 is a leaf; (b) x_2 is a leaf; (c) whenever x_i is a cut-vertex of T, each component of $T - x_i$ contains at least one of the predecessors $x_1, x_2, \ldots, x_{i-1}$ of x_i; and (d) the costs $c(i, \sigma)$ are nonincreasing for $i \geq 2$, that is, $c_2 \geq c_3 \geq \cdots \geq c_k$.

We verify all of these, starting with condition (d).

Lemma 3.1. *Given any edge-weighted tree T and an integer $k \geq 2$, there exists a sequence $\sigma = (x_1, x_2, \ldots, x_k)$ of k vertices of T for which $L(\sigma) = Opt\,(k, T)$ and for which $c(2, \sigma) \geq c(3, \sigma) \geq \cdots \geq c(k, \sigma)$.*

Proof. Consider any set of k distinct vertices of T. It suffices to show that there exists an arrangement $\sigma = (v_1, v_2, \ldots, v_k)$ of those k vertices for which $L(\sigma)$ is maximum (over all arrangements of that fixed set) and for which $c(2, \sigma) \geq c(3, \sigma) \geq \cdots \geq c(k, \sigma)$. Let $\sigma = (x_1, x_2, \ldots, x_k)$ be an arrangement of those k vertices for which $L(\sigma)$ is maximum. Suppose to the contrary that $c(i, \sigma) < c(i+1, \sigma)$ for some $i \in \{1, 2, \ldots, k-1\}$.

Let σ' be the permutation, of the same k vertices, obtained from σ by interchanging x_i and x_{i+1}. We claim that $c(i, \sigma') + c(i+1, \sigma') \geq c(i, \sigma) + c(i+1, \sigma)$. This claim follows from the inequalities $c(i+1, \sigma') \geq c(i, \sigma)$ and $c(i, \sigma') \geq c(i+1, \sigma)$, of which we prove the first (the second being similar). Assume to the contrary that $c(i+1, \sigma') < c(i, \sigma)$. Because the first $i-1$ points of σ and σ' are the same, this assumption implies $c(i+1, \sigma') = \text{dist}(x_i, x_{i+1})$. Hence, we get $c(i+1, \sigma) \leq \text{dist}(x_i, x_{i+1}) = c(i+1, \sigma') < c(i, \sigma)$, yielding the contradiction $c(i+1, \sigma) < c(i, \sigma)$ and proving the claim.

Now consider any vertex $x_j, j \neq i, i+1$. The predecessors of x_i are the same in σ' as in σ, so we have $c(j, \sigma') = c(j, \sigma)$ for all $j \neq i, i+1$. Therefore, $L(\sigma') \geq L(\sigma)$. Because σ maximized L, we have $L(\sigma') = L(\sigma)$, and the lists of the costs c_i are the same for σ as for σ', only in a different order. If the costs in σ' are not yet in the desired nonincreasing order, we can iterate the above interchanging process so as to bring the largest c_i to the front of the list, then the second-largest as next, and so on, terminating in an arrangement in which L is still maximized and has nonincreasing costs, proving the first statement in the lemma. Starting with a sequence for which $L(\sigma) = \text{Opt}(k, T)$, the second statement in the lemma is an immediate consequence. ∎

Next we verify condition (c) above.

Lemma 3.2. *Let T be an edge-weighted tree, $k \geq 2$ an integer, and σ a sequence of k vertices $x_1, x_2, \ldots, x_k$ of T for which $L(\sigma) = \text{Opt}(k, T)$. Suppose x_i is a cut-vertex of T for some i, $1 < i \leq k$. Then each component of $T - x_i$ contains at least one of the predecessors $x_1, x_2, \ldots, x_{i-1}$ of x_i in σ.*

Proof. Suppose σ is a sequence for such a T with $L(\sigma) = \text{Opt}(k, T)$. Suppose to the contrary that some x_i is a cut-vertex of T with $i > 1$, but some component of $T - x_i$ contains none of the predecessors $x_1, x_2, \ldots, x_{i-1}$ of x_i. If no x_j is in that component then we could increase $L(\sigma)$ by replacing x_i in σ by any vertex in that component, contradicting the optimality of σ. Therefore, some x_j is in that component, and we have $j > i$ by assumption. Let us also take j to be the minimum with respect to these properties.

Consider the sequence σ' obtained from σ by interchanging terms x_i and x_j while leaving all other terms the same; that is, let $\sigma'(i) = x_j$, $\sigma'(j) = x_i$ and $\sigma'(r) = \sigma(r)$ for $r \neq i, j$. Then by minimality of j we have $c(i, \sigma') = c(i, \sigma) + \text{dist}(x_i, x_j)$, while $c(j, \sigma') > 0$, and $c(j, \sigma) = \text{dist}(x_i, x_j)$. This implies $c(i, \sigma') + c(j, \sigma') > c(i, \sigma) + c(j, \sigma)$. On the other hand, for $m \neq i, j$ we have $c(m, \sigma') \geq c(m, \sigma)$ when $m < j$, and $c(m, \sigma') = c(m, \sigma)$ when $m > j$, again using minimality of j. It follows that $L(\sigma') > L(\sigma)$, contradicting $L(\sigma) = \text{Opt}(k, T)$. ∎

The following corollary gives us conditions (a) and (b).

Corollary 3.2.1. *Let σ be a sequence of k vertices $x_1, x_2, \ldots, x_k$, $k \geq 2$, in an edge-weighted tree T. If $L(\sigma) = \text{Opt}(k, T)$, then x_1 and x_2 must be leaves of T.*

Proof. Suppose, to the contrary, that x_2 is a cut-vertex. Then it cannot have a predecessor in each component of $T - x_2$ (because it has only one predecessor), contradicting

Lemma 3.2. So x_2 is a leaf. As for x_1, observe that interchanging the first two vertices in σ has no effect on $L(\sigma)$, so x_1 must also be a leaf. ∎

4. OPTIMAL SOLUTIONS FOR THE DISCRETE PATH

In this section we study the SSP problem for the path P_{n+1} having $n+1$ vertices and n edges, our main results being a concise explicit formula solution, and an unexpected characterization we found for optimal sequences (i.e., those realizing $\mathrm{Opt}(n+1, P_{n+1})$). It might be expected that a length k sequence realizing $\mathrm{Opt}(k, P_{n+1})$ for each k can be found by applying the greedy algorithm; that is, by choosing the jth vertex, $1 \le j \le k$, to be a midpoint of any interval joining two successive (on the path) previously chosen vertices. Our first result will show that, indeed, the greedy algorithm always yields such an optimal solution. We also managed to characterize all length $n+1$ sequences which realize $\mathrm{Opt}(n+1, P_{n+1})$, finding to our surprise a rich variety of such sequences including nongreedy ones.

We denote the vertices of P_{n+1} by the integers $0, 1, 2, \ldots, n$, where the edges are the pairs $\{i, i+1\}$, with $1 \le i \le n$. The algorithm below produces a length $n+1$ sequence of the distinct vertices of P_{n+1}, of which each initial segment of length k, $2 \le k \le n+1$, will be shown to realize $\mathrm{Opt}(k, P_{n+1})$. The algorithm is greedy in the following sense. Let the sequence the algorithm produces be $\beta_n = \{x(i): 1 \le i \le n+1\}$, and let S_k be the initial segment $\{x(i): 1 \le i \le k\}$. Then $x(k+1)$ is chosen as a vertex $x \notin S_k$ which maximizes $\min\{\mathrm{dist}(x, y): y \in S_k\}$.

Algorithm GREEDY(n) Input: The discrete path P_{n+1} with vertices $\{0, 1, \ldots, n\}$ listed in left-to-right order.
Output: A permutation $\beta_n = (x(1), x(2), \ldots, x(n+1))$ of the vertices of P_{n+1}.

1. Initialization
$$x(1) = 0, x(2) = n, x(3) = \left\lfloor \tfrac{n}{2} \right\rfloor.$$
2. Recursively apply GREEDY$\left(\left\lfloor \frac{n}{2} \right\rfloor\right)$ and GREEDY $\left(\left\lceil \frac{n}{2} \right\rceil\right)$ to obtain respective output sequences
$$\beta_{\lfloor n/2 \rfloor} = \left(y(1), y(2), \ldots, y\left(\left\lfloor \tfrac{n}{2} \right\rfloor + 1\right)\right) \text{ and}$$
$$\beta_{\lceil n/2 \rceil} = \left(z(1), z(2), \ldots, z\left(\left\lceil \tfrac{n}{2} \right\rceil + 1\right)\right).$$
3. Merge the two sequences $\beta_{\lfloor n/2 \rfloor}$ and $\beta_{\lceil n/2 \rceil}$ by alternating them and making the obvious translation in vertex names. That is, let the desired sequence be $\beta_n = (x(1), x(2), \ldots, x(n+1))$, where $x(i)$, $1 \le i \le 3$, is given by the initialization, and for $4 \le i \le n+1$ is given by
$$x(i) = \left\lfloor \tfrac{n}{2} \right\rfloor + z\left(\left\lceil \tfrac{i+1}{2} \right\rceil\right) \qquad \text{for } i \text{ even, and}$$
$$x(i) = y\left(\left\lfloor \tfrac{i+1}{2} \right\rfloor\right) \qquad \text{for } i \text{ odd.}$$

end

The merging in step 3 implies, by a simple induction, that for any $k \ge 2$ the initial k vertices of the sequence β_n are indeed chosen in the greedy fashion. In fact, this merging also guarantees a balance property of β_n, given in the next lemma.

It is useful to have the following notation. Given a sequence σ of length k and integer $n \le k$ let $L(n, \sigma)$ denote $\sum_{i=1}^{n} c(i, \sigma)$, the nth partial sum of $L(\sigma)$. Also let $I(\sigma)$, the *interior cost* of σ, be given by $I(\sigma) = \sum_{i=3}^{k} c(i, \sigma)$.

Lemma 4.1. *$L(r, \beta_k) + L(r-1, \beta_{k+1}) \le L(r-1, \beta_k) + L(r, \beta_{k+1})$ for any k and $r \le k+1$.*

Proof. The left-hand side of the claimed inequality is $L(r-1, \beta_k) + L(r-1, \beta_{k+1}) + c(r, \beta_k)$, while the right-hand side is $L(r-1, \beta_k) + L(r-1, \beta_{k+1}) + c(r, \beta_{k+1})$.

We will therefore show by induction on k that $c(r, \beta_k) \le c(r, \beta_{k+1})$, where $r \le k+1$. This inequality is easily verified for small k, so assume the statement true for integers $k' < k$. From the algorithm GREEDY(u) we see that for any integer u and any r where $4 \le r \le u+1$ we have $c(r, \beta_u) = c\left(\left\lceil \frac{r+1}{2} \right\rceil, \beta_{\lceil u/2 \rceil}\right)$ when r is even and $c(r, \beta_u) = c\left(\left\lfloor \frac{r+1}{2} \right\rfloor, \beta_{\lfloor u/2 \rfloor}\right)$ when r is odd. Hence, we are reduced to showing that $c\left(\left\lceil \frac{r+1}{2} \right\rceil, \beta_{\lceil k/2 \rceil}\right) \le c\left(\left\lceil \frac{r+1}{2} \right\rceil, \beta_{\lceil (k+1)/2 \rceil}\right)$ and $c\left(\left\lceil \frac{r+1}{2} \right\rceil, \beta_{\lfloor k/2 \rfloor}\right) \le c\left(\left\lceil \frac{r+1}{2} \right\rceil, \beta_{\lfloor (k+1)/2 \rfloor}\right)$. But these follow by the inductive hypothesis. ∎

We can now show the optimality of the sequence β_n produced by the greedy algorithm.

Theorem 4.2. *For the path P_{n+1} of length n, we have $\mathrm{Opt}(k, P_{n+1}) = L(k, \beta_n)$ for any $k \le n+1$. That is, the initial segment of length k in the sequence β_n realizes $\mathrm{Opt}(k, P_{n+1})$.*

Proof. Let $\sigma = \sigma_{n,k} = (x_1, x_2, x_3, \ldots, x_k)$ be any optimal sequence realizing $\mathrm{Opt}(k, P_{n+1})$, where we may assume by Corollary 3.2.1 that x_1 and x_2 are the end points of P_{n+1}. Let $f(k-2, n) = I(\sigma) = \sum_{i=3}^{k} c(i, \sigma)$ be the internal cost of σ, where we have parametrized f in its first input coordinate by the number $(k-2)$ of internal path points which follow x_1 and x_2 in the sequence σ. Recall that the first two points of β_n are also the endpoints of P_{n+1}. So we similarly let $g(k-2, n) = \sum_{i=3}^{k} c(i, \beta_n)$ be the contribution to $L(k, \beta_n)$ from the $k-2$ internal path points among the first k points of β_n.

It suffices to show that $f(r, n) \le g(r, n)$ for all n, with $r \le n-1$. We will prove this by induction on n. The statement being trivially true for small n, assume for some n that $f(x, n') \le g(x, n')$ whenever $n' < n$ and $x \le n'-1$, and consider any $r \le n-1$. In the optimal sequence $\sigma = (x_1, x_2, x_3, \ldots, x_{r+2})$ the vertex x_3 is at some distance $t \le \left\lfloor \frac{n}{2} \right\rfloor$ from one of the end points, say from the "left" one. Of the remaining $r-1$ internal vertices, some number, say q, are internal vertices in the left subpath of length t while the remaining $r-1-q$ are internal vertices in the right subpath of length $n-t$. Also, the contributions of these sets of internal vertices to $L(\sigma)$ must realize $f(q, t)$ and $f(r-1-q, n-t)$ respectively. We therefore have

$$f(r, n) = t + f(q, t) + f(r-1-q, n-t). \qquad (4.1)$$

By the inductive hypothesis, we know that the appropriate initial segments GREEDY(t) and GREEDY($n-t$) produce optimal sequences of length q and length $r-1-q$ in the subpaths of length t and $n-t$, respectively. Hence, we have

$$f(q,t)=\left\lfloor\frac{t}{2}\right\rfloor+f\left(\left\lfloor\frac{q-1}{2}\right\rfloor,\left\lfloor\frac{t}{2}\right\rfloor\right)+f\left(\left\lceil\frac{q-1}{2}\right\rceil,\left\lceil\frac{t}{2}\right\rceil\right),$$

and

$$f(r-1-q,n-t)=\left\lfloor\frac{n-t}{2}\right\rfloor+f\left(\left\lfloor\frac{r-q-2}{2}\right\rfloor,\left\lfloor\frac{n-t}{2}\right\rfloor\right)+f\left(\left\lceil\frac{r-q-2}{2}\right\rceil,\left\lceil\frac{n-t}{2}\right\rceil\right).$$

We substitute these expressions into (4.1). On associating the middle term of each of these two expressions with the last term of the other expression, and replacing t by $\lfloor\frac{t}{2}\rfloor+\lceil\frac{t}{2}\rceil$, equation (4.1) becomes the following, called (4.1.1):

$$\begin{aligned}f(r,n)=&\left\{\left\lfloor\frac{t}{2}\right\rfloor+f\left(\left\lfloor\frac{q-1}{2}\right\rfloor,\left\lfloor\frac{t}{2}\right\rfloor\right)+f\left(\left\lceil\frac{r-q-2}{2}\right\rceil,\left\lceil\frac{n-t}{2}\right\rceil\right)\right\}\\&+\left\{\left\lceil\frac{t}{2}\right\rceil+f\left(\left\lceil\frac{q-1}{2}\right\rceil,\left\lceil\frac{t}{2}\right\rceil\right)+f\left(\left\lfloor\frac{r-q-2}{2}\right\rfloor,\left\lfloor\frac{n-t}{2}\right\rfloor\right)\right\}+\left\lfloor\frac{t}{2}\right\rfloor+\left\lfloor\frac{n-t}{2}\right\rfloor.\end{aligned}$$

