Technische Universität Berlin

FACHBEREICH 3
MATHEMATIK

Maximum Dispersion and Geometric Maximum Weight Cliques

by

Sándor P. Fekete   Henk Meijer

No. 667/2000

Maximum Dispersion and Geometric Maximum Weight Cliques

Sándor P. Fekete* Henk Meijer†

Abstract

We consider geometric instances of the problem of finding a set of k vertices in a complete graph with nonnegative edge weights. In particular, we present algorithmic results for the case where vertices are represented by points in d-dimensional space, and edge weights correspond to rectilinear distances. This problem can be considered as a facility location problem, where the objective is to "disperse" a number of facilities, i.e., select a given number of locations from a discrete set of candidates, such that the average distance between selected locations is maximized. Problems of this type have been considered before, with the best result being an approximation algorithm with performance ratio 2. For the case where k is fixed, we establish a linear-time algorithm that finds an optimal solution. For the case where k is part of the input, we present a polynomial-time approximation scheme.

1 Introduction

A common problem in the area of facility location is the selection of a given number of k locations from a set P of n feasible positions, such that the selected set has optimal distance properties. A natural objective function is the maximization of the average distance between selected points; *dispersion* problems of this type come into play whenever we want to minimize interference between the corresponding facilities. Examples include oil storage tanks, ammunition dumps, nuclear power plants, hazardous waste sites, and fast-food outlets (see [10, 3]). In the latter paper the problem is called the *Remote Clique* problem.

Formally, problems of this type can be described as follows: given a graph $G = (V, E)$ with n vertices, and non-negative edge weights $w_{v_1,v_2} = d(v_1, v_2)$. Given $k \in \{2, \ldots, n\}$, find a subset $S \subset V$ with $|S| = k$, such that $w(S) := \sum_{(v_i,v_j)\in E(S)} d(v_i, v_j)$ is maximized. (Here, $E(S)$ denotes the edge set of the subgraph of G induced by the vertex set S.)

*Department of Mathematics, TU Berlin, 10623 Berlin, GERMANY, fekete@math.tu-berlin.de

†Department of Computing and Information Science, Queen's University, Kingston, Ontario K7L 3N6, CANADA, henk@cs.queensu.ca. Research partially supported by NSERC.

From a graph theoretic point of view, this problem has been called a *heaviest subgraph problem.* Being a weighted version of a generalization of the problem of deciding the existence of a k-*clique*, i.e., a complete subgraph with k vertices, the problem is strongly NP-hard [12]. (It should be noted that Håstad [7] showed that the vertex-oriented problem CLIQUE is in general hard to approximate within $n^{1-\varepsilon}$. The heaviest subgraph problem is edge-oriented, so Håstad's result does not imply an immediate performance bound.)

Related Work

Over recent years, there have been a number of approximation algorithms for various subproblems of this type. Feige and Seltser [5] have studied the graph problem (i.e., edge weights are 0 or 1) and showed how to find in time $n^{O((1+\log \frac{n}{k})/\varepsilon)}$ a k-set $S \subset V$ with $w(S) \geq (1-\varepsilon)\begin{pmatrix} k \\ 2 \end{pmatrix}$, provided that a k-clique exists. They also gave evidence that for $k \simeq n^{1/3}$, semidefinite programming fails to distinguish between graphs that have a k-clique, and graphs with densest k-subgraphs having average degree less than $\log n$.

Kortsarz and Peleg [8] describe a polynomial algorithm with performance guarantee $O(n^{0.3885})$ for the general case where edge weights do not have to obey the triangle inequality. For the case where $k = \Omega(n)$, Asahiro, Iwama, Tamaki, and Tokuyama [2] give a greedy constant factor approximation, while Srivastav and Wolf [11] use semidefinite programming for improved performance bounds. For the case of dense graphs (i.e., $|E| = \Omega(n^2)$) and $k = \Omega(n)$, Arora, Karger, and Karpinski [1] give a polynomial time approximation scheme.

