# Complexity and Approximability of Bicriteria Compact Location Problems [4]

*Sven O. Krumke* [1] *H. Noltemeier* [1] *S. S. Ravi* [2] *Madhav V. Marathe* [3]

November 29, 1995

**Abstract**

We investigate the complexity and approximability of some location problems when two distance values are specified for each pair of potential sites. These problems involve the selection of a specified number of facilities (i.e., a placement of a specified size) to minimize a function of one distance metric subject to a budget constraint on the other distance metric. Such problems arise in several application areas including statistical clustering, pattern recognition and load–balancing in distributed systems. We show that, in general, obtaining placements that are near-optimal with respect to the first distance metric is $\mathcal{NP}$–hard even when we allow the budget constraint on the second distance metric to be violated by a constant factor. However, when both the distance metrics satisfy the triangle inequality, we present approximation algorithms that produce placements which are near-optimal with respect to the first distance metric while violating the budget constraint only by a small constant factor.

**Keywords**: Location Theory, Approximation Algorithms, Computational Complexity, $\mathcal{NP}$–hardness.

**AMS(MOS) subject classification.** 68R10, 68Q15, 68Q25.

[1] University of Würzburg, Am Hubland, 97074 Würzburg, Germany. Email: {krumke, noltemei}@informatik.uni-wuerzburg.de

[2] Department of Computer Science, University at Albany - SUNY, Albany, NY 12222. Email: ravi@cs.albany.edu.

[3] Los Alamos National Laboratory P.O. Box 1663, MS K990, Los Alamos NM 87545. Email: madhav@c3.lanl.gov. The work is supported by the Department of Energy under Contract W-7405-ENG-36.

[4] Parts of the paper appear in two conference papers, namely [KN+95a] and [KN+95b].

# 1 Introduction and motivation

Several fundamental problems in location theory [HM79, MF90] involve finding a placement of facilities obeying certain "covering" constraints. Generally, the goal of such a location problem is to find a placement of minimum cost that satisfies all the specified constraints. The cost of a placement may be specified in several ways including the total cost of all the edges included in the placement (i.e., the total cost of constructing the network of facilities) and the maximum cost of any edge in the placement (i.e., the *diameter*, which models the maximum communication cost between any pair of facilities). Finding a placement of sufficient generality minimizing even one of these measures is often $\mathcal{NP}$–hard [GJ79].

We address a class of location problems, known as *compact location* problems [RKM$^+$93]. The goal of such a problem is to find a "compact" placement of facilities. More precisely, the goal is to find a placement that minimizes some measure of the distance between the selected nodes. In practice, it is often the case that a minimum cost placement must be chosen subject to budget constraints on other cost measures. We consider problems where the goal is to find a placement that minimizes one cost measure subject to a budget constraint on another cost measure. For this reason, we refer to these problems as *bicriteria compact location problems.*

In graph theoretic terms, bicriteria compact location problems can be formalized as follows. Suppose we are given *two* weight–functions $c, d$ on the edges of the network. (For example, the first weight function $c$ may represent the cost of constructing an edge, and the second weight function $d$ may represent the actual transportation– or communication–cost over an edge once it has been constructed). Given such a graph, we define a general bicriteria problem $(\mathcal{A}, \mathcal{B})$ by identifying two minimization objectives of interest from a set of possible objectives. A budget value is specified on the second objective $\mathcal{B}$ and the goal is to find a placement of facilities having minimum possible value for the first objective $\mathcal{A}$ such that this solution obeys the budget constraint on the second objective. For example, consider the *Diameter–Constrained Minimum Diameter Problem* denoted by DC–MDP: Given an undirected complete graph $G = (V, E)$ with two nonnegative integral edge weight functions $c$ (modeling the building cost) and $d$ (modeling the delay or the communication cost), an integer $p$ denoting the number of facilities to be placed, and an integral bound $B$ (on the delay), find a placement of $p$ facilities with minimum diameter under the $c$–cost such that the diameter of the placement under the $d$–costs (the maximum delay between any pair of nodes) is at most $B$.

Bicriteria compact location problems arise in a number of applications such as allocation of manufacturing sites for the components of a system to minimize the cost of distributing raw materials and the cost of transporting components, distributing the activities of a project among geographically dispersed offices so as to minimize the transportation and communi-

cation costs among the offices, etc. An example of a natural bicriteria compact location problem that arises in the area of high performance computing is the following. Consider a computational task consisting of a number of communicating subtasks. At a given time, some of the processing elements may be already allocated and the remaining processing elements are available. The problem is to select a subset of processors from the currently available processors, one per subtask. Processors communicate with each other to exchange data, and these data communication delays increase the time needed to complete the task. Further, the computation may require the processors to be synchronized often, thus adding to the time needed to complete the task. Therefore, it is desirable to select a subset of processors so that the total communication cost among the processors is minimized and the maximum delay due to synchronization during the computation does not exceed a given bound. Bicriteria compact location problems also arise in other areas such as statistical clustering, pattern recognition, load–balancing in distributed systems, etc. (see [HM79, MF90, KN$^+$95a] and the references cited therein).

The remainder of the paper is organized as follows. Section 2 contains preliminaries and the formal definitions of the problems considered in this paper. In Section 3 we summarize the results obtained. In Section 4 we discuss the related research done in this area. Section 5 contains non-approximability results for graphs with arbitrary weights. Section 6 contains our approximation algorithms for the diameter constrained problems. In Section 7 we outline the approximation algorithms for the sum constrained problems. Section 8 discusses some extensions of our approximation algorithms. Finally, Section 9 contains concluding remarks and directions for future research.

## 2 Preliminaries and problem definitions

We consider a complete undirected $n$–vertex graph $G = (V, E)$ with two distance functions $c$ and $d$ specified on the edges. Given an integer $p$, a *placement* $P$ is a subset of $V$ with $|P| = p$. The *set of neighbors* of a vertex $v$ in $G$, denoted by $N(v, G)$, is defined by $N(v, G) := \{w : (v, w) \in E\}$. The *degree* $\deg(v, G)$ of $v$ in $G$ is the number of vertices in $N(v, G)$. For a subset $V' \subseteq V$ of nodes, we denote by $G[V']$ the subgraph of $G$ induced by $V'$. Given a graph $G = (V, E)$, the graph $G^2 = (V, E^2)$ is defined by $(u, v) \in E^2$ if and only if there is a path in $G$ between $u$ and $v$ consisting of at most two edges.

If the edge distances are allowed to be zero, then the optimal solution value may be zero. In a such case, obtaining a solution whose value is within some factor of the optimal solution value is trivially equivalent to finding an optimal solution itself. Therefore, we assume that the values of both the distance functions for any edge are strictly *positive*.

With $\delta \in \{c, d\}$ denoting one of the two edge–weight functions, we use $\mathcal{D}_\delta(P)$ to denote the

*diameter* and $\mathcal{S}_\delta(P)$ to denote the *sum of the distances* between the nodes in the placement $P$; that is

$$\mathcal{D}_\delta(P) = \max_{\substack{e=(v,w) \\ v,w \in P}} \delta(v,w) \qquad \text{and} \qquad \mathcal{S}_\delta(P) = \sum_{\substack{e=(v,w) \\ v,w \in P}} \delta(v,w).$$

Note that the average length of an edge in a placement $P$ equals $\frac{2}{p(p-1)}\mathcal{S}_\delta(P)$. Since the average length of an edge in a placement differs from the total length of all the edges in the placement by only the scaling factor $\frac{2}{p(p-1)}$, finding a placement of minimum average length is equivalent to finding a placement of minimum total length. We use this fact throughout this paper. For the rest of the paper we use $c$-cost of an edge $(u,v)$ to denote $c(u,v)$ and use $d$-cost of an edge $(u,v)$ to denote $d(u,v)$.

As is standard in the literature, we say that a nonnegative edge–weight function $\delta \in \{c,d\}$ satisfies the *triangle inequality*, if we have $\delta(v,w) \leq \delta(v,u) + \delta(u,w)$ for all $v,w,u \in V$. Following [HS86], the *bottleneck graph* bottleneck$(G,\delta,\Delta)$ of $G=(V,E)$ with respect to $\delta$ and a bound $\Delta$ is defined by

$$\text{bottleneck}(G,\delta,\Delta) := (V,E'), \text{ where } E' := \{e \in E : \delta(e) \leq \Delta\}.$$

We now define the problems studied in this paper.