The braces suggest the next step; to relate the quantities inside the first and second braces to f-values on paths of length $\lfloor\frac{t}{2}\rfloor+\lceil\frac{n-t}{2}\rceil$ and length $\lceil\frac{t}{2}\rceil+\lfloor\frac{n+t}{2}\rfloor$, respectively. As background, observe that if in Equation (4.1) we replace t by some positive integer $t'\neq t$, still with $t'\leq\lfloor\frac{n}{2}\rfloor$, then Equation (4.1) becomes an inequality in which the left side is at least as large as the right. We will apply this inequality to the expressions in the two braces. In the first braces, $\lfloor\frac{t}{2}\rfloor$ plays the role of the replacement t' in (4.1), while $\lfloor\frac{t}{2}\rfloor+\lceil\frac{n-t}{2}\rceil$ plays the role of n. Here, the assumption $t\leq\lfloor\frac{n}{2}\rfloor$ implies $\lfloor\frac{t}{2}\rfloor\leq\lceil\frac{n-t}{2}\rceil$, making this application in the first braces valid, so

$$\begin{aligned}&\left\lfloor\frac{t}{2}\right\rfloor+f\left(\left\lfloor\frac{q-1}{2}\right\rfloor,\left\lfloor\frac{t}{2}\right\rfloor\right)+f\left(\left\lceil\frac{r-q-2}{2}\right\rceil,\left\lceil\frac{n-t}{2}\right\rceil\right)\\&\quad\leq f\left(\left\lfloor\frac{q-1}{2}\right\rfloor+\left\lceil\frac{r-q-2}{2}\right\rceil+1,\left\lfloor\frac{t}{2}\right\rfloor+\left\lceil\frac{n-t}{2}\right\rceil\right).\end{aligned}$$

In the second braces, $\lceil\frac{t}{2}\rceil+\lfloor\frac{n-t}{2}\rfloor$ plays the role of n. If $\lceil\frac{t}{2}\rceil\leq\lfloor\frac{n-t}{2}\rfloor$, then we let $\lceil\frac{t}{2}\rceil$ play the role of the replacement t'. If $\lceil\frac{t}{2}\rceil>\lfloor\frac{n-t}{2}\rfloor$ (in which case t is odd and $t=\frac{n}{2}$), then we interchange the $\lceil\frac{t}{2}\rceil$ in the second braces with the last term $\lfloor\frac{n-t}{2}\rfloor$ of (4.1.1), and we let $\lfloor\frac{n-t}{2}\rfloor$ play the role of the replacement t'. This validates the application of the inequality in the (possibly modified) second braces. We get

$$\begin{aligned}&\left\lceil\frac{t}{2}\right\rceil+f\left(\left\lceil\frac{q-1}{2}\right\rceil,\left\lceil\frac{t}{2}\right\rceil\right)+f\left(\left\lfloor\frac{r-q-2}{2}\right\rfloor,\left\lfloor\frac{n-t}{2}\right\rfloor\right)\\&\quad\leq f\left(\left\lceil\frac{q-1}{2}\right\rceil+\left\lfloor\frac{r-q-2}{2}\right\rfloor+1,\left\lceil\frac{t}{2}\right\rceil+\left\lfloor\frac{n-t}{2}\right\rfloor\right),\end{aligned}$$

or the same inequality with the first term $\lceil\frac{t}{2}\rceil$ on the left side replaced by $\lfloor\frac{n-t}{2}\rfloor$.

Combining these upper bounds for the quantities in braces, we get an upper bound for $f(r,n)$ which is analyzed below.

As an abbreviation let $W=q-1$, $X=r-q-2$, $Y=t$, and $Z=n-t$, and then consider cases defined by parity. Our upper bounds for the quantities in braces imply

$$\begin{aligned}f(r,n)\leq&\, f\left(\left\lfloor\frac{W}{2}\right\rfloor+\left\lceil\frac{X}{2}\right\rceil+1,\left\lfloor\frac{Y}{2}\right\rfloor+\left\lceil\frac{Z}{2}\right\rceil\right)\\&+f\left(\left\lceil\frac{W}{2}\right\rceil+\left\lfloor\frac{X}{2}\right\rfloor+1,\left\lceil\frac{Y}{2}\right\rceil+\left\lfloor\frac{Z}{2}\right\rfloor\right)\\&+\left\lfloor\frac{n}{2}\right\rfloor.\end{aligned}\tag{4.2}$$

We use the facts that for integers I and J,

(i) if I and J are both even or both odd, then $\lfloor\frac{I}{2}\rfloor+\lceil\frac{J}{2}\rceil=\lceil\frac{I}{2}\rceil+\lfloor\frac{J}{2}\rfloor=\frac{I+J}{2}$, and
(ii) if I is even and J is odd, then $\lceil\frac{I}{2}\rceil+\lfloor\frac{J}{2}\rfloor=\frac{I+J-1}{2}$, and $\lfloor\frac{I}{2}\rfloor+\lceil\frac{J}{2}\rceil=\frac{I+J+1}{2}$.

CASE 1. W even, X odd, and Y even, Z odd. Then (4.2) implies

$$\begin{aligned}f(r,n)&\leq f\left(\frac{W+X+1}{2}+1,\frac{Y+Z+1}{2}\right)\\&\quad+f\left(\frac{W+X-1}{2}+1,\frac{Y+Z-1}{2}\right)+\left\lfloor\frac{n}{2}\right\rfloor\\&=f\left(\frac{r}{2},\frac{n+1}{2}\right)+f\left(\frac{r}{2}-1,\frac{n-1}{2}\right)+\left\lfloor\frac{n}{2}\right\rfloor\\&\leq g\left(\frac{r}{2},\frac{n+1}{2}\right)+g\left(\frac{r}{2}-1,\frac{n-1}{2}\right)\\&\quad+\left\lfloor\frac{n}{2}\right\rfloor\quad\text{(by induction)}\\&=g(r,n).\end{aligned}$$

CASE 2. W odd, X even, and Y odd, Z even. This case is symmetric with case 1, resulting in the reversal of order of the two f-terms in the second line.

CASE 3. W even, X odd, and Y odd, Z even. Then (4.2) implies

$$\begin{aligned}f(r,n)&\leq f\left(\frac{W+X+1}{2}+1,\frac{Y+Z-1}{2}\right)\\&\quad+f\left(\frac{W+X-1}{2}+1,\frac{Y+Z+1}{2}\right)+\left\lfloor\frac{n}{2}\right\rfloor\\&=f\left(\frac{r}{2},\frac{n-1}{2}\right)+f\left(\frac{r}{2}-1,\frac{n+1}{2}\right)+\left\lfloor\frac{n}{2}\right\rfloor\\&\leq g\left(\frac{r}{2},\frac{n-1}{2}\right)+g\left(\frac{r}{2}-1,\frac{n+1}{2}\right)\\&\quad+\left\lfloor\frac{n}{2}\right\rfloor\quad\text{(by induction)}\end{aligned}$$

$$\leq g\left(\frac{r}{2}, \frac{n+1}{2}\right) + g\left(\frac{r}{2} - 1, \frac{n-1}{2}\right)$$
$$+ \left\lfloor \frac{n}{2} \right\rfloor \text{ (by Lemma 4.1)}$$
$$= g(r, n).$$

CASE 4. W odd, X even, and Y even, Z odd. This case is symmetric with case 3, resulting in the reversal of order of the two f-terms in the second line.

CASE 5. W and X the same parity, and Y and Z the same parity. Then (4.2) implies

$$f(r,n) \leq f\left(\frac{W+X}{2} + 1, \frac{Y+Z}{2}\right)$$
$$+ f\left(\frac{W+X}{2} + 1, \frac{Y+Z}{2}\right) + \left\lfloor \frac{n}{2} \right\rfloor$$
$$= f\left(\frac{r-1}{2}, \frac{n}{2}\right) + f\left(\frac{r-1}{2}, \frac{n}{2}\right) + \frac{n}{2}$$
$$\leq g\left(\frac{r-1}{2}, \frac{n}{2}\right) + g\left(\frac{r-1}{2}, \frac{n}{2}\right)$$
$$+ \frac{n}{2} \text{ (by induction)}$$
$$= g(r, n).$$

CASE 6. W and X the same parity, and Y even, Z odd. Then (4.2) implies

$$f(r,n) \leq f\left(\frac{W+X}{2} + 1, \frac{Y+Z+1}{2}\right)$$
$$+ f\left(\frac{W+X}{2} + 1, \frac{Y+Z-1}{2}\right) + \left\lfloor \frac{n}{2} \right\rfloor$$
$$= f\left(\frac{r-1}{2}, \frac{n+1}{2}\right) + f\left(\frac{r-1}{2}, \frac{n-1}{2}\right) + \left\lfloor \frac{n}{2} \right\rfloor$$
$$\leq g\left(\frac{r-1}{2}, \frac{n+1}{2}\right) + g\left(\frac{r-1}{2}, \frac{n-1}{2}\right)$$
$$+ \left\lfloor \frac{n}{2} \right\rfloor \quad \text{(by induction)}$$
$$= g(r, n).$$

CASE 7. W and X the same parity, and Y odd, Z even. This case is symmetric with case 6, resulting in the reversal of order of the two f-terms in the second line.

CASE 8. W even and X odd, and Y and Z the same parity. Then (4.2) implies

$$f(r,n) \leq f\left(\frac{W+X+1}{2} + 1, \frac{Y+Z}{2}\right)$$
$$+ f\left(\frac{W+X-1}{2} + 1, \frac{Y+Z}{2}\right) + \left\lfloor \frac{n}{2} \right\rfloor$$
$$= f\left(\frac{r}{2}, \frac{n}{2}\right) + f\left(\frac{r}{2} - 1, \frac{n}{2}\right) + \frac{n}{2}$$
$$\leq g\left(\frac{r}{2}, \frac{n}{2}\right) + g\left(\frac{r}{2} - 1, \frac{n}{2}\right) + \frac{n}{2} \text{ (by induction)}$$
$$= g(r, n).$$

CASE 9. W odd, X even, Y and Z the same parity. This case is symmetric with case 8, resulting in the reversal of order of the two f-terms in the second line. ∎

Although the preceding theorem gives us at least one sequence realizing $\mathrm{Opt}(k, P_{n+1})$ for each k, it is natural to ask for a characterization of all such sequences for a given k. We have found a surprising such characterization for the case $k = n+1$, admitting many more sequences than just the greedy one, as a corollary to the theorem that follows.

We know from Corollary 3.2.1 that any such sequence must have $\{x_1, x_2\} = \{0, n\}$. By symmetry we need only consider the case in which $2x_3 \leq n$. As a convenient notation, for positive integers a and b, let $HP(a, b) = \max\{2^p : p$ is an integer, and 2^p divides a or 2^p divides $b\}$.

Theorem 4.3. *Suppose t is an integer with $2 \leq 2t \leq n$. The 3-term sequence $x_1 = 0$, $x_2 = n$, $x_3 = t$ can be extended to an optimal sequence $\sigma = (x_1 = 0, x_2 = n, x_3 = t, x_4, \ldots, x_{n+1})$ of all the vertices of P_{n+1}, (that is, $\mathrm{Opt}(n+1, P_{n+1}) = L(\sigma)$) if and only if $n - 2t \leq HP(t, n-t)$.*

Proof. Our proof will be streamlined by the introduction of some notation. Let $f(n) = L(n+1, \beta_n) - n = \mathrm{Opt}(n+1, P_{n+1}) - n$, the total cost accrued by the last $n-1$ points of the sequence β_n of $n+1$ points in P_{n+1}, these being the internal vertices of P_{n+1} because x_1 and x_2 are the ends. For convenience, we let $f(0) = f(1) = 0$. Note that

$$f(n) = \left\lfloor \frac{n}{2} \right\rfloor + f\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + f\left(\left\lceil \frac{n}{2} \right\rceil\right),$$

by the recursive nature of the sequence of the algorithms GREEDY(u). For whole numbers t and n satisfying $2t \leq n$, let $f(t, n)$ denote $t + f(t) + f(n-t)$, so that for $t > 0$, $f(t, n)$ can be interpreted as the total cost accrued by the last $n-1$ points of a sequence σ of the $n+1$ points of P_{n+1}, the fist two points of σ chosen as the end points, the third at distance t from the nearer end of the path, and the remaining points chosen greedily. Once we decide (about a sequence σ of all $n+1$ vertices of P_{n+1}) that $x_1 = 0, x_2 = n, x_3 = t$, then by the previous theorem we can maximize $L(\sigma)$ by choosing the remaining points greedily. Hence, such a sequence satisfying $\mathrm{Opt}(n+1, P_{n+1}) = L(\sigma)$ will exist if and only if $f(t, n) = f(n)$. In this setting we use the notation $\Delta = \Delta(t, n) = (n-t) - t$, so that $x_3 = t$ partitions the path of length n into paths of lengths t and $n - t = t + \Delta$, these lengths differing by exactly Δ. So, it suffices for us to prove that $f(t, n) = f(n)$ if and only if $\Delta(t, n) \leq HP(t, n-t)$.

Let $C1(t, n)$ denote the condition that $f(t, n) = f(n)$, and let $C2(t, n)$ denote the condition that $\Delta(t, n) \leq HP(t, n-t)$. We proceed by induction on n. For the basis case $n = 2$ (forcing $t = 1$) it is easy to verify that $C1(1,2)$ and $C2(1,2)$ are both true, verifying the basis case. Suppose the claim is true for all $n < m$, and consider some t with $2 \le 2t \le m$, where $\Delta(t,m) = m - 2t$.

CASE 1. $m = 2i$ and $t = 2j$ are even. From $2t \le m$ we have $2j \le i$. Note that $\Delta(t,m) = 2\Delta(j,i)$ and that $HP(t, m-t) = 2\,HP(j, i-j)$.
Now, $f(m) = i + 2f(i)$, while $f(t,m) = 2j + f(2j) + f(2i - 2j) = i + 2[j + f(j) + f(i-j)]$. Therefore, $C1(t,m)$ is true $\Leftrightarrow f(t,m) = f(m) \Leftrightarrow f(i) = j + f(j) + f(i-j) \Leftrightarrow C1(j,i)$ is true. By the inductive hypothesis, $C1(j,i)$ is true $\Leftrightarrow C2(j,i)$ is true $\Leftrightarrow \Delta(j,i) \le HP(j, i-j) \Leftrightarrow 2\Delta(j,i) \le 2\,HP(j, i-j) \Leftrightarrow \Delta(t,m) \le HP(t, m-t) \Leftrightarrow C2(t,m)$ is true. Therefore, $C1(t,m)$ is true $\Leftrightarrow C2(t,m)$ is true, as desired.

CASE 2. $m = 2i$ is even and $t = 2j + 1$ is odd. Note that $HP(t, n-t) = 1$, because t and $n - t$ are odd. If $C2(t,m)$ is true, then it must be that $t = i$, in which case the sequence corresponding to $f(t,m)$ is greedy even at the third step, so has maximum cost, that is, $C1(t,m)$ is true. Suppose instead that $C2(t,m)$ is false, that is, that $t < i$. Then $2j < 2(j+1) \le i$, so we have that $j + f(j) + f(i-j) \le f(i)$, and that $(j+1) + f(j+1) + f(i-(j+1)) \le f(i)$ by Theorem 4.2. Thus, we obtain that

$$\begin{aligned} f(t,m) &= 2j + 1 + f(2j+1) + f(2i - 2j - 1) \\ &= 2j + 1 + j + f(j) + f(j+1) + (i - j - 1) \\ &\quad + f(i - j - 1) + f(i - j) \\ &= i - 1 + [j + f(j) \\ &\quad + f(i-j)] + [(j+1) \\ &\quad + f(j+1) + f(i - (j+1))] \\ &\le i - 1 + f(i) + f(i) < i + f(i) + f(i) = f(m). \end{aligned}$$

That is, $f(t,m) < f(m)$, so $C1(t,m)$ is false.