For the case where edge weights fulfill the triangle inequality, Ravi, Rosenkrantz, and Tayi [10] give a heuristic with time complexity $O(n^2)$ and prove that it guarantees a performance bound of 4. (See Tamir [13] with reference to this paper.) Hassin, Rubinstein, and Tamir [6] give a different heuristic with time complexity $O(n^2 + k^2 \log k)$ with performance bound 2. On a related note, see Chandra and Halldórsson [3], who study a number of different remoteness measures for the subset k, including total edge weight $w(S)$. If the graph from which a subset of size k is to be selected is a tree, Tamir [12] shows that an optimal weight subset can be determined in $O(nk)$ time.

In many important cases there is even more known about the set of edge weights than just the validity of triangle inequality. This is the case when the vertex set V corresponds to a point set P in geometric space, and distances between vertices are induced by geometric distances between points. Given the practical motivation for considering the problem, it is quite natural to consider geometric instances of this type. In fact, it was shown by Ravi, Rosenkrantz, and Tayi in [10] that for the case of Euclidean distances in two-dimensional space, it is possible to achieve performance bounds that are arbitrarily close to $\pi/2 = 1.57...$. For other metrics, however, the best performance guarantee remains at 2.

An important application of our problem is data sampling and clustering, where points are to be selected from a large more-dimensional set. Different metric dimensions of a data point describe different metric properties of a corresponding item. Since these properties are not geometrically related, distances are typically not evaluated by Euclidean distances. Instead, some weighted L_1 metric is used. (See Erkut [4].) For data sampling, a set of points is to be selected that has high average distance. For clustering, a given set of points is to be subdivided into k clusters, such that points from the same cluster are close together, while points from different clusters are far apart. If we do the clustering by choosing k center points, and assigning any point to its nearest cluster center, we have to consider the same problem of finding a set of center points with large average distance, which is equivalent to finding a k-clique with maximum total edge weight.

Main Results

In this paper, we consider point sets P in d-dimensional space, where d is some constant. For the most part, distances are measured according to the rectilinear "Manhattan" norm L_1.

Our results include the following:

- A linear time ($O(n)$) algorithm to solve the problem to optimality in case where k is some fixed constant. This is in contrast to the case of Euclidean distances, where there is a well-known lower bound of $\Omega(n \log n)$ in the computation tree model for determining the diameter of a planar point set, i.e., the special case $d = 2$ and $k = 2$ (see [9]).
- A polynomial time approximation scheme for the case where k is not fixed. This method can be applied for arbitrary fixed dimension d. For the case of Euclidean distances in two-dimensional space, it implies a performance bound of $\sqrt{2} + \varepsilon$, for any given $\varepsilon > 0$.

2 Preliminaries

For the most part of this paper, all points are assumed to be points in the plane. Higher-dimensional spaces will be discussed in the end. Distances are measured using the L_1 norm, unless noted otherwise. The x- and y-coordinates of a point p are denoted by x_p and y_p. If p and q are two points in the plane, then the distance between p and q is $d(p, q) = |x_p - x_q| + |y_p - y_q|$. We say that q is above p in direction (a, b) if the angle between the vector in direction (a, b) and the vector $q - p$ is less than $\pi/2$, i.e., if the inner product $\langle q - p, (a, b)\rangle$ of the vector $q - p$ and the vector (a, b) is positive. We say that a point p is maximal in direction (a, b) with respect to a set of points P if no point in P is above p in direction (a, b). For example, if p is an element of a set of points P and p has a maximal y-coordinate, then p is maximal in direction (0,1) with respect to P, and a point p with minimal x-coordinate is maximal in direction (-1,0) with respect to P. If the set P is clear from the context, we simply state that p is maximal in direction (a, b).