**Definition 2.1** [*Diameter Constrained Minimum Average Placement Problem* (DC–MAP)]

*Input:* An undirected complete graph $G=(V,E)$ with two positive edge weight functions $c,d : E \to \mathbb{Q}^+$, an integer $2 \leq p \leq n$ and a number $\Omega \in \mathbb{Q}^+$ .
*Output:* A set $P \subseteq V$, with $|P| = p$, minimizing the objective

$$\mathcal{S}_c(P) = \sum_{\substack{e=(v,w) \\ v,w \in P}} c(v,w)$$

subject to the constraint

$$\mathcal{D}_d(P) = \max_{\substack{e=(v,w) \\ v,w \in P}} d(v,w) \leq \Omega.$$

**Definition 2.2** [*Diameter Constrained Minimum Diameter Placement Problem* (DC-MDP)]

*Input:* Same as in DC–MAP above.
*Output:* A set $P \subseteq V$, with $|P| = p$, minimizing the objective

$$\mathcal{D}_c(P) = \max_{\substack{e=(v,w) \\ v,w \in P}} c(v,w)$$

subject to the constraint

$$\mathcal{D}_d(P) = \max_{\substack{e=(v,w) \\ v,w \in P}} d(v,w) \leq \Omega.$$

**Definition 2.3** [*Sum Constrained Minimum Diameter Placement Problem* (SC-MDP)]

*Input:* Same as in DC–MAP above.
*Output:* A set $P \subseteq V$, with $|P| = p$, minimizing the objective

$$\mathcal{D}_c(P) = \max_{\substack{e=(v,w) \\ v,w \in P}} c(v,w)$$

and satisfying the budget–constraint

$$\mathcal{S}_d(P) = \sum_{\substack{e=(v,w) \\ v,w \in P}} c(v_i, v_j) \leq \Omega.$$

**Definition 2.4** [*Sum Constrained Minimum Average Placement Problem* (SC–MAP)]

*Input:* Same as in DC–MAP above.
*Output:* A set $P \subseteq V$, with $|P| = p$, minimizing the objective

$$\mathcal{S}_c(P) = \sum_{\substack{e=(v,w) \\ v,w \in P}} c(v,w)$$

subject to the constraint

$$\mathcal{S}_d(P) = \sum_{\substack{e=(v,w) \\ v,w \in P}} d(v,w) \leq \Omega.$$

Given a problem $\Pi$, we use TI-$\Pi$ to denote the problem $\Pi$ restricted to graphs in which both the edge weight functions satisfy the triangle inequality.

Let $\Pi \in \{$DC–MDP, SC–MDP, SC–MAP, DC–MAP, TI–DC–MDP, TI–SC–MDP, TI–DC–MAP, TI–SC–MAP$\}$. One of the goals of our work is to find good approximation algorithms for several of the bicriteria compact location problems introduced here. By an approximation algorithm (heuristic) we mean a *polynomial time* algorithm which produces feasible, but not necessarily optimal, solutions. A **relative** approximation algorithm guarantees a solution which is within a *multiplicative* constant $K$ of the optimal value for every instance of the problem. The multiplicative constant $K$ is referred to as the **performance guarantee** provided by the algorithm. This paper is concerned with the study of relative approximation algorithms for the placement problems defined above. Define an $(\alpha, \beta)$*–approximation algorithm* for $\Pi$ to be a polynomial–time algorithm, which for any instance $I$ of $\Pi$ does one of the following:

(a) It produces a solution within $\alpha$ times the optimal value with respect to the first distance function $c$, violating the constraint with respect to the second distance function $d$ by a factor of at most $\beta$.

(b) It returns the information that no feasible placement exists at all.

Notice that if there is no feasible placement but there is a placement violating the constraint by a factor of at most $\beta$, an $(\alpha, \beta)$–approximation algorithm has the choice of performing either action (a) or (b).

## 3 Summary of results

Here, for the first time in the literature, we study the complexity and approximability of several bicriteria compact location problems. We present both $\mathcal{NP}$–hardness results and approximation algorithms with provable performance guarantees for several bicriteria compact location problems.

As shown in [RKM+93], most of the compact location problems studied in this paper are $\mathcal{NP}$–hard, even without additional constraints. We show that, in general, obtaining placements that are near-optimal with respect to the first cost metric is $\mathcal{NP}$–hard even when we allow the budget constraint on the second cost metric to be violated by a constant factor. These hardness results hold even when one of the cost functions satisfies the triangle inequality.

Given the hardness of approximating the objective function even when one of the cost functions satisfies the triangle inequality we focus our attention on graphs in which *both* the weight functions satisfy the triangle inequality. In the past, a substantial amount of work has been carried out in investigating the approximability of problems arising in network design and location theory when edge weights satisfy the triangle inequality (for example see [RSL77, HS86, RRT94]). We refer the reader to the paper by Bern and Eppstein [BE95] for a comprehensive survey of other geometric location problems. We obtain the following results.

1. There is a polynomial time approximation algorithm for TI-DC-MDP with performance guarantee $(2, 2)$. Furthermore, unless $\mathcal{P} \neq \mathcal{NP}$, no polynomial time approximation algorithm can provide a performance guarantee of $(2-\varepsilon, 2)$ or $(2, 2-\varepsilon)$, for any $\varepsilon > 0$.

2. There is a polynomial time approximation algorithm for TI-DC-MAP with performance guarantee $(2-\frac{2}{p}, 2)$. Furthermore, unless $\mathcal{P} \neq \mathcal{NP}$, no polynomial time approximation algorithm can provide a performance guarantee of $(\alpha, 2-\varepsilon)$, for any $\alpha \geq 1, \varepsilon > 0$.

3. There is a polynomial time approximation algorithm for TI-SC-MDP, with performance guarantee $(2, 2-\frac{2}{p})$. Furthermore, unless $\mathcal{P} \neq \mathcal{NP}$, no polynomial time approximation algorithm can provide a performance guarantee of $(2-\varepsilon, \alpha)$, for any $\alpha \geq 1, \varepsilon > 0$.

4. For all $\gamma > 0$, there is a polynomial time approximation algorithm for TI-SC-MAP with performance guarantee $((1+\gamma)(2-\frac{2}{p}), (1+\frac{1}{\gamma})(2-\frac{2}{p}))$.

As can be seen, the performance guarantees provided by our approximation algorithms are best possible or nearly the best possible in most cases, under the assumption that $\mathcal{P} \neq \mathcal{NP}$. All the four results above can be extended to the case when there weights on vertices. They

can also be extended to the case where we are given a subset $D \subseteq V$ that must be included in the solution. We discuss these extensions in Section 8.

Our results are based on two basic techniques. The first is a combination of two ideas, namely the *power of graphs* approach of Hochbaum and Shmoys [HS86] and the local search approach for approximating single criteria compact location problems introduced in [KN+95a]. The second is an application of a *parametric search technique* discussed in [MR+95] for network design problems.

## 4 Related work

As mentioned earlier, problems involving the placement of $p$ facilities so as to minimize suitable cost measures have been studied extensively in the literature (see [BE95, RKM+93, AI+91] and the references cited therein). These problems can roughly be divided into two main categories. The first category of problems involves selecting a set of $p$ facilities so as to minimize (or maximize) the cost (distance) from the unselected sites to the selected sites. Problems that can be cast in this framework include the $p$–center problem [HS86, DF85], the $p$–cluster problem [HS86, FG88, Go85] and the $p$–median problem [LV92, MF90]. The second category consists of problems where the goal is to select $p$ facilities so as to optimize a certain cost measure defined on the set of selected facilities. Problems that can be cast in this framework include the $p$–dispersion problem [RRT94, Ta91, EN89], the $p$–minimum spanning tree problem [RR+94, GH94, AA+94, BCV95] and the $p$–compact location problem [RKM+93, AI+91, KN+95a].

In contrast, not much work has been done in finding optimal location of facilities when there is more than one constraint. A notable work in this direction is by Bar-Ilan, Kortsarz and Peleg [BKP93] who considered the problem of assigning network centers, with a bound imposed on the number of nodes that any center can service. We refer the reader to [MR+95, RMR+93] for a survey of the work done in the area of algorithms for bicriteria network design and location theory problems.

## 5 Hardness results

In this section, we prove our non-approximability results. The hardness of decision versions of this follows directly from the hardness results in [RKM+93]. Several of our hardness results use the recent hardness results concerning approximability of the MAX-CLIQUE problem. We recall the definition and the related non-approximability result for the sake of completeness.

**Definition 5.1** An instance of the CLIQUE problem consists of an undirected graph $G(V, E)$ and an integer $J \leq |V|$. The question is whether there exists a clique of size $J$ in $G$; that is,

whether there exist $J$ vertices $S = \{v_1, \ldots, v_J\}$ such that $\forall v_i, v_k \in S$ we have $(v_i, v_k) \in E$.

The optimization version of the problem, denoted by MAX-CLIQUE, is to find a clique of maximum size in $G$.

Recently, using a new characterization of $\mathcal{NP}$ in terms of probabilistically checkable proof systems, it has been shown that the MAX CLIQUE problem cannot be efficiently approximated. We state the most recent result obtained by Bellare and Sudan [BS94].

**Theorem 5.2** Unless $\mathcal{P} = \mathcal{NP}$, for all $\varepsilon > 0$ the problem MAX-CLIQUE does not have a polynomial time approximation algorithm with performance guarantee $n^{1/6-\varepsilon}$. ■

We begin with a non–approximability result for TI–DC–MDP.

**Lemma 5.3** For all $\varepsilon, \varepsilon' > 0$, the following statement is true: If there is a polynomial time algorithm A with the property that given any instance of TI–DC–MDP, the algorithm A

- either returns a subset $S \subseteq V$ of at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$ nodes satisfying $\mathcal{D}_d(S) \leq (2-\varepsilon)\Omega$,
- or provides the information that a placement of $p$ nodes having $d$–cost of at most $\Omega$ does not exist,

then $\mathcal{P} = \mathcal{NP}$.

**Proof:** Let $I'$ be an arbitrary instance of MAX–CLIQUE, given by a graph $G' = (V', E')$. Without loss of generality we can assume that $E' \neq \emptyset$. We will give a many-one Turing reduction to the problem TI–DC–MDP.