CASE 3. $m = 2i + 1$ and $t = 2j + 1$ are odd. We have $2(2j+1) \le 2i + 1$, so by parity considerations, $2(2j+1) \le 2i$, so $2j + 1 \le i$. Thus, $2j \le 2j + 1 \le i$ and $2(j+1) \le i + 1$, so we have that $j + f(j) + f(i-j) \le f(i)$ and $j + 1 + f(j+1) + f(i-j) \le f(i+1)$, again by Theorem 4.2. Now

$$\begin{aligned} f(t,m) &= 2j + 1 + f(2j+1) + f(2i - 2j) \\ &= 2j + 1 + j + f(j) + f(j+1) + i - j + 2f(i-j) \\ &= i + [j + f(j) + f(i-j)] + [j + 1 + f(j+1) \\ &\quad + f(i-j)] \le i + f(i) + f(i+1) = f(m). \end{aligned}$$

So, $C1(t,m)$ is true $\Leftrightarrow$ equality holds in the previous inequality $\Leftrightarrow C2(j,i)$ and $C2(j+1, i+1)$ are true (by the inductive hypothesis) $\Leftrightarrow i - 2j \le HP(j, i-j)$ and $i - 2j - 1 \le HP(j+1, i-j)$. We now use the last condition equivalent to $C1(t,m)$ to prove the equivalence of $C1(t,m)$ and $C2(t,m)$.

Suppose $C2(t,m)$ is true, that is, $2i - 4j - 1 \le HP(2j+1, 2i - 2j) = 2\,HP(1, i-j)$. Because $2i - 4j - 1$ is odd and $2\,HP(1, i-j)$ is even, we have $2i - 4j \le 2\,HP(1, i-j)$, so $i - 2j \le HP(1, i-j)$, from which both $i - 2j \le HP(j, i-j)$ and $i - 2j - 1 \le HP(j+1, i-j)$ follow, so $C1(t,m)$ is true, as desired.

Conversely, suppose $C1(t,m)$ is true, that is, that $i - 2j \le HP(j, i-j)$ and $i - 2j - 1 \le HP(j+1, i-j)$. It suffices to show that $i - 2j \le HP(1, i-j)$, because from there it follows that $2i - 4j - 1 \le 2i - 4j \le HP(1, 2i - 2j) = HP(2j+1, 2i - 2j)$, that is, $C2(t,m)$ is true. If j is odd, then $i - 2j \le HP(j, i-j) = HP(1, i-j)$. Suppose j and i are both even. Then, because $i - 2j - 1 \le HP(j+1, i-j) = HP(1, i-j)$, and because $i - 2j - 1$ is odd and $HP(j+1, i-j)$ is even, we have $i - 2j \le HP(1, i-j)$, again as desired. Last, if j is even and i is odd, then $i - 2j - 1 \le HP(j+1, i-j) = 1$, but because $i - 2j - 1$ is even, it follows that $i - 2j = 1 = HP(1, i-j)$, as desired.

CASE 4. $m = 2i + 1$ is odd and $t = 2j$ is even. Because $2(2j) \le 2i + 1$ and $2i + 1$ is odd, we have $2j \le i \le i + 1$. Therefore, $j + f(j) + f(i-j) \le f(i)$ and $j + f(j) + f(i - j + 1) \le f(i+1)$. Thus, we obtain that

$$\begin{aligned} f(t,m) &= 2j + f(2j) + f(2i - 2j + 1) \\ &= 2j + j + 2f(j) + i - j + f(i-j) + f(i - j + 1) \\ &= i + [j + f(j) + f(i-j)] + [j + f(j) \\ &\quad + f(i - j + 1)] \le i + f(i) + f(i+1) = f(m). \end{aligned}$$

So, $C1(t,m)$ is true if and only if $C1(j,i)$ and $C1(j, i+1)$ are both true which (by inductive hypothesis) is true if and only if both $C2(j,i)$ and $C2(j, i+1)$ are true.

Suppose $C2(t,m)$ is true, that is, $2i - 4j + 1 \le HP(2j, 2i - 2j + 1) = 2\,HP(j, 1)$. Then $2i - 4j + 2 \le 2\,HP(j,1)$ [because $2i - 4j + 1$ is odd while $2\,HP(j,1)$ is even], so $i - 2j + 1 \le HP(j,1)$. So, $C2(j,i)$ is true because $i - 2j \le i - 2j + 1 \le HP(j,1) \le HP(j, i-j)$, and $C2(j, i+1)$ is true because $i - 2j + 1 \le HP(j,1) \le HP(j, i - j + 1)$. Therefore $C1(t,m)$ is true, as desired.

Conversely, suppose $C1(t,m)$ is true, so $C2(j,i)$ and $C2(j, i+1)$ are both true, that is, $i - 2j \le HP(j, i-j)$ and $i - 2j + 1 \le HP(j, i - j + 1)$. It suffices to show that $i - 2j + 1 \le HP(j,1)$, because then it would follow that $2i - 4j + 1 \le 2i + 4j + 2 \le HP(2j, 1) = HP(2j, 2i - 2j + 1)$, i.e. $C2(t,m)$ is true. If i and j have the same parity, then from $i - 2j + 1 \le HP(j, i - j + 1) = HP(j,1)$ we are done. Suppose i is odd and j is even. Then from $i - 2j \le HP(j, i-j) = HP(j,1)$, in which the left side is odd and the right side is even, we have $i - 2j + 1 \le HP(j,1)$, as desired. Last, suppose i is even and j is odd. Then from $i - 2j \le HP(j, i-j) = 1$ in which $i - 2j$ is even, we have $i - 2j = 0$, so $i - 2j + 1 = 1 = HP(j,1)$, as desired. ∎

Given a sequence $\sigma = (x_1, x_2, \ldots, x_{n+1})$ of the distinct vertices of P_{n+1} in which $\{x_1, x_2\} = \{0, n\}$, for each $i = 2, 3, \ldots, n$ the set $\{x_1, x_2, \ldots, x_i\}$ naturally partitions the path P_{n+1} into $i - 1$ edge disjoint subpaths. Each subpath has a successive pair of vertices x_j, x_k, for some $1 \le j, k \le i$, as its ends. For each $i = 2, 3, \ldots, n$ the vertex x_{i+1} lies interior to one such subpath, splitting it into two paths. Let the shorter length be t_i, and let the longer path have length $t_i + \Delta_i$.

Corollary 4.3.1. *A sequence $\sigma = (x_1, x_2, \ldots, x_{n+1})$ of the distinct vertices of P_{n+1} satisfies $\mathrm{Opt}(n+1, P_{n+1}) = L(\sigma)$ if and only the following hold: (a) $\{x_1, x_2\} = \{0, n\}$, and (b) For each $i = 2, 3, \ldots, n$ we have $\Delta_i \leq HP(t_i, t_i + \Delta_i)$*

Proof. The requirement $\{x_1, x_2\} = \{0, n\}$ is from Corollary 3.2.1. As we proceed after that, because we eventually split up each subinterval until all intervals have length l, the order in which we split the subintervals is immaterial to whether the sequence maximizes the total cost. We just have to ensure each time we split a subpath that the subdivision follows the criterion of Theorem 4.3.1, if we regard the new vertex as the third vertex used in the subpath, having already used both of its ends. ∎

According to Corollary 4.3.1, an optimal sequence in a discrete path can be quite flexible. For example, in P_{14} let's begin with $x_1 = 0$, $x_2 = 13 = n$. Then choosing $x_3 = 8$, we have $t_2 + \Delta_2 = 8, t_2 = 5$, and $\Delta_2 = 3$. Thus, $\Delta_2 \leq 8 = HP(t_2, t_2 + \Delta_2)$, showing that condition (b) is satisfied. Thus, the three term sequence $x_1 = 0, x_2 = 13, x_3 = 8$ can be extended to a sequence realizing $\mathrm{Opt}(14, P_{14})$. Now the choice $x_4 = 12$, taken from the length 5 interval $[x_3 = 8, x_2 = 13]$, yields $t_3 + \Delta_3, t_3 = 1$ and $\Delta_3 = 3$, so $\Delta_3 \leq 4 = HP(t_3, t_3 + \Delta_3)$. Again, because condition (b) is satisfied, the corollary implies that the four-term sequence $x_1 = 0$, $x_2 = 13$, $x_3 = 8, x_4 = 12$ can be extended to realize $\mathrm{Opt}(14, P_{14})$, and so on.

We now generalize this example to obtain a special sequence π_n, which realizes $\mathrm{Opt}(n+1, P_{n+1})$ while also making an explicit calculation of $\mathrm{Opt}(n+1, P_{n+1})$ especially convenient. Adopting the notation preceding Corollary 4.3.1, suppose inductively that the points $\{x_1, x_2, \ldots, x_i\}$ have been constructed, with $x_1 = 0$, and $x_2 = n + 1$. Choose one of the $i - 1$ resulting induced subintervals of $[0, n]$, say $[x_j, x_k]$, for some $1 \leq j, k \leq i$. We let x_{i+1} be an interior point in this interval, its distance from one end (either one) of the interval being the greatest power of 2 allowable. That is, take x_{i+1} to be at distance $t_i + \Delta_i$ from one end, and distance t_i from the other end, where $t_i + \Delta_i$ is the greatest power of 2 less than the length $x_k - x_j$ of this interval. So we have $t_i + \Delta_i = 2^{\lceil \log_2(2t_i + \Delta_i) \rceil - 1}$ and, hence, $x_{i+1} = x_j + 2^{\lceil \log_2(x_k - x_j) \rceil - 1}$, or $x_{i+1} = x_k - 2^{\lceil \log_2(x_k - x_j) \rceil - 1}$ (and we choose arbitrarily between these two alternatives). For this choice of x_{i+1} we have $HP(t_i, t_i + \Delta_i) = t_i + \Delta_i > \Delta_i$, so that the sequence $\pi_n = (x_1, x_2, \ldots, x_{n+1})$ inductively so constructed realizes $\mathrm{Opt}(n+1, P_{n+1})$ by Corollary 4.3.1.

For a sequence π_n constructed in this manner, let us abbreviate $L(\pi_n)$ by $L(n)$ and recall $I(n) = I(\pi_n) = \sum_{i=3}^{n+1} c(i, \pi_n) = L(\pi_n) - n$, the interior cost of π_n. The next lemma will lead to a formula for $\mathrm{Opt}(n+1, P_{n+1})$ in the theorem following.

Lemma 4.4. *Suppose n is a power of 2, say $n = 2^k$. Then $L(n) = 2^k\left(1 + \frac{k}{2}\right)$.*

Proof. Let $a_k = I(2^k)$. Then by the construction of π_n, clearly a_k satisfies the recurrence relation $a_k = 2a_{k-1} + 2^{k-1}$ (for $k \geq 1$), with initial condition $a_0 = 0$. From this we get $a_k = k \cdot 2^{k-1}$, and the result follows. ∎

Theorem 4.5. *Write n in base-2; $n = 2^{s_1} + 2^{s_2} + \cdots + 2^{s_{k-1}} + 2^{s_k}$, where $0 \leq s_1 < s_2 < \cdots < s_k$. Then $\mathrm{Opt}(n+1, P_{n+1}) = \sum_{i=1}^{k}\left(k - i + 1 + \frac{s_i}{2}\right)2^{s_i}$.*

Proof. It follows from the construction of π_n that $L(n)$ obeys the recurrence $L(n) = n + L(n - 2^{s_k}) + I(2^{s_k})$. Solving this recurrence and using Lemma 4.4 we get $L(n) = [n + (n - 2^{s_k}) + (n - 2^{s_k} - 2^{s_{k-1}}) + \cdots + 2^{s_1}] + [s_k 2^{s_k - 1} + s_{k-1} 2^{s_{k-1} - 1} + \cdots + s_2 2^{s_2 - 1} + s_1 2^{s_1 - 1}]$. The result now follows by observing that $\mathrm{Opt}(n+1, P_{n+1}) = L(n)$ by the construction of π_n and Corollary 4.3.1. ∎

5. SSP FOR CONTINUOUS STRUCTURES AND BOUNDS ON THE PERFORMANCE RATIO OF THE VERTEX GREEDY ALGORITHM

5.1. *Solution of SSP for Continuous Paths*

In this subsection we define the continuous version of the sequential sum problem, in the setting of the metric space associated naturally with any edge-weighted graph G. This will be a tool in obtaining our upper bounds for $L(\sigma)$ for any edge weighted graph.

Let us regard each edge $uv \in E(G)$ of weight w as a continuous interval of length w having end points u and v. Then define the associated "continuous" metric space *con-G* as follows. The underlying point set $V(\text{con-}G)$ of con-G will be the set of all vertices of G (these will be called *original* vertices of G), together with all "internal" points on intervals $\{uv: uv \in E(G)\}$ as just described, where any two such intervals are understood to be internally disjoint. We shall define the metric $\rho(x, y)$ for any pair of points x and y (internal or not) in the natural way as the length of the shortest route between x and y in con-G as follows. Consider any sequence $s = (x = x_0, e_1, x_1, e_2, x_2, \ldots, e_k, x_k = y)$ in con-G starting at x and ending at y, where each $x_i, i \neq 0, k$, is an original vertex of G and each e_i is an edge of G (viewed as an interval) containing x_{i-1} and x_i. Define $\lambda_s(x, y) = \sum_{i=1}^{k} \mathrm{dist}(x_{i-1}, x_i)$, where $\mathrm{dist}(x_{i-1}, x_i)$ is the usual Euclidean distance between x_{i-1} and x_i on the interval corresponding to e_i. Finally, let $\rho(x, y)$ be the minimum of $\lambda_s(x, y)$ over all such sequences s. This defines the metric space con-G associated with G.

As examples, con-K_2 is just an interval of length 1, while con-$K_{1,p}$ is a set of p intervals each of length 1 glued together at a central point, which is also an end point of each of them.

The SSP problem can then be defined on con-G in the obvious way, with $\rho(x, y)$ playing the role of the graph distance $\mathrm{dist}(x, y)$ in G for the discrete case. In particular, for a sequence $\sigma = (x_1, \ldots, x_k)$ of distinct points of con-G, we can let $c(i,\sigma) = \min_{1\le t<i} \rho(x_i, x_t)$ for $i \ge 2$, with $c(1,\sigma) = 0$. Now the notations $L(\sigma) = \sum_{i=1}^{k} c(i,\sigma)$ and $\mathrm{Opt}(k, \text{con-}G) = \max\{L(\sigma) : \sigma = (x_1, \ldots, x_k)$ is a sequence of distinct vertices in $G\}$ carry over directly, where k is finite but allowed to be arbitrarily large for any fixed G. The SSP problem on con-G is then to determine $\mathrm{Opt}(k, \text{con-}G)$ for a given G and k.