The weight of a set of points P is the sum of the distances between all pairs of points in this set, and is denoted by $w(P)$. Similarly, $w(P, Q)$ denotes the total sum of distances between two sets P and Q. For L_1 distances, $w_x(P)$ and $w_x(P, Q)$ denote the sum of x-distances within P, or between P and Q.

3 Cliques of Fixed Size

Let $S = \{s_0, s_1, \ldots, s_{k-1}\}$ be a maximal weight subset of P, where k is a fixed integer greater than 1. We will label the x- and y-coordinates of a point $s \in S$ by some (x_a, y_b) with $0 \le a < k$ and $0 \le b < k$ such that $x_0 \le x_1 \le \ldots \le x_{k-1}$ and $y_0 \le y_1 \le \ldots \le y_{k-1}$. So

$$w(S) \;=\; \sum_{0 \le i < j < k} (x_j - x_i) + \sum_{0 \le i < j < k} (y_j - y_i)$$

We denote the first part of this sum by $w_x(S)$. The second part of this sum is denoted by $w_y(S)$. We can derive the following equation:

$$w_x(S) \;=\; \sum_{i=1}^{k-1} i(k-i)(x_i - x_{i-1}).$$

From this we derive that if k is odd, we have

$$w_x(S) \;=\; (k-1)(x_{k-1} - x_0) + (k-3)(x_{k-2} - x_1) + \;\ldots\; + 2(x_{(k+1)/2} - x_{(k-3)/2})$$

and for even k we have

$$w_x(S) \;=\; (k-1)(x_{k-1} - x_0) + (k-3)(x_{k-2} - x_1) + \;\ldots\; + (x_{k/2} - x_{(k-2)/2}).$$

Similar equations for $w_y(S)$ can be found. These equations allow us to deduce properties of a point s in a maximal weight subset S. For example, suppose $s \in S$ with $s = (x_i, y_j)$. Suppose we add ϵ_x and ϵ_y to the the x- and y-coordinate of s, such that $\delta = (2i + 1 - k)\epsilon_x + (2j + 1 - k)\epsilon_y > 0$. From the above equations we can derive that this increases the value of $w(S)$ by at least δ. Since $w(S)$ is maximal, we derive that there can be no point $p = (x_i + \epsilon_x, y_j + \epsilon_y)$ in $P - S$ for which $(2i + 1 - k)\epsilon_x + (2j + 1 - k)\epsilon_y > 0$. So s is maximal in direction $(2i + 1 - k, 2j + 1 - k)$ with respect to $P - S$. This observation allows us to prove the following lemma.

Lemma 1 *If S is a maximal weight subset of P of cardinality k, then each point in S is maximal in direction $(2i + 1 - k, 2j + 1 - k)$ with respect to $P - S$ for some values of i and j with $0 \le i, j < k$.*

Proof: Suppose a point $p = (x_i, y_j)$ is in S. From the discussion above we derive that p is maximal in direction $(2i + 1 - k, 2j + 1 - k)$ with respect to $P - S$.

□

Theorem 1 *Given a constant value for k, the maximal weight subset S of a set of n points P, such that S has cardinality k, can be found in linear time.*

Proof: Consider all directions of the form $(2i+1-k, 2j+1-k)$ with $0 \leq i, j < k$. For each direction (a, b), find $S_k(a, b)$, a set of k points that are maximal in direction (a, b) with respect to $P - S_k(a, b)$. Compute the set $\cup S_k(a, b)$ and try all possible subsets of size k of this set until a subset of maximal weight is found.

Correctness follows from the fact that Lemma 1 implies that $S \subset \cup S_k(a, b)$. Since k is a constant, each set $S_k(a, b)$ can be found in linear time. Since the cardinality of $\cup S_k(a, b)$ is less than or equal to k^3, the result follows.

□

4 Cliques of Variable Size

In this section we consider the scenario where k is not fixed, i.e., part of the input. We show that there is a polynomial time approximation scheme (PTAS), i.e., for any fixed positive ε, there is a polynomial approximation algorithm that finds a solution that is within $(1 + \varepsilon)$ of the optimum.

