Finding subsets maximizing minimum structures*

Magnús M. Halldórsson† Kazuo Iwano‡ Naoki Katoh§ 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 give a natural greedy approximation algorithm that simultaneously approximates all three problems on metric graphs. For instance, its performance ratio for REMOTE-MST is exactly 4, while it is NP-hard to approximate within a factor of less than 2. We also show a better approximation for graphs induced by Euclidean points in the plane, give an exact algorithm for graphs whose distances correspond to shortest-path distances in a tree, and give hardness and approximability results for general non-metric graphs.

Abbreviated title: Subsets maximizing minimum structures
Key words: Maximum Spanning Tree, Traveling Salesperson, Steiner tree, Dispersion.
Subject classification: 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 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 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.

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

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

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

§Kobe University of Commerce, Gakuen-Nishimachi, Nishi-ku, Kobe 651-21, JAPAN. naoki@kucgw.kobeuc.ac.jp

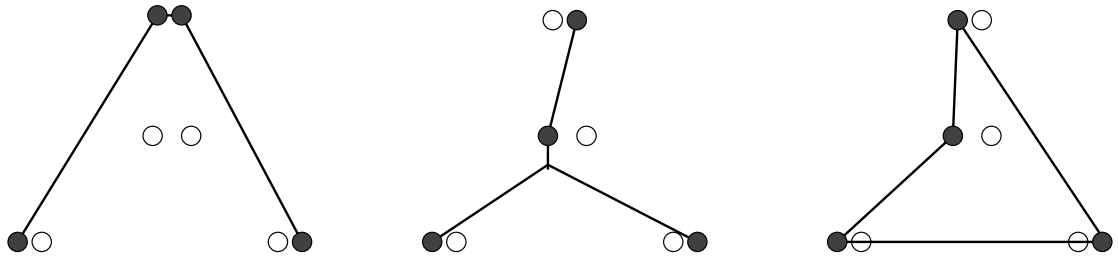

Figure 1: Remote planar point sets.

The problems of finding a minimum weight spanning tree (MST, in short), minimum weight Steiner tree, and minimum weight tour (TSP, or traveling salesperson tour) are fundamental combinatorial structures, which are not 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 [18] 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 [19]. 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 nodeset 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 Section 2: in metric graphs, general graph, Euclidean graphs and tree graphs.

Metric graphs are graphs with 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 an non-complete graph G, where the edge weight of $e(u,v)$ is defined to be the weight of the minimum weight path between u and v of G.

We apply in Section 2.1 the 'greedy furthest-point' 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. For the REMOTE-ST problem, the greedy algorithm attains a ratio of 3.

For REMOTE-MST in general graphs, we give in Section 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 Section 2.4, we give a linear time algorithm for computing REMOTE-ST when the set of edges with non-infinity weights forms a tree in G.

In Section 3, we prove approximation hardness results for the three problems. REMOTE-MST and REMOTE-TSP of general graphs cannot be approximated within a factor of $\Omega(n^{1-\epsilon})$, unless $NP \subseteq ZPP$. Here, n is the number of vertices in the input graph. We generalize the proofs 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. On metric graphs, these problems are also NP-hard to approximate within a factor less than 2.

The REMOTE-ST problem effectively always works on a metric graphs, by using the shortest-distance graph of the input 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.

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	$n^{1-\epsilon}$	$k-1$	2	3	
ST	4/3	3	4/3	3	2.16

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 [10] for a survey, primarily from the management science viewpoint. The focus has been on two structures not specifically 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 [24] showed that the 'greedy furthest-point' algorithm obtains a performance ratio of 2 on metric graphs, improving on a weaker bound of [28]. They also showed that approximating within a factor of less than 2 is NP-hard. Independently, Tamir [26] proved the same upper bound for the same algorithm (see also [27]).

A dispersion problem with the criteria of maximizing the *average distance* between vertices in the k-set was also considered in [24], and they gave a different greedy algorithm with a ratio of 4. Hassin [14] gave an algorithm with a performance ratio of 2. Kortsarz and Peleg [16] 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 $O(n^{3.885})$. While different minimum structures have been proposed in the location theory literature, we are not aware of work analyzing algorithms for such problems.

