HEURISTIC AND SPECIAL CASE ALGORITHMS FOR DISPERSION PROBLEMS

S. S. RAVI, D. J. ROSENKRANTZ and G. K. TAYI
University at Albany-SUNY, Albany, New York
(Received August 1991; revision received June 1992; accepted June 1992)

The dispersion problem arises in selecting facilities to *maximize* some function of the distances between the facilities. The problem also arises in selecting nondominated solutions for multiobjective decision making. It is known to be NP-hard under two objectives: maximizing the minimum distance (**MAX-MIN**) between any pair of facilities and maximizing the average distance (**MAX-AVG**). We consider the question of obtaining near-optimal solutions. For **MAX-MIN**, we show that if the distances do not satisfy the triangle inequality, there is no polynomial-time relative approximation algorithm unless **P** = **NP**. When the distances satisfy the triangle inequality, we analyze an efficient heuristic and show that it provides a performance guarantee of two. We also prove that obtaining a performance guarantee of less than two is NP-hard. For **MAX-AVG**, we analyze an efficient heuristic and show that it provides a performance guarantee of four when the distances satisfy the triangle inequality. We also present a polynomial-time algorithm for the 1-dimensional **MAX-AVG** dispersion problem. Using that algorithm, we obtain a heuristic which provides an asymptotic performance guarantee of $\pi/2$ for the 2-dimensional **MAX-AVG** dispersion problem.

Many problems in location theory deal with the placement of facilities on a network to minimize some function of the distances between facilities or between facilities and the nodes of the network (Handler and Mirchandani 1979). Such problems model the placement of "desirable" facilities such as warehouses, hospitals, and fire stations. However, there are situations in which facilities are to be located to *maximize* some function of the distances between pairs of nodes. Such location problems are referred to as *dispersion* problems (Chandrasekharan and Daughety 1981, Kuby 1987, Erkut and Neuman 1989, 1990, and Erkut 1990) because they model situations in which proximity of facilities is undesirable. One example of such a situation is the distribution of business franchises in a city (Erkut). Other examples of dispersion problems arise in the context of placing "undesirable" (also called obnoxious) facilities, such as nuclear power plants, oil storage tanks, and ammunition dumps (Kuby 1987, Erkut and Neuman 1989, 1990, and Erkut 1990). Such facilities need to be spread out to the greatest possible extent so that an accident at one of the facilities will not damage any of the others. The concept of dispersion is also useful in the context of multiobjective decision making (Steuer 1986). When the number of nondominated solutions is large, a decision maker may be interested in selecting a manageable collection of solutions which are dispersed as far as possible with respect to the objective function values. Other applications of facility dispersion are discussed in Erkut (1990) and Erkut and Neuman (1989, 1990).

Analytical models for the dispersion problem assume that the given network is represented by a set $V = \{v_1, v_2, \ldots, v_n\}$ of n nodes with nonnegative distance (also called *edge weight*) between every pair of nodes. The distances are assumed to be symmetric and so the network can be thought of as an undirected complete graph on n nodes with a nonnegative weight on each edge. The weight of the edge $\{v_i, v_j\}$ $(i \neq j)$ is denoted by $w(v_i, v_j)$. We assume that $w(v_i, v_i) = 0$ for $1 \leq i \leq n$. The objective of the dispersion problem is to locate p facilities $(p \leq n)$ among the n nodes of the network, with at most one facility per node, such that some function of the distances between facilities is maximized. Two of the optimality criteria considered in the literature (Kuby 1987, Erkut 1990, and Erkut and Neuman 1989) are **MAX-MIN** (i.e., maximize the minimum distance between a pair of facilities) and **MAX-AVG** (i.e., maximize the average distance between a pair of facilities). Under either criterion, the problem is known to be NP-hard, even when the distances satisfy the triangle inequality (Wang and Kuo 1988, Hansen and Moon 1988, and Erkut 1990).

Subject classifications: Analysis of algorithms: computational complexity. Facilities/equipment planning: discrete location. Programming: heuristic.
Area of review: OPTIMIZATION.

0030-364X/94/4202-0299$01.25

Although many researchers have studied the dispersion problem (see Erkut and Neuman 1989 for a survey and an extensive bibliography), except for White (1991), the question of whether there are efficient heuristics with provably good performance has not been addressed. This question forms the main focus of this paper. We show that if the distances do not satisfy the triangle inequality, then for any constant $K \geqslant 1$, no polynomial time algorithm can provide a performance guarantee of K for the **MAX-MIN** dispersion problem unless **P** = **NP**. When the distances satisfy the triangle inequality, we analyze a known heuristic and prove that it provides a performance guarantee of 2 for the **MAX-MIN** dispersion problem. This is an improvement over the performance guarantee of 3 proven in White (1991) using a different heuristic. (This improvement was obtained independently by White 1992.) We also show that no polynomial-time algorithm can provide a performance guarantee of less than 2 unless **P** = **NP**. We also analyze an efficient heuristic for the **MAX-AVG** dispersion problem with triangle inequality, and prove that it provides a performance guarantee of 4. An efficient algorithm for the 1-dimensional **MAX-MIN** dispersion problem is presented in Wang and Kuo. We provide an efficient algorithm for the 1-dimensional **MAX-AVG** dispersion problem. We also show how this algorithm can be used to obtain a heuristic with an asymptotic performance guarantee of ≈ 1.571 for the 2-dimensional **MAX-AVG** dispersion problem.

The remainder of this paper is organized as follows. Section 1 contains the formal definitions and a discussion of the previous work on the dispersion problem. Sections 2 and 3 address the dispersion problem under the **MAX-MIN** and **MAX-AVG** criteria, respectively. Section 4 presents our results for the 1- and 2-dimensional dispersion problems. Section 5 contains tables which summarize prior results, our contributions, and open problems.

1. DEFINITIONS AND PREVIOUS WORK

We begin with the specifications of the **MAX-MIN** and **MAX-AVG** dispersion problems in the format of Garey and Johnson (1979).

MAX-MIN Facility Dispersion (MMFD)

Instance: A set $V = \{v_1, v_2, \ldots, v_n\}$ of n nodes, a nonnegative distance $w(v_i, v_j)$ for each pair v_i, v_j of nodes, and an integer p such that $2 \leqslant p \leqslant n$.

Requirement: Find a subset $P = \{v_{i_1}, v_{i_2}, \ldots, v_{i_p}\}$ of V with $|P| = p$, such that the objective function $f(P) = \min_{x,y \in P}\{w(x, y)\}$ is maximized.

MAX-AVG Facility Dispersion (MAFD)

Instance: As in **MMFD**.

Requirement: Find a subset $P = \{v_{i_1}, v_{i_2}, \ldots, v_{i_p}\}$ of V with $|P| = p$, such that the objective function

$$g(P) = \frac{2}{p(p-1)} \sum_{x,y \in P} w(x, y)$$

is maximized.

The objective function for **MAFD** has the above form because the number of edges among the nodes in P is $p(p-1)/2$. Note that maximizing the average distance is equivalent to maximizing the sum of the distances. We point out that maximizing the average would sometimes produce solutions which are far from the optimum with respect to the **MAX-MIN** criterion and vice versa.