The basic idea is to use a suitable subset of m coordinates that subdivide an optimal solution into subsets of equal cardinality. More precisely, we find (by enumeration) a subdivision of an optimal solution into $m \times m$ rectangular cells C_{ij}, each of which must contain a specific number k_{ij} of selected points. From each cell C_{ij}, the points are selected in a way that guarantees that the total distance to all other cells except for the $m - 1$ cells in the same "horizontal" strip or the $m-1$ cells in the same "vertical" strip is maximized. As it turns out, this can be done in a way that the total neglected distance within the strips is bounded by a fraction of $(5m - 9)/(2(m - 1)(m - 2))$ of the weight of an optimal solution, yielding the desired approximation property. See Figure 1 for the overall picture.

For ease of presentation we assume that k is a multiple of m and $m > 2$. Approximation algorithms for other values of k can be constructed in a similar fashion and will be treated in the full paper. Consider an optimal solution of k points, denoted by OPT. Furthermore consider a division of the plane by a set of $m + 1$ x-coordinates $\xi_0 \leq \ldots \leq \xi_1 \leq \xi_m$. Let $X_i := \{p = (x, y) \in \Re^2 \mid \xi_i \leq x \leq \xi_{i+1}, 0 \leq i < m\}$ be the vertical strip between coordinates ξ_i and ξ_{i+1}. By enumeration of possible choices of $\xi_0, \ldots, \xi_m$ we may assume that the ξ_i have the property that, for an optimal solution, from each of the m strips X_i precisely k/m points of P are chosen.

In a similar manner, suppose we know $m + 1$ y-coordinates $\eta_0 \leq \eta_1 \leq \ldots \leq \eta_m$ such that from each horizontal strip $Y_i := \{p = (x, y) \in \Re^2 \mid \eta_i \leq y \leq \eta_{i+1}, 0 \leq i < m\}$ a subset of k/m points are chosen for an optimal solution.

Let $C_{ij} := X_i \cap Y_j$, and let k_{ij} be the number of points in OPT that are chosen from C_{ij}. Since $\sum_{0 \leq i < m} k_{ij} = \sum_{0 \leq j < m} k_{ij} = k/m$, we may assume by enumeration over the possible partitions of k/m into m pieces that we know all the numbers k_{ij}.

