アルゴリズム 44−10
(1995. 3. 17)

設置散乱および遠隔部分グラフについて

バルン チャンドラ
モーナシュ大学

マグナス ハルダースソン
北陸先端科学技術大学院大学 (情)

散乱問題は, 点の集合を可能な限り互いに離して配置する問題である. この問題は, 施設の位置の選定や経営決定学の分野に, 多くの応用をもつ. 現在までの重要な研究は, 二つの特定の散乱の尺度に着目してきている. 我々は, 現実の問題に動機づけられ, いくつかの自然な遠隔の尺度を導入し, 考察する. そして, 自明でないパーフォマンスの境界をもつ, 最初のアルゴリズムを示す.

設置散乱、ネットワーク最適化、近似アルゴリズム

Facility Dispersion and Remote Subgraphs

Barun Chandra
Department of Computer Science
Monash University
Victoria, Australia
barun@cs.monash.edu.au

Magnús M. Halldórsson
School of Information Science
Japan Adv. Institute of Science and Tech.
Ishikawa, Japan
magnus@jaist.ac.jp

Dispersion problems involve arranging a set of points as far away from each other as possible. They have numerous applications in the location of facilities and in management decision science. Must work to date has focused on two particular measures of the dispersion. We study and introduce several natural measures of remoteness, motivated by real-life problems, and present the first algorithms with non-trivial performance bounds.

facility dispersion, network optimization, approximation algorithms

1 Introduction

As the proud and agressive owner of the McWoofer burger chain, you are given the opportunity to build p new franchises to be located from any of n choice locations available. After ensuring that the available slots are all attractive in terms of cost, visibility etc, what would your criteria be for locating the franchises *relative to each other*?

Locating two identical burger joints next to each other would not increase the number of customers, and thus halve the amount of business that either of them could do if apart. Non-competitiveness is a concern here, which can be alleviated by properly *dispersing* the facilities.

Dispersion problems. The franchise location example is one of many problems where we seek a subset of points that are, in some sense, as *remote* from each other as possible. Dispersion has found applications to diverse areas: locating undesirable or interfering facilities; aiding decision analysis with multiple objectives; marketing a set of products with different attributes; providing good starting solutions for 'grand-tour' TSP heuristics. Dispersion is also of combinatorial interest, as a measure of remote subgraphs.

Which measure of remoteness should be applied? The focus of the literature has been on the minimum distance between any pair in the selected set, with some work done on the average pairwise distance. The proper measure is very much a question of the problem under study, and several of the applications we consider give rise to quite different notions of remoteness.

In this paper, we give the first approximate algorithms for dispersion problems under several measures of remoteness. We unify these and the other dispersion problems in the literature by a novel formalization, where each dispersion problem P corresponds to a certain class of graphs Π. This by itself suggests various interesting new dispersion problems.

Applications. Location theory is a branch of management science/operations research that deals with the optimal location of facilities. Most of that work deals with desirable facilities, where nearness to users or each other is preferable. More recently, some number of papers have considered the opposite objective of placing the facilities far from users or each other. We are interested in intra-facility dispersion cases.

Strategic facilities that are to be protected from simultaneous enemy attacks is one example suggested by Moon and Chaudhry [9]. This could involve oil tanks [9] missile silos, or ammunition dumps [4], which would be preferable to keep separated from each other to minimize the damage of a limited attack. Limiting the range and possible spread of fire or accidents at hazardous installations is also helped by proper spacing [8].

Non-competition is another motivation for dispersal, as in the case of the burger chain example. This may apply to other types of franchises such as gasoline stations, or to the location of radio transmitters with the objective of minimizing interference. Dispersal has also been seen to be desirable in order to obtain a more effective and/or fair coverage of a region. In fact, White [12] cites some example of government regulars to that effect, including firehouses and ambulance stations in New York City.

Yet another dispersal issue in facility location involves undesirable interaction between all facilities that grows inversely with distance [4]. This may apply to dormitories at a university, or chairs during an examination.

All of the above applications suggest a metric sensitive to the largely two-dimensional nature of our world. This is not the case, however, for the problems outside the location area.