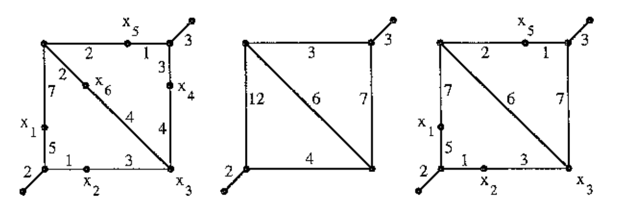
This continuous version of the SSP is naturally related to the discrete version defined previously. Let $e = xy$ be any edge in a weighted graph G. An *elementary refinement* of G is any graph G' obtained from G by "inserting" a point v of degree 2 "along" e while preserving total edge weight. More precisely, define G' by $V(G') = V(G) \cup \{v\}, E(G') = (E(G) - \{e\}) \cup \{vx, vy\}$, where the weights of the new edges vx and vy are positive and satisfy $w(vx) + w(vy) = w(e)$, and all other edge weights in G' remain what they were in G. A graph H obtained from G by any sequence of elementary refinements is called a *refinement* of G. Here, we allow G to be a refinement of itself. Then $\mathrm{Opt}(k, \text{con-}G) = \max\{\mathrm{Opt}(k, H)$: H is a refinement of G having at least k points$\}$, so $\mathrm{Opt}(k, \text{con-}G)$ depends on k and the underlying topology of G, but not on where in G its degree 2 points are.

Inverse to the operation of refinement there is the operation of reduction. When H is a refinement of G, we will say that G is a *reduction* of H. When G is the unique reduction of H having no points of degree 2 (such a G is sometimes called "homeomorphically irreducible"), then we denote G by *red*(H). Note that this definition allows red(H) to have loops (arising from the reduction of certain cycles), but our interest is confined to the case where red(H) is without loops. See Figure 4 for examples of reduction and refinement.

In Figure 1, the sequence σ of 10 points from con-$K_{1,3}$ (where each edge of $K_{1,3}$ has weight 1) are placed at distances from the center that are multiples of $\frac{1}{3}$. The fact that $L(\sigma) = \frac{22}{3}$ implies that $\mathrm{Opt}(10, \text{con-}K_{1,3}) \ge \frac{22}{3}$, and that $\mathrm{Opt}(10, T) \ge \frac{22}{3}$ for some refinement T of $K_{1,3}$ having at least 10 points (in fact one possible such T is the tree shown in Fig. 1).

We begin the study of the continuous SSP by showing that the natural greedy algorithm solves the SSP problem in con-K_2. Toward that end, for each integer $n \ge 0$ let $f(n) = \sum_{k=1}^{n} 2^{-1-\lfloor \log_2(k) \rfloor} = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{16} + \cdots + 2^{-1-\lfloor \log_2(n) \rfloor}$, where we set $f(0) = 0$.

We begin with a concavity property of the function f.



a) The weighted graph G and sequence $\sigma = (x_1, x_2, \ldots, x_6)$ b) red(G) c) The initial cover subsequence $\mathrm{Cov}(\sigma, I) = (x_1, x_2, x_3, x_5)$ for the interval partition consisting of edges of red(G)

FIG. 4. red(G) and an initial cover of G; graphs (a) and (c) are refinements of (b).

Lemma 5.1. *Suppose integers s,t and real number α satisfy $0 \le t \le s$ and $0 \le \alpha \le \frac{1}{2}$. Then $\alpha + \alpha f(t) + (1-\alpha) f(s-t) \le f(s+1)$, with equality if and only if $\alpha = \frac{1}{2}$ and $t = \lfloor s/2 \rfloor$ or $\lceil s/2 \rceil$.*

Proof. Set $a = \lfloor s/2 \rfloor$, and $b = \lceil s/2 \rceil$. First observe that $f(s+1) = \frac{1}{2} + \frac{1}{2} f(a) + \frac{1}{2} f(b)$, because $\frac{1}{2} f(a)$ is the sum of the terms of even index k in the sum defining $f(s+1)$, while $\frac{1}{2} f(b)$ is the sum of the terms of odd index $k \ge 3$ in the sum defining $f(s+1)$. Next, note that $\alpha + \alpha f(t) + (1-\alpha) f(s-t)$ is a linear function in α (for s, t fixed), so its maximum value over the interval $0 \le \alpha \le \frac{1}{2}$ is achieved at $\alpha = 0$ or $\alpha = \frac{1}{2}$ or both, that maximum value then being $\max\{f(s-t), \frac{1}{2} + \frac{1}{2} f(t) + \frac{1}{2} f(s-t)\}$. Now, $f(s-t) < f(s+1)$, because f is increasing, so it suffices to prove that $\frac{1}{2} + \frac{1}{2} f(t) + \frac{1}{2} f(s-t) \le f(s+1)$, with equality if and only if $t = a$ or b. But we have already observed that equality holds at values $t = a$ and b. Further, f is increasing and "concave down" as a function on integers, so $\frac{1}{2} f(t) + \frac{1}{2} f(s-t)$ achieves its maximum at precisely those inputs where t and $s-t$ are as nearly equal as allowed, which in this case is at the integer values $t = a$ and b. ∎

In con-K_2, consider the infinite sequence of points $0, 1, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}, \frac{1}{16}, \frac{3}{16}, \frac{5}{16}, \frac{7}{16}, \frac{9}{16}, \frac{11}{16}, \frac{13}{16}, \frac{15}{16}, \frac{1}{32}, \ldots$. Let μ_n be the finite subsequence formed by the first $n \ge 2$ terms of the preceding sequence. It is easy to verify $L(\mu_n) = 1 + f(n-2)$, and that μ_n is greedy. Observe now that if $M \ge 0$ is an integer with $2^k - 1 \le M < 2^{k+1} - 1$, then $f(M) = \frac{k}{2} + \frac{M - 2^k + 1}{2^{k+1}}$ with $k = \lfloor \log_2(M+1) \rfloor$. Hence, for $n \ge 3, L(\mu_n) = 1 + f(n-2) = 1 + \frac{k}{2} + \frac{n-1-2^k}{2^{k+1}} \le \frac{3}{2} + \frac{k}{2}$, where $k = \lfloor \log_2(n-1) \rfloor$. In the following theorem we prove for this topologically simple metric space that the greedy sequence μ_n produces the greatest cost among sequences of the same length.

Theorem 5.2. *For positive integers $n \ge 3$ we have $\mathrm{Opt}(n, \text{con-}K_2) = L(\mu_n) = 1 + \frac{k}{2} + \frac{n-1-2^k}{2^{k+1}} \le \frac{3}{2} + \frac{k}{2}$, where $k = \lfloor log_2(n-1) \rfloor$.*

Proof. We need only prove the first equality. The claim is trivially true when $n \le 2$. For $n \ge 2$, let $\tau = (v_1, v_2, \ldots, v_n)$ be a sequence of points in con-K_2, with $L(\tau) = \mathrm{Opt}(n, \text{con-}K_2)$. Because $L(\mu_n) = 1 + f(n-2)$, it suffices to show by induction on n that

$$L(\tau) \le 1 + f(n-2) \text{ for all } \tau \text{ of length } n \text{ in con-}K_2. \quad (5.1)$$

The basis case $n = 2$ has already been handled. We assume inductively that (5.1) holds for $2 \le n \le s+2$, and consider the case $n = s+3$.

We first note that without loss of generality $v_1 = 0$ and $v_2 = 1$. To see this, suppose for contradiction that v_2 is neither 0 nor 1, and that without loss of generality $v_1 < v_2$. If each v_i for $i \ge 3$ is less than v_2 then clearly the sequence is not optimal, because we can increase $L(\tau)$ by changing v_2 to be 1 and keeping all other v_i the same, contradicting the optimality of τ. Therefore, some v_i exceeds v_2, and we can let v_j be the first such v_i in τ. But upon creating τ' from τ by interchanging the values of v_2 and v_j and keeping all other v_i the same, we reach the contradiction that $L(\tau') > L(\tau)$. Therefore, v_2 must be an end of the interval [0,1]. Because interchanging the order of v_1 and v_2 does not alter the cost of the sequence (and therefore yields an optimal sequence), v_1 must also be an end of the interval [0,1]. Therefore, without loss of generality, $v_1 = 0$ and $v_2 = 1$ as claimed.

Let $\alpha = v_3$, and assume without loss of generality that $0 \leq \alpha \leq \frac{1}{2}$, where t of the points $v_4, v_5, \ldots, v_{s+3}$ are in $[0, \alpha]$, the remaining $s - t$ in $[\alpha, 1]$. We now apply the inductive hypothesis twice; first to the subsequence of $t + 2$ points consisting of $0, \alpha$, followed by the t points in $(0, \alpha)$, and second to the subsequence of $s - t + 2$ points consisting of $1, \alpha$, followed by the $s - t$ points in $(\alpha, 1)$. In each application, we are working with scaled versions of the interval [0, 1], one with a scaling factor α, and the other with a factor $1 - \alpha$. Then by the inductive hypothesis, $L(\tau) \leq 1 + \alpha + \alpha f(t) + (1 - \alpha) f(s - t)$, so that by Lemma 5.1 we have $L(\tau) \leq 1 + f(s + 1) = 1 + f(n - 2)$, proving (5.1). ∎

5.2. *Upper Bounds on the Performance Ratio of the Greedy Algorithm for the OST*

We now use our result (Theorem 5.2) on Opt(n, con-K_2) to derive an upper bound for $L(\sigma)/w(\sigma)$ in any weighted graph G and any sequence σ from $V(G)$, recalling that $w(\sigma)$ denotes the weight of a minimum Steiner tree interconnecting the points of σ. Such a bound is then also an upper bound on the performance ratio of the greedy algorithm for the OST.

Start with any subgraph H of G spanning the vertices of σ, where later H will be taken to be a minimum Steiner tree for σ. Our basic idea is to partition con-H into a collection $I = \{I_1, I_2, \ldots, I_p\}$ of intervals (i.e., scaled copies of con-K_2), where any two intervals are internally disjoint but may have either one or two end points in common. Letting σ_i be the induced subsequence of σ whose points lie on I_i, we use the values of $L(\sigma_i)$ derived from Theorem 5.2 to give an upper bound for $L(\sigma)$ itself.

We call such a collection I an *interval partition* of H, and we let the *weight*(I_i) be the length of I_i (see Fig. 5). Each point $v \in \sigma$ is viewed as belonging to some interval I_i through the

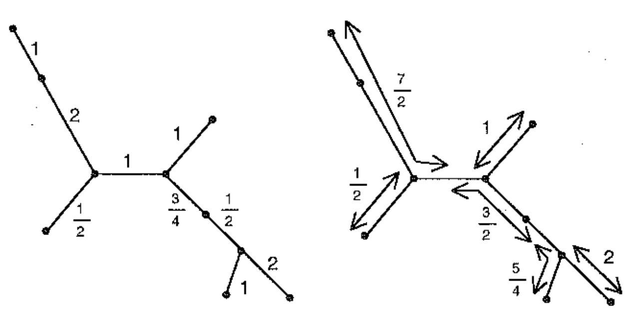


FIG. 5. An interval partition of a tree.

identification of con-G with G described earlier. For each $I_i \in I$, let $x(I_i)$ be the first point of σ (if such a point exists) contained in I_i. Note that it is possible to have distinct i and j for which $x(I_i) = x(I_j)$ because a point can be in more than one interval of the partition by being an end point of several intervals. We refer to the set $\{x(I_i) : I_i \in I\}$ as an *initial σ-cover* of I, and denote it by $Cov(\sigma, I)$ (see Fig. 4c).

Theorem 5.3. *Let G be an edge-weighted graph, and σ a sequence of n vertices of G. Let H be any subgraph of G contains the vertices of σ, and let $I = \{I_1, I_2, \ldots, I_p\}$ be an interval partition of H with weight$(I_i) = w_i$ for $1 \leq i \leq p$. Set*

$$W = \sum_{i=1}^{p} w_i = \text{weight}(H) \text{ and } K = \prod_{i=1}^{p} w_i^{w_i}.$$

Then

$$L(\sigma) \leq \frac{1}{2} \log_2\left(K \cdot \left(\frac{n}{W}\right)^{W}\right) + \sum_{x \in \mathrm{Cov}(\sigma, I)} c(x, \sigma) + \frac{3}{2} W.$$

Proof. For each point $x \in \sigma$, assign x to one of the at most $\deg_H(x)$ intervals of I containing x, calling it $I_{\alpha(x)}$. Note that because there may be several such candidate intervals, it is possible that $x = x(I_i)$ while $\alpha(x) \neq i$. Let $V(i) = \{x \in \sigma : i = \alpha(x)\}$ be the set of points of σ assigned in this way to I_i, so that $\{V(i)\}$ is a partition of the set of points of σ. Let $R(i) = V(i) - x(I_i)$, and $n_i = |V(i)|$. For each $x \in R(i)$, let $d(x) = \min\{\text{dist}(x, y) : y \in V(i) \text{ and } y \text{ precedes } x \text{ in } \sigma\}$. Then

$$L(\sigma) = \sum_{i=1}^{p} \sum_{x \in V(i)} c(x, \sigma)$$

$$= \sum_{x \in \mathrm{Cov}(\sigma, I)} c(x, \sigma) + \sum_{i=1}^{p} \sum_{x \in R(i)} c(x, \sigma),$$

where the sum

$$\sum_{x \in R(i)} c(x, \sigma)$$

is understood to be 0 if $R(i)$ is empty.

Consider the sum

$$Q = \sum_{i=1}^{p} \sum_{x \in R(i)} c(x, \sigma)$$

above on the right. For each i we have

$$\sum_{x \in R(i)} c(x, \sigma) \leq \sum_{x \in R(i)} d(x) \leq \frac{1}{2} w_i \log_2(n_i) + \frac{3}{2} w_i,$$

using Theorem 5.2 and scaling by a factor of w_i.

Thus,

$$Q \leq \frac{1}{2} \sum_{i=1}^{p} w_i \log_2(n_i) + \frac{3}{2} W.$$

We now maximize this upper bound for Q, viewed as a function of the p variables n_i, subject to the constraint $\sum_{i=1}^{p} n_i = n$. Applying Lagrange multipliers we find that the maximum is achieved at the values $n_i = \frac{w_i}{W}n$. Substituting these values into the upper bound for Q we obtain

$$Q \leq \frac{1}{2}\sum_{i=1}^{p} w_i \log_2\left(\frac{w_i n}{W}\right) + \frac{3}{2}W$$
$$= \frac{1}{2}\log_2\left(K \cdot \left(\frac{n}{W}\right)^W\right) + \frac{3}{2}W,$$

where K is as in the statement of the theorem. The theorem follows. ∎

The following upper bound on the performance ratio of the DGA follows immediately.

Corollary 5.3.1. *Let G be an edge-weighted graph, and σ a sequence of n vertices of G. Let T be a minimum Steiner tree in G for the vertices of σ, and let $I = \{I_1, I_2, \ldots, I_p\}$ be an interval partition of T with weight $(I_i) = w_i$ for $1 \leq i \leq p$. Set $W = \sum_{i=1}^{p} w_i = weight(T)$, and $K = \prod_{i=1}^{S} w_i^{w_i}$. Then*

$$\frac{L(\sigma)}{w(\sigma)} \leq \frac{1}{2W}\log_2\left(K \cdot \left(\frac{n}{W}\right)^W\right) + \frac{1}{W}\left(\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\sigma)\right) + \frac{3}{2}$$
$$= \frac{1}{2}\log_2(n) + \frac{1}{2W}\sum_{i=1}^{p} w_i \log_2(w_i) - \frac{1}{2}\log_2(W) + \frac{1}{W}\left(\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\sigma)\right) + \frac{3}{2}.$$

Proof. The inequality follows directly from Theorem 5.3, by letting T play the role of H. The equality follows by algebra. ∎

The flexibility in choosing an interval partition of T allows us to view the above bounds on $L(\sigma)/w(\sigma)$ in various ways, some leading to an improvement on known bounds and a proof of a conjectured bound on the performance ratio of the DGA in certain special cases. To start on this, we need a basic lemma concerning the induced cost of subsequences in arbitrary graphs. The straightforward proof is omitted.