The distances specified in an instance of **MMFD** or **MAFD** satisfy the *triangle inequality* if for any three distinct nodes v_i, v_j, and v_k, $w(v_i, v_j) + w(v_j, v_k) \geqslant w(v_i, v_k)$. The set P of nodes at which an algorithm places the p facilities is called a *placement*. Given a placement P for an **MMFD** instance, the quantity $f(P)$ defined by

$$f(P) = \min_{x,y \in P}\{w(x, y)\} \tag{1}$$

is called the *solution value* corresponding to P. Similarly, given an **MAFD** instance and a placement P, the solution value $g(P)$ corresponding to P is defined by

$$g(P) = \frac{2}{p(p-1)} \sum_{x,y \in P} w(x, y). \tag{2}$$

Both **MMFD** and **MAFD** are known to be NP-hard, even when the edge weights satisfy the triangle inequality (Wang and Kuo 1988, Hansen and Moon 1988, and Erkut 1990). Much of the work on the dispersion problem reported in the literature (see the bibliography in Erkut and Neuman 1989) falls into two categories. Papers in the first category deal with branch-and-bound algorithms and heuristics (see Kuby 1987, Erkut, Baptie and von Hohenbalken 1990, Erkut 1990, Erkut and Neuman 1989, 1990, and the references cited therein). However, except for White (1991), only experimental studies of the performance of the heuristics have been reported. White (1991) presents a heuristic for **MMFD** when the nodes are points in d-dimensional Euclidean space and the distance between a pair of points is their Euclidean distance. He shows that the heuristic always produces a placement whose solution value is within a factor of 3 of the optimum solution value. In the next section, we improve that result by considering a different heuristic which guarantees a placement whose solution value is within a factor of 2 of the optimal solution value for any instance of **MMFD** in which the distances satisfy the triangle inequality. We also show that unless **P** = **NP**, no polynomial-time algorithm can provide a better performance guarantee.

Papers in the second category deal with restricted versions and variants of **MMFD** and **MAFD**. For example, 1- and 2-dimensional versions of **MMFD** were studied by Wang and Kuo. They present a polynomial algorithm for the 1-dimensional **MMFD** and prove that the 2-dimensional **MMFD** is NP-hard. We note that problems **MMFD** and **MAFD** defined above are *discrete* in nature because each facility must be placed at one of the given nodes. Researchers have also considered *continuous* versions of the dispersion problems (Church and Garfinkel 1978, Chandrasekharan and Daughety 1981, and Tamir 1991) where facilities may be placed at any point on the edges of a given network. In Chandrasekharan and Daughety a polynomial algorithm is presented for the continuous version of **MMFD** on tree networks. Church and Garfinkel present a polynomial algorithm for locating one facility on an edge of a connected (but not necessarily complete) network to maximize a weighted sum of the distances from the facility to the nodes of the network. Tamir presents an improved algorithm for the same problem. In addition, he establishes the NP-hardness of the continuous versions of **MMFD** and **MAFD** and presents results concerning performance guarantees for the continuous version of **MMFD**. Dasarathy and White (1980) assume the nodes of the network to be points in k-dimensional space and consider the problem of finding a point within a given convex polyhedron to maximize the minimum Euclidean distance between the point and the nodes. They present polynomial algorithms for $k = 2$ and $k = 3$. For $k = 2$, this problem is referred to as the *largest empty circle* (LEC) in the computational geometry literature (see subsection 6.4 of Preparata and Shamos 1985). The LEC problem can be solved in $O(n \log n)$ time, which is known to be optimal (Preparata and Shamos). A problem similar to LEC but with a different distance function is studied in Melachrinoudis and Cullinane (1986). A weighted version of the problem for $k = 2$ is studied in Erkut and Öncü (1991). For a discussion of other variants, we refer the reader to Erkut and Neuman (1989, 1990).

In this paper, we consider only the discrete versions of the dispersion problems. Our focus is on the analysis of heuristics for **MMFD** and **MAFD**. By a heuristic we mean a *polynomial-time* approximation algorithm which produces feasible, but not necessarily optimal, solutions. Heuristics are commonly classified as *absolute* or *relative* depending on the types of performance guarantees that can be established for them (Horowitz and Sahni 1984). An *absolute* approximation algorithm guarantees a solution that is within an *additive* constant of the optimal value for every instance of the problem. A *relative* approximation algorithm guarantees a solution that is within a *multiplicative* constant of the optimal value for every instance of the problem. It is easy to show, using the technique presented in Garey and Johnson (pp. 138–139), that there are no absolute approximation algorithms for **MAFD** or for **MMFD**, unless **P** = **NP**. So we restrict our attention to the study of relative approximation algorithms.

2. NEAR-OPTIMAL SOLUTIONS TO MAX-MIN FACILITY DISPERSION

We first consider **MMFD** without requiring the distances to satisfy the triangle inequality and prove a negative result concerning relative approximation algorithms.

Theorem 1. *If the distances are not required to satisfy the triangle inequality, then there is no polynomial-time relative approximation algorithm for* **MMFD** *unless* **P** = **NP**.

Proof. Suppose that A is a polynomial-time approximation algorithm which provides a performance guarantee of $K \geqslant 1$ for **MMFD**. We show that A can be used to devise a polynomial-time algorithm for a problem which is known to be NP-complete. This contradicts the assumption that **P** $\neq$ **NP** and, hence, will establish the theorem.

The known NP-complete problem used here is **CLIQUE**, whose definition is as follows (Garey and Johnson).

Problem CLIQUE

Instance: An undirected graph $G(N, E)$ and a positive integer $J \leqslant |N|$.

Question: Does G contain a *clique* of size $\geqslant J$ (i.e., is there a subset $N' \subseteq N$ such that $|N'| \geqslant J$ and every pair of vertices in N' is joined by an edge in E)?

Consider an arbitrary instance of **CLIQUE** defined by a graph $G(N, E)$ and integer J. Let $n = |N|$ and

$N = \{x_1, x_2, \ldots, x_n\}$. We construct an instance of **MMFD** (without triangle inequality) as follows. The node set $V = \{v_1, v_2, \ldots, v_n\}$ of the **MMFD** instance is in one-to-one correspondence with N. The number of facilities p is set equal to J and the distances are defined as follows. Let $w(v_i, v_j) = K + 1$ if $\{x_i, x_j\}$ is in E; otherwise, let $w(v_i, v_j) = 1$. Clearly, this construction can be carried out in polynomial time. We will show that for the resulting **MMFD** instance, the solution value of a placement produced by A is *greater than* 1 iff G has a clique of size J.

Suppose that G has a clique of size J. Let $\{x_{i_1}, x_{i_2}, \ldots, x_{i_J}\}$ denote the vertices of the clique. Consider the placement $P = \{v_{i_1}, v_{i_2}, \ldots, v_{i_J}\}$. By our definition of the distances, the weight of every edge in P is equal to $K + 1$. Therefore, the solution value of the placement P is also $K + 1$. Since A provides a performance guarantee of K, the solution value of the placement returned by A is at least $(K + 1)/K$, which is *greater than* 1.

Now suppose that G does not have a clique of size J. In this case, notice that no matter which subset of $p = J$ nodes is chosen as the placement, there will always be at least one pair of nodes v_i and v_j with $w(v_i, v_j) = 1$. Therefore, the solution value corresponding to *any* placement is at most 1. In particular, the solution value of a placement produced by A is also at most 1. Thus, by merely comparing the solution value of the placement produced by A with 1, we can solve an arbitrary instance of the **CLIQUE** problem. This completes the proof.

