FINDING SUBSETS MAXIMIZING MINIMUM STRUCTURES*

MAGNÚS M. HALLDÓRSSON†, KAZUO IWANO‡, NAOKI KATOH§, AND TAKESHI TOKUYAMA†

Abstract. We consider the problem of finding a set of k vertices in a graph that are in some sense *remote*. Stated more formally: "Given a graph G and an integer k, find a set P of k vertices for which the total weight of a *minimum structure* on P is *maximized*." In particular, we are interested in three problems of this type, where the structure to be minimized is a Spanning Tree (REMOTE-MST), Steiner Tree (REMOTE-ST), or Traveling Salesperson tour (REMOTE-TSP).

We study a natural greedy algorithm that simultaneously approximates all three problems on metric graphs. For instance, its performance ratio for REMOTE-MST is exactly 4, while this problem is NP-hard to approximate within a factor of less than 2. We also give a better approximation for graphs induced by Euclidean points in the plane, present an exact algorithm for graphs whose distances correspond to shortest-path distances in a tree, and prove hardness and approximability results for general graphs.

Key words. minimum spanning tree, traveling salesperson, Steiner tree, dispersion.

AMS subject classifications. 58Q25, 05C85, 05C05.

1. Introduction.

Let $G[P]$ denote the subgraph of a graph G induced by a vertex subset P. We are interested in the following problem:

> REMOTE-MST: Given a complete undirected edge-weighted graph $G = (V, E)$ and integer k, find a subset P of V of cardinality k such that the cost of the minimum weight spanning tree on $G[P]$ is maximized.

We also study the problems REMOTE-TSP and REMOTE-ST, where the objective is to maximize the minimum traveling salesman tour and the minimum Steiner tree of the induced subgraph, respectively. These problems are illustrated in Figure 1.1.

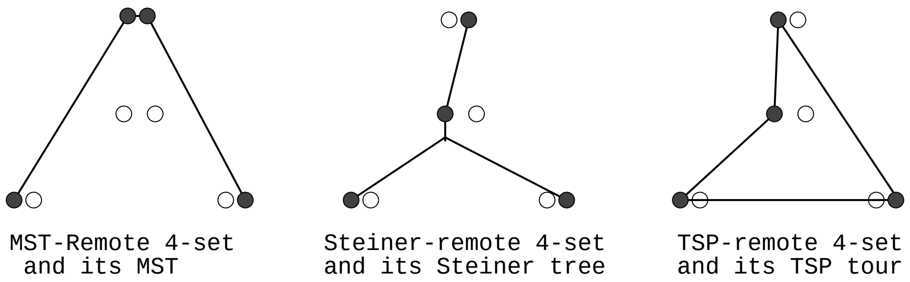

FIG. 1.1. *Remote planar point sets.*

Minimum weight spanning trees (MST, in short), minimum weight Steiner trees, and minimum weight tours (TSP, or traveling salesperson tours) are fundamental combinatorial structures, and the problems of finding such optimal structures are not

*Earlier version of this paper appeared in *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms, 1995.*

† Science Institute, University of Iceland, IS-107 Reykjavik, Iceland. `mmh@hi.is`

‡IBM Tokyo Research Laboratory, Yamato, Kanagawa 242, JAPAN. `iwano@trl.ibm.co.jp`, `ttoku@trl.ibm.co.jp`

§Kyoto University. `naoki@is-mj.archi.kyoto-u.ac.jp` Research performed at Kobe University of Commerce.

only useful in applications but also a rich source of research on exact and approximate algorithms. All of these problems consist of finding a subset *maximizing* the total weight of edges of minimum combinatorial structures constructed from the subsets. Except for REMOTE-ST, these structures are contained in the subgraph induced by the subset.

From a practical point of view, the REMOTE-MST (or REMOTE-ST) k-set of a network can be viewed as the set of k nodes among which communicating information is most expensive. Thus, the remote subsets can be applied to the evaluation of the communication performance of networks.

They can also be applied to clustering problems. Indeed, we originally faced these problems when trying to find a good "starting tour" of a large TSP instance (a circuit board drilling problem [19] that occurred at a manufacturing plant) with more than 10,000 non-uniformly distributed cities. To obtain a short approximate TSP tour by construction heuristics, it is effective to start with a subtour (starting tour) consisting of a relatively small number of sample cities capturing the global structure of the point distribution [20]. For this purpose, random sampling is not suitable, since it may miss some critical cities, and approximate TSP tours constructed from the associated starting tour often respond poorly to improvements by local search heuristics. The exact or approximate REMOTE-MST and/or REMOTE-TSP solutions seem to give better starting tours.

General framework. The problems under study can be generalized to the following framework. Let Π be a minimization problem whose solution is a subset of the edge set satisfying a particular property with respect to a given subset P of vertices. Let the cost of a solution be the sum of the weight of the edges in the solution. Let $\pi(P)$ denote the minimum cost value for a node set P. We are interesting in the following problem:

> REMOTE-Π: Given a graph $G = (V, E)$ and integer k, find a subset P of V of cardinality k such that $\pi(P)$ is maximized.

Our results. We present approximation algorithms in §2, for in metric graphs, general graphs, Euclidean graphs and trees.

Metric graphs are graphs with non-negative weights that satisfy the triangular inequality: for any three nodes u, v, w, $d(u, v) + d(v, w) \geq d(u, w)$. The *distance* of the edge $e(u, v)$, denoted $d(u, v)$, is the weight of the edge. One example of a metric graph is the *shortest-path distance graph* $D(G)$ of a graph G, where the weight of the edge (u, v) in $D(G)$ is defined to be the weight of the minimum weight path between u and v of G.

We apply in §2.1 a known greedy algorithm to obtain simultaneous approximations of all three problems in metric graphs. We obtain performance ratios of 4 for REMOTE-MST and 3 for REMOTE-TSP, both of which are tight for this algorithm, while the ratio for REMOTE-ST is at most 3 and at least 2.46.

For REMOTE-MST in general graphs, we give in §2.2 an algorithm that finds a solution within a factor of $k - 1$ from optimal.

Euclidean graphs are a special class of metric graphs, where the vertices correspond to points in the plane and the weight of an edge is the Euclidean distance between the points. The results obtained for the metric case, in combination with results on the *Steiner ratio* in the plane, yield asymptotic ratios of 2.31 (2.16) for the REMOTE-MST (REMOTE-ST) problems, respectively.

Finally, in §2.4, we give linear time algorithms for computing REMOTE-ST and REMOTE-MST when the set of edges in G with non-infinity weights forms a tree.

Π	General		Metric		$\mathbf{R}^2$
	l.b.	u.b.	l.b.	u.b.	u.b.
MST	$n^{1-\epsilon}$	$k-1$	2	4	2.25
TSP	∞		2	3	
ST	4/3	3	4/3	3	2.16

TABLE 1.1
Approximability of remote problems

In §3, we prove approximation hardness results for the three problems. Let n denote the number of vertices in the input graph. REMOTE-MST of general graphs cannot be approximated within a factor of $\Omega(n^{1-\epsilon})$, for any $\epsilon > 0$, unless $NP \subseteq ZPP$ (i.e. unless polynomially bounded zero-error randomized algorithms exist for all problems in NP). This proof is generalized to the remoteness versions of *degree-constrained subgraph* problems, with or without connectivity requirement. These problems include MST, TSP, minimum weight matching, cycle cover, degree-constrained spanning tree, and a number of other well-studied problems. For REMOTE-TSP we can prove a still harder inapproximability bound, since like the ordinary TSP problem, it cannot be approximated on general graphs within *any* ratio, unless $P = NP$.