For each $2 \leq k \leq |V|$, we construct an instance $I^{(k)}$ of TI–DC–MDP as follows: Let $G^{(k)} = (V', E_c)$ ($E_c = \{(u, v) : u, v \in V, u \neq v\}$) and define $c^{(k)}, d^{(k)} : E_c \to \mathbb{N}$ by $c^{(k)}(e) := 1$ for all $e \in E_c$ and

$$d^{(k)}(e) := \begin{cases} 1 & \text{if } e \in E' \\ 2 & \text{else} \end{cases}$$

It is trivial to see that both weight functions obey the triangle inequality. We let $\Omega^{(k)} := 1$ and $p^{(k)} := k$. Notice that the size of an instance $I^{(k)}$ is still polynomial in the size of $I'$, and that we have constructed only polynomially many (namely $O(|V|)$) instances.

Now consider an instance $I^{(k)}$. Note that any placement $P$ of $p^{(k)} = k$ nodes that has diameter $\mathcal{D}_{d^{(k)}}(P) \leq (2-\varepsilon)\Omega = (2-\varepsilon)$ must have diameter 1. Also, any subset $S \subseteq V$ having diameter $\mathcal{D}_{d^{(k)}}(S) = 1$ must form a clique in the original graph $G'$.

Assume that the original graph $G'$ has a clique $C$ of size $p^{(k)} = k$. Then this clique will satisfy $\mathcal{D}_{c^{(k)}}(C) = \mathcal{D}_{d^{(k)}}(C) = 1 = \Omega'$. By our assumption, the algorithm A must return a set

$S$ of at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$ nodes with $d$-diameter at most $(2-\varepsilon)\Omega = (2-\varepsilon) < 2$. Thus, as noted above, the algorithm must find a placement of diameter 1, and this set will form a clique in the original graph $G'$.

If there is no clique of size $p^{(k)} = k$ in $G'$, any placement $P$ of $p^{(k)} = k$ nodes in $G'$ must include at least one edge $e$ of length $d^{(k)}(e) = 2 > (2-\varepsilon)$. Now, according to our assumptions about A, the algorithm has the choice of either returning a set of size at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$ that will form a clique in the original graph or providing the information that there is no placement $P$ of diameter at most $\Omega = 1$.

Thus, the output of the algorithm A can be used to either obtain the information that $G'$ does not contain a clique of size $p^{(k)} = k$ or that $G'$ does have a clique of size at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$.

Now, we run A for all the instances $I^{(k)}$ $(2 \leq k \leq |V|)$. Since the size of each instance $I^{(k)}$ is polynomial in the size of $I'$ and we only have $O(|V|)$ instances, this will result in an overall polynomial time algorithm, according to our assumptions about A. Let $m := \max\{k : \text{A returns a set } S \text{ of diameter } 1\}$. Then, by our observations from above, we can conclude that $G'$ has a clique of size at least $\frac{2m}{|V|^{1/6-\varepsilon'}}$ and that there is no clique of size $m+1$ in $G'$. Hence, we can approximate the maximum clique number of $G'$ by a factor of at most $\frac{m+1}{2m} \cdot |V|^{1/6-\varepsilon'} \leq |V|^{1/6-\varepsilon'}$. By the results in [BS94] this implies $\mathcal{P} = \mathcal{NP}$. ■

We can interchange the roles of $c$ and $d$ in the proof of the last proposition to show that the optimal value of the TI-DC-MDP problem cannot be approximated by a factor of $(2-\varepsilon)$. Moreover, replacing 2 by a suitable function $f \in \Theta(2^{\text{poly}(|V|)})$, which given an input length of $\Omega(|V|)$ is polynomial time computable, it can be seen that, if the triangle inequality is not required to hold, there can be no polynomial time approximation with performance ratio $O(2^{\text{poly}(|V|)})$ for either the optimal function value or the constraint (modulo $\mathcal{P} = \mathcal{NP}$). Thus we obtain:

**Theorem 5.4** Unless $\mathcal{P} = \mathcal{NP}$, for all $\varepsilon > 0, \varepsilon' > 0$ the problem TI–DC–MDP does not have a polynomial time approximation algorithm that finds a placement of at least $2p/|V|^{1/6-\varepsilon'}$ facilities and which has a performance guarantee of $(\alpha, 2-\varepsilon)$ or $(2-\varepsilon, \beta)$.

Unless $\mathcal{P} = \mathcal{NP}$, for any $f, g \in O(2^{\text{poly}(|V|)})$, if the $c$-costs and $d$-costs do not satisfy triangle inequality, then the problem TI–DC–MDP does not have a polynomial time $(f(|V|), g(|V|))$–approximation algorithm. ■

The results of Lemma 5.3 also give a slightly stronger hardness result than the one stated in the last Theorem. Namely, we can conclude that, for any fixed $\alpha, \beta \geq 1$, unless *both* the

$c$–costs and the $d$–costs satisfy the triangle inequality, we can not obtain a polynomial time $(\alpha, \beta)$–approximation algorithm for the problem DC-MDP.

We now turn to TI–DC–MAP and DC–MAP problems.

**Theorem 5.5** Unless $\mathcal{P} = \mathcal{NP}$, any fixed $\alpha, \beta \geq 1$, if the $c$-costs and the $d$-costs do not satisfy triangle inequality, the problem DC–MAP does not have a polynomial time $(\alpha, \beta)$–approximation algorithm.

Unless $\mathcal{P} = \mathcal{NP}$, for any fixed $\alpha \geq 1$ and $\varepsilon > 0$, the problem TI–DC–MAP does not have a polynomial time $(\alpha, 2 - \varepsilon)$–approximation algorithm.

**Proof:** We first consider the DC–MAP problem. Suppose there is a polynomial approximation algorithm A with a performance guarantee of $(\alpha, \beta)$ for some $\alpha, \beta \geq 1$. We will show that A can be used to solve an arbitrary instance of the CLIQUE problem in polynomial time, contradicting the assumption that $\mathcal{P} \neq \mathcal{NP}$.

Let the graph $G = (V, E)$ and the integer $J$ form an arbitrary instance of CLIQUE. Construct the following instance $I$ of DC–MAP. The vertex set for $I$ is $V$ itself. For all $u, v \in V$ $(u \neq v)$, let $c(u, v) = 1$; also, let $d(u, v) = 1$ if $(u, v) \in E$ and $d(u, v) = \beta + 1$ otherwise. Finally set $p = J$ to complete the construction. In the remainder of this proof, we refer to any edge in the instance $I$ with $d$ value equal to $\beta + 1$ as a *long* edge; other edges are referred to as *short* edges.

If $G$ has a clique of size $J$, then the nodes which form this clique constitute an optimal solution to the DC–MAP instance $I$ with sum (under $c$–cost) equal to $J(J-1)/2$ and diameter (under $d$–cost) equal to 1. Since $\mathcal{A}$ provides a performance guarantee of $(\alpha, \beta)$, the solution returned by A cannot include any long edges. If $G$ does not have a clique of size $J$, then every subset of $J$ nodes must include at least one long edge. Therefore, by merely examining the solution produced by A, we can solve the CLIQUE problem.

We use the same construction for TI–DC–MAP except that for every long edge, the $d$ value is chosen as 2. This modification ensures that both the cost functions satisfy the triangle inequality. ■

Again, the proof of Theorem 5.5 can be used to show that, if only one of the cost functions $c$ and $d$ does not obey the triangle inequality, no polynomial time $(\alpha, \beta)$–approximation for any fixed $\alpha, \beta \geq 1$ can exist, unless $\mathcal{P} = \mathcal{NP}$.

Using ideas similar to those used in the proof of Theorems 5.4 and 5.5, the following hardness results for the sum–constrained problems (i.e., SC–MDP and SC–MDP) can be obtained.

**Theorem 5.6** Unless $\mathcal{P} = \mathcal{NP}$, for all fixed $\alpha, \beta \geq 1$, if the $c$- and $d$-costs do not satisfy triangle inequality, the problems SC–MAP and SC–MDP do not have a polynomial time $(\alpha, \beta)$–

approximation algorithms.

Unless $\mathcal{P} = \mathcal{NP}$, for all fixed $\varepsilon > 0$ and $\beta \geq 1$, for the problem TI–SC–MDP does not have a polynomial time $(2 - \varepsilon, \beta)$–approximation algorithm. ■

# 6 Approximation algorithms for diameter constrained problems

## 6.1 Basic technique

We discuss the basic technique used in approximating the diameter constraint compact location problems. Hochbaum and Shmoys [HS86] introduced a general approach for approximating a number of minimum bottleneck problems when the costs satisfy triangle inequality. We combine this approach with the local search heuristic for compact location problem developed in [KN$^+$95a] to develop a general approach for approximating the diameter constraint compact location problems.

To illustrate our ideas, consider the problem TI–DC–MAP. Let $\Omega$ be the bound on diameter of the placement with respect to $d$-cost. It is clear that in the subgraph induced by an optimal placement no edge has $d$-cost more than $\Omega$. Thus we can prune all the edges in the graph with $d$-costs more than $\Omega$. Since finding an optimal placement that minimizes the the sum of distances with respect to $c$-cost is still "hard" in the modified graph, we resort to finding an near optimal placement. Because edge weights satisfy triangle inequality, it follows that the placement does not violate the constraint by a factor of more than 2.

Although the basic idea is simple, the method yields near tight bounds on the performance guarantees of the resulting algorithms. We believe that this approach would be useful in approximating other bicriteria minimization problems when the weights obey the triangle inequality.

## 6.2 Approximating TI–DC–MAP

We present an approximation algorithm for the problem TI–DC–MAP, that provides a performance guarantee of $(2 - 2/p, 2)$. The algorithm is shown in Figure 1, where the cost measure $\mathcal{M}_c$ corresponds to $\mathcal{S}_c$.