Even though Theorem 1 provides a strong negative result, it is not applicable in many practical situations because distances often satisfy the triangle inequality. Therefore, it is of interest whether there is an efficient relative approximation algorithm for **MMFD** when the distances satisfy the triangle inequality (**MMFD-TI**). The remaining theorems in this section precisely characterize the performance guarantees obtainable for **MMFD-TI**.

A greedy heuristic (which we call **GMM**) for **MMFD-TI** is shown in Figure 1. This heuristic is essentially the same as the "furthest point outside the neighborhood" heuristic, described in Steuer (Chapter 11), using a different format. An experimental study of the performance of this heuristic is carried out in Erkut and Neuman (1990). In describing this heuristic, we use P to denote the set of nodes at which **GMM** places the p facilities. The heuristic begins by initializing P to contain a pair of nodes in V which are joined by an edge of maximum weight. Subsequently, each iteration of **GMM** chooses a node v

```
Step 1. Let $v_i$ and $v_j$ be the endpoints of an edge of maximum weight.
Step 2. $P \leftarrow \{v_i, v_j\}$.
Step 3. while ( $|P| < p$ ) do
        begin
            a. Find a node $v \in V - P$ such that $\min_{v' \in P} \{w(v, v')\}$ is maximum
               among the nodes in $V - P$.
            b. $P \leftarrow P \cup \{v\}$.
        end
Step 4. Output $P$.
```

Figure 1. Details of heuristic **GMM**.

from $V - P$ such that the minimum distance from v to a node in P is the largest among all the nodes in $V - P$. In each step, ties are broken arbitrarily. Heuristic **GMM** terminates when $|P| = p$. The solution value of the placement P produced by **GMM** is equal to $\min_{x,y \in P}\{w(x, y)\}$.

We now present an example to illustrate the **GMM** heuristic. Consider an **MMFD-TI** instance with five nodes (denoted by v_1, v_2, v_3, v_4, and v_5) and let $p = 3$. The edge weights which satisfy the triangle inequality are: $w(v_1, v_2) = 3$, $w(v_2, v_3) = w(v_2, v_4) = w(v_2, v_5) = 1$, and all other edges are of weight 2. To begin, **GMM** will place two of the facilities at v_1 and v_2 because $w(v_1, v_2) = 3$ is the maximum edge weight. Now, no matter where the third facility is placed, the solution value of the placement is 1 because each of the remaining nodes (v_3, v_4, and v_5) has an edge of weight 1 to v_2. However, an optimal placement consists of the three nodes v_3, v_4, and v_5 and has a solution value of 2. Thus, for this example, the solution value produced by **GMM** differs from the optimal value by a factor of 2. Our next theorem shows that the performance of **GMM** is never worse. (This result was obtained independently by White 1992.) Moreover, we will also show (Theorem 3) that unless **P** = **NP**, no polynomial-time heuristic can provide a better performance guarantee.

Theorem 2. *Let I be an instance of* **MMFD-TI**. *Let $OPT(I)$ and $GMM(I)$ denote, respectively, the solution values of an optimal placement and that produced by* **GMM** *for the instance I. Then $OPT(I)/GMM(I) \leqslant 2$.*

Proof. Consider the set-valued variable P in the description of **GMM**. Let $f(P) = min_{x,y \in P}\{w(x, y)\}$. We will show by induction that the condition

$$f(P) \geqslant OPT(I)/2 \tag{3}$$

holds after each addition to P. Since $GMM(I) = f(P)$ after the last addition to P, the theorem then follows.

Since the first addition inserts two nodes joined by an edge of the largest weight into P, (3) clearly holds after the first addition. So, assume that the condition holds after k additions to P for some $k \geq 1$. We will prove that the condition holds after the $(k + 1)$st addition to P as well.

To that end, let $P^* = \{v_1^*, v_2^*, \ldots, v_p^*\}$ denote an optimal placement. For convenience, we use l^* for $OPT(I)$. The following observation is an immediate consequence of the definition of the solution value corresponding to a placement for an **MMFD** instance.

Observation 1. *For every pair v_i^*, v_j^* of distinct nodes in P^*, $w(v_i^*, v_j^*) \geq l^*$.*

Let $P_k = \{x_1, x_2, \ldots, x_{k+1}\}$ denote the set P after k additions. (Note that $|P_k| = k + 1$, because the first addition inserts two nodes into P.) Since **GMM** adds at least one more node to P, the following is a trivial observation.

Observation 2. $|P_k| = k + 1 < p$.

For each $v_i^* \in P^*$ $(1 \leq i \leq p)$, define $S_i^* = \{u \in V \mid w(v_i^*, u) < l^*/2\}$. That is, S_i^* is the set of all nodes whose distances from v_i^* are *less than* $l^*/2$. The following claim provides two useful properties of these sets.

Claim 1

a. For $1 \leq i \leq p$, S_i^* is nonempty.
b. For $i \neq j$, S_i^* *and* S_j^* are disjoint.

Proof of Claim 1

Part a: This is obvious, because $v_i^* \in S_i^*$ for $1 \leq i \leq p$.

Part b: Suppose that $S_i^* \cap S_j^* \neq \emptyset$ for some $i \neq j$. Let $u \in S_i^* \cup S_j^*$. Thus, $w(v_i^*, u) < l^*/2$ and $w(v_j^*, u) < l^*/2$. By Observation 1, $w(v_i^*, v_j^*) \geq l^*$. These three inequalities together imply that the triangle inequality does not hold for the three nodes u, v_i^* and v_j^*. Claim 1b follows.

We now continue with the main proof. Since P_k has *less than* p nodes (Observation 2) and there are p disjoint sets $S_1^*, S_2^*, \ldots, S_p^*$, there must be at least one set, say S_r^* (for some r, $1 \leq r \leq p$), such that $P_k \cap S_r^* = \emptyset$. Therefore, by the definition of S_r^*, we must have for each $u \in P_k$, $w(v_r^*, u) \geq l^*/2$. Since v_r^* is available for selection by **GMM**, and **GMM** selects a node $v \in V - P_k$ for which $\min_{v' \in P_k} w(v, v')$ is a maximum among the nodes in $V - P_k$, it follows that (3) holds even after the $(k + 1)$st addition to P. This completes the proof of Theorem 2.

Our next theorem shows that if **P** $\neq$ **NP**, **GMM** provides the best possible performance guarantee obtainable in polynomial time for **MMFD-TI.**

Theorem 3. *If* **P** $\neq$ **NP**, *no polynomial-time relative approximation algorithm can provide a performance guarantee of* $(2 - \epsilon)$ *for any* $\epsilon > 0$ *for* **MMFDT-TI.**

Proof. We use a construction similar to that presented in the proof of Theorem 1, except that the edge weights are chosen as follows. Let $w(v_i, v_j) = 2 - \epsilon/2$ if $\{x_i, x_j\}$ is in E; otherwise, let $w(v_i, v_j) = 1$. Using the fact that $0 < \epsilon < 1$, it is easy to verify that the resulting distances satisfy the triangle inequality. The proof that the solution value of a placement produced by A for **MMFD-TI** is greater than 1 iff G has a clique of size J is virtually the same as that of Theorem 1.