On metric graphs, these problems are also NP-hard to approximate within a factor less than 2. Without loss of generality we may assume that the input to the REMOTE-ST problem is a metric graph. We show it to be hard to approximate within a ratio less than 4/3.

We summarize the main approximability results of the paper in the following table. It lists the results obtained for each of the MST, TSP, and ST remote problems, with lower and upper bounds for approximability in general graphs, metric graphs, and Euclidean graphs.

Related work. Problems of maximizing minimum structures have applications to the location of undesirable facilities. For instance, hazardous facilities like nuclear plants or ammunition dumps should be located as far from each other as possible to minimize vulnerability. A not insubstantial body of literature has developed on the subject – see [11] for a survey, primarily from a management science viewpoint. The focus has been on two structures not dealt with in this paper: the minimum weight of any edge in the k-set, and the average, or equivalently the sum, of the weights of edges between pairs in the k-set. For the former problem, known as the *k-Dispersion problem*, Ravi, Rosenkrantz and Tayi [25] showed that the 'greedy furthest-point' algorithm obtains a performance ratio of 2 on metric graphs, improving on a weaker bound of [29]. They also showed that approximating within a factor of less than 2 is NP-hard. Independently, Tamir [27] proved the same upper bound for the same algorithm (see also [28]).

A dispersion problem with the criteria of maximizing the *average distance* between vertices in the k-set was also considered in [25], and they gave a different greedy algorithm with a ratio of 4. Hassin et al. [15] gave an algorithm with a performance ratio of 2. Kortsarz and Peleg [17] considered this latter problem on general weighted graphs, under the name *Heavy Subgraph Problem*, and gave a sequence of algorithms that converges with a performance ratio of $O(n^{0.3885})$. While different minimum structures have been proposed in the location theory literature, we are not aware of work analyzing algorithms for such problems.

Problems on Euclidean graphs can be regarded as belonging to computational geometry. The problems of finding a subset with cardinality k of a planar point set maximizing the perimeter or area of convex hull (minimum perimeter enclosing polygon) of the subset has been studied in the literature and nearly linear time algorithms are known [2, 3, 7]. However, the authors know no previous results on computing subsets maximizing other minimum structures.

Problems of finding subsets *minimizing* the minimum weight of a combinatorial structure are more common [1, 10, 24, 14]. In particular, the problem of finding the k-set minimizing the weight of the minimum MST was studied by Ravi et al. [24], who proved NP-hardness and gave the first approximations. The performance ratios have recently been improved to 3 for general graphs [14] and $1+\epsilon$ for Euclidean graphs [21, 5].

Chandra and Halldórsson [8] continued the work started in this paper, and analyzed a number of other remote problems. In particular, they gave a $O(\log k)$-approximate algorithm for two problems suggested in a previous version of the current paper: computing a k-set maximizing the minimum weight matching, and the k-*defense* problem, where the objective $\pi(P)$ is $\sum_{v\in P}\min_{u\in P-\{v\}} d(u,v)$.

Notation. A *spanning tree* of a node set P is a subtree of G whose node set is P. A *Steiner tree* of P is a spanning tree of a *superset* of P. A *tour* of P is a cycle that contains all the vertices of P. The weight of a tree or a tour is the sum of the weight of the edges in it.

We denote the minimum spanning tree, minimum Steiner tree, and TSP tour of P by $MST(P)$, $ST(P)$, and $TSP(P)$, respectively. The weights of these minimum solutions are denoted by $mst(P)$, $st(P)$, and $tsp(P)$. For a graph H, the maximum cost of $MST(P)$ over all k-node sets P is denoted by r-mst(H). In general, for a problem Π and node set P, the minimum structure and the minimum value are denoted by $\Pi(P)$ and $\pi(P)$, respectively, and the optimal value of REMOTE-Π (i.e. the maximum weight of the minimum Π-structure) is denoted by r-$\pi(H)$.

The *approximation ratio* of an algorithm for REMOTE-MST on a given input graph G is the ratio of the largest MST weight of a set of k points to the MST weight of the k-set output by the algorithm. The same holds for other problems. The *performance ratio* ρ of the algorithm is the maximum approximation ratio over all instances. A problem is *approximable within a factor of* t if there exists a polynomial time algorithm for the problem with a performance ratio at most t. A problem Π_1 is *as hard to approximate* as problem Π_2 if, an approximation of Π_2 within a factor of $f(n)$ implies an approximation of Π_1 within a factor of $O(f(n))$.

Given a graph G and value γ, the bi-valued network $H_{G,\gamma}$ is a complete graph on the same vertex set as G, where the weight of an edge is 1 if the edge is in G, and γ otherwise. Let $G[P]$ denote the subgraph of G induced by a vertex subset P. Namely, $P\subset V(G)$ and $E(G[P]) = \{(v,u) \mid (v,u)\in E(G) \text{ and } v,u\in P\subseteq V(G)\}$. The distance graph $D(G)$ of a graph G has the weight of an edge (u,v) equal to the length of the shortest path from u to v in G.

2. Algorithms.

2.1. Metric graphs. In this section, we assume that $G=(V,E)$ is metric unless otherwise stated. Let the distance between a node u and a set of nodes be the minimum distance between u and any node in the set, $d(v,P)=\min_{p\in P} d(v,p)$.

Central to our approach is the concept of an *anticover*.

DEFINITION 2.1. *A set P of vertices $p_1, p_2, \ldots$ is an* anticover *of a graph if,*

1. $d(p_i,p_j)\geq r$*, for* $i\neq j$*, and*

2. $\min_i\{d(v, p_i)\} \leq r$ *for any node* $v \in V$.

The radius *of* P *is the largest value* r *for which* P *is an anticover. The* size *of an anticover is its number of vertices.*

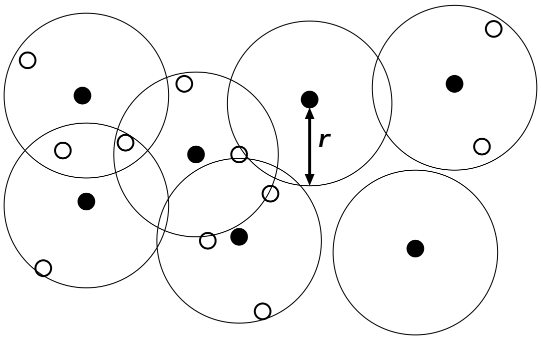

FIG. 2.1. *Anticover (black points) of size 7 of a Euclidean graph.*

An anticover is illustrated in Figure 2.1. An anticover can be constructed efficiently by the "greedy furthest-point" algorithm [12, 29, 25] given in Fig. 2.2.

Greedy(G)
  pick an arbitrary node v
  $P \leftarrow \{v\}$
  for $i \leftarrow 2$ to k
    $v \leftarrow$ node in $V - P$ furthest from P
    $P \leftarrow P \cup \{v\}$
end

FIG. 2.2. *The "greedy furthest-point" algorithm*

It is easy to see that the node set found by Greedy is an anticover of size k, and that its radius is the distance between the node v chosen last and $P - \{v\}$.

We apply Greedy to obtain simultaneous constant-factor approximations of the remote MST, TSP and Steiner problems. The same algorithm was also applied to approximate the k-Dispersion problem [29, 25], as well as the Euclidean k-clustering problem [12], indicating a level of universality of this approach and an applicability to multi-objective computing.

THEOREM 2.2. *An anticover of size* k *is a* 4*-approximation of* REMOTE-MST, *and a* 3*-approximation of* REMOTE-ST *and* REMOTE-TSP.