White [12] considers dispersion problem motivated by *multiple objective analysis* in decision theory. Given a potential set of *actions* for a decision maker, we are to find a fixed-size subset of these that are as dispersed as possible, for further consideration by the decision-makers. White lists several studies that have used dispersal as a filter of the possible choices, involving e.g. oil drilling, media selection, and forestry management

Dispersion also has application to product development. The marketing of new but related products is helped by diversity [2]. From dimensions including price, quality, shape, packaging, etc., a set of products can be produced which are likely to gain greater market coverage if they are easily distinguished rather than all very similar.

Dispersion Formulations. Let us now formulate the dispersion problems that have been considered in the literature.

The input is an integer p and a network $G = (V, V \times V)$ with a distance function d on the edges satisfying the triangular inequality ($d(u,v) \leq d(u,z) + d(z,v)$). The output is a set V' of p vertices. The objective is a sum of some of the edges in the subgraph induced by V'.

Remote-Edge	$\min_{v,u \in V'} d(u,v)$
Remote-Clique	$\sum_{v,u \in V'} d(u,v)$
Remote-Star	$\min_{v \in V'} \sum_{u \in V'} d(u,v)$
Remote-PseudoForest	$\sum_{v \in V'} \min_{u \in V'} d(u,v)$

Our choice of names comes from the graph structure induced by the solutions to these problems. The names used in the literature are quite different and varied. Remote-Edge is known as *p-Dispersion* [2, 8] and *Max-Min Facility Dispersion* [10]; Remote-Clique as *Maxisum Dispersion* [8] and *Max-Avg Facility Dispersion* [10]; Remote-Star as *MaxMinSum dispersion* [4]; and Remote-PseudoForest as *p-Defense* [9] and *MaxSumMin dispersion* [4].

Related work. A considerable body of work has appeared on facility dispersion problems in the management science and operations research literature; [9, 2, 3, 4, 12, 8]. Most previous work has focused on either easily solvable tree networks, or empirical studies of heuristics, Only recently have some of these heuristics been analyzed analytically.

For Remote-Edge, Tamir [11], White [12, 13] and Ravi, Rosenkrantz and Tayi [10] independently showed that a "furthest-point greedy" algorithm is 2-approximate. The latter also showed that obtaining an approximation strictly less than 2 was NP-hard.

For Remote-Clique Ravi et al. gave a greedy algorithm that they showed came within a factor of 4, while Hassin, Rubinstein and Tamir [6] have recently given elegant proofs of two 2-approximate algorithms. This problem has also been studied for non-metric graphs under the name Heavy Subgraph Problem by Kortsarz and Peleg [7] and they presented a sequence of algorithms that converge with a performance ratio of $O(n^{3.865})$.

No analytic bounds have been previously given for either Remote-Star or Remote-Pseudoforest problems. Moon and Chaudhry [9] suggested the star problem. Erkut and Neuman [4] gave an branch-and-bound algorithm that solves all four of these problems simultaneously.

All of these problem can be seen to be NP-hard by a reduction from Clique. The same reduction also establishes that Remote-Edge cannot be approximated within a constant smaller than 2 [10]. Further, when the weights are not constrained to be metric, the problem is as hard to approximate as Max Clique, which implies that $n^{1/6}$ approximation is NP-hard [1]. Reductions from Clique also yield the same hardness for the PseudoForest problem [5]. On the other hand, no hardness results are known for Remote-Clique and Remote-Star.

Our formalism and further problems. We suggest a unifying formalism for describing these and several other dispersal problems. In each problem, the output is a vertex subset and the objective is a function of the subgraph induced on that vertex set. More precisely, the objective function is a sum of a certain set of edges within that subgraph, and this edge set is of minimum weight among edge subsets satisfying certain graph property.

For instance, in the Remote-Star problem, the objective function is a sum of the edge weights of a minimum-weight *star* spanning the vertex set. In general, for a property Π of graphs, the objective function for Remote-Π is the weight of the minimum-weight subgraph satisfying property Π within the induced subgraph on X.

A *pseudo-forest* is the undirected equivalent of a directed graph where each vertex is of out-degree one. Hence, each component contains as many edges as vertices. Hence, the Remote-Pseudoforest problem.

We are led to considering other Remote-Π problems for other important graph properties. Remote-Tree and Remote-Cycle was considered by Halldórsson, Iwano, Katoh, and Tokuyama [5], under the names

Remote-MST and Remote-TSP, respectively. They showed that the greedy furthest-point algorithm approximates both problem within a factor of 4, while obtaining a ratio less than 2 is NP-hard. They proposed Remote-Matching as an open problem.