3. NEAR-OPTIMAL SOLUTIONS TO MAX-AVG FACILITY DISPERSION

In this section, we discuss a relative approximation algorithm for **MAFD** under the triangle inequality assumption (**MAFD-TI**). This heuristic, which we call **GMA,** is shown in Figure 2. It is identical to the **GMM** heuristic of Figure 1, except that in Step 3a, we choose a node $v \in V - P$ for which $\sum_{v' \in P} w(v, v')$ is maximum among all the nodes in $V - P$. Note that the solution value of the placement P produced by **GMA** is equal to $2/p(p - 1) \sum_{x,y \in P} w(x, y)$.

An experimental study of this heuristic is carried out in Erkut and Neuman (1990). We focus on determining the performance guarantee provided by this heuristic. Our next theorem shows that **GMA** is indeed a relative approximation algorithm for **MAFD-TI.** Before presenting that theorem, we introduce some notation which will also be used in Section 4. Let A and B be disjoint subsets of V. Define $W(A) = \sum_{x,y \in A} w(x, y)$ and $W(A, B) = \sum_{x \in A, y \in B} w(x, y)$. (Note that $W(A, B) = W(B, A)$.) Also, for $x \in A$, let $W(x, B) = \sum_{y \in B} w(x, y)$. We now

```
Step 1. Let $v_i$ and $v_j$ be the endpoints of an edge of maximum weight.
Step 2. $P \leftarrow \{v_i, v_j\}$.
Step 3. while ( $|P| < p$ ) do
        begin
            a. Find a node $v \in V - P$ such that $\sum_{v' \in P} w(v, v')$ is maximum
               among the nodes in $V - P$.
            b. $P \leftarrow P \cup \{v\}$.
        end
Step 4. Output $P$.
```

Figure 2. Details of heuristic **GMA.**

give two lemmas which are used several times in the proof of the performance bound for the **GMA** heuristic. The first is an immediate consequence of the pigeon hole principle (Roberts 1984).

Lemma 1. *Let A and B be nonempty and disjoint subsets of V. Then there is a node $x \in A$ such that $W(x, B) \geq W(A, B)/|A|$.*

Lemma 2. *Given an instance of* **MAFD-TI**, *let A and B be nonempty and disjoint subsets of V with $|B| \geq 2$. Then $W(A, B) \geq |A| W(B)/(|B| - 1)$.*

Proof. Let v be an arbitrary node in A. Let $|B| = t$ and $B = \{b_1, b_2, \ldots, b_t\}$. Since $t = |B| \geq 2$, there is at least one edge in B. For each edge $\{b_i, b_j\}$ in $B (i \neq j)$, we have by the triangle inequality,

$$w(v, b_i) + w(v, b_j) \geq w(b_i, b_j) \quad 1 \leq i < j \leq t. \qquad (4)$$

If we sum the inequalities shown in (4), the left-hand side of the sum is $(t - 1) W(v, B)$ because each edge weight $w(v, b_i)(1 \leq i \leq t)$ appears exactly $(t - 1)$ times in the sum. The right-hand side of the sum is simply $W(B)$. Therefore, we get $(t - 1) W(v, B) \geq W(B)$, or

$$W(v, B) \geq W(B)/(t - 1) = W(B)/(|B| - 1). \qquad (5)$$

Since inequality (5) holds for each $v \in A$, we get $W(A, B) \geq |A| W(B)/(|B| - 1)$.

Theorem 4. *Let I be an instance of* **MAFD-TI**. *Let $OPT(I)$ and $GMA(I)$ denote, respectively, the solution values of an optimal placement and that produced by* **GMA** *for the instance I. Then $OPT(I)/GMA(I) \leq 4$.*

Proof. We show by induction that after each addition, the average weight of an edge in P is at least $OPT(I)/4$.

The statement is clearly true after the first addition (which brings two nodes into P) because an edge of maximum weight is added to P. So, assume that $p \geq 3$ and the statement holds after k additions for some $k \geq 1$. We will prove that the statement holds after the $(k + 1)$st addition as well.

For convenience, we use l^* for $OPT(I)$. Let P_k denote the set P after k additions. We have the following two-part observation. The first part is due to the fact that **GMA** adds at least one more node to P. The second is an immediate consequence of the inductive hypothesis.

Observation 3

a. $|P_k| = k + 1 \leq p - 1$.
b. $W(P_k) \geq k(k + 1)/2 \; l^*/4$.

To prove the inductive hypothesis for P_{k+1}, it suffices to show that there is a node $x \in V - P_k$ such that $W(x, P_k) \geq (k + 1)l^*/4$, because this condition in conjunction with Observation 3b implies that the average edge weight is at least $l^*/4$ after the $(k + 1)$st addition to P as well. We now state this condition formally as a claim and present its proof.

Claim 2. *There is a node $x \in V - P_k$ such that $W(x, P_k) \geq (k + 1)l^*/4$.*

Proof of Claim 2. Let P^* denote the set of p nodes in an optimal placement. By the definition of l^*, we have

$$W(P^*) = \frac{p(p - 1)}{2} l^*. \qquad (6)$$

We have two cases to consider, depending upon whether or not P_k and P^* are disjoint.

Case 1. (P_k and P^* are disjoint.) We apply Lemma 2 with P_k as the set A and P^* as the set B. (We can do so because P_k and P^* are disjoint and $|P^*| = p \geq 3$.) We get

$$\begin{aligned} W(P_k, P^*) &\geq |P_k| W(P^*)/(p - 1) \\ &= (k + 1)pl^*/2 \text{ (using 6).} \end{aligned} \qquad (7)$$

Since $W(P_k, P^*) = W(P^*, P_k)$, we have $W(P^*, P_k) \geq (k + 1)pl^*/2$. Now, by Lemma 1, there must be a node $x \in P^*$ such that

$$\begin{aligned} W(x, P_k) &\geq W(P^*, P_k)/p \\ &\geq (k + 1)l^*/2 \text{ (using 7).} \end{aligned}$$

Thus, the node x satisfies a condition which is even stronger than that required by Claim 2.

Case 2. (P_k and P^* are not disjoint.) Let $Y = P^* \cap P_k$ and let $X = P^* - Y$. Since Y is nonempty, we must have:

$$|X| \leq p - 1. \qquad (8)$$

Since X and Y are disjoint and $P^* = X \cup Y$, we have

$$\begin{aligned} W(P^*) &= \frac{p(p - 1)}{2} l^* \\ &= W(X) + W(Y) + W(X, Y). \end{aligned} \qquad (9)$$

We have two subcases depending on $|X|$.

Case 2a. ($|X| \leq 1$.) Note that $X \neq \emptyset$, otherwise, $Y = P^* = P_k$ and so $|P_k| = p$, contradicting Observation 1a. Therefore, in this subcase, we need to consider only the possibility that $|X| = 1$. Then, $P_k = Y$ and so P_k consists of $p - 1$ nodes from P^*. Let x be the node in X. Since **GMA** selects a node $v \in V - P_k$ for which $\sum_{v' \in P_k} w(v, v')$ is a maximum among the nodes in $V - P_k$, and x is available for selection, it follows that **GMA** will produce an optimal solution in this subcase.