Proof. Let P be an anticover of G, and let r denote its radius. Let Q be any set of k points.

Any pair of points in P is of distance at least r, so

$$mst(P) \geq (k-1)r. \tag{2.1}$$

Each point q in Q is of distance at most r from P, thus the tree obtained by connecting Q to $MST(P)$ via the shortest edge is of weight at most $mst(P) + kr$. That is,

$$st(Q) \leq st(P \cup Q) \leq mst(P) + kr.$$

The ratio (Steiner ratio) of the weight of an MST of a set of k points to that of its Steiner tree is at most $2(k-1)/k$. It follows that,

$$\frac{mst(Q)}{mst(P)} \;\leq\; 2\frac{k-1}{k}(1+\frac{kr}{(k-1)r}) \;\leq\; 4-\frac{2}{k}.$$

Similarly, $st(P) \geq \frac{k}{2}r$ because of the Steiner ratio, and

$$st(Q) \leq st(P \cup Q) \leq st(P) + kr.$$

Hence, a performance ratio of 3 follows.

Furthermore,

$$tsp(P) \;\geq\; kr.$$

Connecting each point of Q to its nearest point in P by a pair of directed edges (with different directions), we can form a tour of $P \cup Q$ of length at most $tsp(P) + 2kr$. Thus,

$$tsp(Q) \leq tsp(P \cup Q) \leq tsp(P) + 2kr \leq 3 \cdot tsp(P). \qquad \square$$

The Steiner ratio $2(k-1)/k$ holds even if the tree is restricted to be a path, thus the results hold equally for degree-constrained versions of the problems.

While the analysis of the approximation ratio in Theorem 2.2 obtained by Greedy appears loose, it is actually asymptotically optimal for both REMOTE-MST and REMOTE-TSP. We give lower bounds on the performance of Greedy that holds for any choice of the initial starting vertex.

THEOREM 2.3. *The performance ratio of* Greedy *for* REMOTE-MST *on metric graphs is asymptotically 4.*

Proof. We construct a family of instances, for which Greedy is destined to perform poorly independent of its choice of a starting vertex.

Let G_t be an unweighted (i.e. unit-weighted) graph, with vertex set $\{c, p_1, p_2, \ldots, p_t, q_1, q_2, \ldots, q_t\}$. Let $p_1, \ldots, p_t, c$ be connected into a path, and let each q_i be connected to both p_1 and p_2. G_t contains no further edges.

Let $G'_{t,z}$ be the graph formed by taking z copies of G_t, with a single c vertex common to all copies (Figure 2.3). Thus, we have a connected graph on $2tz + 1$ nodes. For convenience, we use notations such as p_1-vertex, p-vertex, q-vertex, and c-vertex. In order to force the algorithm to prefer the p-vertices, we perturb the distances between vertices as follows: the lengths $d(c, p_t)$ are stretched to $1 + 2\epsilon$, and the lengths $d(p_i, p_{i+1})$ to $1 + \epsilon$ for $i \geq 1$.

The hard instance is the distance graph $D(G'_{t,z})$, with z sufficiently large. Observe that the distance between q-vertices in different copies is $2t(1+\epsilon)$, while the distances between p_1 vertices is $2t(1+\epsilon) + 2\epsilon$. Thus, a p_1 vertex is the furthest vertex from any set of at most $z-1$ vertices.

Let $k = tz$. The set of the first z vertices selected by Greedy contains at least $(z-1)$ p_1-vertices. Thus, Greedy cannot select a q-vertex adjacent to a selected p_1-vertex. Consequently, the number of q-vertices which Greedy can select is at most t. Also, Greedy must select the vertex c, whose neighbors are all of distance at least $1 + 2\epsilon$. Thus, ignoring the ϵ terms, $mst(P) \leq zt + 2(t-1)$ for any set P of k points selected by Greedy.

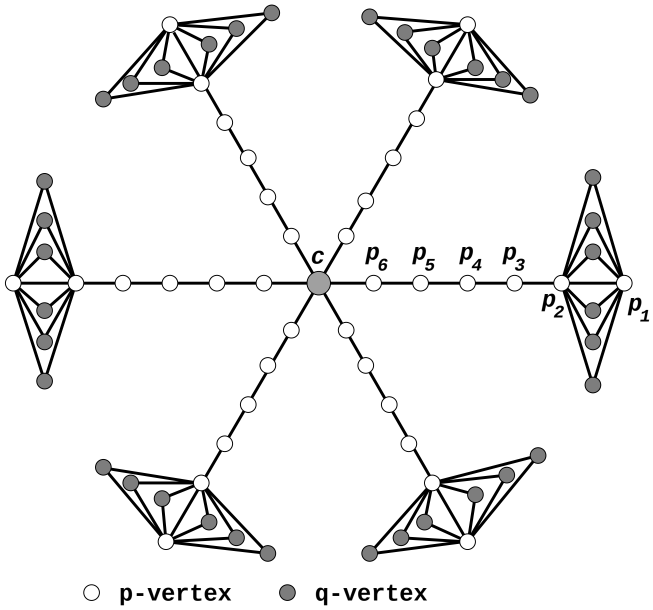

FIG. 2.3. *Lower bound example for* Greedy.

Let Q consist of the tz different q-vertices. Let q_i and q_i' be vertices in different copies of G_t. Then,

$$\begin{aligned} mst(Q) &= z(t-1)d(q_i, q_{i+1}) + (z-1)d(q_t, q_1') \\ &= 2z(t-1) + (z-1)(2t) \\ &= 4zt - 2z - 2t. \end{aligned}$$

If $z = t$, we have that

$$\rho \geq \frac{mst(Q)}{mst(P)} = 4 - O(\frac{1}{\sqrt{k}}). \qquad \square$$

Although the above lower bound is applicable only to the solution generated by Greedy, we conjecture that 4, rather than the lower bound of 2 that we will give in Theorem 3.4, is the best possible performance ratio for the problem.

One plausible approach for improving on the approximation produced by Greedy is to post-process the greedy solution with local improvement changes. Having obtained an anticover P of radius at most r, it may be possible to move individual points further away from the other points. That is, for a point $v \in P$ with $d = d(v, P - \{v\})$, there may exist a point $u \in V - P$ such that $d(u, P - \{v\}) > d$. This would improve the bounds, using a strengthening of (2.1) to $mst(P) \geq \sum_{v \in P} d(v, P - \{v\})(k-1)/k$.

The hard instances constructed above demolish that hope, since no single point can be moved further away. These instances can also be easily modified to ensure that no b points can be moved further away, for any fixed b.

THEOREM 2.4. *The performance ratio of* Greedy *for* REMOTE-TSP *is asymptotically 3.*

Proof. Our construction is based on the graphs $G'_{t,z}$ of the preceding theorem. Assume z is even, and consider an arbitrary matching of z copies of G_t into $z/2$ pairs.

Assign each edge between each pair G_t and $G_{t'}$ the weight $\alpha = \sqrt{t}$. Among these, we add an additional ϵ weight to the edges incident on p_1-vertices, to ensure they will always be favored.

Our graph $G''_{t,z}$ is the graph obtained by adding the above edges to the original $G'_{t,z}$. Then, Greedy selects the same set P as in Theorem 2.3, and there is a tour of P using edges from $MST(P)$ as well as $z/2$ matching edges between p_1 vertices. Thus, $tsp(P) = zt + o(zt)$. On the other hand, $tsp(Q) \geq 3zt$ for the set Q consisting of the q-vertices. □

THEOREM 2.5. *The performance ratio of* Greedy *for the* REMOTE-ST *problem is at least* $32/13 \geq 2.46$.