Summary of results. In the current paper, we present the following results:

1. $O(\log p)$-approximate algorithms for Remote-PseudoForest and Remote-Matching, and matching lower bounds for these algorithms.
2. 2-approximate algorithm for Remote-Star
3. 2 or 4-approximate algorithm for various related *min-max* problems.

In contrast with *all* previous results for these problems, our algorithms for the Remote-PseudoForest and Remote-Matching problems are neither *greedy* in nature nor does do they use a one-shot computation of a simple structure. In fact, we show that any greedy algorithm is bound to fail miserably, and that the solutions to these problem do not relate well to any of the easily computable structures studied before.

As an interesting comparison in the power of the various algorithms, we have similarly analyzed some of the corresponding *min-max* problems.

Overview of paper. In the next section, we consider Remote-PseudoForest and Remote-Matching which behave similarly for most types of algorithms, and present the first algorithms non-trivial worst-case performance bounds. In section 3, we use recent results of Hassin, Rubinstein and Tamir [6] to give good approximation for Remote-Star. In order to compare the power of the various algorithms, we study in section 4 some of the corresponding *min-max* problems, and obtain constant-factor approximations.

Remark. The results presented constitute preliminary work in progress.

1.1 Notation

For a vertex set $X \subseteq V$, let $\Pi(X)$ ($\pi(X)$) denote the maximum (minimum) weight set of edges in the induced subgraph $G[X]$ that form a graph satisfying property Π, respectively. In particular, we consider $\mathsf{STAR}(X)$ and $\mathsf{star}(X)$ (max- and min-weight spanning star), $\mathsf{pf}(X)$ (min-weight pseudoforest), $\mathsf{CYCLE}(X)$ (max-weight tour), $\mathsf{TREE}(X)$ and $\mathsf{tree}(X)$ (max- and min-weight spanning tree), and $\mathsf{MAT}(X)$ and $\mathsf{mat}(X)$ (min- and max-weight matching).

For a set of edges E', let $wt(E')$ denote the sum of the weights of the edges of E'. Let HEU be the vertex set selected by the algorithm in question, and let OPT be the vertex set of an optimal solution. Let $AvgEDGE(X)$ be the average weight of an edge in $G[X]$.

2 Remote-PseudoForest and Remote-Matching

We first consider the problem where we want to select p vertices so as to maximize the minimum weight pseudoforest (pf). A pseudoforest is a collection of directed edges so that the outdegree of each vertex is at least one, and hence $wt(\mathsf{pf})$ is the sum of the nearest neighbor distances. More formally, given a set of vertices W, and $x \in W$, define the nearest neighbor of x in W, $NN_W(x)$, to be the vertex (other than x) in W which minimizes $d(x,y)$. $wt(\mathsf{pf}(W))$ is defined to be $\sum_{x\in W} d(x, NN_W(x))$. With a reduction from Clique it is easy to see that Remote-PseudoForest is NP-Hard, in the general case it is at least as hard to approximate as the independent set problem, and even with the triangle inequality it is hard to approximate to better than a factor of 2 ([5]). We assume the triangle inequality.

A technique which has worked for some dispersion problems is the greedy approach. For example, the furthest-point greedy algorithm (henceforth called GREEDY) works by successively selecting a vertex whose distance to the already selected vertices is maximized. GREEDY works well for Remote-Edge ([11, 12, 13, 10]) and Remote-Tree and Remote-Cycle [5]. However, for Remote-PseudoForest and Remote-Matching, GREEDY can do very badly, as can be seen from the following example: $V = \{a_1, a_2, b_1, b_2, \ldots, b_{n-2}\}$,

$d(a_i, b_j) = 1$, $d(a_1, a_2) = 2\epsilon$, $d(b_i, b_j) = \epsilon$, $p = 4$. An optimal selection of the vertices is any one of the a's and any three of the b's with $wt(\mathsf{pf}) > 1$, while GREEDY will select a_1, a_2 and two of the b's with $wt(\mathsf{pf}) = 6\epsilon$. Similar results hold for Remote-Matching.

Upper Bounds We present an algorithm for selecting p vertices for Remote-PseudoForest; the same algorithm (i.e. choosing the same set of vertices) works well for Remote-Matching also.