Lemma A. *Let α and σ be sequences of distinct vertices in an edge-weighted graph G, with α a subsequence of σ. Then (a) If $\alpha(1) = \sigma(1)$, then $\sum_{x\in\alpha} c(x,\sigma) \leq \sum_{x\in\alpha} c(x,\alpha)$; (b) $\sum_{x\in\alpha} c(x,\sigma) \leq \sum_{x\in\alpha-\{\alpha(1)\}} c(x,\alpha) + c(\alpha(1),\sigma)$, where as usual $\sigma(1)$ [resp. $\alpha(1)$] denotes the first point of σ (resp. α).*

We can now further develop our results on $L(\sigma)/w(\sigma)$, exploiting the flexibility in the choice of interval partition of the Steiner tree T in Corollary 5.3.1.

Corollary 5.3.2. *With the same hypothesis and notation as in Corollary 5.3.1 we have;*

(a) If $w_i = \frac{W}{p}$ for all i, $1 \leq i \leq p$, then

$$\frac{L(\sigma)}{w(\sigma)} \leq \frac{1}{2}\log_2(n) - \frac{1}{2}\log_2(p) + \frac{1}{W}\left(\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\sigma)\right) + \frac{3}{2}$$
$$\leq \frac{1}{2}\log_2(n) + \frac{1}{2}\log_2(p) + O(1).$$

(b) Suppose red(T) has s edges, and red(G) has r edges. Then

$$\frac{L(\sigma)}{w(\sigma)} \leq \tfrac{1}{2}\log_2(n) + \lceil \log_2(s)\rceil + \tfrac{3}{2}$$
$$\leq \tfrac{1}{2}\log_2(n) + \lceil \log_2(r)\rceil + \tfrac{5}{2}.$$

(c) Again, with r being the number of edges in red(G), suppose $n \geq r^k$. Then

$$\frac{L(\sigma)}{w(\sigma)} \leq \left(\frac{1+\frac{2}{k}}{2}\right)\cdot \log_2(n) + O(1).$$

Proof. For (a), the first inequality follows from Corollary 5.3.1 upon substituting

$$w_i = \frac{W}{p} \text{ into } \frac{1}{2W}\sum_{i=1}^{P} w_i \log_2(w_i)$$

and simplifying.

The second inequality of (a) will follow from showing that

$$\frac{1}{W}\left(\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\sigma)\right) \leq \lceil \log_2(p)\rceil.$$

Let T^* be a minimum Steiner tree in G for the set $\mathrm{Cov}(\sigma, I)$, and set $W^* = \mathrm{weight}(T^*)$. Also, let α be the subsequence of σ consisting of just the vertices of $\mathrm{Cov}(\sigma, I)$ (i.e., in the same relative order as they appear in σ). Recall that $\alpha(1)$ and $\sigma(1)$ are the first vertices of α and σ, respectively. Now, because every point of σ belongs to some interval of I, and $\alpha(1)$ is the first point in the initial σ-cover of I, it follows that $\alpha(1) = \sigma(1)$. Hence, by Lemma A we have

$$\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\sigma) \leq \sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\sigma).$$

By inequality 2.2 (from Theorem 2 in [4], we have

$$\frac{1}{W^*}\left(\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\alpha)\right) \leq \lceil \log_2 |\mathrm{Cov}(\sigma, I))|\rceil \leq \lceil \log_2(p)\rceil.$$

Nothing that $W \geq W^*$, because T is a subtree of G spanning $\mathrm{Cov}(\sigma, I)$, we have

$$\frac{1}{W}\left(\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\sigma)\right) \leq \frac{1}{W^*}\left(\sum_{x\in \mathrm{Cov}(\sigma,I)} c(x,\alpha)\right) \leq \lceil \log_2(p) \rceil.$$

For (b), apply the second inequality of (a) to the interval partition I of T whose intervals are precisely the edges of $\mathrm{red}(T)$. There are s such edges, giving the first inequality of (b).

For the second inequality of (b), we need only show that $s \leq 2r$. Now by the connectedness of T, each edge xy of $\mathrm{red}(G)$, when viewed as an interval of con-red(G) with ends x and y, may "host" up to two distinct edges of $\mathrm{red}(T)$. This is because xy may contain at most two edges, xv_1 and yv_2, where v_1 and v_2 are end points of $\mathrm{red}(T)$ lying in the interval xy, such that xv_1 and yv_2 are disjoint subintervals of xy in con-red(G). Hence, $s \leq 2r$ as desired.

Part (c) follows immediately from the second inequality of part (b) using the upper bound $r \leq n^{1/k}$, combining the logarithms, and simplifying. ∎

The tightness of the upper bounds in Corollaries 5.3.1 and 5.3.2 is explored in the Appendix. We analyze there the values of $L(\sigma)$ for the graphs $K_{1,p}$ and con-$K_{1,p}$ and certain sequences σ, showing that the upper bounds can be matched by lower bounds up to a small constant additive term.

In the concluding remarks of [4] the problem of finding the worst-case performance ratio for DGA was posed, where DGA is the dynamic greedy algorithm for solving OST described in our first section. While the lower and upper bounds (2.1) and (2.2) for this performance ratio of $1 + \frac{1}{2}\lfloor \log_2(n-1) \rfloor$ and $\lceil \log_2(n) \rceil$, respectively, were known for on-line sequences of n vertices, the authors conjectured that $\frac{1}{2}\log_2(n)$ is actually correct as an upper bound, persumably up to a constant additive term. But recall that any upper bound for $L(\sigma)/w(\sigma)$ is also an upper bound for the performance ratio of VG, and hence of DGA on sequences of length $|\sigma|$. Part (c) of Corollary 5.3.2 therefore implies that this conjecture is true when the instance graph G is taken from any collection of graphs in which r, the number of edges in $\mathrm{red}(G)$, is bounded by a constant. We may summarize this as follows.

Corollary 5.3.3. *Let K be a constant, and let Ω be any collection of graphs such that for any $G \in \Omega$ the number of edges in red(G) satisfies $e(\mathrm{red}(G)) \leq K$. Then for any $G \in \Omega$ and any sequence σ of n vertices in G we have $L(\sigma)/w(\sigma) \leq \frac{1}{2}\log_2(n) + O(1)$. Hence, the performance ratio of VG and hence of DGA on sequences of length n taken from this collection of graphs is upper bounded by $\frac{1}{2}\log_2(n) + O(1)$.*

Some natural examples of collections Ω satisfying the hypothesis of Corollary 5.3.2 are the set of all refinements of some fixed irreducible graph G (i.e., where $\mathrm{red}(G) = G$), the set of all refinements of k-regular graphs on v vertices with k and v bounded by some constant K, and generally any collection Ω of graphs whose set of reductions $\{\mathrm{red}(G): G \in \Omega\}$ contains at most a finite number of isomorphism types (or "topologies"). Thus, such a collection Ω could well be infinite (as in the preceding examples), but any $G \in \Omega$ is a refinement of one of at most a finite number of irreducible graphs which "generate" Ω by refinement.

We do not know whether the $\frac{1}{2}\log_2(n) + O(1)$ upper bound conjecture is true in general without the hypothesis of Corollary 5.3.3. But part (c) of Corollary 5.3.2 does yield an improvement on the known upper bound $\lceil \log_2(n) \rceil$ for the performance ratio of DGA applied to a graph G when n is large enough compared to r; that is, when the number of degree 2 points in G is large enough compared to r. Indeed, as k increases (in Corollary 5.3.2c), we get progressively better improvements on $\lceil \log_2(n) \rceil$, which converge to the conjectured $\frac{1}{2}\log_2(n)$ upper bound.

6. CONCLUDING REMARKS

We have examined the on-line Steiner problem, through a study of the parameter $\mathrm{Opt}(n, G)$ and the upper bound $L(\sigma)/w(\sigma)$ for the performance ratio of the vertex greedy algorithm for the OST. We computed $\mathrm{Opt}(n, P_k)$ and its continuous analogue $\mathrm{Opt}(n, \text{con-}K_2)$, both by the greedy algorithm and by a characterization of optimal sequences, the latter leading to an explicit formula in the discrete case. The result on $\mathrm{Opt}(n, \text{con-}K_2)$ was used in deriving a general upper bound on $L(\sigma)/w(\sigma)$, which in turn, yielded the corollary $L(\sigma)/w(\sigma) \leq \frac{1}{2}\log_2(n) + O(1)$, where $|\sigma| = n$, for any collection of graphs whose homeomorphic reductions have a number of edges bounded by some constant. This proves the conjectured performance bound of $\frac{1}{2}\log_2(n) + O(1)$ in [4] on DGA for all graphs, when restricted to any such collection. Our results also imply upper bounds on $L(\sigma)/w(\sigma)$ for any graph G that improve on the known $\lceil \log_2(n) \rceil$ upper bound and converge to the conjectured $\frac{1}{2}\log_2(n) + O(1)$ as the proportion of degree 2 points among all the points grows.

In the Appendix we show the tightness of the general bound by examining the stars con-$K_{1,p}$. There we also develop constructions that lower bound the maximum of $\mathrm{Opt}(n, T)/\mathrm{weight}(T)$ over all trees on n vertices, in particular finding examples for which this ratio is larger than the same ratio for the continuous path con-K_2.

Finally, we should mention the cycle C_n on n points with all edges having weight 1. Then $\mathrm{Opt}(k, C_n) = \mathrm{Opt}(k+1, P_{n+1}) - n$ and $\frac{1}{n}\mathrm{Opt}(k, \text{con-}C_n) = \mathrm{Opt}(k+1, \text{con-}K_2) - 1$, as can be seen by viewing the cycle C_n as a path of length n with end points identified.

Many issues remain to be explored; for example the following.

1. Maximize $\mathrm{Opt}(k, T)$ over all trees T having weight 1. By Theorem 2.1 each lower bound for this maximum is also a lower bound for the worst case performance ratio $C_A(k)$ for all algorithms A for the on-line Steiner problem.

2. Determine the complexity of the SSP problem when restricted to trees. Is it polynomial time solvable?
3. Determine $\mathrm{Opt}(n, \text{con-}K_{1,p})$ for all n.
4. Proving the conjectured bound $L(\sigma)/w(\sigma) \le \frac{1}{2}\log_2(n) + O(1)$ for trees naturally implies the same bound for any graph G because distances in any spanning tree of G are at least as large as they are in G. Hence, the first inequality of Corollary 5.3.2a shows that proving

$$\sum_{x \in \mathrm{Cov}(\sigma, I)} c(x, \sigma) \le W \cdot \log_2(p)$$

for any weighted tree T would imply the conjectured $\frac{1}{2}\log_2(n) + O(1)$ bound for all graphs, where p is the size of some interval partition of T with equal interval weights (call this a "balanced partition"). This in turn reduces to proving $L(\sigma) \le W \cdot \log_2(p)$ for any sequence σ satisfying $|\sigma| \le p$, where p is the size of some balanced partition of T. Thus, a natural starting place is to prove $L(\sigma) \le W \cdot \log_2(|E(T)|)$ in any tree with edges all having equal weight, where $|\sigma| \le |E(T)|$ and the balanced partition of T consists of the edges of T.

APPENDIX: CONSTRUCTIONS SHOWING TIGHTNESS OF BOUNDS

Tightness for the Bounds of Corollaries 5.3.1 and 5.3.2

To examine the tightness of the bounds in Corollary 5.3.1, we now consider the sequential sum problem for con-$K_{1,p}$, where the edges of the orignal $K_{1,p}$ are equally weighted, say with weight 1. Apart from tightness of the bounds, we also believe that determining $\mathrm{Opt}(k, \text{con-}K_{1,p})$ is the next natural step after our determination of $\mathrm{Opt}(k, \text{con-}K_2)$. We will need the following notation. Let z be the center point of con-$K_{1,p}$. Let $e_1, e_2, \ldots, e_p$ be the p end points of the branches $B_1, B_2, \ldots, B_p$ of con-$K_{1,p}$, respectively; that is, of the p paths of length 1 from the center z to the e_i. Denote by (i, d) the point of con-$K_{1,p}$ lying on the path B_i at distance $d, 0 \le d \le 1$, from the center z (see Fig. 1). Thus, for $1 \le i \le p$, $z = (i, 0)$ for all i.

We construct a greedy sequence δ_p of points in con-$K_{1,p}$, modeled on the sequence $\tau = (\tau(1), \tau(2), \ldots) = (0, 1, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}, \frac{1}{16}, \ldots)$ which was optimal for con-K_2. Begin by letting $\delta_p(1) = (1, 1) = (1, \tau(2))$, $\delta_p(2) = (2, 1) = (2, \tau(2)), \ldots, \delta_p(p) = (p, 1) = (p, \tau(2))$, and $\delta_p(p+1) = z$. Assume inductively for $s \ge 1$ that we have defined $\delta_p(r)$ for every $1 \le r \le ps + 1$. Then let $\delta_p(ps + 1 + j) = (j, \tau(s + 2))$ for $1 \le j \le p$, thereby defining $\delta_p(r)$ for every $1 \le r \le p(s + 1) + 1$. Notice that the restriction of δ_p to each branch simply reproduces the sequence τ, and that δ_p alternates evenly and identically among the p branches. For example, the values of $\delta_p(r)$ for $p + 2 \le r \le 2p + 1$ are $(1, \frac{1}{2}), (2, \frac{1}{2}), \ldots, (p, \frac{1}{2})$, and for $2p + 2 \le r \le 3p + 1$ they are $(1, \frac{1}{4}), (2, \frac{1}{4}), \ldots, (p, \frac{1}{4})$, etc.

Trivially, $\mathrm{Opt}(k, \text{con-}K_{1,p}) \ge L(k, \delta_p)$. Given that δ_p is a natural, greedy sequence, one might wonder whether this lower bound is optimal. Taking $k = 10$ and $p = 3$ we can see that it is not, as evidenced by the sequence $\sigma = \{(1, 1), (2, 1), (3, 1), (1, \frac{1}{3}), (2, \frac{1}{3}), (3, \frac{1}{3}), (1, \frac{2}{3}), (2, \frac{2}{3}), (3, \frac{2}{3}), z\}$. Then $L(\sigma) = \frac{22}{3}$, which exceeds $L(10, \delta_3) = \frac{29}{4}$. Again, see Figure 1. But while evidently $L(k, \delta_p)$ is not always optimal, the next theorem gives a slight improvement on the upper bound for $\mathrm{Opt}(k, \text{con-}K_{1,p})$ implied by Corollary 5.3.1 (taking $w(\sigma) = p$ in that corollary).

Theorem A.1. $\mathrm{Opt}(k, \text{con-}K_{1,p}) \le L(k - p + 1, \delta_p) + p - 1$ *for* $k \ge 2p$.

Proof. Let $\sigma = (x_1, x_2, \ldots, x_k)$ be a sequence of points in con-$K_{1,p}$ which realizes $\mathrm{Opt}(k, \text{con-}K_{1,p})$; that is, $L(\sigma) = \mathrm{Opt}(k, \text{con-}K_{1,p}) = \sum_{i=1}^{k} c(i, \sigma)$. Further, let $Z_i = \{x_j : x_j \in B_i\}$ be the set of points in σ lying on the ith branch of con-$K_{1,p}$ for $1 \le i \le p$. As a convention, if $x_i = z$ for some i, then by symmetry we assume that the nearest predecessor of x_i lies on branch B_1, in which case we consider x_i to be a point of B_1.