Proof. Let $z = k/2$. Let ϵ be a number less than $z^{-1/2}$.

Let T be an edge-weighted tree with $V(T) = \{c, p, q, u_1, u_2, r\}$. Let $d(p, u_1) = d(p, u_2) = 12$, $d(p, r) = 7$, $d(c, r) = 1$, and $d(r, q) = 5 + \epsilon$ for a positive real number ϵ. We extend this tree by adding edges to obtain a complete graph T' in which the distance between any pair of vertices is minimum of 16 and the shortest distance within this tree.

We construct a graph H containing z copies $T'(1), ..., T'(z)$ of T'. Let the copy of a vertex v in $T'(i)$ be denoted by $v(i)$. The vertices $c(1), c(2), ..., c(z)$ are located on a path with the distance between $c(i)$ and $c(j)$ is $|i - j|z^{-1/2}$. Define $dist_H(x, y)$ to be the minimum of 16 and the shortest path distance on H.

We extend H by adding edges to obtain a graph G whose distance function is denoted by $dist_G$. $dist_G$ equals $dist_H$ within each $T'(i)$ $(i = 1, 2, .., z)$, while for vertices $v(i)$ and $w(j)$ in distinct copies of T',

$$dist_G(v(i), w(j)) = \begin{cases} dist_H(p(i), p(j)) & v = p,\ w = p, \\ \min\{dist_G(p(i), p(j)) - \epsilon, dist_H(v(i), p(j))\} & v \neq p,\ w = p, \\ \min\{dist_G(p(i), p(j)) - 2\epsilon, dist_H(v(i), w(j))\} & v \neq p,\ w \neq p. \end{cases}$$

We can easily check that $dist_G$ satisfies the triangle inequality. By construction, $p(j)$ is the node in $T'(j)$ that is the farthest from $p(i)$, for any $i \neq j$. Also, if $|j-i| > 16z^{1/2}$, then $p(j)$ is the node in $T'(j)$ that is the farthest from any other node in $T'(i)$.

Greedy applied to G first selects a node, say $c(i_0)$, and then picks all $p(j)$ for which $|j - i_0| > 16z^{1/2}$. So far, the distance from the farthest node to the current vertex set is at least $16 + z^{1/2}$. Next, it picks at most two nodes (typically, $u_1(j)$ and $u_2(j)$) in each $T'(j)$ satisfying $|j - i| < 16z^{1/2}$.

Now, the distance from the farthest node to the current vertex set is reduced to $12 + \epsilon$, and the algorithms selects all $q(j)$ for which $|j - i_0| > 16z^{1/2}$. The algorithm has by now selected $k - O(z^{1/2})$ nodes; the choice of the remaining $O(z^{1/2})$ is irrelevant to our analysis.

The length of the spanning tree of the output of the algorithm is $(13+\epsilon)z+O(z^{1/2})$. On the other hand, if we pick $u_1(i)$ and $u_2(i)$ for $i = 1, 2, .., z$, the length of the spanning tree is $32z - O(z^{1/2})$. Hence, as z goes to infinity, the ratio between the cost of the optimal solution to that of the greedy solution approaches $32/13 \approx 2.4615$. □

The precise determination of the performance ratio of Greedy for REMOTE-ST remains an open problem.

2.2. General graphs. We give an approximation algorithm for REMOTE-MST on general graphs, with a performance ratio of $k - 1$.

For a graph G and a positive weight α, define G_α to be the subgraph of G on $V(G)$ with edges whose weight is less than α.

HeavyEdge(G)
  Determine the largest α such that
    G_α is not $(n-k+1)$-vertex-connected.
  Let C be a cutset of G_α of size $n-k$.
  Output $P = V - C$.
end

The desired α can be found by binary search on the at most $\binom{n}{2}$ different edge-weights. Since P, the subgraph induced by C, is not connected in G_α, an MST of P must contain an edge of weight at least α. On the other hand, if edges of weight α are added to G_α, any k-set must be connected. Thus,

$$\text{r-mst}(G) \le (k-1)\alpha \le (k-1)mst(P).$$

COROLLARY 2.6. HeavyEdge *has performance ratio of* $k-1$ *for* REMOTE-MST.

For the Steiner tree problem, it suffices to consider the distance graph of the input graph, which satisfies the triangular inequality. Thus, we obtain the following corollary of Theorem 2.2.

COROLLARY 2.7. REMOTE-ST *of a general graph can be approximated within a factor of 3.*

2.3. Euclidean graphs. Let P be a set of n points $\{p_1, \ldots, p_n\}$ in the plane. The Euclidean graph of P is the complete graph on the node set P where the weight of an edge (p_i, p_j) is the Euclidean distance $d(p_i, p_j)$. We consider algorithms for approximating REMOTE-MST and REMOTE-ST of this graph.

The anticover defined in the previous section gives a geometric covering of P by k circles of radius r, each of which is centered by a point in P. Since $st(P) \ge \sqrt{3}mst(P)/2$ [9] in the Euclidean case, we immediately obtain the following.

COROLLARY 2.8. *An anticover is a* $\frac{4k-2}{\sqrt{3}(k-1)}$*-approximation of* REMOTE-MST *and a* $\frac{2k+\sqrt{3}(k-1)}{\sqrt{3}(k-1)}$*-approximation of* REMOTE-ST *in Euclidean graphs.*

Thus, the approximation ratios are asymptotically at most $4/\sqrt{3} \approx 2.309$ for REMOTE-MST, and $(2+\sqrt{3})/\sqrt{3} \approx 2.155$ for the REMOTE-ST.

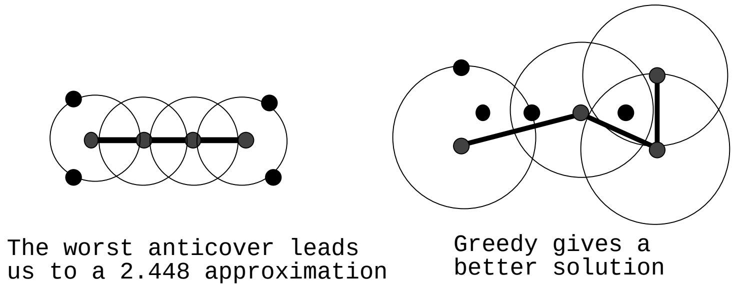

FIG. 2.4. *Approximation by circle covers.*

Unlike in the metric case, it seems that the approximation ratio depends on the choice of the anticover. For the example in Figure 2.4, the worst anticover has a $(2\sqrt{3}+4)/3) \approx 2.448$ approximation ratio, which is near to the upper bound $14/3\sqrt{3} \approx 2.694$ for the REMOTE-MST 4-set.

2.4. Tree networks. In this section, we consider graphs in which the set of edges with finite weights forms a tree. Let T be a weighted tree on n nodes. Define $G(T)$ to be a complete graph of order n, where the weight of an edge is the same as in T if it exists in T and ∞ otherwise.

We first give efficient algorithms for the REMOTE-ST k-set of $G(T)$. Clearly, we should select k leaves. If $k = 2$, the problem is the diameter path problem on a tree – finding the pair of vertices of maximum distance – and can be solved in linear time. If k exceeds the number of leaves of T, every set of k nodes containing all leaves forms the (unique) optimal REMOTE-ST k-set. The following lemma (essentially given in Peng et al. [23] for a slightly different problem) is the key observation:

LEMMA 2.9. *Any optimal* REMOTE-ST $(k-1)$*-set is contained in an optimal* REMOTE-ST k*-set.*

Proof. Let S_1 be an optimal REMOTE-ST $(k-1)$-set, and let S_2 be any optimal REMOTE-ST k-set. Let T_1 (T_2) be the Steiner tree spanned by S_1 (S_2), and let $W(T_1)$ ($W(T_2)$) be its weight. Then, $T_1 \cap T_2$ is a (possibly empty) tree.