### Performance guarantee

**Theorem 6.1** Let $I$ be any instance of TI–DC–MAP such that an optimal solution $P^*$ of total $c$–cost $OPT(I) = \mathcal{S}_c(P^*)$ exists. Then the algorithm HEUR–FOR–DIA–CONSTRAINT called with $\mathcal{M}_c = \mathcal{S}_c$ returns a placement $P$ satisfying $\mathcal{D}_d(P) \leq 2\Omega$ and $\mathcal{S}_c(P)/OPT(I) \leq 2 - 2/p$.

**Procedure** HEUR–FOR–DIA–CONSTRAINT

1 $G' := \text{bottleneck}(G, d, \Omega)$
2 $V_{cand} := \{v \in G' : \deg(v, G') \geq p - 1\}$
3 **if** $V_{cand} = \emptyset$ **then return** "certificate of failure"
4 Let $best := +\infty$
5 Let $P_{best} := \emptyset$
6 **for** each $v \in V_{cand}$ **do**
7 Sort the neighbors $N(v, G')$ of $v$ according to their $c$–distance from $v$
8 Assume now that $N(v, G') = \{w_1, \ldots, w_r\}$ with $c(v, w_1) \leq \cdots \leq c(v, w_r)$
9 Let $P(v) := \{v, w_1, \ldots, w_{p-1}\}$
10 **if** $\mathcal{M}_c(P(v)) < best$ **then** $P_{best} := P(v)$
11 **output** $P_{best}$

Figure 1: Details of the heuristic for TI–DC–MAP.

**Proof:** Consider an optimal solution $P^*$ such that $\mathcal{D}_d(P^*) \leq \Omega$. By definition, this placement forms a clique of size $p$ in $G' := \text{bottleneck}(G, d, \Omega)$. Hence, for any node $v \in P^*$ $|N(v, G')| \geq p$ and $V_{cand} \neq \emptyset$. Thus, the heuristic with $G'$ as defined above will not output a "certificate of failure".

Next, observe that HEUR-FOR-DIA-CONSTRAINT selects $p$ nodes from the graph $G'$. Let $P(v)$ be any such set of nodes found by the heuristic. Then $P(v)$ forms a clique in $(G')^2$. By the definition of $G'$ as a bottleneck graph with respect to $d$, the bound $\Omega$ and the assumption that the edge weights obey the triangle inequality, it follows that no edge $e$ in $(G')^2$ has $d$–weight more than $2\Omega$. Thus, for any *set* $P(v)$ of nodes selected by the heuristic from $G'$ has the property that $\mathcal{D}_d(P(v)) \leq 2\Omega$.

Next, we establish the performance guarantee with respect to the objective function value. To this end, define the following for each node $v \in P^*$: $S_v := \sum_{\substack{w \in P^* \\ w \neq v}} c(v, w)$. We have $\mathcal{S}_c(P^*) = \frac{1}{2} \sum_{v \in P^*} S_v$. Now let $v \in P^*$ be so that $S_v$ is a minimum among all nodes in $P^*$. Then clearly

$$OPT(I) = \mathcal{S}_c(P^*) \geq \frac{p}{2} S_v. \tag{1}$$

As mentioned earlier, $v \in V_{cand}$. Consider the step of the algorithm HEUR–FOR–DIA–CONSTRAINT in which it examines $v$. Let $N(v) := P(v) \setminus \{v\}$ denote the set of $p-1$ nearest neighbors of $v$ in $G'$ with respect to $c$. Then we have

$$\sum_{\substack{w \in N(v) \\ w \neq v}} c(v, w) \leq S_v, \tag{2}$$

by definition of $N(v)$ as the set of nearest neighbors. Let $w \in N(v)$ be arbitrary. Then

$$\sum_{u \in N(v) \cup \{v\} \setminus \{w\}} c(w, u) \quad = \quad c(w, v) + \sum_{u \in N(v) \setminus \{w\}} c(w, u)$$

$$
\begin{aligned}
&\leq c(w,v) + \sum_{u \in N(v)\setminus\{w\}} (c(w,v) + c(v,u)) \\
&= (p-1)c(w,v) + \sum_{u \in N(v)\setminus\{w\}} c(v,u) \\
&= (p-2)c(v,w) + \sum_{u \in N(v)} c(v,u) \\
&\overset{(2)}{\leq} (p-2)c(v,w) + S_v. \qquad (3)
\end{aligned}
$$

Now using (3) and again (2), we obtain

$$
\begin{aligned}
\mathcal{S}_c(P(v)) &= \mathcal{S}_c(N(v) \cup \{v\}) \\
&= \tfrac{1}{2}\left( \sum_{u \in N(v)} c(v,u) + \sum_{w \in N(v)} \sum_{u \in N(v)\cup\{v\}\setminus\{w\}} c(w,u) \right) \\
&\overset{(2)}{\leq} \tfrac{1}{2} S_v + \tfrac{1}{2} \sum_{w \in N(v)} \sum_{u \in N(v)\cup\{v\}\setminus\{w\}} c(w,u) \\
&\overset{(3)}{\leq} \tfrac{1}{2} S_v + \tfrac{1}{2} \sum_{w \in N(v)} ((p-2)c(v,w) + S_v) \\
&= \tfrac{1}{2} S_v + \tfrac{p-2}{2} S_v + \tfrac{p-1}{2} S_v \\
&= (p-1) S_v \\
&\overset{(1)}{\leq} (2 - 2/p) OPT(I).
\end{aligned}
$$

As the algorithm chooses the placement $P_{best}$ with the least $\mathcal{S}_c$, the claimed performance guarantee follows. ■

**Lower bound example**

We provide an example of an instance $I$ of TI-DC-MAP, where the $(2-2/p, 2)$–performance of our heuristic is asymptotically tight. In this example, the node set $V$ consists of $p+1$ subsets $B_0, \ldots, B_p$. The set $B_0 = \{v_1, \ldots, v_p\}$ forms an optimal placement of $p$ nodes and the distances between the nodes in $B_0$ are $c(v_i, v_j) = 1$ and $d(v_i, v_j) = 1$. All other sets $B_i$ $(i > 1)$ consist of $p-1$ nodes and we have that $c(u,w) = 2 - 2\varepsilon$ and $d(u,w) = 2$ $(u, w \in B_i)$.

Let $v_i$ be any node in the optimum placement $B_0$. we set $c(v_i, w) = 1 - \varepsilon$ and $d(v_i, w) = 1$ for all $w \in B_i$, while we let $c(v_i, w') = 2 - \varepsilon$ and $d(v_i, w) = 2$ for all $w' \in B_j, j \neq i$.

Finally, for each edge $(u, v)$ such that $u$ and $v$ belong to distinct sets $B_i$, we set the $c(u,) = 3 - 2\varepsilon$ and $d(u,v) = 2$. It is easily verified that the distances $c$ and $d$ obey the triangle inequality. The instance $I$ just described is illustrated in Figure 2.

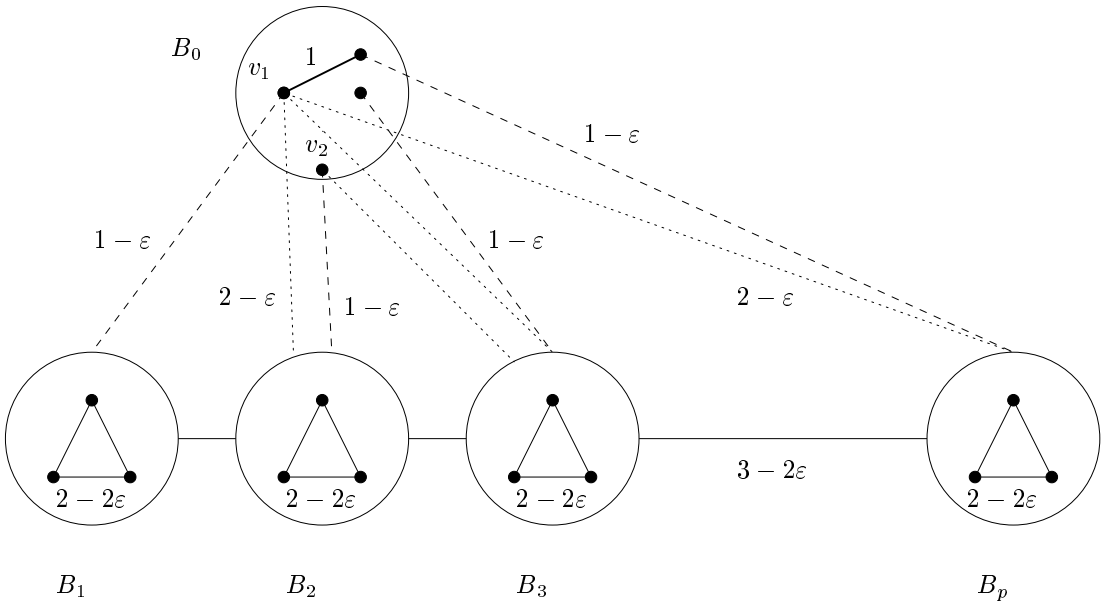


Figure 2: Lower Bound Example for the Heuristic for TI-DC-MAP

For the bound $\Omega := 1$, the set $B_0$ forms an optimum placement with objective function value $\mathcal{S}_c(B_0) = \frac{p(p-1)}{2}$ and diameter $\mathcal{D}_d(B_0) = 1$. In the first step, the algorithm calculates the bottleneck graph $G' = \mathit{bottleneck}(G, d, 1)$. This graph now only contains the edges between the nodes in $B_0$ and the edges of the form $(v_i, w)$ with $w \in B_i$. It is now easy to see that $V_{cand} = B_0$. For each node $v_i \in B_0$ the set of $p-1$ nearest neighbors consists of the set $B_i$ and $\mathcal{D}_d(\{v_i\} \cup B_i) = 2 = 2\Omega$. Moreover $\mathcal{S}_c(\{v_i\} \cup B_i) = (p-1)(1-\varepsilon) + \frac{(p-1)(p-2)}{2}(2-2\varepsilon)$ and $\lim_{\varepsilon \to 0} \frac{\mathcal{S}_c(\{v_i\} \cup B_i)}{\mathcal{S}_c(B_0)} = 2 - 2/p$.