Our first step is to show that the first p points $x_1, x_2, \ldots x_p$ may be taken to be the end points $e_1, e_2, \ldots, e_p$. Now the proofs and conclusions of Lemmas 3.1 and 3.2 carry over to any tree structured metric space. Hence, we may suppose that x_1, x_2 are leaves, say $x_1 = e_1$ and $x_2 = e_2$. Suppose that $x_3 \ne e_i$, $i \ge 3$. Then x_3 must be a cut vertex. By the monotonicity condition of Lemma 3.1 we know that x_3 lies on some branch different than B_1 or B_2, say B_3. But now Lemma 3.2 forces a predecessor of x_3 in σ to lie on B_3, a contradiction. Repeating this argument we have $x_i = e_i$ for $1 \le i \le p$.

Next we define a bound $d(x_i)$ on $c(i, \sigma)$. For $1 \le i \le p$, let $d(x_i) = 0$. For $i \ge p + 1$ let $d(x_i) = \min\{\mathrm{dist}(x_i, x_j) : j < i, x_j \text{ on the same branch of con-}K_{1,p} \text{ as } x_i\}$. Clearly, $c(i, \sigma) \le d(x_i)$ for $i \ge p + 1$.

First, we relate $L(\sigma)$ on con-$K_{1,p}$ to $L(\tau)$ on con-K_2. Because $c(i, \sigma) = 2$ for $2 \le i \le p$, we have

$$L(\sigma) \le 2(p - 1) + \sum_{i=1}^{k} d(x_i). \tag{1}$$

Now bound the sum in (1) by

$$\sum_{i=1}^{k} d(x_i) = \sum_{j=1}^{p} \sum_{x_i \in B_j} d(x_i) \le \sum_{j=1}^{p} L(|Z_j|, \tau).$$

The terms of $L(\tau)$ are nonicreasing for $i \ge 2$, so the last sum is maximized when the $|Z_j|$ are as nearly equal as possible subject to summing to k. Accordingly, for each $r \ge 2p$, define $S_p(r) = \sum_{j=1}^{p} L(t_j(r), \tau)$, where $t_j(r)$ are the unique integers (after possible reordering) satisfying $\sum_{j=1}^{p} t_j(r) = r$ and $|t_i(r) - t_j(r)| \le 1$. Using (1) we then get

$$L(\sigma) \le \sum_{j=1}^{p} L(t_j(k), \tau) + 2(p - 1) = S_p(k) + 2(p - 1). \tag{2}$$

It remains to compare $S_p(k) = \sum_{j=1}^{p} L(t_j(k), \tau)$ to $L(k-2, \delta_p)$. Now $S_p(2p) = p, S_p(2p+1) = p + \frac{1}{2}, L(p+1, \delta_p) = 2p-1, L(p+2, \delta_p) = 2p - \frac{1}{2}$, and in general $S_p(2p+r) - S_p(2p) = L(p+1+r, \delta_p) - L(p+1, \delta_p)$ for $r > 0$. It follows that $S_p(2p+r) = L(p+1+r, \delta_p) - p + 1$, or changing variables $S_p(k) = L(k-p+1, \delta_p) - p + 1$. On combining this expression for $S_p(k)$ with the bound (2) we get

$$L(\sigma) \le L(k-p+1, \delta_p) - p + 1 + 2(p-1)$$
$$= L(k-p+1, \delta_p) + p - 1.$$
∎

Corollary A.1.1. *For $k \ge 2p+1$ we have*

$$2p - 1 + \frac{p}{2}\left\lfloor \log_2\left(\frac{k-1}{p}\right)\right\rfloor + N(k) \le \mathrm{Opt}(k, \text{con-}K_{1,p})$$
$$\le 3p - 2 + \frac{p}{2}\left\lfloor \log_2\left(\frac{k-p}{p}\right)\right\rfloor + N(k-p+1),$$

where

$$N(r) = \frac{r - 1 - p.2^{L+1}}{2^{L+2}}$$

and

$$L = \left\lfloor \log_2\left(\frac{r-1}{p}\right)\right\rfloor - 1$$

for any integer $r \ge 2p$.

Proof. We begin with a calculation of $L(k, \delta_p)$, whose first p nonzero terms are $c(2, \delta_p) = 2, c(3, \delta_p) = 2, \ldots, c(p, \delta_p) = 2, c(p+1, \delta_p) = 1$. The next p terms are all $\frac{1}{2}$, the next $2p$ terms are $\frac{1}{4}$, the next $4p$ terms are $\frac{1}{8}$, and so on. Starting from $c(p+2, \delta_p)$, the terms are arranged in successive blocks of length $p \cdot 2^i$, $i \ge 0$, with constant term value $\frac{1}{2^{i+1}}$. If we let $p \cdot 2^L$ be the length of the longest complete block of identical terms contributing to $L(k, \delta_p)$, then because there are $k-1$ nonzero terms in $L(k, \delta_p)$ we have that $p + \sum_{i=0}^{L} p \cdot 2^i \le k - 1$. By maximality of L we have $p + \sum_{i=0}^{L+1} p \cdot 2^i > k - 1$. It follows that $L = \lfloor \log_2(\frac{k-1}{p})\rfloor - 1$. Now the contribution to $L(k, \delta_p)$ of a block of length $p \cdot 2^i$ is $\frac{p \cdot 2^i}{2^{i+1}} = \frac{p}{2}$. Hence, adding together the first $p+1$ terms, plus the contribution of the $L+1$ complete blocks, plus the terms (of constant value $\frac{1}{2^{L+2}}$) remaining beyond the last complete block, we obtain

$$L(k, \delta_p) = 2p - 1 + \frac{p}{2}\left\lfloor \log_2\left(\frac{k-1}{p}\right)\right\rfloor + N(k). \quad (3)$$

Combining this with the previous theorem and the trivial lower bound $\mathrm{Opt}(k, \text{con-}K_{1,p}) \ge L(k, \delta_p)$, the corollary follows. ∎

The preceding bounds for $\mathrm{Opt}(k, \text{con-}K_{1,p})$ allow us to see how strong are the bounds in Corollary 5.3.2. We apply this corollary to the sequence δ_p, letting both G and T be $K_{1,p}$, with interval partition consisting of the edges of $K_{1,p}$. In the upper bound, the initial cover sum satisfies $\sum_{x \in \mathrm{Cov}(\delta_p, I)} c(x, \sigma) = 2(p-1)$, because $\mathrm{Cov}(\delta_p, I)$ consists of the endpoints of $K_{1,p}$. Hence, the first inequality of Corollary 5.3.2a (with $W = p$, $w_i = 1$ for all i, and $k = n$) gives the bound $\frac{L(k,\delta_p)}{p} \le \frac{1}{2}\log_2\left(\frac{k}{p}\right) + \frac{7}{2} - \frac{2}{p}$. To get a nearly matching lower bound, consider (3). The term $N(k)$ has its integer local maxima at integers $k = 2^s p$, with s an integer, where its values are $N(k) = \frac{p}{2} - 2^{-s}$. With these values of k we get $L(k, \delta_p) = \frac{p}{2}\lfloor \log_2(\frac{k-1}{p})\rfloor + \frac{5p}{2} - 1 - 2^{-s}$. Allowing for the rounding down in $\lfloor \log_2(\frac{k-1}{p})\rfloor$ by at most 1, we find that (3) implies $\frac{L(k,\delta_p)}{p} \ge \frac{1}{2}\log_2(\frac{k-1}{p}) + \frac{3}{2} + o(1)$ as k grows at these special values. This matches the first upper bound of Corollary 5.3.2a to within an additive term of $2 + o(1)$. On the other hand, using integers k of the form $k = 2^s p + p$ we get $N(k-p+1) = 0$, and then the upper bound on $\frac{L(k,\delta_p)}{p}$ implied by Corollary A.1.1 improves on the first upper bound of Corollary 5.3.2a) by an additive term very close to $\frac{1}{2} + \frac{1}{p}$ as k grows at these values. An even tighter match between lower and upper bounds will come after our next theorem.

We have already seen that $L(k, \delta_p)$ fails to be optimal for the graph con-$K_{1,p}$ when $k = 10$. We will now see that this failure occurs infintely often. Specializing to the case $p = 3$ to simplify the calculations, where we let $\delta = \delta_3$, we will construct a sequence $\{\gamma(i)\}, i \ge 1$, of vertices in con-$K_{1,3}$ for which $L(k, \gamma) > L(k, \delta)$ for infinitely many k. On the other hand, it is not true that $L(k, \gamma)$ is optimal for all k sufficiently large because we will show that the reverse inequality holds for infinitely many k.

Let $\{s(i)\}, i \ge 1$, be a sequence of vertices in con-$K_{1,3}$. For a pair of distinct vertices $s(a) = (x, j)$ and $s(b) = (y, j)$, $x < y$, in the sequence lying on the same branch B_j (where now z is understood to lie on all three brances *simultaneously*), define the *segment* $(s(a), s(b))$ to be the set of points $\{(z, j) : x < z < y\}$. Now for an integer $k \ge \max\{a, b\}$, we say that $s(a)$ and $s(b)$ are *k-consecutive* if there exists no vertex $s(m)$, $m \le k$, lying in the segment $(s(a), s(b))$.

The sequence $\{\gamma(i)\}$, $1 \le i \le n$, (where n is a free parameter) is constructed by the following procedure.

Initialization:

1. For $1 \le i \le 3$; $\gamma(i) = (1, i)$, and $\gamma(3+i) = (\frac{3}{1}, i)$.
2. Let $\gamma(7) = z$.

For $k \ge 8$ do

1. Among pairs $\gamma(r) = (a, i)$ and $\gamma(s) = (b, i)$ of $(k-1)$-consecutive points with $r, s < k$, choose a pair maximizing $\mathrm{dist}(\gamma(i), \gamma(j))$.
2. Define $\gamma(k) = (\frac{a+b}{2}, i)$ (that is, we take $\gamma(k)$ to be the midpoint of the segment $(\gamma(r), \gamma(s))$).
3. If $k = n$, stop.
4. $k \leftarrow k + 1$

od.

Notice that the sequence $\{\gamma(k)\}$ is "greedy" for $k \geq 8$, because $\gamma(k)$ for such k is chosen so that $c(k, \gamma)$ is the maximum possible given the previously constructed points $\gamma(i)$, $1 \leq i \leq k-1$. In fact, the only nongreedy term in this sense is $\gamma(4) = (\frac{1}{3}, 1)$ (giving $c(4, \gamma) = \frac{2}{3}$), where the greedy choice would have been $\gamma(4) = z$ (giving $c(4, \gamma) = 1$). By contrast, the sequence $\{\delta(i)\}$ is greedy for all terms (i.e., for $i \geq 2$). Our next theorem shows that the values of $L(k, \delta)$ and $L(k, \gamma)$ alternate exceeding each other infinitely often as k grows.

Theorem A.2. *Let $\{m_i\}$ and $\{n_j\}, i, j \geq 1$, be the sequences of integers defined by $m_i = 10 + 9(2^{i+1} - 1)$, and $n_j = 4 + 3(2^{j+1} - 1)$. Then (a) $L(n_j, \delta) > L(n_j, \gamma)$ for $j > 1$, and (b) $L(m_i, \gamma) > L(m_i, \delta)$ for $i > 1$.*

Proof. The sequences $\{c(k, \delta)\}$ and $\{c(k, \gamma)\}, k \geq 1$, of terms contributing to $L(\delta)$ and $L(\gamma)$ can be partitioned into successively longer blocks of constant terms. We will show that the indices k of last terms $c(k, \delta)$ (resp. $c(k, \gamma)$) of these blocks form the sequence $\{n_j\}$ (resp. $\{m_i\}$). Evaluating $L(k, \delta)$ and $L(k, \gamma)$ at these indices k yields claims (a) and (b).

The first four terms of $\{c(i, \delta)\}$ are 0,2,2,1. After that we have three terms $c(i, \delta), 5 \leq i \leq 7$, of constant value $\frac{1}{2}$, followed by 6 terms $c(i, \delta), 8 \leq i \leq 13$, of constant value $\frac{1}{4}$, and so on. In general, past the first four terms, we find a block of $3 \cdot 2^i$ many terms $c(i, \delta), 1+3 \cdot 2^j < i \leq 1+3 \cdot 2^{j+1}, j \geq 0$, of constant value $\frac{1}{2^{j+1}}$. Let us call such a block of terms the jth *δ-block*. Then the number of terms of $L(\delta)$ up to (and including) the jth δ-block is $n_j = 4+3 \cdot \sum_{i=0}^{j} 2^i = 1 + 3 \cdot 2^{j+1}$. The jth δ-block contributes $\frac{3 \cdot 2^j}{2^{j+1}} = \frac{3}{2}$ to $L(n_j, \delta)$, so after adding in the first four terms we obtain $L(n_j, \delta) = 5 + \frac{3}{2}(j+1)$.

The initial 10 terms of $\{c(i, \delta)\}$ are $0, 2, 2, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}$. Past these we can similarly identify the block of $9 \cdot 2^i$ many terms $c(k, \gamma), 1 + 9 \cdot 2^i < k \leq 1 + 9 \cdot 2^{i+1}, i \geq 0$, all a having constant value of $\frac{1}{3 \cdot 2^{i+1}}$, as the ith γ-block. Then the number of terms of $L(\gamma)$ up to (and including) the ith γ-block is $m_i = 10+9 \cdot \sum_{k=0}^{i} 2^k = 1 + 9 \cdot 2^{i+1}$. As with δ, the ith γ-block contributes $\frac{3}{2}$ to $L(m_i, \gamma)$, so after adding in the first 10 terms we obtain $L(m_i, \gamma) = \frac{22}{3} + \frac{3}{2}(i+1)$.

Let us now calculate $L(m_i, \delta)$ and $L(n_j, \gamma)$ to show the claimed inequalities.

For $L(m_i, \delta)$, we ask what is the maximum j for which the set of summands comprising $L(m_i, \delta)$ contains the jth δ-block; that is, the maximum j satisfying $1+3 \cdot 2^{j+1} \leq 1+9 \cdot 2^{i+1}$. Simple arithmetic shows this to be $j = i+1$. The summands if $L(m_i, \delta)$ beyond this $(i+1)$'st δ-block (of which there must be $m_i - n_{i+1} = 1+9 \cdot 2^{i+1} - 1 - 3 \cdot 2^{i+2} = 3 \cdot 2^{i+1}$ many) belong to the $(i+2)$'nd δ-block, and hence, all have constant value $\frac{1}{2^{i+3}}$. Hence, we get

$$
\begin{aligned}
L(m_i, \delta) = L(n_{i+1}, \delta) + \frac{1}{2^{i+3}} 3.2^{i+1} &= \frac{23}{4} + \frac{3}{2}(i+2) \\
&< \frac{22}{3} + \frac{3}{2}(i+1) = L(m_i, \gamma),
\end{aligned}
$$

proving (b).