The edge sets $F_1 = T_1 - T_1 \cap T_2$ and $F_2 = T_2 - T_1 \cap T_2$ are forests. Each connected component (a tree) in the forest has a root, which is a vertex in $T_1 \cap T_2$. If $T_1 \cap T_2$ is empty, we pick arbitrary vertices x in T_1 and y in T_2, and consider the path from x to y in T, and define the root of T_1 (resp. T_2) as its nearest node to y (resp. x) on the path.

A leaf of these forests must be either in $S_2 - S_1$ or $S_1 - S_2$. For each leaf v of these forests, let $h(v)$ be the weight of the path from v to the root in the component T_0 containing v, and $l(v)$ be the weight of the path to the nearest branch from v in T_0 (if there is no branch, to the root). Then, $h(v) \geq l(v)$, and equality holds if and only if T_0 is a path.

Consider a leaf v in $S_1 - S_2$ and a leaf w in $S_2 - S_1$. Unless $h(v) = l(v) = h(w) = l(w)$, either $h(w) > l(v)$ or $h(v) > l(w)$ holds. If $h(w) > l(v)$, we consider the set $S_1 - \{v\} + \{w\}$, and observe that the spanning tree of this set has the weight $W(T_1) + h(w) - l(v) > W(T_1)$. This contradicts the assumption that S_1 is an optimal REMOTE-ST $(k-1)$-set. Similarly, if $h(v) > l(w)$, we derive a contradiction to the fact that S_2 is an optimal REMOTE-ST k-set. Therefore, $h(v) = l(v) = h(w) = l(w)$ holds for all pairs of leaves v in F_1 and w in F_2. We write l for $l(v)$.

Hence, the total weight of T_1 is $W(T_1 \cap T_2) + |S_1 - S_2|l$, and that of T_2 is $W(T_1 \cap T_2) + |S_2 - S_1|l$. By definition, $|S_2 - S_1| = |S_1 - S_2| + 1$. Therefore, if we choose any $w \in S_2 - S_1$, the Steiner tree of $S_1 \cup \{w\}$ has the same length as that of S_2; hence $S_1 \cup \{w\}$ is a REMOTE-ST k-set, and it contains S_1. Thus, we obtain the lemma. □

This enables us to compute an REMOTE-ST k-set (more precisely, its spanning tree) by a greedy algorithm: Starting from any diameter path, find the farthest leaf from the current tree, and update the tree by adding the leaf and the path from the leaf. This takes $O(kn)$ time. We can give a better algorithm, which is an analogue of an algorithm of Shioura and Uno [26] for the problem in [23].

Consider a subtree H of T. Then, $T - H$ forms a forest F of rooted, directed trees with roots drawn from the vertices of H. We partition the edges of F into a set of directed paths such that for any internal node w on a path leading to a leaf l, the weight of the subpath from w to l is maximum over all directed paths from w to a leaf. This is easy to compute bottom-up in linear time for each tree in T by selecting for each internal node the path of maximum weight through a child. Define for each leaf l in F the *benefit* of l to be the weight of the path incident on l.

In particular, we consider a diameter path P as H. The above process computes all the benefits of leaves of $T - P$ in linear time. Then, we can observe the following lemma:

LEMMA 2.10. *The greedy algorithm adds the leaves with the largest $k-2$ benefits of $T - P$ to P to obtain a* REMOTE-ST *k-set.*

Proof. For any RST j-set and its Steiner tree H, the leaf v in $T - H$ of largest benefit is the one that is farthest from H. Moreover, for the tree H' obtained by joining the path from v to H, the benefits of $T - H'$ are by construction the same as the benefits of $T - H$ (except, of course, for v). Thus, we have the lemma. □

We can select the k largest benefits using a linear time selection algorithm, and hence, we have the following theorem:

THEOREM 2.11. *The* REMOTE-ST *k-set of $G(T)$ can be computed in $O(n)$ time.*

We next consider the REMOTE-MST problem on trees. REMOTE-MST of $G(T)$ is not a well-defined problem, since we can almost always find a subset P whose MST has infinity weight. Instead, we add a *connectivity condition* to the definition of a remote k-set P, such that $mst(P)$ is maximized on the condition that $mst(P) \neq \infty$. Namely, $MST(P)$ must be an induced subtree of P in T. Thus, the problem becomes a special case (where all edge weights are non-positive) of the *minimum weighted $(k-1)$-cardinality tree* problem defined by Fischetti et al. [13] if we reverse the sign of all weights of T. We can thus apply their $O(k^2 n)$ time dynamic programming algorithm. Moreover, we can improve it to $O(kn)$ time.

THEOREM 2.12. *The minimum weighted $(k-1)$-cardinality tree of a weighted tree can be computed in $O(kn)$ time. Hence, the* REMOTE-MST *k-set of $G(T)$ under the connectivity condition can be computed in $O(kn)$ time.*

Proof. We only give a proof for the computation of an optimal REMOTE-MST k-set. For a node v, a *v-optimal* REMOTE-MST *j-set* refers to a j-set that induces an optimal REMOTE-MST among those j-sets constrained to contain v.

Fix any internal node r as the root of T. For a subtree T_0 of T and its highest node (root of T_0) r_0, we consider the following set called the *profile* of T_0:

> The weights of the optimal REMOTE-MST j-sets containing r_0 (we call r_0-optimal REMOTE-MST j-sets) in T_0 for $j = 1, 2, \ldots, \min k, |T_0|$, and the weight of the optimal REMOTE-MST k-set of T_0 if $k > |T_0|$.

We give a dynamic programming algorithm that sweeps the tree bottom-up to compute the profile of T.

If r has only one child r_1, then the r-optimal j-set of T is the union of r with the r_1-optimal $j-1$-set of T_1. Otherwise, let r_1 be any child of r rooting the subtree T_1 and let T_2 be the tree obtained by cutting T_1 and the adjoining edge from T.

The minimum spanning tree of the optimal REMOTE-MST j-set containing r in T can be obtained by joining the r_1-optimal REMOTE-MST j_1-set of T_1 and the r_2-optimal REMOTE-MST j_2-set of T_2 for a suitable pair j_1 and j_2 satisfying $j_1 + j_2 = k$. The weight of the tree is simply the sum of these two parts. Thus, the weight of the r-optimal REMOTE-MST j-set, for $j = 1, 2, \ldots k$, can be computed in $O(k^2)$ time by examining all combinations.

We can improve the analysis of the time complexity. We say that a node u of T is *heavy* if both of its descendant trees have at least $k/2$ nodes, and *light* otherwise. The number of heavy nodes is at most n/k. We separately charge for operations at the heavy nodes, which is of cost $O(kn)$ in total. Let $f(n)$ be the cost for operations at all light nodes.

At a light node r rooting T with subtrees T_1 and T_2, T_i has n_i nodes and $m_i =$

$n_i - 1$ edges. Since r is a branching node, $m_i \geq 1$ $(i = 1, 2)$. T has $m = n - 1$ edges, thus $m \geq m_1 + m_2$ holds.