Case 2b. ($|X| > 1$ (i.e., $|X| \geq 2$).) Here, from (9), we can conclude that either $W(X) \geq W(P^*)/2$ or $W(Y) + W(X, Y) \geq W(P^*)/2$. We consider these two possibilities separately.

Case 2b.i. ($W(X) \geq W(P^*)/2$) We use Lemma 2 with P_k as set A and X as set B (we can do so because $|X| \geq 2$) to get

$$\begin{aligned} W(X, P_k) &= W(P_k, X) \\ &\geq \frac{(k+1)}{|X| - 1} W(X) \\ &\geq \frac{(k+1)}{|X| - 1} W(P^*)/2 \\ &= \frac{(k+1)}{|X| - 1} p(p-1)l^*/4 \text{ (using 6).} \qquad (10) \end{aligned}$$

From Lemma 1, there is a node $x \in X$ such that $W(x, P_k) \geq W(X, P_k)/|X|$. Combining this observation with (10), we get

$$W(x, P_k) \geq \frac{p(p-1)}{|X|(|X| - 1)} (k+1)l^*/4.$$

Since $p - 1 \geq |X|$ (from 8), the quantity $p(p-1)/(|X|(|X| - 1))$ is greater than 1. Therefore, we get $W(x, P_k) > (k+1)l^*/4$ as required.

Case b.ii. ($W(Y) + W(X, Y) \geq W(P^*)/2$) First note that if $|Y| = 1$, then $W(Y) = 0$ and so $W(X, Y) \geq W(P^*)/2$. Since $Y \subseteq P_k$ and the edge weights are nonnegative, we must have $W(X, P_k) \geq W(X, Y)$. Therefore, $W(X, P_k) \geq W(P^*)/2 = p(p-1)l^*/4$, and by Lemma 1, there must be a node $x \in X$ such that

$$\begin{aligned} W(x, P_k) &\geq \frac{p(p-1)}{|X|} l^*/4 \\ &\geq pl^*/4 \text{ (from 8)} \\ &> (k+1)l^*/4 \text{ (from observation 1a)} \end{aligned}$$

and so Claim 2 holds when $|Y| = 1$. Therefore, for the remainder of this proof, we assume that $|Y| \geq 2$.

Since in this subcase we are assuming that $W(Y) + W(X, Y) \geq W(P^*)/2$, and as observed above, $W(X, Y) \leq W(X, P_k)$, we have

$$W(Y) \geq \frac{W(P^*)}{2} - W(X, P_k). \qquad (11)$$

Let $Q = P_k - Y$. Note that Q and Y are disjoint and so $|Q| = |P_k| - |Y| = (k+1) - |Y|$. We apply Lemma 2 with Q as the set A and Y as set B (recall that $|Y| \geq 2$). We get

$$W(Q, Y) \geq \frac{(k+1-|Y|)}{(|Y| - 1)} W(Y). \qquad (12)$$

Since $P_k = Q \cup Y$ and the sets Q and Y are disjoint, we also have,

$$\begin{aligned} W(P_k) &\geq W(Y) + W(Q, Y) \\ &\geq W(Y)\left[1 + \frac{k+1-|Y|}{|Y| - 1}\right] \text{ (from 12)} \\ &= W(Y)\frac{k}{(|Y| - 1)}. \qquad (13) \end{aligned}$$

We now apply Lemma 2 with X as the set A and P_k as the set B. We get,

$$\begin{aligned} &W(X, P_k) \\ &\geq |X| W(P_k)/(|P_k| - 1) \\ &= |X| W(P_k)/k \text{ (from Observation 1a)} \\ &\geq |X| W(Y)/(|Y| - 1) \text{ (from 13)} \\ &\geq \frac{|X|}{(|Y| - 1)} [W(P^*)/2 - W(X, P_k)] \\ &\qquad\qquad \text{(from (11)} \quad (14) \end{aligned}$$

Rearranging (14) we get,

$$W(X, P_k)\left[\frac{|X| + |Y| - 1}{|Y| - 1}\right] \geq \frac{|X|}{|Y| - 1} W(P^*)/2.$$

Noting that $|X| + |Y| = |P^*| = p$, substituting for $W(P^*)$ from (6), and eliminating the common denominator $|Y| - 1$, we get

$$W(X, P_k)(p-1) \geq |X| p(p-1)l^*/4.$$

That is, $W(X, P_k) \geq |X| pl^*/4$. From this inequality and Lemma 1, we conclude that there must be node $x \in X$ such that $W(x, P_k) \geq pl^*/4 > (k+1)l^*/4$ (from Observation 1a).

This completes the proof of Claim 2 and also that of Theorem 4.

Theorem 4 shows that the performance guarantee provided by **GMA** is no worse than 4. However, the guarantee may well be less. The following result shows that the guarantee cannot be less than 2.

Theorem 5. *For any instance I of* **MAFD-TI**, *let $OPT(I)$ and $GMA(I)$ denote, respectively, the solution values of an optimal placement and that produced by* **GMA** *for the instance I. For any $\epsilon > 0$, there is an instance I_ϵ for which $OPT(I_\epsilon)/GMA(I_\epsilon) \geq 2 - \epsilon$.*

Proof. Consider the **MAFD** instance I described below. This instance has a total of $2p$ nodes and p facilities are to be located. For convenience in description, we partition the set of $2p$ nodes into three sets called X, Y, and Z. The set X has p nodes (denoted by $x_1, x_2, \ldots, x_p$), the set Y has $p - 2$ nodes (denoted by $y_1, y_2, \ldots, y_{p-2}$), and Z has two nodes (denoted by z_1 and z_2). The edge weights are chosen as follows. For any distinct nodes x_i and $x_j \in X$, $w(x_i, x_j) = 2$. Also $w(z_1, z_2) = 2$. All other edge weights are 1. It is straightforward to verify that the distances satisfy the triangle inequality. The set X is an optimal placement and its solution value ($OPT(I)$) is 2 (because every edge in X has a weight of 2). We can force **GMA** to place the first two facilities at z_1 and z_2 because $w(z_1, z_2) = 2$ is a maximum edge weight. It is easy to verify that in the subsequent ($p - 2$) steps, **GMA** can be forced to choose all the ($p - 2$) nodes from the set Y. Thus, we force **GMA** to return $Z \cup Y$ as the placement. The solution value ($GMA(I)$) corresponding to this placement is given by

$$GMA(I) = \frac{2}{p(p-1)}\left[2 + \left(\frac{p(p-1)}{2} - 1\right)\right]$$

because in the placement, one edge (namely, that between z_1 and z_2) is of weight 2 and each remaining edge is of weight 1. A bit of simplification shows that $GMA(I) = 1 + 2/(p(p - 1))$. The ratio $OPT(I)/GMA(I)$ is given by

$$\frac{OPT(I)}{GMA(I)} = \frac{2}{1 + 2/(p(p-1))}.$$

Clearly, the ratio can be made arbitrarily close to 2 by choosing p to be sufficiently large.