## 6.3 Approximating TI–DC–MDP

Using the results in [RKM+93] in conjunction with the results in [MR+95] we can devise an approximation algorithm with a performance guarantee $(4, 4)$ for TI-DC-MDP. Here, we present an improved heuristic for this problem. This heuristic provides a performance guarantee of $(2, 2)$. In view of Theorem 5.4, this is the best approximation we can expect to obtain in polynomial time. The heuristic for approximating TI-DC-MDP is the same as HEUR-FOR-DIA-CONSTRAINT given in Figure 1, except that the measure $\mathcal{M}_c := \mathcal{D}_c$.

### Performance guarantee

**Theorem 6.2** Let $I$ be any instance of of TI–DC–MDP such that an optimal solution $P^*$ of diameter $OPT(I) = \mathcal{D}_c(P^*)$ under the $c$–cost exists. Then the algorithm HEUR-FOR-DIA-

**Procedure** HEUR–FOR–SC–MDP

1 Sort the edges of $G$ in ascending order with respect to $c$
2 Assume now that $c(e_1) \leq c(e_2) \leq \cdots \leq c(e_{\binom{n}{2}})$
3 Let $P_{best}$ :="certificate of failure"
4 $i := 1$
5 **do**
6 $\quad G_i := \text{bottleneck}(G, c, c(e_i))$
7 $\quad P_{best} := \mathsf{Test}(G_i, d, \Omega)$
8 $\quad i := i + 1$
9 **until** $P_{best} \neq$ "certificate of failure"
10 **output** $P_{best}$

Figure 3: Main Procedure for the Approximation of TI-SC-MDP

CONSTRAINT, called with $\mathcal{M}_c := \mathcal{D}_c$, returns a placement $P$ satisfying $\mathcal{D}_d(P) \leq 2\Omega$ and $\mathcal{D}_c(P)/OPT(I) \leq 2$.

The proof of Theorem 6.2 is deferred until Section 8, where we will state and prove an extension of this result (Theorem 8.5).

#### Lower bound example

The lower bound example in Figure 2 can be used to show that the approximation ratio indicated in Theorem 6.2 is tight for the heuristic. Again, the optimum placement consists of the nodes in $B_0$ which has both $c$– and $d$–cost 1. The placement returned by the heuristic is one of the sets $\{v_i\} \cup B_i$, which has $c$–cost $2 - 2\varepsilon$ and $d$–cost 2.

## 7 Approximation algorithms for sum constrained problems

In this section, we study approximation algorithms for bicriteria compact location problems where the objective is to minimize either the diameter $\mathcal{D}_c$ or the sum of the distances $\mathcal{S}_c$ subject to constraints of sum type.

### 7.1 Approximating TI-SC-MDP

We proceed to present a heuristic for TI-SC-MDP. The basic technique remains the same as discussed in Subsection 6.1. The main procedure shown in Figure 3 uses the test procedure from Figure 4.

#### Performance guarantee

**Lemma 7.1** Let $I$ be an instance of TI-SC-MDP such than there is an optimal placement $P^*$. If the test procedure $\mathsf{Test}(G_i, d, \Omega)$ returns a "certificate of failure", then we have $OPT(I) > c(e_i)$.

**Procedure** Test$(G, d, \Omega)$

1 $V_{cand} := \{v \in G : \deg(v, G) \geq p - 1\}$
2 **if** $V_{cand} = \emptyset$ **then return** "certificate of failure"
3 Let $best := +\infty$
4 Let $P_{best} := \emptyset$
6 **for** each $v \in V_{cand}$ **do**
7 Sort the neighbors $N(v, G)$ of $v$ according to their $d$–distance from $v$
8 Assume now that $N(v, G) = \{w_1, \ldots, w_r\}$ with $d(v, w_1) \leq \cdots \leq d(v, w_r)$
9 Let $P(v) := \{v, w_1, \ldots, w_{p-1}\}$
10 **if** $\mathcal{S}_d(P(v)) < best$ **then** $P_{best} := P(v)$
$best := \mathcal{S}_d(P(v))$
11 **if** $best > (2 - 2/p)\Omega$ **then return** "certificate of failure" **else return** $P_{best}$

Figure 4: Test–Procedure

**Proof:** Assume for the sake of a contradiction that Test$(G_i, d, \Omega)$ returns a "certificate of failure", and $OPT(I) \leq c(e_i)$.

Then by definition of the bottleneck graph $G_i = bottleneck(G, c, c(e_i))$, the optimum placement $P^*$ forms a clique of size $p$ in $G_i$ such that $\mathcal{S}_c(P^*) \leq \Omega$. Any node $v \in P^*$ will be included in $V_{cand}$. We define $S_v := \sum_{\substack{w \in P^* \\ w \neq v}} d(v, w)$, and let $v \in P^*$ be so that $S_v$ is a minimum among all nodes in $P^*$. Then $\Omega \geq \mathcal{S}_d(P^*) \geq \frac{p}{2} S_v$. Now using basically the same arguments as in the proof of Theorem 6.1 we conclude that $\mathcal{S}_d(P(v)) \leq (2 - 2/p)\Omega$.

Since $v \in V_{cand}$, the set $P(v)$ is one of the placements considered by the test procedure. Hence Test cannot return a "certificate of failure". This contradiction completes the proof of the lemma. ■

The following lemma is a straightforward consequence of the fact that any placement $P(v)$ considered by the Test$(G_i, d, \Omega)$ is a clique in $(G_i)^2$:

**Lemma 7.2** Assume that for some $i$ the test procedure Test$(G_i, d, \Omega)$ does not return a "certificate of failure". Then the placement $P$ returned satisfies $\mathcal{D}_c(P) \leq 2c(e_i)$. ■

Using the results from Lemma 7.1 and Lemma 7.2 we can establish the performance guarantee of the approximation algorithm for TI-SC-MDP.

**Theorem 7.3** Let $I$ denote any instance of TI–SC–MDP and assume that there is an optimal placement $P^*$ of diameter $OPT(I) = \mathcal{D}_c(P^*)$. Then HEUR–FOR–SC–MDP with the test procedure Test returns a placement $P$ with $\mathcal{S}_d(P) \leq (2 - 2/p)\Omega$ and $\mathcal{D}_c(I)/OPT(I) \leq 2$.

**Proof:** By construction, any placement returned by the test procedure Test violates the constraint $\mathcal{S}_d(P) \leq \Omega$ by a factor of at most $(2 - 2/p)$. The performance guarantee with

**Procedure** HEUR–FOR–SC–MAP

1 Use a binary search to find the smallest integer $T \in [0, p^2 \max\{\, c(e) : e \in E \,\}]$ such that Sum–Test($T$)=Yes.

2 **output** the placement generated by Sum–Test($T$).

Figure 5: Main procedure for TI–SC–MAP.

respect to the objective function immediately follows from Lemma 7.1 and Lemma 7.2. ■

**Lower bound example**

As a lower bound example take again the instance depicted in Figure 2 but interchange the $c$– and $d$–costs and set the bound $\Omega$ to be $\frac{p(p-1)}{2}$.

After having sorted the edges according to their $c$–costs, the heuristic will run the procedure Test on the bottleneck graph $bottleneck(G, c, 1)$, which again consists only of the edges between the nodes in $B_0$ and of the edges of the form $(v_i, w)$ with $w \in B_i$. Any placement $\{v_i\} \cup B_i$ violates the constraint $\mathcal{S}_d(P) \leq \Omega$ by a factor of less than $2 - 2/p$, so the test procedure will not return a certificate of failure; instead it will return a placement of that form. This placement has $c$–cost 2, since it includes two nodes from the same set $B_i$. Therefore, as $\varepsilon > 0$ tends to 0, the constraint violation can be made arbitrarily close to $2 - 2/p$.

## 7.2 Approximating TI–SC–MAP

We proceed to present a heuristic for TI–SC–MAP. The basic idea behind the approximation algorithm is to use a *parametric search* technique to reduce the problem to that of solving the minimum sum problem for one modified weight function. Then, by appropriate scaling and rounding techniques, this solution can be transformed back into a near optimal solution for the original bicriteria problem. The main procedure shown in Figure 5 uses the test procedure from Figure 6. Step 3 of the main procedure, that is, computing a $(2 - 2/p)$–approximation for the problem of minimizing $\mathcal{S}_h$, can be done by using the algorithm from Section 6.2, where we set $c := h$, $d := 1$ and $\Omega := 1$. We also note that $\gamma$ is a fixed quantity that specifies the accuracy requirement.

For a value of $T$, let $OPT_h(T)$ denote the sum of the distances of an optimal placement of $p$ nodes with respect to the distance function $h(v, w) := c(v, w) + \frac{T}{\Omega} d(v, w) = c(v, w) + \mu d(v, w)$; i.e.,

$$OPT_h(T) = \min_{\substack{P \subseteq V \\ |P|=p}} \mathcal{S}_h(P).$$

Then we have the following lemma:

**Lemma 7.4** The function $\mathcal{R}(T) = \frac{OPT_h(T)}{T}$ is monotonically nonincreasing on $\mathbf{Q} \setminus \{0\}$.