The profile for T_i has $\min(n_i, k)$ weights of r_i-optimal REMOTE-MST sets. Hence, to compute the profile of T, we only need to examine $\min(n_1, k)\min(n_2, k)$ combinations, which takes $O(\min(n_1, k)\min(n_2, k)) = O(\min(m_1, k)\min(m_2, k))$ time. Thus, the cost function $f(m)$ (up to a constant factor) follows the formula $f(m) \leq f(m_1) + f(m_2) + \min(m_1, k)\min(m_2, k)$.

Consider $g(m) = \min\{2km, m^2\}$. We shall verify that $g(m)$ satisfies $g(m) \geq g(m_1) + g(m_2) + \min(m_1, k)\min(m_2, k)$. Assume without loss of generality that $m_2 \leq m_1$. Thus, m_2 is smaller than $k/2$ at a light node.

Case 1: If $2k \geq m$, then $g(m) = m^2 \geq (m_1 + m_2)^2 \geq m_1^2 + m_2^2 + m_1 m_2$.

Case 2: If $m \geq 2k$, then $g(m) = 2km \geq 2km_1 + km_2 + km_2 > \min(2k, m_1)m_1 + m_2^2 + km_2$.

Hence, $f(m) < cg(m)$ for some constant c, thus is $O(km)$. Since $m = n - 1$, the complexity is $O(kn)$. ◻

The same algorithm can compute REMOTE-MST k-sets (with connectivity condition) of decomposable graphs, such as series-parallel graphs, in $O(kn)$ time.

3. Hardness. The decision version of REMOTE-MST (to decide whether there exists a set of k vertices whose MST weight is more than a given threshold) is obviously in NP. Instead of showing NP-hardness, we show approximation-hardness for both general and metric graphs.

We shall be primarily interested in approximating the remote problems within a function independent of k. Thus, we ask about the worst-case performance ratio as k ranges from 1 through n. Let $\alpha(G)$ denote the independence number of G, or the size of a maximum independent set.

THEOREM 3.1. *Approximating* REMOTE-MST *is as hard as approximating* INDEPENDENT SET.

Proof. Let g be the gap in the approximability of INDEPENDENT SET. Thus, for some value R, determining if $\alpha(G) = R$ or $\alpha(G) \leq R/g$ is hard.

Let k be R, and let γ be a value greater than k. We construct a bi-valued graph $H = H_{G,\gamma}$ on the same vertex set as G, with the weight of an edge being 1 if contained in G and γ otherwise. Refer to Figure 3.1.

If there is an independent set of size k in G, then that set has a value r-mst $= (k-1)\gamma$. On the other hand, suppose r-mst$(H) \geq (k-1)\gamma/g$. Notice that this is at least $(k/g - 1)\gamma + (k - k/g)$, since $\gamma \geq k$. Then, there is a subset P of k vertices such that $MST(P)$ contains at least $k/g - 1$ edges of weight γ. Let $G[P]$ be the the subgraph in G induced by P. It follows that $G[P]$ must contain at least k/g connected components. Hence, $\alpha(G) \geq \alpha(G[P]) \geq k/g$.

It follows that

$$\alpha(G) = k \Rightarrow \text{r-mst}(H) = (k-1)\gamma$$
$$\alpha(G) \leq k/g \Rightarrow \text{r-mst}(H) \leq (k-1)\gamma/g.$$

Thus, a gap in the approximation of INDEPENDENT SET carries over to REMOTE-MST. ◻

Håstad has recently strengthened the approximation hardness of INDEPENDENT SET to $n^{1-\epsilon}$, for any $\epsilon > 0$ [16]. This assumes that $NP \not\subseteq ZPP$, or that randomized polynomial-time algorithms do not exist for NP.

We now generalize the hardness proof for REMOTE-MST to other problems. Given a graph and integers ℓ and u, the *degree-constrained subgraph problem (DCS)* is

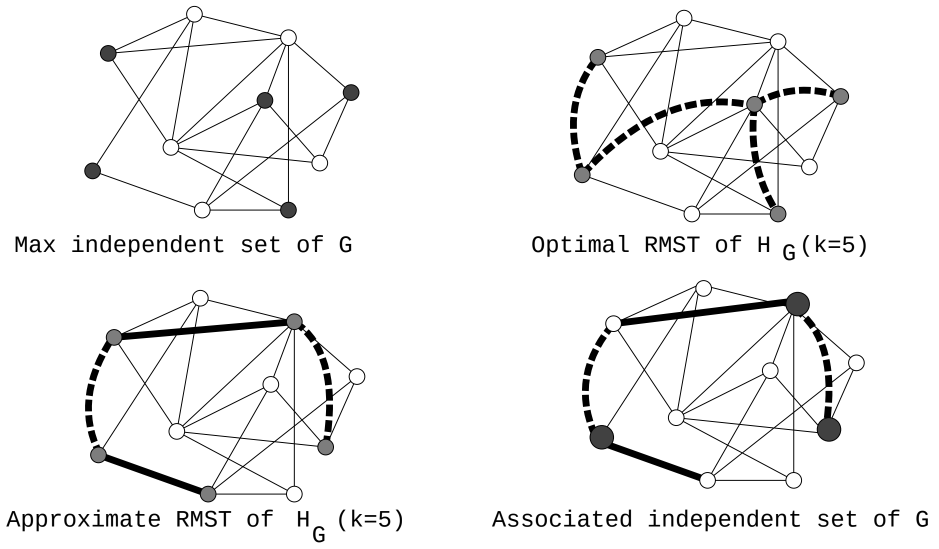

FIG. 3.1. *Graphs in Theorem 3.1.*

to find a subgraph of minimum weight such that the degree of each vertex is between ℓ and u, inclusive. Note that u may be only the trivial bound of $n-1$. The DCS minimization problem can be solved via a reduction to non-bipartite matching [18], and it subsumes the assignment problem, and problems of covering the vertices with cycles or paths. If the subgraph must additionally be connected, we have *Connected-DCS (CDCS) problems*, which include TSP, MST, and the Degree-Constrained MST problems.

Assume from now that Π is any such DCS problem, with degree lower bound ℓ. For a given $\gamma > 1$, let $H = H_{G,\gamma}$ be the bi-valued network on G, that has the weight of an edge being 1 if the edge is in G, and γ otherwise. Fix some optimal Π-solution to H, and let $\Pi(H)$ denote its set of edges. We sometimes abuse notation by denoting Π for $\Pi(H)$.

LEMMA 3.2. *Let G be a graph, and $\gamma \geq 1$. Let $Heavy$ denote the set of γ-weight edges in $\Pi(H)$. Then,*

$$\alpha(G) \geq \frac{|Heavy|}{\ell^2+1}.$$

Proof. The statement is trivial if $|Heavy| \leq \ell^2 + 1$, thus we assume the contrary. Also, we assume that the number k of vertices in $\Pi(H)$ is greater than any constant power of ℓ.

Let $Conn$ denote some minimal set of edges from $Heavy$ such that $Conn \cup (\Pi(H) - Heavy)$ is connected and spans H, if Π is a problem requiring connectivity. Otherwise, let $Conn$ be the emptyset. Let $Slack = Heavy - Conn$. Let S denote the set of vertices incident on fewer than ℓ edges in $\Pi(H) - Slack$.

We first observe that

$$\alpha(G) \geq |Conn| + 1, \tag{3.1}$$

since G must contain that many connected components. To satisfy the lemma, it now suffices to bound the independence number in terms of the other heavy edges, by $\alpha(G) \geq |Slack|/\ell^2$.