4. DISPERSION PROBLEMS IN ONE AND TWO DIMENSIONS

The 1-dimensional dispersion problems are restricted versions of **MMFD** and **MAFD**, where the node set V consists of a set of n points (denoted by $x_1, x_2, \ldots, x_n$) on a line. Thus $w(x_i, x_j) = |x_i - x_j|$. We denote these problems by **1D-MMFD** and **1D-MAFD**, respectively. Similarly, in the case of the 2-dimensional dispersion problems (denoted by **2D-MMFD** and **2D-MAFD**, respectively), the node set V is a set of n points in $\mathcal{R}^2$ and the distance between a pair of points is the Euclidean distance. It is known that **1D-MMFD** can be solved in polynomial time using a dynamic programming approach and that **2D-MMFD** is NP-hard (Wang and Kuo). Accordingly, we consider **1D-MAFD** and **2D-MAFD** in this subsection.

4.1. A Polynomial-Time Algorithm for 1D-MAFD

Our polynomial-time algorithm for **1D-MAFD** is also based on a dynamic programming approach. It runs in $O(\max\{n \log n, pn\})$ time. In the development of the dynamic programming formulation for this problem, we use the notation introduced in Section 3. In studying the formulation, the reader should bear in mind that maximizing the average distance is equivalent to maximizing the sum of the distances.

We begin by sorting the points into increasing order. Let $V = \{x_1, x_2, \ldots, x_n\}$ denote the points in sorted order. Consider a point x_j and an integer $k \leq \min(j, p)$. For each such combination of x_j and k, the dynamic programming algorithm considers choosing k points from $\{x_1, x_2, \ldots, x_j\}$ to maximize a certain quantity described below. Let $C = A \cup B$ be a set of p points consisting of a subset $A \subseteq \{x_1, \ldots, x_j\}$ such that $|A| = k$, and a subset $B \subseteq \{x_{j+1}, \ldots, x_n\}$ such that $|B| = p - k$. For each pair of points x_u in A and x_v in B, we have

$$w(x_u, x_v) = w(x_j, x_u) + w(x_j, x_v). \tag{15}$$

If we sum (15) over the points in B, we get

$$W(x_u, B) = (p - k)w(x_j, x_u) + W(x_j, B). \tag{16}$$

If we sum (16) over the points in A, we get

$$W(A, B) = (p - k)W(x_j, A) + kW(x_j, B). \tag{17}$$

Hence,

$$\begin{aligned} W(C) &= W(A \cup B) \\ &= W(A) + W(B) + W(A, B) \\ &= W(A) + (p - k)W(x_j, A) \\ &\quad + W(B) + kW(x_j, B). \end{aligned} \tag{18}$$

The optimization goal of **1D-MAFD** is to maximize $W(C)$. Suppose that for a given k, we want to choose a set C maximizing $W(C)$, but subject to the constraint that A contains k points and B contains $p - k$ points. Equation (18) shows that under this constraint, the choice of B has no effect on the choice of A. For a given subset $A \subseteq \{x_1, \ldots, x_j\}$ such that $|A| = k$, define

$$f_{k,j}(A) = W(A) + (p - k)W(x_j, A). \tag{19}$$

Let $OPT_{k,j}$ be a k-element subset of $\{x_1, \ldots, x_j\}$ that maximizes $f_{k,j}$. Note that $OPT_{p,n}$ is the solution to the **1D-MAFD** problem instance. The dynamic programming algorithm computes all the values of $f_{k,j}(OPT_{k,j})$, and then uses these values to find an optimal solution to the **1D-MAFD** instance. The algorithm finds the values of $f_{k,j}(OPT_{k,j})$ by considering increasing values of j. For a given value of j, the algorithm considers increasing values of k.

Now consider how to compute $f_{k,j}(OPT_{k,j})$. Note that $OPT_{k,j}$ either includes the point x_j or excludes it. if $x_j \notin OPT_{k,j}$, then $OPT_{k,j}$ must be $OPT_{k,j-1}$, for which we have, using (19),

$$\begin{aligned} & f_{k,j}(OPT_{k,j-1}) \\ &= W(OPT_{k,j-1}) + (p-k)W(x_j, OPT_{k,j-1}) \\ &= W(OPT_{k,j-1}) + (p-k)W(x_{j-1}, OPT_{k,j-1}) \\ &\quad + k(p-k)(x_j - x_{j-1}) \\ &= f_{k,j-1}(OPT_{k,j-1}) + k(p-k)(x_j - x_{j-1}). \end{aligned} \tag{20}$$

If $x_j \in OPT_{k,j}$, then $OPT_{k,j}$ must be $OPT_{k-1,j-1} \cup \{x_j\}$, and again using (19) we have,

$$\begin{aligned} & f_{k,j}(OPT_{k-1,j-1} \cup \{x_j\}) \\ &= W(OPT_{k-1,j-1}) + (p-k)(W(x_{j-1}, OPT_{k-1,j-1}) \\ &\quad + (k-1)(x_j - x_{j-1})) + W(x_{j-1}, OPT_{k-1,j-1}) \\ &\quad + (k-1)(x_j - x_{j-1}) \\ &= W(OPT_{k-1,j-1}) + (p-k+1)W(x_{j-1}, OPT_{k-1,j-1}) \\ &\quad + (k-1)(p-k+1)(x_j - x_{j-1}) \\ &= f_{k-1,j-1}(OPT_{k-1,j-1}) \\ &\quad + (k-1)(p-k+1)(x_j - x_{j-1}). \end{aligned} \tag{21}$$

Equations (20) and (21) show that given the sets $OPT_{k,j-1}$ and $OPT_{k-1,j-1}$, and the values of $f_{k,j-1}(OPT_{k,j-1})$ and $f_{k-1,j-1}(OPT_{k-1,j-1})$, we can compute $OPT_{k,j}$ and $f_{k,j}(OPT_{k,j})$. To complete the dynamic programming formulation, we note that the boundary conditions are $OPT_{0,j} = \emptyset$, $f_{0,j}(OPT_{0,j}) = 0 (1 \leq j \leq n)$, $OPT_{1,1} = \{x_1\}$, and $f_{1,1}(OPT_{1,1}) = 0$.

The details of the algorithm are shown in Figure 3. The algorithm first computes (Step 4) the entries of the array F (which corresponds to the function f in the above dynamic programming formulation) and then uses these entries to construct an optimal placement P (Step 5). This implementation obviates the need for the sets $OPT_{k,j}$ used in the formulation. The running time of the algorithm is $O(n \log n)$ for sorting plus $O(pn)$ to carry out the dynamic programming (there are $O(pn)$ entries to compute, and each entry can be computed in constant time). Thus, the overall running time is $O(\max(n \log n, pn))$.

The above discussion is summarized in the following theorem which will be used in the next subsection.

Theorem 6. *An optimal solution to any instance of* **1D-MAFD** *given by a set V of n points and an integer $p \leq n$ can be obtained in $O(\max\{n \log n, pn\})$ time.*

4.2. A Heuristic for 2D-MAFD

It is open whether **2D-MAFD** is NP-hard. Note that **GMA** (Section 3) provides a performance guarantee of 4 for **2D-MAFD.** Here, we first present a heuristic for **2D-MAFD** which provides a performance guarantee of $4(\sqrt{2} - 1) \approx 1.657$ and then show how this heuristic can be modified to obtain another heuristic which provides an asymptotic performance guarantee of $\pi/2 \approx 1.571$. These heuristics use our polynomial algorithm for **1D-MAFD.**

We assume that an instance of **2D-MAFD** is given by a set $V = \{v_1, v_2, \ldots, v_n\}$ of n points (where each point v_i is specified by a pair of coordinates (x_i, y_i)) and an integer $p \leq n$. The steps of this heuristic (called **PROJECT_4**) are shown in Figure 4. The performance guarantee provided by **PROJECT_4** is indicated in the following theorem.