If the input is a complete graph with nodes corresponding to a set of points in Euclidean space and edge weights corresponding to the Euclidean distance between the pairs of points, the problems can be regarded as belonging to computational geometry. The problem 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 [2, 3, 6]. 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, 9, 23, 13]. In particular, the problem of finding the k-set minimizing the weight of the minimum MST was recently studied by Ravi et al. [23], who proved NP-hardness and gave the first approximations. The performance ratios have recently been improved to the best possible 3 for general graphs [13] and $1+\epsilon$ for Euclidean graphs [20].

Chandra and Halldórsson [7] have 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 $\sum_{v \in P} \min_{u \in P} d(u,v)$.

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 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 nodeset 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* r-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 r-anticover. The* size *of an anticover is its number of vertices.*

An anticover is illustrated in Figure 2.

An anticover can be constructed efficiently by the following "greedy furthest-point" algorithm.

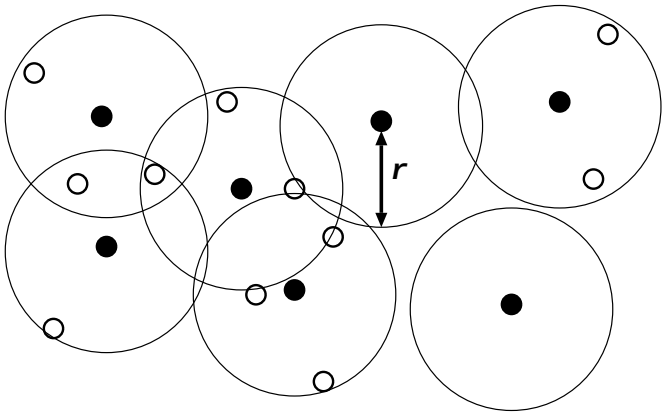

Figure 2: Anticover (black points) of size 7 of a Euclidean graph.

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

It is easy to see that the nodeset 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 [24], as well as the Euclidean k-clustering problem [11], indicating a level of universality of this approach and an applicability to multi-objective computing.

Theorem 2.1 *An anticover 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

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

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,

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

∎

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 2.1 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 2.2 *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 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 $2t$, while the distances between p_1 vertices is $2t(1 + \epsilon)$. 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.

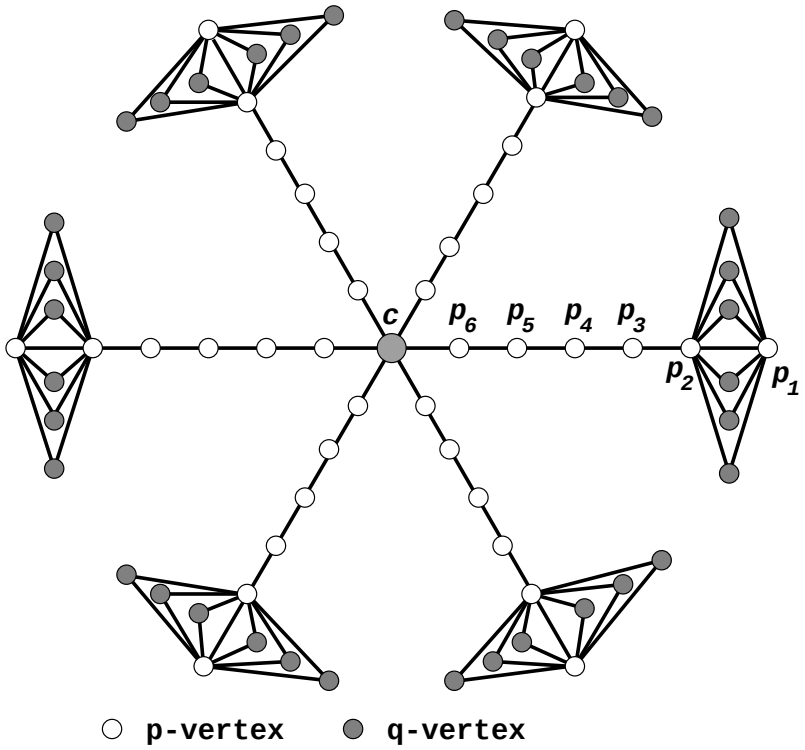

Figure 3: Lower bound example for Greedy.