**Procedure** Sum–Test($T$)

1 Let $\mu := \frac{T}{\Omega}$.

2 **for** each pair $(v, w)$ of nodes define the distance function $h(v, w)$ by $h(v, w) := c(v, w) + \mu d(v, w)$.

3 Compute a $(2 - 2/p)$–approximation for the problem of finding a set of $p$ nodes minimizing $\mathcal{S}_h$.

4 Let $P_T$ be a set of $p$ nodes with $\mathcal{S}_h(P_T) \leq (2 - 2/p) \cdot \min_{\substack{P \subseteq V \\ |P|=p}} \mathcal{S}_h(P)$.

5 **if** $\mathcal{S}_h(P_T) \leq (2 - 2/p)(1 + \gamma)T$ **then output** Yes **else output** No.

Figure 6: Test procedure used for TI–SC–MAP.

**Proof:** Suppose for the sake of contradiction that for two values of $T$, say $T_1$ and $T_2$ with $T_1 < T_2$, we have that $\mathcal{R}(T_1) < \mathcal{R}(T_2)$. Let $P_1$ and $P_2$ denote optimal placements of $p$ nodes under $h$ when $T = T_1$ and $T = T_2$ respectively. For $i \in \{1, 2\}$, let $C_i$ and $D_i$ denote the costs of placement $P_i$ under $c$ and $d$ respectively. Thus, we have that $\mathcal{R}(T_i) = \frac{C_i}{T_i} + \frac{D_i}{\Omega}$ for $i \in \{1, 2\}$.

Consider the cost under $h$ of the placement $P_1$ when $T = T_2$. By the definition of $C_1$ and $D_1$, it follows that the cost of $P_1$ is $C_1 + \frac{D_1 \cdot T_2}{\Omega}$. Thus the value of $\mathcal{R}(T_2)$ is at most this cost divided by $T_2$ which is $\frac{C_1}{T_2} + \frac{D_1}{\Omega}$. This in turn is less than $\frac{C_1}{T_1} + \frac{D_1}{\Omega}$, since $T_1 < T_2$. But $\frac{C_1}{T_1} + \frac{D_1}{\Omega}$ is exactly $\mathcal{R}(T_1)$, and this contradicts the assumption that $\mathcal{R}(T_1) < \mathcal{R}(T_2)$. ■

Now we can establish the result about the performance guarantee of the heuristic. Let $OPT(I) = \mathcal{S}_c(P^*)$ denote the function value of an optimal placement $P^*$ of $p$ nodes. To simplify the analysis, we assume that $OPT(I)/\gamma$ is an integer. This can be enforced by first scaling the cost function $c$ so that all values are integers and then scaling again by $\gamma$.

**Theorem 7.5** Let $I$ denote any instance of TI–SC–MAP and assume that there is an optimal placement $P^*$ with $OPT(I) = \mathcal{S}_c(P^*)$. Then HEUR–FOR–SC–MAP with the test procedure Sum–Test returns a placement $P$ with $\mathcal{S}_d(P) \leq (1 + \gamma)(2 - 2/p)\Omega$ and $\mathcal{S}_c(I)/OPT(I) \leq (2 - 2/p)(1 + 1/\gamma)$.

**Proof:** Consider the call to the procedure Sum–Test when $T = T^* = OPT(I)/\gamma$. Notice that $T^*$ is an integer by our assumption. The $h$–cost of the placement $P^*$ is then $OPT(I) + \frac{T^*}{\Omega}\Omega = OPT(I) + T^* = (1 + \gamma)T^*$. Thus we have $OPT_h(T^*) \leq (1 + \gamma)T^*$ and the $(2 - 2/p)$–approximation $P_T$ that is computed in step 3 will satisfy $\mathcal{S}_h(P_T) \leq (2 - 2/p)OPT_h(T^*) \leq (2 - 2/p)(1 + \gamma)T^*$.

Thus, we observe that the procedure will return Yes and that $\mathcal{R}(T^*) \leq 1 + \gamma$. Further, the value $T$ found by the binary search in the main procedure satisfies $T \leq T^*$, since $T$ is the minimum value such that Sum–Test($C'$) returns Yes. Let $P_T$ be the corresponding

placement that is returned by Sum–Test. Then we have

$$\mathcal{S}_c(P_T) \leq \mathcal{S}_h(P_T) \leq (2 - \frac{2}{p})(OPT(I) + \frac{T}{\Omega}\Omega) \leq (2 - \frac{2}{p})(1 + \frac{1}{\gamma}) \cdot OPT(I).$$

Moreover, we see that

$$\frac{T}{\Omega}\mathcal{S}_d(P_T) \leq \mathcal{S}_h(P_T) \leq (2 - \frac{2}{p})(1 + \gamma)T,$$

and multiplying the last chain of inequalities by $\Omega/T$ yields

$$\mathcal{S}_d(P_T) \leq (2 - 2/p)(1 + \gamma)\Omega$$

and this completes the proof. ■

# 8 Extensions

We now briefly discuss some extensions of our algorithms presented in the previous sections.

## 8.1 Extension to the node–weighted case

We show how to extend HEUR–FOR–DIA–CONSTRAINT to apply to the case when we additionally have weights for each node in $G$, and the minimization objective is a function of both the edge weights and node weights. For the sake of brevity, we will illustrate our ideas for such an extension by considering one specific problem, namely TI-DC-MAP. The approximation algorithms for the other problems can be extended in a similar fashion.

The input consists again of a complete undirected graph $G = (V, E)$ with edge–weight functions $c, d$ and weights $\omega(v)$, for each $v \in V$. The goal is to find a placement $P$, with $|P| = p$, such that the objective function

$$\mathcal{S}_c^{wt}(P) = \sum_{\substack{e=(v,w) \\ v,w \in P}} c(v, w) + \sum_{v \in P} \omega(v)$$

is minimized subject to the same constraint as for the DC-MAP problem, namely $\mathcal{D}_d(P) \leq \Omega$. We denote this extension of the DC-MAP problem by DC-MAP$^{wt}$.

To obtain an approximate solution for TI-DC-MAP$^{wt}$, we transform a given instance $I$ of TI-DC-MAP$^{wt}$ into an instance $I'$ of TI-DC-MAP as discussed in the proof of the following lemma.

**Lemma 8.1** Given any instance $I$ of TI-DC-MAP$^{wt}$, there are linear time transformations $f$ and $g$ such that both of the following conditions hold.

1. $f$ constructs an instance $I'$ of TI-DC-MAP from $I$.

2. Given a placement $P'$ of $p$ nodes for $I'$ with diameter $D$ and objective function value $C$, $g$ constructs a placement $P$ of $p$ nodes for $I$ such that $P$ has diameter $D$ and objective function value $C$.

**Proof:** Given the instance $I$, the transformation $f$ constructs an instance $I'$ of DC-MAP as follows. The nodes in $G'(V', E')$ are in 1-1 correspondence with the nodes of $G$. The $d'$-cost on each edge in $G'$ is the same as the $d$-cost of the corresponding edge in $G$. The $c'$-cost for an edge $(v', w')$ in $G'$ is defined as follows:

$$c'(v', w') := c(v, w) + \frac{1}{2p}(\omega(v) + \omega(w)).$$

It is easy to check that the triangle inequality is satisfied for $c'$. Next consider a placement $P'$ of $p$ nodes in $I'$. The transformation $g$ chooses as its placement $P$ the $p$ nodes in $I$ corresponding to the $p$ nodes in $I'$. By a straightforward calculation, it can be seen that for any placement $P$ of $p$ nodes

$$\sum_{\substack{e'=(v',w') \\ v',w' \in P'}} c'(v', w') = \sum_{v \in P} \omega(v) + \sum_{\substack{e=(v,w) \\ v,w \in P}} c(v, w).$$

■

Combining Lemma 8.1 with Theorem 6.1, we obtain a $(2, 2 - 2/p)$–approximation for TI-DC-MAP$^{wt}$.

## 8.2 Extension to distinguished nodes

We now turn to the extension of our techniques to another variant of the problems. In this variant (referred to as placement with *distinguished nodes*), we are given a subset $D \subseteq V$ that must be included in the solution. As in node weighted case, such extensions can be done for all the four problems considered. For a problem $\Pi \in \{$ TI-DC-MAP, TI-DC-MDP, TI-SC-MAP, TI-SC-MDP, we use $\Pi^D$ to denote the problem with distinguished nodes. Since the basic approach for solving each of the problems $\Pi^D$ is similar, we illustrate our ideas by considering two problems, namely TI-DC-MAP$^D$ and TI-DC-MDP$^D$. We have chosen these two problems since one of them (TI-DC-MDP) involves two similar cost measures while the other (TI-DC-MAP) involves two different cost measures.

### Approximating TI-DC-MAP$^D$

A formal statement of the problem DC-MAP$^D$ is as follows.