To calculate $L(n_j, \gamma)$, we similarly ask for the maximum i for which the set of summands comprising $L(n_j, \gamma)$ contains the ith γ-block; that is, the maximum i satisfying $1 + 9 \cdot 2^{i+1} \leq 1 + 3 \cdot 2^{j+1}$. This maximum is $i = j - 2$. Here the number of summands of $L(n_j, \gamma)$ past the $(j-2)$'nd γ-block is $n_j - m_{j-2} = 3 \cdot 2^{j-1}$, all belonging to the $(j-1)$'st γ-block and thus all having constant value $\frac{1}{3 \cdot 2^j}$. It follows that

$$
\begin{aligned}
L(n_j, \gamma) = L(m_{j-2}, \gamma) + \frac{3 \cdot 2^{j-1}}{3 \cdot 2^j} &= \frac{22}{3} + \frac{3}{2}(j-1) + \frac{1}{2} \\
&< 5 + \frac{3}{2}(j+1) = L(n_j, \delta),
\end{aligned}
$$

proving (a). ∎

Working with con-$K_{1,3}$ (in effect, any refinement of $K_{1,3}$) as our underlying graph, we note that $\frac{L(m_i, \gamma)}{3}$ is within the additive term of roughly 1.18 of the first inequality upper bound of Corollary 5.3.2a for i sufficiently large.

Constructing Trees T *with [Opt(n,*T*)/weight(*T*)] Achieving the (1/2* $\log_2(n)$*) Upper Bound*

The conjectured bound $\frac{L(\sigma)}{w(\sigma)} \leq \frac{1}{2}\log_2(|\sigma|) + O(1)$ for all sequences σ of vertices of a graph G is achieved by $G = P_r$ for each r (Theorems 4.2 and 4.5) and by its continuous analog con-K_2 (Theorem 5.2), recalling that $\text{Opt}(r, \text{con-}K_2) = \frac{1}{2}\log_2(r) + O(1)$. Taking a hint from the conjecture, one might ask whether there exist non-path weighted trees T and sequences σ on $V(T)$, with $|V(T)|$ and $|\sigma|$ arbitrarily large, such that

$$\frac{L(\sigma)}{w(\sigma)} > \text{Opt}(|\sigma|, \text{con-}K_2). \qquad (*)$$

If the conjecture is false, then there must exist infinitely many trees T and corresponding sequences σ for which $\frac{L(\sigma)}{w(\sigma)} > \text{Opt}(|\sigma|, \text{con-}K_2) + h(|\sigma|)$ where $h(n)$ grows without bound as n grows. If the conjecture is true, then the left side of (*) (for any T and σ) can exceed the right by at most an absolute constant. In this appendix we construct infinitely many non-path trees T and corresponding sequences σ satisfying (*), but in all these examples the left side exceeds the right by at most a constant (in fact, a small one) as $|\sigma|$ grows. We were unable to find a sequence of examples where the left exceeds the right by an unbounded amount as $|\sigma|$ grows. Thus, the path topology does not maximize the ratio $\frac{L(\sigma)}{w(\sigma)}$, but the conjecture may still be true.

In our examples we take σ to include all vertices in T, so then $\frac{L(\sigma)}{w(\sigma)} = \frac{L(\sigma)}{\text{weight}(T)}$. So with this in mind we define the *normalized cost* $L'(\sigma)$ of a sequence $\sigma = (x_1, x_2, \ldots, x_k)$ of vertices of a tree T with $V(T) = k$ by $L'(\sigma) = \frac{L(\sigma)}{\text{weight}(T)}$.

For future reference recall that if we let $f(n) = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \cdots + \frac{1}{2^{-1-\lfloor \log n \rfloor}}$, for $n > 0$, with $f(0) = 0$, then by Theorem 5.2 we have $\text{Opt}(n, \text{con-}K_2) = 1 + f(n-2)$. Recall that if $M \geq 0$ is an integer with $2^k - 1 \leq M < 2^{k+1} - 1$, then $f(M) = \frac{k}{2} + \frac{M - 2^k + 1}{2^{k+1}}$.

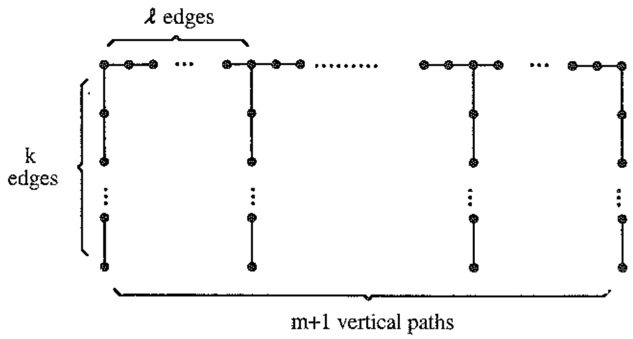


FIG. 6. $CT(k,r,m)$.

The Cascade Tree. Let k, r, and m be positive integers. Let P be a path on the $rm+1$ vertices $\{v_0, v_1, \ldots, v_{rm}\}$, taken in consecutive order from one end to the other. For $0 \le i \le m$, let Q_i be a path of length k on the $k+1$ vertices $\{w_{i,0}, w_{i,1}, \ldots, w_{i,k}\}$.

The *Cascade Tree*, denoted $CT(k,r,m)$, is constructed as follows. For each $i, 0 \le i \le m$, attach Q_i onto P by identifying the vertex $w_{i,0}$ of Q_i with the vertex v_{ir} of P. Here, the subscript ir denotes the product of i with r.) Thus, $CT(k,r,m)$ has the $m+1$ leaves $w_{0,k}, w_{1,k}, \ldots, w_{m,k}$, has $m-1$ vertices of degree 3 (namely $w_{1,0}, w_{2,0}, \ldots, w_{m-1,0}$) and all the remaining vertices have degree 2. There are $m(r+k)+(k+1)$ vertices altogether. See Figure 6 illustrating $CT(k,r,m)$.

We will be interested in the Cascade Trees with $(k,r,m) = (2^s, 2^{s+1}, 2^p)$, for suitably chosen nonnegative integers s and p. Denote these trees by $G(s,p)$, having $n = 3 \cdot 2^{s+p} + 2^s + 1$ vertices.

Choosing a sequence σ of all the vertices of $G(s,p)$ is equivalent to labeling the vertices with the integers $1, 2, 3, \ldots, n$, and for ease of explanation, we choose to adopt this viewpoint. Label $w_{0,k}$ with "1" and $w_{m,k}$ with "2" (in other words, let $\sigma(1) = w_{0,k}$, and $\sigma(2) = w_{m,k}$). We next label the remaining $m-1$ leaves as we would by applying the greedy algorithm to the ordered set $\{w_{1,k}, w_{2,k}, \ldots, w_{m-1,k}\}$. Now note that $G(s,p)$ contains m paths T_i, for $1 \le i \le m$, (overlapping, i.e., sharing edges) with end points $w_{i-1,k}$ and $w_{i,k}$. Each T_i has length 2^{s+2}, a power of 2, and its end points have been labeled.

The next m labels we use for the midpoints of $T_1, T_2, \ldots, T_m$ in succession. These new labeled vertices further subdivide each T_i into two subpaths ($2m$ in all, each half as long as a T_i), with labeled end points. All but two of the midpoints of these new subpaths are the vertices of degree 3 of $G(s,p)$, and we use the next $m+1$ labels for these midpoints. Now all the labeled vertices subdivide $G(s,p)$ into, and are the end points of, $3m+1$ nonoverlapping subpaths with labeled end points, and we label the remaining vertices of $G(s,p)$ by using the greedy algorithm on the interiors of these subpaths. See Figure 7 for the labeling of $G(1,1)$.

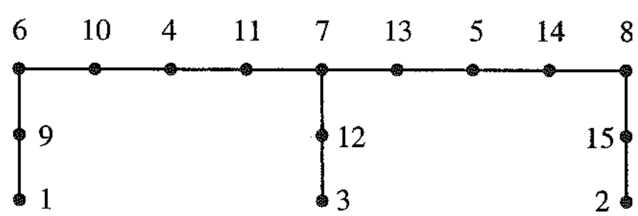


FIG. 7. Labeling of $G(1,1)$.

TABLE 1. Values of h(0,p) for $0 \le p \le 3$.

p	0	1	2	3
n	5	8	14	26
$h(0,p)$	2	$\frac{17}{7} \approx 2.429$	$\frac{37}{13} \approx 2.846$	$\frac{81}{25} \approx 3.24$

We now determine the cost of this labeling σ. For fixed s, let a_p denote the interior cost, $I(\sigma)$. Note that $G(s,0)$ is a path of length 2^{s+2}, so by Lemma 4.4, $a_0 = (s+2)2^{s+1}$. For $p > 0$, note that $c(3,\sigma) = 2 \cdot 2^s + 2^{s+1} \cdot 2^p/2 = 2^{s+1} + 2^{s+p}$. Hence,

$$a_p = 2a_{p-1} + 2^{s+1} + 2^{s+p} - 2^s - s2^{s-1},$$

where the last two terms are there to avoid double-counting the interior cost of all the vertices on the subpath of length 2^s from $\sigma(3)$ to the nearest vertex of degree 3. Solving this recurrence, we obtain

$$a_p = 2^{s-1}[(3s+10+2^p) \cdot 2P + (s-2)], \quad \text{for } p \ge 0.$$

Let $h(s,p) = L'(\sigma)$, the normalized cost of σ. Then $(n-1)h(s,p) = 2^{s+1} + 2^{s+p+1} + a_p$, and we get

$$h(s,p) = \frac{2+s+(3s+14+2p)2^p}{2(3 \cdot 2^p + 1)} = \frac{s}{2} + h(0,p).$$

Some values of $h(0,p)$ are tabulated below. For convenience, the number of vertices n is also given (Table 1).

On the other hand, with $n = 3 \cdot 2^{s+p} + 2^s + 1$ let $f(s,p) = \mathrm{Opt}(n, \text{con-}K_2)$. Recall also the function $f(n)$ satisfying $\mathrm{Opt}(n, \text{con-}K_2) = 1 + f(n-2)$. Note that $n-2 = (3 \cdot 2^p + 1) \cdot 2^s - 1$, so because $2^{p+2} = 4 \cdot 2^p > 3 \cdot 2^p + 1 > 2^{p+1}$, we have

$$2^{s+p+1} - 1 < n - 2 < 2^{s+p+2} - 1.$$

Hence, $f(s,p) = 1 + f(n-2) = 1 + \frac{s+p+1}{2} + \frac{n-1-2^{s+p+1}}{2^{s+p+2}} = \frac{s}{2} + f(0,p)$.

Some values of $f(0,p)$ are tabulated below. (Table 2).

Evidence for the following can be seen from the tables, and is easy to show.

TABLE 2. Values of f(0,p) for $0 \le p \le 3$.

p	0	1	2	3
n	5	8	14	26
$f(0,p)$	2	$\frac{19}{8} = 2.375$	$\frac{45}{16} \approx 2.813$	$\frac{105}{32} \approx 3.281$

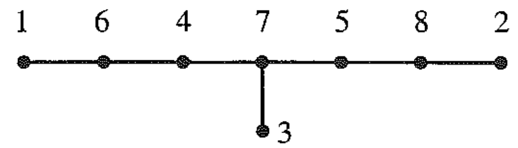


FIG. 8. A tree T on eight points for which $\mathrm{Opt}(8,T) > \mathrm{Opt}(8,P_8)$.

Theorem A.3. *For each $s \geq 0, h(s,p) > f(s,p)$ if and only if $p = 1$ or 2. Furthermore, the difference $h(s,p) - f(s,p)$ is approximately* 0.05357 *for $p = 1$, and approximately 0.03365 for $p = 2$.* ∎

As a consequence, we have the following.

Corollary A.3.1. *An example exists [namely, $G(0,1)$ in Fig. 8] for which $h(s,p) = (1+\varepsilon)f(s,p)$, where $\varepsilon \geq$* 0.02255. ∎

More generally, consider the Cascade Trees with $(k,r,m) = (2^s, 2^{s+1}(2^t - 1), 2^p)$, for suitably chosen non-negative integers s and p and $t \geq 1$. Denote these trees by $G(s,t,p)$, having $n = (2^{t+1} - 1) \cdot 2^{s+p} + 2^s + 1$ vertices. Note that $G(s,1,p) = G(s,p)$.

We label $G(s,t,p)$, obtaining the sequence σ, as for $G(s,p)$ above. Again, for fixed s and t, let a_p denote the interior cost $I(\sigma)$. Note that $G(s,t,0)$ is a path of length 2^{s+t+1}, so by Lemma 4.4, $a_0 = (s+t+1)2^{s+t}$. Here, we have, for $p > 0$,

$$a_p = 2a_{p-1} + 2^{s+1} + 2^{s+p}(2^t - 1) - 2^s - s2^{s-1}.$$

Solving gives $a_p = 2^{s-1}[[(2s+2t+2+2p) \cdot 2^t - (s+2p-2)] \cdot 2^p + (s-2)]$, for $p \geq 0$.

Let $g(s,t,p) = L'(\sigma)$. Then $(n-1)g(s,t,p) = 2^{s+1} + 2^{s+p+1}(2^t - 1) + a_p$.

So,

$$\begin{aligned} g(s,t,p) &= \frac{s}{2} + g(0,t,p) \\ &= 1 + \frac{s}{2} + \frac{p}{2} + \frac{t}{2} + \frac{(t-p)2^p - (t+p) + 2^{p+t+1}}{2(2^{p+t+1} - 2^p + 1)}. \end{aligned}$$

Still with $n = (2^{t+1} - 1) \cdot 2^{s+p} + 2^s + 1$, let $f(s,t,p) = \mathrm{Opt}(n, \text{con-}K_2)$. Note that $n - 2 = 2^{s+p}(2^{t+1} - 1) + 2^s - 1$, from which we get

$$2^{s+p+t} - 1 < n - 2 < 2^{s+p+t+1} - 1.$$

Hence, $f(s,t,p) = 1 + f(n-2) = \frac{s}{2} + f(0,t,p) = 1 + \frac{s}{2} + \frac{p}{2} + \frac{t}{2} + \frac{2^t - 1}{2^{t+1}} + \frac{1}{2^{p+t+1}}$.

For which t, and p is $g(s,t,p) > f(s,t,p)$? Some values of $g(s,t,p) - f(s,t,p)$ are tabulated below. Note that, when $p = 0$, $g(s,t,p) = f(s,t,p)$. (Table 3).

The graph $G(0,2,2)$, achieving the maximum difference $g(0,2,2) - f(0,2,2)$ of 0.07651 among these examples, has 30 vertices. Evidence for the following can be seen from Table 3.

TABLE 3. Values of $g(s,t,p) - f(s,t,p)$ for $1 \leq p \leq 7$, $1 \leq t \leq 7$; arbitrary s: Only positive values are given, rounded to five decimal places.

$t\downarrow \backslash p\rightarrow$	1	2	3	4	5	6	7
1	0.05357	0.03365	—	—	—	—	—
2	0.06250	0.07651	0.05674	0.01542	—	—	—
3	0.04738	0.06327	0.05882	0.04199	0.01793	—	—
4	0.03150	0.04344	0.04341	0.03634	0.02524	0.01187	—
5	0.01962	0.02753	0.02852	0.02555	0.02034	0.01389	0.00676
6	0.01175	0.01666	0.01765	0.01642	0.01396	0.01083	0.00732
7	0.00685	0.00979	0.01052	0.01003	0.00887	0.00734	0.00561

Theorem A.4. *Except for $p = 3$ and $t = 1$, if $t \geq p - 2$, then $g(s,t,p) > f(s,t,p)$, for all s.*

The proof of Theorem A.4 follows as a corollary to the following.

Theorem A.5. *Let $p \geq 1$ be fixed, and let $K = \frac{2^p+1}{2^p-1}$. Then $g(s,t,p) > f(s,t,p)$ if either of the following conditions hold.*

(i) $t \geq Kp - 5/2$, or
(ii) $p = 4$ and $t = 2$.