Theorem 7. *Let I be an instance of* **2D-MAFD.** *Let $OPT(I)$ and $PROJECT\_4(I)$ denote, respectively, the solution values of an optimal placement and that produced by* **PROJECT_4** *for the instance I. Then $OPT(I)/PROJECT\_4(I) \leq 4(\sqrt{2} - 1)$.*

Proof. Recall that maximizing average distance is equivalent to maximizing the sum of the distances. So we will present the proof in terms of the sum of the distances between pairs of points.

Let P^* be an optimal placement. For convenience, we use the term *edge* to refer to the line segment between a pair of (distinct) points in P^*. For an edge $e \in P^*$, let $l(e)$ denote its length (i.e., the Euclidean distance between the end points of e). Note that $OPT(I) = \sum_{e \in P^*} l(e)$. Given an edge e, let $l_x(e)$, $l_y(e)$, $l_v(e)$, and $l_w(e)$ denote the magnitudes of the projections of e on the X, Y, V and W axes, respectively. We have the following claim.

Claim 3. $l_x(e) + l_y(e) + l_v(e) + l_w(e) \geq (1 + \sqrt{2})l(e)$.

Proof of Claim 3. Since we are considering the sum of the projections, assume without loss of generality that the angle θ of e with respect to the X axis is between 0° and 45°, as shown in Figure 5. From that figure, we have

$l_x(e) = l(e)\cos\theta$, $l_y(e) = l(e)\sin\theta$,

$l_v(e) = l(e)\cos(45° - \theta)$, and $l_w(e) = l(e)\sin(45° - \theta)$.

```
(*-- In the following, array F represents the function $f$ in the formulation. --*)
Step 1. Sort the given points, and let $\{x_1, x_2, \ldots, x_n\}$ denote the points in increasing order.
Step 2. for $j := 1$ to $n$ do
            F $[0, j] \leftarrow 0$;
Step 3. F $[1,1] \leftarrow 0$.
Step 4. (*-- Compute the value of an optimal placement --*)
        for $j := 2$ to $n$ do
            for $k := 1$ to min $(p, j)$ do
                begin
                    $t_1 \leftarrow F[k, j-1] + k(p-k)(x_j - x_{j-1})$;
                    $t_2 \leftarrow F[k-1, j-1] + (k-1)(p-k+1)(x_j - x_{j-1})$;
                    if $t_1 > t_2$, then (*-- do not include $x_j$ --*)
                        $F[k, j] \leftarrow t_1$
                    else (*-- Include $x_j$ --*)
                        $F[k, j] \leftarrow t_2$;
                end;
Step 5. (*-- Construct an optimal placement --*)
        $P \leftarrow \{x_1\}$; $k \leftarrow p$; $j \leftarrow n$;
        while $k > 1$ do
            begin
                if $F[k, j] = F[k-1, j-1] + (k-1)(p-k+1)(x_j - x_{j-1})$,
                  then (*-- $x_j$ to be included in optimal placement --*)
                    begin
                        $P \leftarrow P \cup \{x_j\}$; $k \leftarrow k-1$;
                    end;
                $j \leftarrow j-1$;
            end;
Step 6. Output $P$.
```

Figure 3. Details of the algorithm for **1D-MAFD.**

```
Step 1. Obtain the projections of the given set $V$ of points on each of the four axes defined by the
        equations $y = 0$ ($X$ axis), $y = x$ ($V$ axis), $x = 0$ ($Y$ axis), and $y = -x$ ($W$ axis).
Step 2. Find optimal solutions to each of the four resulting instances of 1D-MAFD.
Step 3. Return the placement corresponding to the best of the four solutions found in Step 2.
```

Figure 4. Details of heuristic **PROJECT_4.**

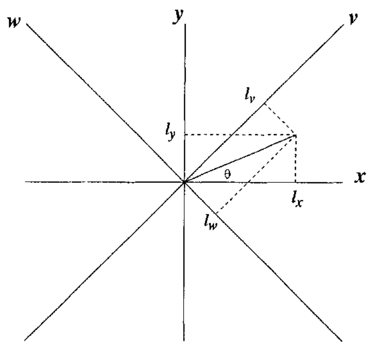

Figure 5. Proof of claim 1.

Let $s(e) = l_x(e) + l_y(e) + l_v(e) + l_w(e)$. Using well known trigonometric identities and the fact that sin 45° = cos 45° = $1/\sqrt{2}$, it is not difficult to verify that

$$s(e) = l(e)[\sin \theta + (1 + \sqrt{2})\cos \theta]. \tag{22}$$

From (22), it is easy to verify that the minimum value of $s(e)$ occurs when $\theta = 0°$ or $\theta = 45°$ and that this minimum value is $(1 + \sqrt{2})l(e)$. This completes the proof of Claim 3.

We now continue with the main proof. Note that Claim 3 holds for each edge in P^*. So if we sum up the result of Claim 3 over all the edges in P^*, we get

$$\begin{aligned}\sum_{e \in P^*} l_x(e) + l_y(e) + l_v(e) + l_w(e) \\ \geq (1 + \sqrt{2}) \sum_{e \in P^*} l(e) \\ = (1 + \sqrt{2}) OPT(I). \qquad (23)\end{aligned}$$

If we let $S_x = \sum_{e \in P^*} l_x(e)$, and use analogous definitions for S_y, S_v, and S_w, we get from (23),

$$S_x + S_y + S_v + S_w \geq (1 + \sqrt{2}) OPT(I). \qquad (24)$$

From the last inequality, it follows that

$$\max(S_x, S_y, S_v, S_w) \geq \frac{1 + \sqrt{2}}{4} OPT(I). \qquad (25)$$

Since **PROJECT_4** chooses the best placement from optimal solutions for the four **1D-MAFD** instances, it follows that $PROJECT\_4(I) \geq \max(S_x, S_y, S_v, S_w)$. We thus have from inequality (25),

$$\frac{OPT(I)}{PROJECT\_4(I)} \leq 4/(1 + \sqrt{2}) = 4(\sqrt{2} - 1)$$

as indicated in the statement of the theorem.

The performance guarantee provided by **PROJECT_4** is approximately 1.657. By projecting the given set of points on more axes, it is possible to achieve an asymptotic performance guarantee of $\pi/2 \approx 1.571$, as shown below.

Theorem 8. *Let I be an instance of* **2D-MAFD** *and let OPT(I) denote the solution value of an optimal placement for I. Then, for any fixed* $\epsilon > 0$ *there is a polynomial-time approximation algorithm* P_ϵ *which produces a placement with solution value* $P_\epsilon(I)$ *such that* $OPT(I)/P_\epsilon(I) \leq (\pi/2 + \epsilon)$.

Proof. Given $\epsilon > 0$, let k be the smallest even positive integer satisfying the condition

$$k \tan(\pi/2k) \leq \pi/2 + \epsilon. \qquad (26)$$

Such a k exists because $\lim_{k \to \infty} k \tan(\pi/2k) = \pi/2$, and this limit is approached from above. Algorithm P_ϵ first projects the given set of points on k axes (denoted by $X_1, X_2, \ldots, X_k$) such that the angle between any pair of successive axes is π/k. It then solves the **1D-MAFD** problem on each of the k axes and outputs the best of the placements found. In view of (26), Theorem 8 would follow by proving that $OPT(I)/P_\epsilon(I)$ is bounded by $k \tan(\pi/2k)$.