**Procedure** HEUR–DISTINGUISHED

1 $G' := \text{bottleneck}(G, d, \Omega)$
2 $V_{cand} := \{v \in G' : v \notin D \text{ and } |N(v, G') \setminus D| \geq p-1\} \cup \{v \in D : |N(v, G') \setminus D| \geq p\}$
3 **if** $V_{cand} = \emptyset$ **then return** "certificate of failure"
4 Let $best := +\infty$
5 Let $P_{best} := \emptyset$
6 **for** each $v \in V_{cand}$ **do**
7 Sort the set $N_D(v, G') := N(v, G') \setminus \{D\}$ according to their $c$–distance from $v$
8 Assume now that $N_D(v, G') = \{w_1, \ldots, w_r\}$ with $c(v, w_1) \leq \cdots \leq c(v, w_r)$
9 **if** $v \in D$ **then** let $P(v) := \{v, w_1, \ldots, w_p\}$ **else** let $P(v) := \{v, w_1, \ldots, w_{p-1}\}$
10 **if** $\mathcal{M}_c(P(v) \cup D) < best$ **then** $P_{best} := P(v)$
11 **output** $P_{best}$

Figure 7: Modified Heuristic for TI–DC–MAP$^D$.

**Definition 8.2** [*Diameter Constrained Minimum Average Placement Problem with Distinguished Nodes* (DC–MAP$^D$)]

*Input:* An undirected complete graph $G = (V, E)$ with two positive edge weight functions $c, d : E \to \mathbb{Q}^+$, a subset $D \subset V$ of nodes, an integer $2 \leq p \leq |V \setminus D|$ and a number $\Omega \in \mathbb{Q}^+$ .
*Output:* A set $P \subseteq V$, with $|P| = p$, minimizing the objective

$$\mathcal{S}_c(P \cup D) = \sum_{\substack{e=(v,w) \\ v,w \in P \cup D}} c(v, w)$$

subject to the constraint

$$\mathcal{D}_d(P \cup D) = \max_{\substack{e=(v,w) \\ v,w \in P \cup D}} d(v, w) \leq \Omega.$$

If the set $D$ is empty, the problem reduces to DC-MAP. We show how to adopt HEUR–FOR–DIA–CONSTRAINT in Figure 1 to devise approximation algorithm for TI-DC-MAP$^D$. The modified heuristic HEUR-DISTINGUISHED is given in Figure 7.

We now establish the performance guarantee of the modified heuristic. As the proof parallels the one for Theorem 6.1, we just sketch the main steps.

**Theorem 8.3** Let $I$ be any instance of TI–DC–MAP$^D$ such that an optimal solution $P^*$ of total $c$–cost $OPT(I) = \mathcal{S}_c(P^* \cup D)$ exists. Then the algorithm HEUR–DISTINGUISHED called with $\mathcal{M}_c = \mathcal{D}_c$ returns a placement $P$ satisfying $\mathcal{D}_d(P) \leq 2\Omega$ and $\mathcal{S}_c(P)/OPT(I) \leq 2 - 2/(p + |D|)$.

**Proof:** Consider an optimal solution $P^*$ such that $\mathcal{D}_d(P^* \cup D) \leq \Omega$. It can be seen that any node from $P^* \cup D$ is in $V_{cand}$ and thus the heuristic will not output a certificate of failure. The fact that the diameter constraint $\mathcal{D}_d(P \cup D) \leq \Omega$ is violated by a factor of at most 2

is again a consequence of the fact that any placement $P(v) \cup D$ considered by the heuristic forms a clique in $(G')^2$.

Defining $\tilde{S}_v := \sum_{\substack{w \in P^* \cup D \\ w \neq v}} c(v,w)$ $(v \in P^* \cup D)$ and choosing $v \in P^* \cup D$ with minimum $\tilde{S}_v$, we have

$$OPT(I) = \mathcal{S}_c(P^* \cup D) = \frac{1}{2} \sum_{v \in P^* \cup D} \tilde{S}_v \geq \frac{p + |D|}{2} \tilde{S}_v.$$

Consider the iteration of the main loop when the heuristic considers $v$. Let $N(v) := P(v) \setminus \{v\} = \{w_1, \ldots, w_k\}$ be the set of nearest neighbors of $v$ in $V - D$, where $k = p$ if $v \in D$ and $k = p - 1$ if $v \in P^*$. Then we have $S'_v := \sum_{w \in (P(v) \cup D) \setminus \{v\}} c(v,w) \leq \tilde{S}_v$, by the fact that $N(v)$ is the set of $k$ nearest neighbors of $v$. Now,

$$\begin{aligned}
\mathcal{S}_c(P(v) \cup D) &= \sum_{w \in (P(v) \cup D) - \{v\}} c(v,w) + \frac{1}{2} \sum_{u \in (P(v) \cup D) - \{v\}} \sum_{w \in (P(v) \cup D) - \{v,u\}} c(u,w) \\
&= S'_v + \frac{1}{2} \sum_{u \in (P(v) \cup D) - \{v\}} \sum_{w \in (P(v) \cup D) - \{v,u\}} c(u,w) \\
&\leq S'_v + \frac{1}{2} \sum_{u \in (P(v) \cup D) - \{v\}} \sum_{w \in (P(v) \cup D) - \{v,u\}} (c(v,u) + c(v,w)) \\
&= S'_v + \frac{1}{2} \sum_{u \in (P(v) \cup D) - \{v\}} \left[ \sum_{w \in (P(v) \cup D) - \{v,u\}} c(v,u) + \sum_{w \in (P(v) \cup D) - \{v,u\}} c(v,w) \right] \\
&= S'_v + \frac{1}{2} \sum_{u \in (P(v) \cup D) - \{v\}} [(p + |D| - 2)c(v,u) + S'_v - c(v,u)] \\
&= S'_v + \frac{1}{2} \left( (p + |D| - 2)S'_v + (p + |D| - 1)S'_v - S'_v \right) \\
&= (p + |D| - 1)S'_v \\
&\leq (p + |D| - 1)\tilde{S}_v \\
&\leq (p + |D| - 1)\frac{2}{p + |D|} \cdot OPT(I) = \left(2 - \frac{2}{p + |D|}\right) \cdot OPT(I).
\end{aligned}$$

■

**Approximating** TI-DC-MDP$^D$

A formal statement of DC-MDP$^D$ is given below.

**Definition 8.4** [*Diameter Constrained Minimum Diameter Placement Problem with Distinguished Nodes* (DC-MDP$^D$)]

*Input:* As for DC-MAP$^D$.

*Output:* A set $P \subseteq V$, with $|P| = p$, minimizing the objective

$$\mathcal{D}_c(P \cup D) = \max_{\substack{e=(v,w) \\ v,w \in P \cup D}} c(v,w)$$

subject to the constraint

$$\mathcal{D}_d(P \cup D) = \max_{\substack{e=(v,w) \\ v,w \in P \cup D}} d(v,w) \leq \Omega.$$

As in the case of DC-MDP, where we simply used the heuristic designed for DC-MAP with the measure $\mathcal{M}_d := \mathcal{D}_c$, we can again use the heuristic HEUR-DISTINGUISHED for DC-MAP$^D$ and merely set $\mathcal{M}_d := \mathcal{D}_d$. We then obtain the following extension of Theorem 6.2.

**Theorem 8.5** Let $I$ be any instance of of TI–DC–MDP$^D$ such that an optimal solution $P^*$ of diameter cost $OPT(I) = \mathcal{D}_c(P^* \cup D)$ exists. Then the algorithm HEUR-DISTINGUISHED, called with $\mathcal{M}_c := \mathcal{D}_c$, returns a placement $P$ satisfying $\mathcal{D}_d(P \cup D) \leq 2\Omega$ and $\mathcal{D}_c(P)/OPT(I) \leq 2$.

**Proof:** Consider an optimal solution $P^*$ such that $\mathcal{D}_d(P^* \cup D) \leq \Omega$. It can be seen that $P^* \cup D \subseteq V_{cand}$, and thus $V_{cand}$ is non–empty. Consequently, the heuristic will not output a "certificate of failure".

Any placement considered in the heuristic HEUR-DISTINGUISHED will turn into a clique in $(G')^2$, where $G' = \text{bottleneck}(G, d, \Omega)$. Thus as in the proof of Theorem 6.1, we conclude again that any placement considered by the heuristic will violate the diameter constraint by a factor of at most two.

Now consider an arbitrary node $v \in P^*$. Clearly $v \in V_{cand}$. Consider the step of the algorithm HEUR-DISTINGUISHED in which it considers $v$. By definition of $N_D(v, G')$, for any $w \in \{w_1, \ldots, w_p\}$ we have $c(v,w) \leq OPT(I)$. Moreover, trivially $c(v,d) \leq OPT(I)$ for any $d \in D$, since $v \in P^*$. Thus for $w, w' \in P(v) \cup D$ we have $c(w,w') \leq c(v,w) + c(v,w') \leq 2\,OPT(I)$ by the triangle inequality. Consequently, $\mathcal{D}_c(P(v)) = \mathcal{D}_c(N(v) \cup \{v\}) \leq 2\,OPT(I)$.

Now, since the algorithm HEUR-DISTINGUISHED chooses a placement with minimum diameter among all the placements produced, the claimed performance guarantee with respect to $c$-cost follows. ■

The approximation algorithms for TI-SC-MDP and TI-SC-MAP can also be extended to handle the case of distinguished nodes. By using the ideas in this section and the techniques in Section 6, we obtain the following theorems.