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

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

∎

Although the above lower bound is applicable only to the solution generated by Greedy, we conjecture that 4, rather than lower bound of 2 obtained in Theorem 3.1, 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 r-anticover P, 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 (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 2.3 *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 3.2, 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 2.4 *The performance ratio of* Greedy *for the* REMOTE-ST *problem is at least* 2.4.

Proof. Let H be an edge-weighted graph with $V(H) = \{c, p_1, p_2, q_1, q_2, r\}$. Let $d(p_1, q_1) = d(p_1, q_2) = 2$, $d(p_1, r) = 1.5$, and $d(c, r) = d(r, p_2) = 0.5$, and let the distance between other pairs of vertices be the shortest distance within this tree. The hard instances H_z we construct, consist of z copies of H sharing the same c vertex, with distance between vertices of different copies determined by shortest distance.

Let $k = 2z$. Let P (Q) be the set of p_i (q_i) vertices, respectively. We may assume Greedy selects all the p_1 vertices, followed by the p_2 vertices, for a cost of $st(P) = (2 + 1/2)k/2 = 5/4k$. On the other hand, $st(Q) = 6/2k = 3k$. Hence, $\rho \geq st(Q)/st(P) = 2.4$. ∎

The best construction we have for REMOTE-ST (omitted) has approximation ratio of Greedy is $38/15 \approx 2.533$. Thus, 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 *at most* α.

HeavyEdge(G)
  Determine the largest α such that
    G_α is not $(n-k)$-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 the subgraph induced P 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,

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

Corollary 2.5 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 2.1.

Corollary 2.6 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 nodeset 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 $st(P) \geq \sqrt{3}mst(P)/2$ [8] in the Euclidean case, we immediately obtain the following.

Corollary 2.7 *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.

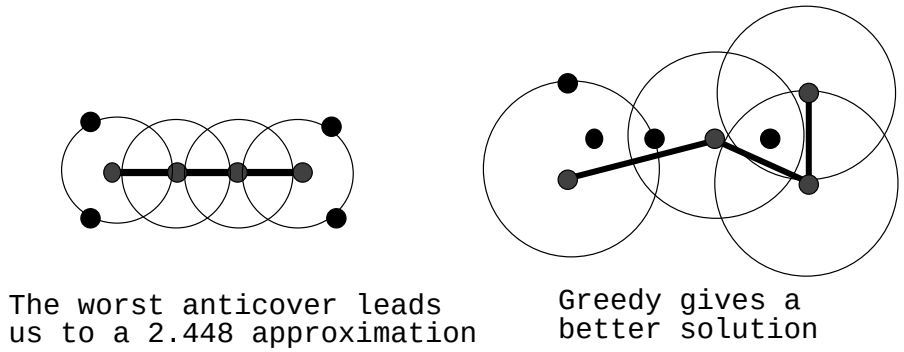

Figure 4: Approximation by circle covers.

Unlike in the metric case, it seems that the approximation ratio depends on the choice of the anticover. For the example in Figure 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.

Note: If we consider the anticover created by Greedy, we can obtain a slightly better analysis than Corollary 2.7. A key difference from the metric case is that the length of a minimum Steiner tree of m points in a unit circle must be less than $m - 0.1(m-3)$ if $m > 3$ for the Euclidean case

(we omit the proof). We can further squeeze the approximation ratio to $(1.95)(2/\sqrt{3}) \approx 2.2517$ for the REMOTE-MST problem by modifying the algorithm itself; indeed, this can be attained by modifying the output of Greedy. Unfortunately, the current proof requires a lengthy and ugly case-study, and we do not include it in this paper.

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. We assign ∞ to each edge of the complement graph T^c of T in the complete graph of order n, and define $G(T) = T \cup T^c$. We first give an $O(n)$ time algorithm for the REMOTE-ST k-set of $G(T)$.