Proof. It is elementary to show that $g(s,t,p) > f(s,t,p) \Leftrightarrow kp - t - 1 < 2 - 2^{-t} + 2^{-p-t}$. Hence, if $t \geq Kp - 5/2$, then $kp - t - 1 \leq 3/2$. On the other hand, $t \geq 1$, so that $2 - 2^{-t} + 2^{-p-t} > 2 - 2^{-t} \geq 3/2$, and we are done. If $p = 4$ and $t = 2$, then it is easy to check that $Kp - t - 1 < 2 - 2^{-t} + 2^{-p-t}$. ∎

Proof of Theorem A.4. It is easy to see that $Kp - p \leq 1/2$ if $p \geq 5$. Hence, if $p \geq 5$, $t \geq p - 2 \Rightarrow t \geq Kp - 5/2$, whence the result follows from Theorem A.5(i).

Similarly by Theorem A.5(i), if $p = 1$, then $K = 3$, so $g(s,t,p) > f(s,t,p)$ for all $t \geq 1$; if $p = 2$, then $K = 5/3$, so $g(s,t,p) > f(s,t,p)$ for all $t \geq 1$; if $p = 3$, then $K = 9/7$, so $g(s,t,p) > f(s,t,p)$ for all $t \geq 2$; and finally, if $p = 4$, then $K = 17/15$, so $g(s,t,p) > f(s,t,p)$ for all $t \geq 2$, using Theorem A.5(ii). This proves the Theorem A.4. ∎

Corollary A.5.1. *An example exists for which $g(s,t,p) = (1+\varepsilon)f(s,t,p)$, where $\varepsilon \geq 0.02255$. This example is $G(0,1,1) = G(0,1)$, as obtained previously.*

The Waterfall Trees. One way to view the Cascade Tree is as a path, with copies of some other path "attached" at regular intervals, from end to end, on the original path. So one way to generalize these is to take a path and attach other (not necessarily isomorphic) paths at intervals (not necessarily regular) on the original. This we do now, but with the attached paths having certain well-chosen lengths, and the intervals chosen carefully.

We will describe the construction inductively. The idea behind it is to end up with the following picture. Take a path of a certain length, attach to it paths of lengths that are powers of two (not necessarily the same lengths), such that the

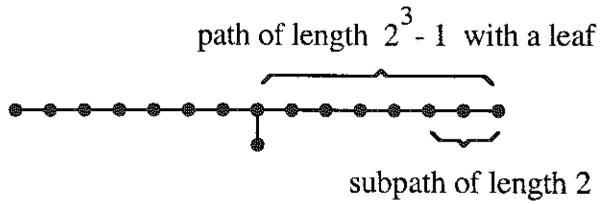


FIG. 9. $W(3{,}0)$.

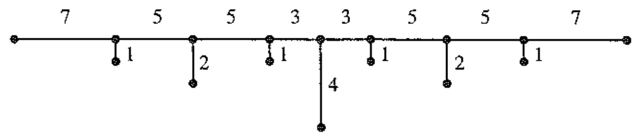


FIG. 11. $W(3{,}2)$.

shortest distances between pairs of leaves is a constant power of two, 2^k, say, for suitable k. The length of the original path is determined by the lengths of the attached paths and by k.

Inductively, we define the *Waterfall Tree*, $W(k,s)$, $s \geq 0$, $k \geq s+1$, having two disjoint induced subpaths of length $2^k - 1$, each containing a leaf of $W(k,s)$, as follows. For each $k \geq 1$, $W(k,0)$ is obtained by taking two copies of a path of length 2^k, $\{v_0, v_1, \ldots, v_{2^k}\}$ and $\{w_0, w_1, \ldots, w_{2^k}\}$, say, and identifying the vertices and edges of the initial segments (paths) of lengths $1 = 2^0$, namely identify v_0 with w_0, v_1 with w_1, and the edge (v_0, v_1) with (w_0, w_1). Hence, $W(k,0)$ has $2^{k+1} - 1$ edges. Note that $W(k,0)$ has two edge-disjoint induced subpaths of length $2^k - 1$, each containing a leaf.

For arbitrary $W(k,s)$, with $s \geq 0, k \geq s+1$, we start with two copies of $W(k, s-1)$. Each copy has an induced subpath of length $2^k - 1$, containing a leaf, and so, because $k \geq s+1$, within each of these subpaths there is an induced subpath of length 2^s containing a leaf. Call one of these subpaths $\{v_0, v_1, \ldots, v_{2^s}\}$ on one copy of $W(k, s-1)$ and $\{w_0, w_1, \ldots, w_{2^s}\}$ on the other copy, where v_0 and w_0 are the leaves. We form $W(k,s)$ by identifying the vertices and edges of these subpaths, namely by identifying v_0 with w_0, v_1 with $w_1, \ldots, v_{2^s}$ with w_{2^s}, and identifying the edges joining successive pairs of these. Note that the number of edges of $W(k,s)$ is 2^s less than twice the number of edges on $W(k, s-1)$. See the Figures 9, 10, and 11 for diagrams of $W(3{,}0)$, $W(3{,}1)$, and $W(3{,}2)$, which illustrate this construction for $k=3$, and $s = 0, 1$, and 2. (The points of degree 2 in these examples are not included in Figures 9 and 10 for simplicity.).

We remark that $W(k,0)$ is isomorphic to the Cascade Tree $G(0, k-1, 1)$.

From the construction, it is easy to see that $W(k,s)$ has $2^{s+1} + 1$ leaves, $t_0, t_1, \ldots, t_{2^{s+1}}$, say, which can be ordered so that $d(t_{i-1}, t_i) = 2^k$, for $1 \leq i \leq s+1$. Further, one of the two disjoint induced subpaths of length $2^k - 1$ contains t_0 and the other contains $t_{2^{s+1}}$. Solving a simple recurrence shows that $W(k,s)$ has $2^{s+k+1} - (s+1)2^s$ edges (and so $n = n(k,s) = 2^{s+k+1} - (s+1)2^s + 1$ vertices).

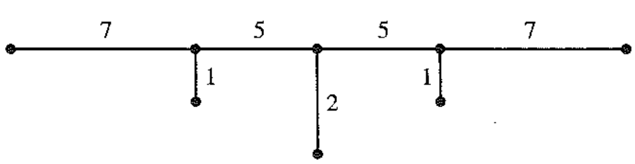


FIG. 10. $W(3{,}1)$.

We now describe a labeling of all the vertices of $W(k,s)$ to obtain a sequence σ. We first label the leaves as we would the ordered set $\{t_0, t_1, \ldots, t_{2^{s+1}}\}$ using the greedy algorithm. We then proceed as we did for the Cascade Trees above. We label the (noninduced) subpaths joining the leaves t_{i-1} and t_i, for $1 \leq i \leq s$, one vertex at a time, as we would using the greedy algorithm, at each stage adding one label, for every i, before proceeding to the next label for any i. Figure 12 shows the labeling of $W(3{,}2)$.

Let $cw(k,s) = L'(\sigma)$. Then $cw(k,s)$ can be determined as for the Cascade Trees by solving a suitable recurrence relation. The details are left to the reader. We get

$$cw(k,s) = 1 + \frac{s}{2} + \frac{(k+1)2^k - (s+8)(s+1)/4}{2^{k+1} - (s+1)}.$$

Let $fw(k,s) = \mathrm{Opt}(n(k,s), \text{con-}K_2)$. Then, because $2^{s+k} - 1 < n - 2 < 2^{s+k+1} - 1$, we have

$$fw(k,s) = 1 + \frac{s}{2} + \frac{(k+1)2^k - (s+1)}{2^{k+1}}.$$

Some values of $cw(k,s) - fw(k,s)$ are tabulated below. (Table 4).

The graph $W(4,2)$, achieving the maximal difference $cw(4,2) - fw(4,2)$ of 0.09375 among these examples, has 117 vertices. Evidence for the following can be seen from Table 45 the proof of which is left to the reader.

Theorem A.6. *Except for $s = 0, k = 1$, and $s = 1, k = 2$, we have $cw(k,s) > fw(k,s)$.*

Corollary A.6.1. *An example exists [namely, $W(3,1)$ with 29 vertices] for which $cw(k,s) = (1+\varepsilon)fw(k,s)$, where $\varepsilon \geq$* 0.02645.

There are clearly other directions in which we can generalize the above. In the inductive step of the construction

TABLE 4. Values of $cw(k,s) - fw(k,s)$ for $1 \leq k \leq 7$, $0 \leq s \leq 6$, and $k \geq s+1$: Only nonnegative values are given, rounded to five decimal places.

$k\downarrow \backslash s \rightarrow$	0	1	2	3	4	5	6
1	—						
2	0.05357	0					
3	0.06250	0.08929	0.07212				
4	0.04738	0.07917	0.09375	0.08929			
5	0.03150	0.05544	0.07147	0.07917	0.07813		
6	0.01962	0.03547	0.04744	0.05544	0.05939	0.05917	
7	0.01175	0.02159	0.02951	0.03547	0.03945	0.04144	0.04140

5/12/05

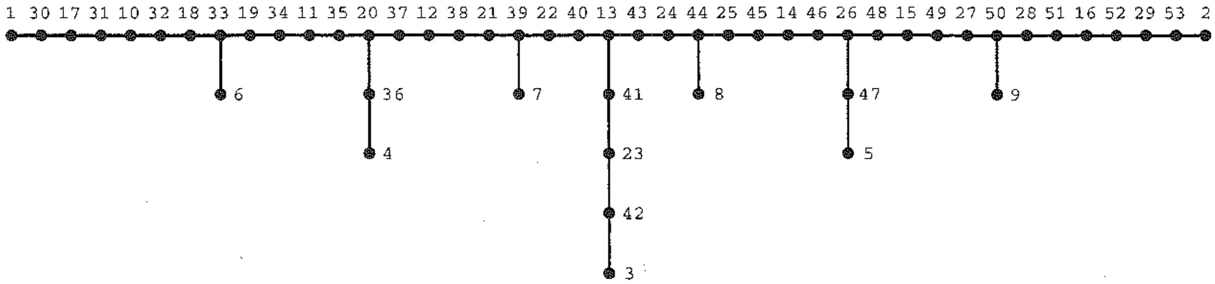


FIG. 12. Labeling of $W(3,2)$.

of the Waterfall Tree we also tried identifying subpaths on 2^t edges, where $t < s$ (or possibly $t > s$, for appropriate k). In the few cases we considered the differences in cost, although positive and relatively large, have not been as large as for $W(4,2)$ above.

Suppose we have a tree T on k edges, and a labeling sequence σ of all the vertices of the tree, starting with two leaves t_1 and t_2 say, giving the sequence. Suppose that the normalized cost $L'(\sigma)$ of σ is larger than $\mathrm{Opt}(k+1, \text{con-}K_2)$. Consider the following "doubling" construction.

Take two copies of T and identify the vertices corresponding to t_2. Call the resulting tree $2T$, having $2k$ edges. Now define the following sequence, which we call 2σ, of all vertices of $2T$ as follows. Label the tree $2T$ by first labeling the two vertices corresponding to t_1 in the two copies of T; that is, let $2\sigma(1)$ and $2\sigma(2)$ be the two vertices corresponding to t_1. Next label the identified vertex corresponding to t_2 [i.e., let $2\sigma(3)$ be the identified vertex t_2], Finally, label the remaining vertices using the labeling of σ, from $\sigma(3)$ onwards, on each of the two copies of T in succession. That is, let $2\sigma(i) = \sigma(i-1), 4 \le i \le k+2$, where σ is restricted to one copy of T, and let $2\sigma(i) = \sigma(i-k), k+3 \le i \le 2k+1$, where now σ is restricted to the other copy of T.

In light of some of the previous examples, it might have been hoped that this "doubling" construction may generate a tree $2T$, and sequence 2σ whose cost exceeds that of conK_2 by more than the original tree T. That this does not occur can be seen from the following theorem.

Theorem A.7. *Let T be a tree on $k+1$ vertices, and σ a sequence of all the vertices of T for which $L'(\sigma) > \mathrm{Opt}(k+1, \text{con-}K_2)$. Following the above notation, let $2T$ be the tree on $2k+1$ vertices obtained by "doubling" T, and 2σ the sequence of all vertices of $2T$. Then*

$$L'(2\sigma) - \mathrm{Opt}(2k+1, \text{con-}K_2) \le L'(\sigma) - \mathrm{Opt}(k+1, \text{con-}K_2),$$

with equality if and only if $T = \text{con-}K_2$.

Proof. Let d denote the distance from t_1 to t_2 in T, and recall the notation $I(\sigma)$ for the internal cost of σ. Because $c(2, 2\sigma) = 2d$ and $c(3, 2\sigma) = d$, we have $L(2\sigma) = 2d + d + 2I(\sigma) = 3d + 2L(\sigma) - 2d = d + 2L(\sigma)$. It follows that $L'(2\sigma) = \frac{d+2kL'(\sigma)}{2k} = L'(\sigma) + \frac{d}{2k}$. Because $d \le k$, we then get $L'(2\sigma) - L'(\sigma) \le \frac{1}{2}$, with equality if and only if $d = k$.

Recall the sequences μ_n realizing Opt (n,con-K_2) for any $n \ge 2$ defined just after Lemma 5.1. These sequences are nested, so that μ^{k+1} is the initial segment of length $k+1$ in the sequence μ_{2k+1}. From the construction of these sequences, the "leftover" terms $\mu_{2k+1}(i), k+2 \le i \le 2k+1$, are just the midpoints of all the subintervals of $[0,1]$ induced by the points of μ^{k+1}, because the latter points are also $\mu_{2k+1}(i), 1 \le i \le k+1$. Hence, $\mathrm{Opt}(2k+1, \text{con-}K_2) - \mathrm{Opt}(k+1, \text{con-}K_2) = \frac{1}{2}$. Finally, this gives $L'(2\sigma) - L'(\sigma) \le \frac{1}{2} = \mathrm{Opt}(2k+1, \text{con-}K_2) - \mathrm{Opt}(k+1, \text{con-}K_2)$, with equality if and only if $T = \text{con-}K_2$. ∎

REFERENCES

[1] N. Alon and Y. Azar, On-line steiner trees in the Euclidean plane, Discrete Comput Geometry 10 (1993) 113–121.

[2] B. Awerbuch, Y. Azar, and Y. Bartal, on-line generalized Steiner problem, Proc. SODA (1996) 68–74.

[3] P. Berman and C. Coulston, on-line algorithms for Steiner tree problems (extended abstract) STOC 1997, 344–353.

[4] M. Imase and B. Waxman, The dynamic Steiner tree problem, SIAM Journal of Discrete Mathematics, Vol. 4, No. 3 (1991) 369–384.

[5] F.K. Hwang, D.S. Richards, and P. Winter, "The Steiner tree problem," Annals of Disc Mathematics, Vol. 53, North Holland, Amsterdam, 1992.

[6] J. Plesnik, A lower bound for the Steiner tree problem in graphs, Math Slovaca, 31(2) (1981) 155–163.

[7] H. Takahashi and A. Matsuyama, An approximate solution for the Steiner problem in graphs, Math Jpn. 24 (1980) 573–577.

[8] H. Prömel and A. Steger, The Steiner tree problem—A tour through graphs, algorithms, and complexity, Vieweg Publishers, Braunschweig/Wiesbaden, 2002.

[9] J. Westbrook and D.C.K. Yan, The performance of greedy algorithms for the on-line Steiner tree and related problems, Math Sys Theory 28 (1995) 451–468.

[10] J. Westbrook and D.C.K. Yan, Linear bounds for on-line Steiner problems, Informa Process Lett 55 (1995) 59–63.