We take a two step approach to problem. In the first step we select some number ($\leq p$) of vertices so as to have a large pf on these vertices. This is done by considering the sequence of vertices selected by GREEDY, and chosing some prefix of this sequence according to a simple optimality criteria. In the second step, we choose the remaining vertices in such a way that $wt(\mathsf{pf})$ does not decrease too much. This is done by ensuring that the additional vertices which are selected don't "kill" too many of the vertices chosen in the first step. For simplicity, we assume that $p \leq n/4$, where n is the total number of vertices. It is easy to see that the algorithm can be easily modified when this is not the case, as long as p is less than some constant fraction of n.

The PREFIX Algorithm :

Step 1 : Let $A_p = (x_1, x_2, \ldots, x_p)$ be the sequence of points selected by GREEDY, and let the minimum distances be $r_1, r_2, \ldots, r_{p-1}$, i.e. $r_i = \min_{j=1}^{i}\{d(x_{i+1}, x_j)\}$. Let $q \in \{1, 2, \ldots, p-1\}$ be the value which maximizes $q \cdot r_q$. Let $A_q = \{x_1, \ldots, x_{q+1}\}$ be a prefix subsequence of A_p.

Step 2 : Let S_i be a sphere of radius $r_q/2$ centered at x_i, $i = 1, \ldots, q+1$. Note that these spheres are disjoint. Let $B = V - \cup_{i=1,\ldots,q+1} S_i$.

Case 1 : $|B| \geq p - (q+1)$. Let B' be a set of any $p - (q+1)$ vertices from B. $HEU = A_q \cup B'$.

Case 2 : $|B| < p - (q+1)$. Pick a set $A' \subset A_q$ such that $|A'| \leq 2(q+1)/3$ and $|\cup_{x \in A'} S_x - A'| \geq p - (q+1)$. Let B' be a set of any $p - (q+1)$ vertices from $\cup_{x \in A'} S_x - A'$. $HEU = A_q \cup B'$.

Theorem 2.1 *Let HEU be the p vertices selected by PREFIX, and let OPT be an optimum set of p vertices.*

$$\frac{wt(\mathsf{pf}(OPT))}{wt(\mathsf{pf}(HEU))} = O(\log p).$$

Proof. We first argue as to the correctness of the algorithm, i.e. we can find the $p - (q+1)$ points. This is clearly true in Case 1, so assume we are in Case 2. Since B is the set of vertices outside the spheres and $|B| < p - (q+1)$, $|\cup_{x \in A_q} S_x| > n - p + (q+1)$. Out of these vertices, $q+1$ are in A_q, so the number of remaining vertices, $|\cup_{x \in A_q} S_x - A_q|$, is at least $n - p \geq 3n/4$. Since these vertices are distributed in $q+1$ spheres, the average density is at least $3n/4(q+1)$, and hence we can find $A' \subset A_q$, $|A'| \leq \lceil (q+1)/3 \rceil \leq 2(q+1)/3$ such that $|\cup_{x \in A'} S_x - A'| \geq n/4 \geq p > p - (q+1)$.

We now show that

$$wt(\mathsf{pf}(HEU)) \geq \frac{wt(\mathsf{pf}(OPT))}{48 \log p}.$$

We first argue that

$$wt(\mathsf{pf}(HEU)) \geq \frac{wt(\mathsf{tree}(A_p))}{6 \log p}. \tag{1}$$

We will show this for Case 2; the argument for Case 1 is similar. Let $A'' = A_q - A'$, so $|A''| \geq (q+1)/3$. By the disjointedness of the spheres, for any $a \in A''$, $b \in B' \cup A'$, $d(a, b) \geq r_q/2$. Also, for any $a, a' \in A''$, $a \neq a'$, $d(a, a') \geq r_q$. Hence,

$$wt(\mathsf{pf}(HEU)) = \sum_{x \in HEU} d(x, NN_{HEU}(x)) \geq \sum_{x \in A''} d(x, NN_{HEU}(x)) \geq \sum_{x \in A''} r_q/2 \geq \frac{q+1}{3}\frac{r_q}{2} > \frac{q r_q}{6} \tag{2}$$

Consider the spanning tree T' on A_p which is constructed as follows. Put the edge x_1, x_2 in T'. Consider the remaining vertices $x_3, \ldots, x_p$ in order. When a vertex x_i is considered, the minimum distance from x_i

to $\{x_1, \ldots, x_{i-1}\}$ is r_{i-1}; connect x_i to a vertex which achieves this distance. By the choice of q, $r_i \leq \frac{qr_q}{i}$, for $i = 1, \ldots, p-1$. Hence,

$$wt(\mathsf{tree}(A_p)) \leq wt(T') = \sum_{i=1}^{p-1} r_i \leq \sum_{i=1}^{p-1} \frac{qr_q}{i} = qr_q H_{p-1} \leq qr_q \log p. \tag{3}$$

Equation 1 follows from equations 2 and 3.