Theorem 2.8 *The* REMOTE-ST *k-set of $G(T)$ can be computed in $O(n)$ time.*

Proof. Clearly, we should select k leaves. 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. If $k = 2$, the problem is the diameter path problem on a tree, and can be solved in linear time. We can apply the incremental strategy developed by Peng et al. [22] for computing a *k-tree core*. A key fact is that any optimal REMOTE-ST $(k-1)$-set must be contained in an optimal REMOTE-ST k-set. A direct modification of the algorithm of Peng et al. [22] runs in $O(\min\{kn, n \log n\})$ time, and the one of the improved algorithm of Shioura and Uno [25] runs in $O(n)$ time. ∎

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. If we modify the definition of the remote k-set P so that $mst(P)$ is maximized on the condition that $mst(P) \neq \infty$ (we call it connectivity condition), $MST(P)$ 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 *weighted $(k-1)$-cardinality tree* problem defined by Fischetti et al. [12] if we reverse the sign of all weights of T. We can thus apply Fischetti et al's $O(k^2 n)$ time dynamic programming algorithm. Moreover, we can improve it to $O(kn)$.

Theorem 2.9 *The 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 optimal REMOTE-MST k-set. For simplicity, we assume that T is a rooted binary tree (we can easily modify the algorithm for non-binary trees). We cut a rooted tree T at the nearest branch v to the root r, and obtain two subtrees T_1 and T_2. $T_1 \cup T_1 = T$ and $T_1 \cap T_2 = \{r\}$. Suppose we have the optimal REMOTE-MST k-set and the optimal REMOTE-MST j-sets containing v, for $j = 1, 2, \ldots, k$ of each of T_1 and T_2. Then, the optimal REMOTE-MST k-set of T, together with the optimal REMOTE-MST j-set containing r for $j = 1, 2, \ldots, k$, can be computed in $O(k^2)$ time, by combining those of T_1 and T_2.

We improve this time complexity as follows: We say that a node u of T is *heavy* if both of its descendent trees have at least $k/2$ nodes, and *light* otherwise. The number of heavy

nodes is at most n/k. We separately charge the cost of the operations at the heavy nodes, which is $O(kn)$ in total. Let $f(n)$ be the cost for operations at all light nodes of T. At a light node, suppose that the size of T_i is n_i. Then, the computing time at the node is $O(\min(n_1, k)\min(n_2, k))$. Thus, the cost function $f(n)$ (up to a constant factor) follows the formula $f(n) \leq f(n_1) + f(n_2) + \min(n_1, k)\min(n_2, k)$.

We can see that $g(n) = \min\{2kn, n^2\}$ satisfies $g(n) \geq g(n_1) + g(n_2) + \min(n_1, k)\min(n_2, k)$.

Case 1: If $2k \geq n$, the formula follows from $n^2 \geq n_1^2 + n_2^2 + n_1 n_2$.

Case 2: If $n \geq 2k \geq n_1 \geq k > n_2$, $2kn \geq n_1^2 + n_2^2 + kn_2$ easily follows.

Case 3: If $n_1 \geq 2k$ and $k > n_2$, $2kn \geq 2kn_1 + n_2^2 + kn_2 = 2k(n_1 + n_2) - n_2(k - n^2)$.

Hence, $f(n) < cg(n)$ for some constant c, thus 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.

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 5.