Observe that each edge in $Slack$ has an endpoint in S, and furthermore, it has an endpoint in S that is of degree exactly ℓ in $\Pi(H)$. Otherwise, this edge would be superfluous to $\Pi(H)$, as connectivity and degree requirements are satisfied without it. This implies that

$$|Slack| \leq \ell|S|.$$

To complete the lemma, we need to show that all the edges in $G[S]$ must be in $\Pi(H)$. That implies that each vertex in S is incident on at most $\ell - 1$ edges in $G[S]$, and any maximal independent set of $G[S]$ is of size at least $\frac{1}{\ell}|S|$. Hence, the independence number of the whole graph is no less, and

$$\alpha(G) \geq \alpha(G[S]) \geq \frac{1}{\ell}|S| \geq \frac{|Slack|}{\ell^2} \tag{3.2}$$

as desired.

CLAIM 1. $E(G[S]) \subseteq \Pi(H)$

Suppose on the contrary that there were vertices x, y in S such that $(x, y) \in G$ but $(x, y) \notin \Pi(H)$. Let (x, x'), (y, y') be edges from $Heavy - Conn$ (where x' and y' are not necessarily distinct).

We consider three cases depending on the degrees of x' and y' (in $\Pi(H)$). If x' and y' are distinct and both of degree greater than ℓ, then let $\Pi' = (\Pi - \{(x, x'), (y, y')\}) \cup \{(x, y)\}$. If one of x' and y', say x', is of degree greater than ℓ, then let $\Pi' = (\Pi - \{(x, x'), (y, y')\}) \cup \{(x, y), (y', z)\}$, where z is some vertex of degree ℓ non-adjacent to y' (and such a vertex must exist since there must be at least $\ell + 1$ vertices of degree ℓ).

Otherwise, the number of heavy edges in Π such that either x' or y' is either incident on the edge or adjacent (via an edge in $\Pi(H)$) to one of its endpoints is at most ℓ^2. Thus, there must exist a third edge (x'', y'') from $Heavy - Conn$ such that x' and x'' are non-adjacent, as well as y' and y''. Let $\Pi' = (\Pi - \{(x, x'), (y, y'), (x'', y'')\}) \cup \{(x, y), (x', x''), (y', y'')\}$.

In all cases, the edges removed from Π are from $Heavy - Conn$, and thus Π' is connected, and degree constraints are preserved. Hence, Π' is a valid solution of lesser cost, contradicting the minimality of Π. The claim and the lemma then follow. ☐

THEOREM 3.3. *Approximating* REMOTE-DCS *and* REMOTE-CONNECTED-DCS *problems is as hard as approximating* INDEPENDENT SET, *for any fixed value of* ℓ.

Proof. Let γ be a number greater than uk, and let $H = H_{G,\gamma}$.

If there is an independent set of size k in G, then r-$\pi(H) \geq \frac{\ell}{2}k\gamma$.

On the other hand, suppose r-$\pi(H) \geq \frac{\ell}{2}k\gamma/g$. Then, there is a subset P of k vertices such that $\Pi(P)$ contains at least $z \geq \frac{\ell}{2}k/g$ edges of weight γ. By Lemma 3.2, $\alpha(G[P]) \geq z/(\ell(\ell+1)) \geq k/(2(\ell+1)g) = \frac{\ell}{2}k\gamma/g'$, where $g' = g/(\ell(\ell+1))$.

It follows that

$$\alpha(G) = k \Rightarrow \text{r-}\pi(H) = \tfrac{\ell}{2}k\gamma$$
$$\alpha(G) \leq k/g \Rightarrow \text{r-}\pi(H) \leq \tfrac{\ell}{2}k\gamma/g'. \qquad \square$$

Similarly, these problems are also hard to approximate in metric graphs within a factor of $2 - \delta$, for any $\delta > 0$. We prove this here only for properties for which all feasible solutions have the same number of edges; the general case is quite tedious, especially for other connected properties.

THEOREM 3.4. *Let* Π *be a DCS problem with* $\ell = u$ *(e.g. TSP), or a connected property with* $\ell = 1$ *(i.e.* (DEGREE-CONSTRAINED) MST*). Then,* REMOTE-Π *is hard to approximate within a factor of* $2 - o(1)$ *in the metric space with distances 1 and 2.*

Proof. Let $\gamma = 2$, $H = H_{G,\gamma}$. Observe that any feasible solution to Π has the same number e of edges: $\ell k/2$ in the former case, and $k-1$ in the latter case

If there is an independent set of size k in G, then r-$\pi(H) = 2e$. On the other hand, suppose r-$\pi(H) \geq e(1+\delta)$. Then, there is a subset P of k vertices such that $\pi(P) \geq e(1+\delta)$. Thus, $\Pi(P)$ contains at least $e\delta$ edges of weight 2. By Lemma 3.2,

$$\alpha(G) \geq \frac{e\delta}{\ell(\ell+1)}.$$

Let $\delta' = \delta k/[(k-1)\ell(\ell+1)]$. Then,

$$\alpha(G) = k \Rightarrow \text{r-}\pi(H) = 2e,$$
$$\alpha(G) < \delta' k \Rightarrow \text{r-}\pi(H) < e(1+\delta)$$

Hence, the problem is hard to approximate within $2 - 1/f(n)$, where $f(n)$ is a function growing with n. ☐

Theorem 3.3 can also be extended to problems involving t-connectivity (for $t = k^{o(1)}$). It can also be extended to other remote-Π problems that satisfy the following property: If F is a feasible solution to Π and (v,u) and (x,y) are edges in that solution, then $F - \{(v,u),(x,y)\} \cup \{(v,x),(u,y)\}$ is also a feasible solution to Π.

One example is when $\pi(P) = \sum_{v \in P} \min_{u \in P} d(u,v)$. The corresponding remote problem, that of finding a k-vertex set P maximizing this quantity, was considered by Moon and Chaudhry [22] under the name *k-Defense problem.* The above reduction shows that approximating it within $n^{1-\epsilon}$ in general graphs is hard.

The REMOTE-TSP problem is yet harder; like the underlying TSP problem, it cannot be approximated within any representable function.

THEOREM 3.5. *Let* W *be a polynomial representable value. Then, approximating* REMOTE-TSP *in general graphs within a factor* W *is* NP*-complete.*

Proof. We give a reduction from Hamilton circuit, for $k = n$.

Given a graph G on n vertices, construct the complete weighted graph H with vertex set $V(G) \cup \{u_1, \ldots, u_n\}$. Define the edge weights by

$$w(u,v) = \begin{cases} (Wn)^2 & \text{if } u,v \in V(G),\ (u,v) \notin E(G), \\ W & \text{if } u,v \in V(H) - V(G), \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

Observe that H can be represented in size polynomial in n.

Suppose G does not contain a Hamilton circuit. Then, $V(G)$ is a remote k-set whose minimum salestour is of cost at least $(Wn)^2$. On the other hand, if G does contain a Hamilton circuit, then $V(H) - V(G)$ is a k-set whose minimum salestour is of cost Wn, while any other k-set has lesser cost. It follows, that an algorithm that can approximate REMOTE-TSP within a factor less than Wn will decide the Hamilton circuit problem. ☐

For REMOTE-ST, one can always assume that graph G is metric, since the minimum Steiner tree of a node set P in G can be realized in the shortest-path distance graph $D(G)$.

THEOREM 3.6. *Approximating* REMOTE-ST *within a factor of* $4/3-\delta$ *is* NP*-hard for any* $\delta > 0$.