The following problem was considered in [5]. Find a set of p points F_p such that $wt(\mathsf{tree}(F_p))$ is maximized. In [5] (Theorem 3.1) it was shown that $wt(\mathsf{tree}(A_p)) \geq wt(\mathsf{tree}(F_p))/4$. Clearly, $wt(\mathsf{tree}(F_p)) \geq wt(\mathsf{tree}(OPT))$ and it is easy to see that $wt(\mathsf{tree}(OPT)) \geq wt(\mathsf{pf}(OPT))/2$. From these previous equations we get

$$wt(\mathsf{tree}(A_p)) \geq wt(\mathsf{pf}(OPT))/8. \tag{4}$$

The theorem now follows from equations 4 and 1.

∎

For Remote-Matching, let HEU be the same set of vertices as selected above by PREFIX (with the assumption that p is even). It is well known that $wt(\mathsf{tree}(Y)) \geq wt(\mathsf{cycle}(Y))/2$ and it is easy to see that (as in the analysis of Christofide's algorithm) $wt(\mathsf{cycle}(Y))/2 \geq wt(\mathsf{mat}(Y))$. From this we get $wt(\mathsf{tree}(OPT)) \geq wt(\mathsf{mat}(OPT))$. The rest of the analysis is similar to the analysis for Remote-PseudoForest, and we get a similar result:

Theorem 2.2 *Let OPT be an optimum set of p vertices selected to maximize the minimum weight matching.*

$$\frac{wt(\mathsf{mat}(OPT))}{wt(\mathsf{mat}(HEU))} = O(\log p).$$

Lower bound We omit the proof of the following results.

Theorem 2.3 *The performance ratio of PREFIX for both* Remote-PseudoForest *and* Remote-Matching *is* $\Omega(\log p)$.

3 Star

We assume for simplicity that p is even.

A *maximum-weight p-matching* is a set of p vertices, for which the maximum-weight matching is maximized. It can be found efficiently via ordinary matching computation by appropriately modifying the graph [6].

Our algorithm is the same as Hassin, Rubinstein and Tamir [6] used for the Remote-Clique problem:

Select the points of a maximum weight p-matching.

They proved the following lemma (Lemma 2.1 in [6]):

Lemma 3.1 (HRT) *For any vertex set X,*

$$AvgEDGE(X) \;\leq\; \lfloor p/2 \rfloor wt(\mathsf{MAT}(X)).$$

Lemma 3.2 *For any vertex set X,*

$$wt(\mathsf{star}(X)) \;\leq\; (p-1) AvgEDGE(X).$$

Proof. We claim that equality holds for the average weight of a (spanning) star in $G[X]$, and hence the inequality. To see this, note that each edge is included in exactly two of the p stars, and the sum of the stars is therefore twice the sum of weights of all the edges. ∎

Lemmas 3.2 and 3.1 yield:

Lemma 3.3 *For any vertex set X,*

$$wt(\mathsf{star}(X)) \leq (2-2/p)wt(\mathsf{MAT}(X)).$$

Theorem 3.4 *The performance ratio of the algorithm is at most $2-2/p$.*

Proof. From the triangular inequality, the optimality of the p-matching, and the preceding lemma:

$$wt(\mathsf{star}(HEU)) \geq wt(\mathsf{MAT}(HEU)) \geq wt(\mathsf{MAT}(OPT)) \geq (2-2/p)wt(\mathsf{star}(OPT))$$

∎

4 Min-Max problems

We now consider the complementary problems of finding vertex subsets that *minimize* the corresponding *maximum* weight structures. Hassin et al [6] gave such algorithms for the Clique and Edge problems; we present here, using similar techniques, constant-factor approximations for the Tree, Cycle, Pseudoforest, and Matching problems.