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

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

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

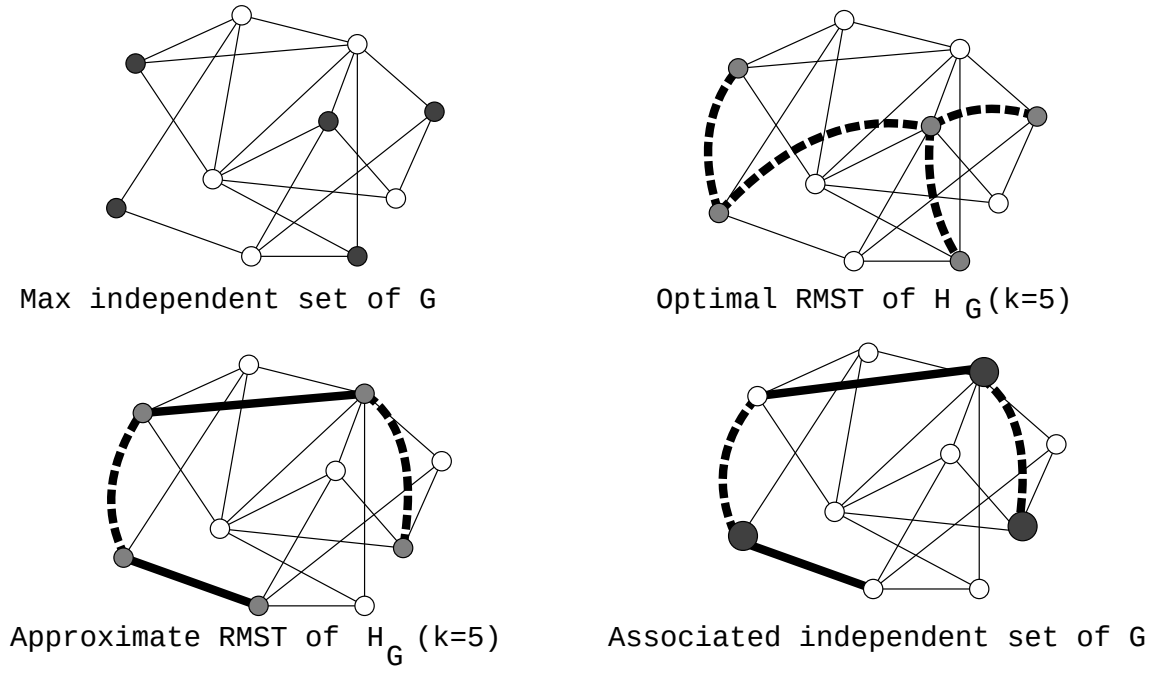

Figure 5: Graphs in Theorem 3.1.

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

We now generalize the hardness proof for REMOTE-MST to other problems. Given a graph and integers ℓ, u, the *degree-constrained subgraph problem (DCS)* is to find a subgraph of minimum weight such that the degree of each vertex is between ℓ and u. Note that u could possibly be trivially bounded by n. This minimization problem can be solved via a reduction to non-bipartite matching [17], and it subsumes the assignment problem, and the problems of covering the vertices with cycles or with paths. If the subgraph must additionally be connected, we have a *Connected-DCS (CDCS) problem*, which includes TSP, MST, and the Degree-Constrained MST problems as special cases.

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 notations 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. We can assume $|Heavy| > \ell^2 + 1$ without loss of generality. Also, the number k of vertices in $\Pi(H)$ is greater than any constant power of ℓ.

We start with some definitions. If Π is a problem requiring connectivity, let $Conn$ denote some minimal set of edges from $Heavy$ such that $Conn \cup (\Pi(H) - Heavy)$ is connected and spans H; otherwise, let $Conn$ be the emptyset. Let $Slack$ denote the set of vertices incident on edges in $Heavy - Conn$. Recall that $G[Slack]$ is the subgraph of G induced by the vertices $Slack$.

We first observe that

$$\alpha(G) \geq |Conn| + 1 \tag{2}$$

since G must contain that many connected components. Thus, we may assume that $Heavy - Conn$ contains at least ℓ^2 edges.

Claim 1 *Each vertex is incident on at most ℓ edges in $Heavy - Conn$.*

Suppose on the contrary that a vertex x was incident on $\ell+1$ or more edges in $Heavy-Conn$. All of its neighbors must be of degree ℓ, as otherwise an edge would be redundant. Thus, there exist two non-adjacent neighbors y and z (along heavy edges). Notice that x is additionally incident on at least one edge in $\Pi(H) - (Heavy - Conn)$, thus its degree is at least $\ell + 2$. Let $\Pi' = (\Pi(H) - \{(x, y), (x, z)\}) \cup \{(y, z)\}$ and observe that Π' is a valid solution to the Π-problem on H: the degree condition of the vertices holds, and connectivity is not affected, since the removed edges are from $Heavy - Conn$. Since Π' is of less cost, this contradicts the assumption that Π is a minimum cost Π-structure on H.

Claim 2 $E(G[Slack]) \subseteq \Pi(H)$

Suppose on the contrary that there were vertices x, y in $Slack$ 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'. 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 $\ell + 1$ vertex of degree ℓ).

Otherwise, the number of heavy edges to which both x' and y' are incident or adjacent 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. Also, the degree constraints are preserved. Hence, Π' is a valid solution of lesser cost, contradicting the minimality of Π. The claim then follows.