The proof is very similar to that of Theorem 7. Let P^* denote an optimal placement and consider an edge e of length $l(e)$ in P^*. Without loss of generality, let θ $(0 \leq \theta \leq \pi/k)$ be the angle of e with respect to the X_1 axis. Let $l_i(e)$ denote the magnitude of the projection of e on the X_i axis $(1 \leq i \leq k)$, and let $s(e) = \sum_{i=1}^{k} l_i(e)$. It is easy to verify that

$$s(e) = l(e)\left[\sum_{r=1}^{k/2} \cos(r\pi/k - \theta) + \sum_{r=0}^{k/2-1} \cos(r\pi/k + \theta)\right]. \qquad (27)$$

Tedious, but straightforward, calculations show that

$$s(e) = l(e)\left[\frac{\sin\theta + \sin(\pi/k - \theta)}{1 - \cos(\pi/k)}\right]. \qquad (28)$$

From (28), it is easy to verify that the minimum value of $s(e)$ occurs when $\theta = 0$ or $\theta = \pi/k$ and that this minimum value is $l(e)/\tan(\pi/2k)$. Thus $s(e) \geq l(e)/\tan(\pi/2k)$. The remainder of the proof to show that $OPT(I)/P_\epsilon(I) \leq k \tan(\pi/2k)$ is virtually the same as of Theorem 7 (keeping in mind that the number of axes is k instead of 4).

We note that Theorem 8 generalizes the bound of Theorem 7 in that choosing $\epsilon = 4(\sqrt{2} - 1) - \pi/2$ (≈ 0.0861) leads to projection on $k = 4$ axes. Also, to achieve a performance guarantee of 1.571 (an approximate value of $\pi/2$), 80 projections are needed; in general, the number of projections goes to ∞, as ϵ approaches 0.

5. CONCLUSIONS

The results of this paper, prior results, and open problems are summarized in two tables. Table I shows the complexity results for solving these problems optimally, while Table II shows performance guarantee results for heuristics. In these tables, prior results are indicated through appropriate citations; our results are indicated by specifying the corresponding theorems.

ACKNOWLEDGMENT

An extended abstract containing some of the results in this paper appears in the *Proceedings of the 1991 Workshop on Algorithms and Data Structures* (*WADS*-91), Ottawa, Canada, August 1991. The research of the first author was supported by NSF grant CCR-89-05296, and the research of the second author was supported by NSF grants CCR-88-03278 and CCR-90-06396. The research of the third author was supported by the Faculty Research Award Program of the University at Albany.

We thank Professor Binay K. Bhattacharya (Simon Fraser University, Burnaby, Canada) for bringing the paper by Wang and Kuo (1988) to our attention. We

Table I
Complexity Results for Dispersion Problems

Problem	MAX-MIN	MAX-AVG
General	NP-hard (Erkut 1990)	NP-hard (Hansen and Moon 1988)
Triangle Inequality	NP-hard (Erkut 1990)	NP-hard (Hansen and Moon 1988)
1D Version	$O(\max\{pn, n \log n\})$ (Wang and Kuo 1988)	$O(\max\{pn, n \log n\})$ (Theorem 6)
2D Version	NP-hard (Wang and Kuo 1988)	Open

Table II
Performance Guarantee Results for NP-hard Dispersion Problems

	MAX-MIN		MAX-AVG	
Problem	Upper Bound (Best Known Guarantee)	Lower Bound (Intrinsic Limit, Assuming $P \neq NP$)	Upper Bound (Best Known Guarantee)	Lower Bound (Intrinsic Limit, Assuming $P \neq NP$)
General	—	No guaranteed ratio (Theorem 1)	Open	Open
Triangle Inequality	2 (Theorem 2)	2 (Theorem 3)	4 (Theorem 4)	Open
2D Version	2 (Theorem 2)	Open	$\pi/2$ asymp. (Theorem 8)	Open

are indebted to the referees for their careful reading of our manuscript and their valuable comments. In particular, one of the referees suggested presenting the conclusions in Section 5 in the form of tables.

REFERENCES

Church, R. L., and R. S. Garfinkel. 1978. Locating an Obnoxious Facility on a Network. *Trans. Sci.* **12,** 107–118.

Chandrasekharan, R., and A. Daughety. 1981. Location on Tree Networks: *p*-Centre and *n*-Dispersion Problems. *Math. Opns. Res.* **6,** 50–57.

Dasarathy, B., and L. J. White. 1980. A Maxmin Location Problem. *Opns. Res.* **28,** 1385–1401.

Erkut, E. 1990. The Discrete *p*-Dispersion Problem. *Eur. J. Opnl. Res.* **46,** 48–60.

Erkut, E., and S. Neuman. 1989. Analytical Models for Locating Undesirable Facilities. *Eur. J. Opnl. Res.* **40,** 275–291.

Erkut, E., and S. Neuman. 1990. Comparison of Four Models for Dispersing Facilities. *INFOR* **29,** 68–85.

Erkut, E., and T. S. Öncü. 1991. A Parametric 1-Maximin Location Problem. *J. Opnl. Res., Soc.* **42,** 49–55.

Erkut, E., T. Baptie and B. von Hohenbalken. 1990. The Discrete *p*-Maxian Location Problem. *Comput. and Opns. Res.* **17,** 51–61.

Garey, M. R., and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, San Francisco.

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

Hansen, P., and I. D. Moon. 1988. Dispersing Facilities on a Network. Presentation at the TIMS/ORSA Joint National Meeting, Washington, D.C.

Horowitz, E., and S. Sahni. 1984. *Fundamentals of Computer Algorithms.* Computer Science Press, Rockville, Maryland.

Kuby, M. J. 1987. Programming Models for Facility Dispersion: The *p*-Dispersion and Maxisum Dispersion Problems. *Geog. Anal.* **19,** 315–329.

Melachrinoudis, E., and T. P. Cullinane. 1986. Locating an Undesirable Facility With a Minimax Criterion. *Eur. J. Opnl. Res.* **24,** 239–246.

Preparata, F. P., and Shamos, M. I. 1985. *Computational Geometry: An Introduction.* Springer-Verlag, New York.

Roberts, F. S. 1984. *Applied Combinatorics.* Prentice-Hall, Englewood Cliffs, New Jersey.

Steuer, R. E. 1986. *Multiple Criteria Optimization: Theory and Application.* John Wiley, New York.

Tamir, A. 1991. Obnoxious Facility Location on Graphs. *SIAM J. Disc. Math.* **4,** 550–567.

White, D. J. 1991. The Maximal Dispersion Problem and the 'First Point Outside the Neighborhood' Heuristic. *Comput. and Opns. Res.* **18,** 43–50.

White, D. J. 1992. The Maximal Dispersion Problem. *J. Applic. Math. in Bus. and Ind.* (to appear).

Wang, D. W., and Y. S. Kuo. 1988. A Study of Two Geometric Location Problems. *Infor. Proc. Letts.* **28,** 281–286.