**Theorem 8.6** There is a polynomial time algorithm A, which given any instance $I$ of TI-SC-MDP$^D$ such that there is an optimal placement $P^*$ of diameter $OPT(I) = \mathcal{D}_c(P^* \cup D)$, returns a placement $P$ with $\mathcal{S}_d(P) \leq \left(2 - \frac{2}{p+|D|}\right)\Omega$ and $\mathcal{D}_c(I)/OPT(I) \leq 2$. ■

**Theorem 8.7** For any fixed $\gamma > 0$, there is a polynomial time algorithm A, which given any instance $I$ of TI-SC-MAP$^D$ such that there is an optimal placement $P^*$ with $OPT(I) = \mathcal{S}_c(P^* \cup D)$, returns a placement $P$ with $\mathcal{S}_d(P) \leq (1+\gamma)\left(2 - \frac{2}{p+|D|}\right)\Omega$ and $\mathcal{S}_c(I)/OPT(I) \leq \left(1+\frac{1}{\gamma}\right)\left(2 - \frac{2}{p+|D|}\right)$. ■

# 9 Concluding remarks

We introduced and studied the complexity and approximability of several natural bicriteria compact location problems. Our results demonstrate that when distance functions obey the triangle inequality, the problems are provably easier to approximate.

Table 1 summarizes our results. The table contains hardness results and performance ratios for finding compact placements for different pairs of minimization objectives. The horizontal entries denote the objective function. For example the entry in row $i$, column $j$ denotes the performance guarantee for the problem of minimizing objective $j$ with a budget on objective $i$.

| $\rightarrow$ Objective<br>$\downarrow$ Budget | Diameter | Sum |
|---|---|---|
| Diameter | approximable within $(2,2)$<br>not approximable within<br>$(2-\varepsilon,2)$ or $(2,2-\varepsilon)$ | approximable within $(2-\frac{2}{p},2)$<br>not approximable within<br>$(\alpha,2-\varepsilon)$ |
| Sum | approximable within $(2,2-\frac{2}{p})$<br>not approximable within $(2-\varepsilon,\alpha)$ | approximable within<br>$((1+\gamma)(2-\frac{2}{p}),(1+\frac{1}{\gamma})(2-\frac{2}{p}))$ |

**Table 1.** Performance guarantee results for bicriteria compact location in a complete graph with both the edge weights obeying the triangle inequality. $\gamma > 0$ is a fixed accuracy parameter. The non-approximability results stated assume that $\mathcal{P} \neq \mathcal{NP}$. As discussed in Section 8, these results can be extended to handle node weights and also the case of distinguished nodes.

The results in this paper raise the following questions.

1. Reference [RRT94] studies approximation algorithms for dispersion problems. It would be of interest to investigate approximation algorithms for bicriteria versions of practical dispersion problems. A slightly different bicriteria formulation involving node and edge weights for dispersion problems is considered in [RRT95].

2. In a companion paper [KN$^+$95c], we study bicriteria compact location problems when the underlying graph is a tree. It is of interest to consider other restricted graph classes which arise in the context of parallel computing and devise efficient algorithms for the

bicriteria problems studied here. The $n$-dimensional hypercubes and $n$-dimensional meshes are two examples of natural graph classes for which bicriteria compact location problems could be investigated.

3. The compact location problem is an instance of a more general class of problems where the goal is to find an embedding of the problem topology (graph) onto a host computer topology so as to minimize the total communication cost. Placements produced using the algorithms presented in this paper may lead to heavy congestion in one or more of the links. Therefore, it is of interest to model and investigate compact location problems taking the congestion of links into account.

**Acknowledgments:** We thank Professor R. Ravi (Carnegie-Mellon University), Dr. V. Radhakrishnan (Hewlett-Packard), R. Sundaram (M.I.T) and Professor D. J. Rosenkrantz (University at Albany - SUNY) for valuable discussions which lead to the research described in this paper.

## References


[AI+91] A. Aggarwal, H. Imai, N. Katoh, and S. Suri, Finding $k$ points with Minimum Diameter and Related Problems. *J. Algorithms*, 12(1):38–56, March 1991.

[AMO93] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[An72] H. C. Andrews, *Introduction to Mathematical Techniques in Pattern Recognition*, Wiley-Interscience, New York, NY, 1972.

[AA+94] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala, Improved approximation guarantees for minimum-weight $k$-trees and prize-collecting salesmen. *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing* (STOC'95), pp. 277–376.

[BCV95] A. Blum, P. Chalasani and S. Vempala, A Constant-Factor Approximation for the $k$-MST Problem in the Plane. *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing* (STOC'95), pp. 294–302.

[BKP93] J. Bar-Ilan, G. Kortsarz and D. Peleg, How to Allocate Network Centers. *J. Algorithms*, Vol. 15, No. 3, Nov. 1993, pp 385-415.

[BS94] M. Bellare and M. Sudan, Improved Non–Approximability Results. in *Proceedings of the 26th annual ACM Symposium on the Theory of Computing (STOC'94)*, May 1994, pp 184–193.

[BE95] M. Bern and D. Eppstein, Approximation Algorithms for Geometric Problems. Unpublished manuscript, January 1995.

[DF85] M. E. Dyer and A. M. Frieze, A Simple Heuristic for the $p$–Center Problem. *Operations Research Letters*, 3(6):285–288, Feb. 1985.

[EN89] E. Erkut and S. Neuman, Analytical Models for Locating Undesirable Facilities. *European J. Operations Research*, 40:275–291, 1989.

[FG88] T. Feder and D. Greene, Optimal Algorithms for Approximate Clustering. In *ACM Symposium on Theory of Computing (STOC'88)*, May 1988, pp 434–444.

[GH94] N. Garg and D. S. Hochbaum, An $O(\log k)$ Approximation Algorithm for the $k$ Minimum Spanning Tree Problem in the Plane. In *Proc. 26th ACM Symp. on Theory of Computing* (STOC'94), Montreal, May 1994, pp 432–438.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness.* W. H. Freeman and Co., San Francisco, CA, 1979.

[Go85] T. F. Gonzalez, Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science*, 38:293–306, 1985.

[HM79] G. Y. Handler and P. B. Mirchandani, *Location on Networks: Theory and Algorithms.* MIT Press, Cambridge, MA, 1979.

[HS86] D. S. Hochbaum and D. B. Shmoys, A Unified Approach to Approximation Algorithms for Bottleneck Problems. *Journal of the ACM*, 33(3):533–550, July 1986.

[KN+95a] S. O. Krumke, H. Noltemeier, S. S. Ravi and M. V. Marathe, Compact Location Problems with Budget and Communication Constraints. In *1st International Conference on Computing and Combinatorics,* X'ian, China, LNCS Volume 959, Springer Verlag, August, 1995, Pages 510-519.

[KN+95b] S. O. Krumke, H. Noltemeier, S. S. Ravi and M. V. Marathe, Bicriteria Problems in Location Theory. In *21st Workshop in Graph Theoretic Concepts in Computer Science (WG'95)*, Aachen, Germany, June 1995.

[KN+95c] S. O. Krumke, H. Noltemeier, S. S. Ravi and M. V. Marathe, Bicriteria Compact Location Problems for Trees (in preparation), Nov. 1995.

[Lee82] D. T. Lee, On $k$–nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput.*, C–31:478–487, 1982.

[LV92] J. H. Lin and J. S. Vitter, $\varepsilon$–Approximations with Minimum Packing Constraint Violation. In *ACM Symposium on Theory of Computing (STOC'92)*, May 1992, pp 771–781.

[MR+95] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz and H. B. Hunt III, Bicriteria Network Design Problems. In *Proceedings of the 22nd International Colloquium on Automata Languages and Programming (ICALP'95)*, LNCS 944, Springer Verlag, July 1995, pp 487–498.

[MF90] P. B. Mirchandani and R. L. Francis, *Discrete Location Theory.* Wiley–Interscience, New York, NY, 1990.

[PS85] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction.* Springer–Verlag Inc., New York, NY, 1985.

[RKM+93] V. Radhakrishnan, S. O. Krumke, M. V. Marathe, D. J. Rosenkrantz and S. S. Ravi, Compact Location Problems. In *Proceedings of the 13th Conference on the Foundations of Software Technology and Theoretical Computer Science (FST&TCS'93)*, Volume 761 of *LNCS*, pages 238–247, December 1993.

[RMR+93] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz and H. B. Hunt III, Many birds with one stone: Multi-objective approximation algorithms. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing (STOC'93)*, May 1993, pages 438–447.

[RRT94] S. S. Ravi, D. J. Rosenkrantz and G. K. Tayi, Heuristic and Special Case Algorithms for Dispersion Problems. *Operations Research*, Vol. 42, No. 2, March-April 1994, pp 299–310. (An extended abstract of this paper appeared in *Proceedings of the 1991 Workshop on Algorithms and Data Structures* (WADS'91), Ottawa, Lecture Notes in CS, Vol. 519, Springer-Verlag, New York, NY, August 1991, pp 355–366.)

[RRT95] D. J. Rosenkrantz, S. S. Ravi and G. K. Tayi, Capacitated Versions of Dispersion Problems (in preparation), Nov. 1995.

[RR+94] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz and S. S. Ravi, Spanning trees short or small. In *Proceedings, Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'94)*, Jan. 1994, pp 546–555. (Complete version to appear in *SIAM Journal on Discrete Mathematics.*)

[RSL77] D. J. Rosenkrantz, R. E. Stearns and P. M. Lewis II, An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM J. Computing*, Vol. 6, No. 3, 1977, pp 563–581.

[ST93] D. B. Shmoys and E. Tardos, Scheduling unrelated parallel machines with costs. In *Proceedings of the 4th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'93)*, Jan. 1993, pp 438–447.

[Ta91] A. Tamir, Obnoxious Facility Location on Graphs. *SIAM J. Discrete Mathematics*, Vol. 4, 1991, pp 550–567.