Proof. Given a graph $G = (V, E)$, we construct a graph H as follows. Replace each edge of G by a path with two-edges, and connect the middle vertices of the paths into a clique. More formally, H contains a vertex for each vertex v_i in V as well as each edge e_j in E. A vertex v_i is adjacent only to those vertices e_j for which v_i intersects e_j in G. Vertices e_j are completely connected into a clique.

The input to REMOTE-ST is the distance graph $D(H)$ of H. If we consider two vertices in G, they will be of distance 2 in H if they are adjacent in G, and of distance 3 in H if they are non-adjacent in G.

An independent set in G corresponds to a set of vertices in H that have no neighbors in common. Hence, the cost of the minimum Steiner tree of that set in $D(H)$ is $2(k-1)$.

A *loner* in a Steiner tree is a leaf whose neighbor is not adjacent to another leaf. Suppose there are two loners in a Steiner tree of $D(G)$ that were adjacent in G. Then, the four edges connecting them to the remaining tree could be replaced by three edges all incident on the corresponding edge-vertex in $D(G)$. Hence, given a k-set P, we can easily find a Steiner tree of P where loners form an independent set in G. If p is the number of loners, then the cost of the Steiner tree constructed will be at most $\frac{3}{2}(k-p-1) + 2p = \frac{3}{2}(k-1) + \frac{1}{2}p$.

If, now, we could guarantee finding a k-set where the minimum Steiner tree is of size at least $\frac{3}{2}k + \frac{1}{2}p$, it follows that the independence number of G is at least p. By the hardness of the independent set problem, it is hard to decide whether r-st(G) is 2 or $\frac{3}{2} + o(1)$. ◻

4. Concluding remarks. If we remove the cardinality condition from the REMOTE-MST problem, we have the following problem:

> REMOTE-MST SUBSET problem: Find a subset Q of V such that $mst(Q)$ is maximized.

The REMOTE-MST subset problem can be considered to be an *inverse problem* to the Steiner problem. Whereas the Steiner problem asks for a superset Q' of P minimizing $MST(Q')$, the REMOTE-MST subset problem calls for a subset Q of V maximizing $MST(Q)$.

In the metric case, returning V as the solution trivially gives an approximation equal to the Steiner ratio, or 2 for general metric graphs and $2/\sqrt{3}$ for Euclidean graphs. We pose the question of improved ratios as an open problem.

Another open problem concerns the complexity classification of REMOTE-ST and REMOTE-TSP. They are in Σ_p^2, at the second level of the polynomial time hierarchy, and are NP-hard, from our results. We conjecture that they are also hard for Σ_p^2.

Other open problems include: proving NP-hardness of REMOTE-MST (and perhaps MAX-SNP-hardness) in the Euclidean plane, and giving better bounds for the approximation ratios for each problem. In particular, a good approximation algorithm for REMOTE-ST will be very useful in applications. Also, a fast algorithm would be needed; when we apply approximate REMOTE-TSP k-sets to large-scale TSP heuristics, sub-quadratic time algorithm is essential.

REFERENCES

[1] A. AGGARWAL, H. IMAI, N. KATOH, AND S. SURI, *Finding k points with minimum diameter and related problems*, J. Algorithms, 12 (1991), pp. 38–56.

[2] A. Aggarwal, M. Klawe, S. Moran, P. Shor, and R. Wilber, *Geometric applications of a matrix-searching algorithm*, Algorithmica, 2 (1987), pp. 195–208.

[3] A. Aggarwal, B. Schieber, and T. Tokuyama, *Finding a minimum weight K-link path in graphs with monge property and applications*, Disc. Computational Geom., 12 (1994), pp. 263–280.

[4] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. Assoc. Comput. Mach., 45 (1998), pp. 501–555.

[5] S. Arora, *Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems*, J. Assoc. Comput. Mach., 45 (1998), pp. 753–783.

[6] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, *Greedily finding a dense subgraph*, In Proc. 5th Scandinavian Workshop on Algorithm Theory (1996), Springer LNCS 1097, pp. 136-145.

[7] J. Boyce, D. Dobkin, R. Drysdale, and L. Guibas, *Finding extremal polygons*, SIAM J. on Computing, 14 (1985), pp. 134–147.

[8] B. Chandra and M. Halldórsson, *Facility dispersion and remote subgraphs*, In Proc. Fifth Scandinavian Workshop on Algorithm Theory (1996), Springer LNCS #1097, pp. 53–65.

[9] D.-Z. Du and F. K. Hwang, *Proof of the Gilbert-Pollak conjecture on the Steiner ratio*, Algorithmica, 7 (1992), pp. 121–135.

[10] D. Eppstein, *New algorithms for minimum area k-gons*, In Proc. 3rd ACM-SIAM SODA (1992), pp. 83–87.

[11] E. Erkut and S. Neuman, *Analytical models for locating undesirable facilities*, Europ. J. Oper. Res, 40 (1989), pp. 275–291.

[12] T. Feder and D. H. Greene, *Optimal algorithms for approximate clustering*, In Proc. 20th ACM STOC (1988), pp. 434–444.

[13] M. Fischetti, H. W. Hamacher, K. Jørnsten, and F. Maffioli, *Weighted K-cardinality trees: Complexity and polyhedral structure*, Networks, 24 (1994), pp. 11–21.

[14] N. Garg, *A 3-approximation of the minimum tree spanning k vertices*, In Proc. 37th IEEE FOCS (1996), pp. 302–308.

[15] R. Hassin, S. Rubinstein, and A. Tamir, *Approximation algorithms for maximum facility dispersion*, Oper. Res. Let., pp. 133-137, 1997.

[16] J. Håstad, *Clique is hard to approximate within $n^{1-\epsilon}$*, In Proc. 37th IEEE FOCS (1996), pp. 627–636.

[17] G. Kortsarz and D. Peleg, *On choosing a dense subgraph*, In Proc. 34th IEEE FOCS (1993), pp. 692–701.

[18] E. Lawler, *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.

[19] S. Misono and K. Iwano, *Circuit board drilling problem*, Technical Report, Inf. Proc. Soc. Japan, 93-AL-33 (1993), pp. 95–102.

[20] ———, *Experiments on TSP real instances*, IBM Tokyo Research Laboratory, Research Report, RT0153 (1996).

[21] J. Mitchell, Guillotine subdivisions approximate polygonal subdivisions: *A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems,* SIAM J. Comp., to appear.

[22] I. D. Moon and S. S. Chaudhry, *An analysis of network location problems with distance constraints*, Management Science, 30 (1984), pp. 290–307.

[23] S. Peng, A. B. Stephens, and Y. Yesha, *Algorithms for a core and k-tree core of a tree*, J. Algorithms, 15 (1993), pp. 143–159.

[24] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, and S. S. Ravi, *Spanning trees short or small*, SIAM J. Disc. Math, 9:2 (1996), 178–200.

[25] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi, *Facility dispersion problems: Heuristics and special cases*, Operations Res., 42 (1994), pp. 299–310.

[26] A. Shioura and T. Uno, *A linear time algorithm for finding a k-tree core*, J. Algorithms, 23:2 (1997), pp. 281–290.

[27] A. Tamir, *Obnoxious facility location on graphs*, SIAM J. Disc. Math., 4 (1991), pp. 550–567.

[28] A. Tamir, *Comments on the paper 'Facility dispersion problems: Heuristics and special cases, by S.S. Ravi, D.J. Rosenkrantz, and G.K. Tayi'*, Operations Res., 46:1, 1998.

[29] D. J. White, *The maximal dispersion problem and the "first point outside the neighborhood" heuristic*, Computers Ops. Res., 18 (1991), pp. 43–50.