4.1 Matching

A *p-star* is a set of p edges with a common end point; hence, they induce the star graph $K_{1,p-1}$. A minimum weight p-star in a graph can be found efficiently by adding the p largest weights incident on each given vertex and selecting the largest of these.

Our algorithm for this min-max p-matching problem is as in Section 3.1 of [6]: Find a p-star of minimum weight.

Theorem 4.1 $wt(\mathsf{MAT}(HEU)) \leq (2-2/p)wt(\mathsf{MAT}(OPT))$

Proof. Using the triangular inequality, the optimality of the p-star, and Lemma 3.3:

$$wt(\mathsf{MAT}(HEU)) \leq wt(\mathsf{star}(HEU)) \leq wt(\mathsf{star}(OPT)) \leq (2-2/p)wt(\mathsf{MAT}(OPT)).$$

∎

4.2 Cycle

Lemma 4.2 *For any vertex set X,*

$$wt(\mathsf{star}(X)) \leq (p-1)/p \cdot wt(\mathsf{CYCLE}(X)).$$

Proof. Some cycle must be of weight at least p times the average weight of an edge, and the rest follows from Lemma 3.2. ∎

Theorem 4.3 $wt(\mathsf{CYCLE}(HEU)) \leq (2-2/p)wt(\mathsf{CYCLE}(OPT))$

Proof. Using the triangular inequality, the optimality of the p-star, and the preceding lemma:

$$\begin{aligned} wt(\mathsf{CYCLE}(HEU)) &\leq 2wt(\mathsf{star}(HEU)) \\ &\leq 2wt(\mathsf{star}(OPT)) \\ &\leq (2-2/p)wt(\mathsf{CYCLE}(OPT)). \end{aligned}$$

∎

4.3 Tree, Pseudo-Forest

We now turn to the algorithm of Section 3.2 of [6]: Find the set of points of a p-star, whose maximum weight edge is minimized.

Let w denote the weight of the maximum weight edge on the selected star.

Theorem 4.4 $wt(\mathsf{TREE}(HEU)) \leq (4 - 4/p)wt(\mathsf{TREE}(OPT))$

Proof. The weight of the heaviest edge in HEU is at most $2w$, by the triangular inquality. Hence:

$$wt(\mathsf{TREE}(HEU)) \leq 2(p-1)w.$$

On the other hand, some pair of vertices v, w in OPT are of distance at least w, and the other $k-2$ points are of distance at least $w/2$ from either v or w. Hence:

$$wt(\mathsf{TREE}(OPT)) \geq (1/2)pw.$$

∎

The same argument holds for Min Max Pseudo-Forest (ratio $4-4/(k+1)$), or any Min Max Pi-subgraph, where Π is satisfied by a diameter-3 tree.

References

[1] M. Bellare and M. Sudan. Improved non-approximability results. In *Proc. 26th Ann. ACM Symp. on Theory of Computing*, pages 184–193, May 1994.

[2] E. Erkut. The discrete p-dispersion problem. *Europ. J. Oper. Res*, 46:48–60, 1990.

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

[4] E. Erkut and S. Neuman. Comparison of four models for dispersing facilities. *INFOR*, 29:68–85, 1990.

[5] M. M. Halldórsson, K. Iwano, N. Katoh, and T. Tokuyama. Finding subsets maximizing minimum structures. In *Proc. Sixth ACM-SIAM Symp. on Discrete Algorithms*, Jan. 1995.

[6] R. Hassin, S. Rubinstein, and A. Tamir. Notes on dispersion problems. Dec. 1994.

[7] G. Kortsarz and D. Peleg. On choosing a dense subgraph. In *Proc. 34th IEEE Symp. on Found. of Comp. Sci.*, pages 692–701, Nov. 1993.

[8] M. J. Kuby. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Geog. Anal.*, 19(4):315–329, Oct. 1987.

[9] I. D. Moon and S. S. Chaudhry. An analysis of network location problems with distance constraints. *Management Science*, 30(3):290–307, Mar. 1984.

[10] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.

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

[12] D. J. White. The maximal dispersion problem and the "first point outside the neighborhood" heuristic. *Computers Ops. Res.*, 18(1):43–50, 1991.

[13] D. J. White. The maximal dispersion problem. *J. Applic. Math. in Bus. and Ind.*, 1992.