From these claims, we have that the neighborhood of each vertex in $Slack$ is incident on at most ℓ^2 edges from $Heavy - Conn$. Thus, a greedy creation of a maximal independent set eliminates at most ℓ^2 edges in each step.

$$\alpha(G) \geq \alpha(G[Slack]) \geq \frac{|Heavy - Conn|}{\ell^2} \tag{3}$$

Combining (2) and (3) , we have that

$$\alpha(G) \geq \max\left(\frac{|Heavy - Conn|}{\ell^2}, |Conn|\right) \geq \frac{|Heavy|}{\ell+1}.$$

∎

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 $H = H_{G,\gamma}$.

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

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

∎

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* $\ell = u$*, or a connected property with* $\ell = 1$ *(i.e.* Degree-Constrained MST*). Then, 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,

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

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 [21] under the name *k-Defense problem*. The above reduction shows that approximating it within $n^{1-\epsilon}$ in general graphs is hard.

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

Proof. Given 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), 38–56.

[2] A. Aggarwal, M. Klawe, S. Moran, P. Shor, and R. Wilber. Geometric Applications of a Matrix-Searching Algorithm. *Algorithmica* **2** (1987), 195–208.

[3] A. Aggarwal, B. Schieber, and T. Tokuyama. Finding a Minimum Weight K-link Path in Graphs with Monge Property and Applications. *Discrete and Computational Geometry* 12 (1994), 263–280.

[4] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and Hardness of Approximation Problems. In *Proc. 33rd IEEE FOCS* (1992), 14–23.

[5] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama Greedily Finding a Dense Subgraph In *Proc. 5th SWAT* (1996), Springer LNCS 1097, pp. 136-145.

[6] J. Boyce, D. Dobkin, R. Drysdale, and L. Guibas. Finding Extremal Polygons. *SIAM J. on Computing* **14** (1985), 134–147.

[7] B. Chandra and M. Halldórsson. Facility Dispersion and Remote Subgraphs. In *Proc. Fifth SWAT* (1996), Springer LNCS #1097, 53–65.

[8] D.-Z. Du and F. K. Hwang. An Approach for Proving Lower Bounds: Solution of Gilbert-Pollak's Conjecture of Steiner Ratio. In *Proc. 31st IEEE FOCS* (1990), 76–85.

[9] D. Eppstein. New Algorithms for Minimum Area k-gons. In *Proc. 3rd ACM-SIAM SODA* (1992), 83–87.

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

[11] T. Feder and D. H. Greene. Optimal Algorithms for Approximate Clustering. In *Proc. 20th ACM STOC* (1988), 434–444.

[12] M. Fischetti, H. W. Hamacher, K. Jørnsten, and F. Maffioli. Weighted K-Cardinality Trees: Complexity and Polyhedral Structure. *Networks* **24** (1994), 11–21.

[13] N. Garg. A 3-Approximation of the Minimum Tree Spanning k Vertices. To appear in *Proc. 37th IEEE FOCS* (1996).

[14] R. Hassin, S. Rubinstein, and A. Tamir. Approximation Algorithms for Maximum Facility Dispersion. Manuscript, August 1996.

[15] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. To appear in *Proc. 37th IEEE FOCS* (1996).

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

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

[18] S. Misono and K. Iwano. Circuit Board Drilling Problem. Technical Report, Inf. Proc. Soc. Japan, 93-AL-33 (1993), 95–102.

[19] S. Misono and K. Iwano, Experiments on TSP Real Instances. IBM Tokyo Research Laboratory, Research Report, RT0153 (1996).

[20] J. Mitchell. *Manuscript*, 1996.

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

[22] S. Peng, A. B. Stephens, and Y. Yesha. Algorithms for a Core and k-Tree Core of a Tree. *J. Algorithms* **15** (1993), 143–159.

[23] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrants, and S. S. Ravi. Spanning Tree Short or Small. *SIAM J. Disc. Math* **9**:2 (1996), 178–200.

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

[25] A. Shioura and T. Uno. A Linear Time Algorithm for the k-Tree Core of a Tree. *Preprint* (1994).

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

[27] A. Tamir. Comments on the paper 'Facility dispersion problems: Heuristics and special cases, by S.S. Ravi, D.J. Rosenkrantz, and G.K. Tayi'. To appear in *Operations Research*, 1996.

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