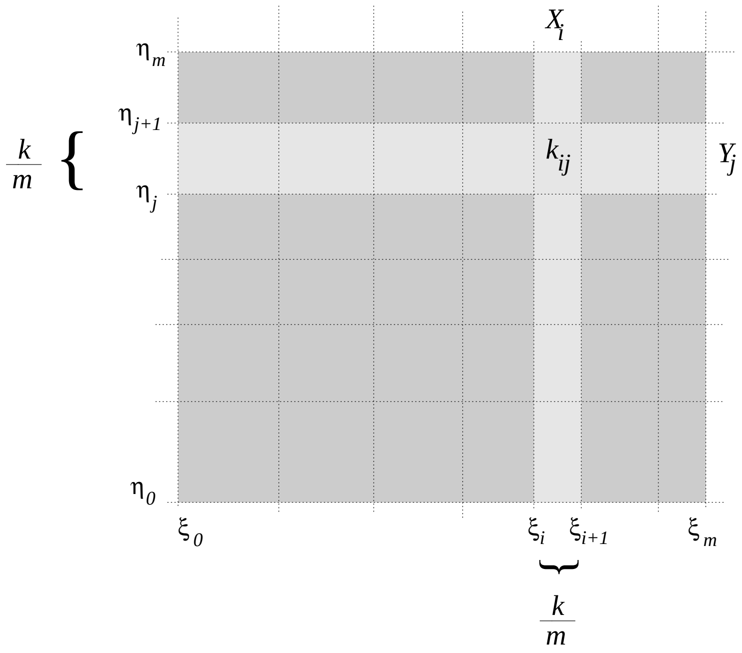

Figure 1: Subdividing the plane into cells

Finally, define the vector $\nabla_{ij} := ((2i + 1 - m)k/m, (2j + 1 - m)k/m)$. (As we will see, ∇_{ij} is the gradient of the total sum of distances from a point in C_{ij} to points not in X_i or Y_j.) Now our approximation algorithm is as follows: from each cell C_{ij}, choose the k_{ij} points that are maximal in direction ∇_{ij}. (Overlap between the selections from different cells is avoided by proceeding in lexicographic order of cells, and choosing the k_{ij} points among the candidates that are still unselected.) Let HEU be the point set selected in this way.

It is clear that HEU can be computed in polynomial time. We will proceed by a series of lemmas to determine how well $w(HEU)$ approximates $w(OPT)$. In the following, we consider the distances involving points from a particular cell C_{ij}. Let HEU_{ij} be the set of k_{ij} points that are selected from C_{ij} by the heuristic, and let OPT_{ij} be a set of k_{ij} points of an optimal solution that are attributed to C_{ij}. Let $S_{ij} = OPT_{ij} \cap HEU_{ij}$. Furthermore we define $\overline{S}_{ij} = HEU_{ij} \setminus OPT_{ij}$, and $\tilde{S}_{ij} = OPT_{ij} \setminus HEU_{ij}$. Let $HEU_{i\bullet}$, $OPT_{i\bullet}$, $HEU_{\bullet j}$ and $OPT_{\bullet j}$ be the set of k/m points selected from X_i and Y_j by the heuristic and an optimal algorithm respectively. Finally $\overline{HEU}_{i\bullet} := HEU \setminus HEU_{i\bullet}$, $\overline{HEU}_{\bullet j} := HEU \setminus HEU_{\bullet j}$, $\overline{OPT}_{i\bullet} := OPT \setminus OPT_{i\bullet}$ and $\overline{OPT}_{\bullet j} := OPT \setminus OPT_{\bullet j}$.

Lemma 2

$$\begin{aligned} & w_x(HEU_{ij}, \overline{HEU}_{i\bullet}) + w_y(HEU_{ij}, \overline{HEU}_{\bullet j}) \\ \geq\ & w_x(OPT_{ij}, \overline{OPT}_{i\bullet}) + w_y(OPT_{ij}, \overline{OPT}_{\bullet j}). \end{aligned}$$

Proof: Consider a point $p \in \overline{S}_{ij}$. Thus, there is a point $p' \in \tilde{S}_{ij}$ that was chosen by the heuristic instead of p. Let $h = (h_x, h_y) = p' - p$. When replacing p in OPT by p', we increase the x-distance to the ik/m points "left" of C_{ij} by h_x, while decreasing the x-distance to $(m-i-1)k/m$ points "right" of C_{ij} by h_x. In the balance, this yields a change of $((2i+1-m)k/m)h_x$. Similarly, we get a change of $((2j+1-m)k/m)h_y$ for the y-coordinates. By definition, we have assumed that the inner product $\langle h, \nabla_{ij}\rangle \geq 0$, so the overall change of distances is nonnegative.

Performing these replacements for all points in $OPT \setminus HEU$, we can transform OPT to HEU, while increasing the sum of distances $w_x(OPT_{ij}, \overline{OPT}_{i\bullet}) + w_y(OPT_{ij}, \overline{OPT}_{\bullet j})$ to the sum
$w_x(HEU_{ij}, \overline{HEU}_{i\bullet}) + w_y(HEU_{ij}, \overline{HEU}_{\bullet j})$.
□

In the following three lemmas we show that the difference between the weight of an optimal solution $w(OPT)$ and the right hand side of the inequality in Lemma 2 is a small fraction of $w(OPT)$.

Lemma 3

$$\sum_{0<i<m-1} w_x(OPT_{i\bullet}) \;\leq\; \frac{w_x(OPT)}{2(m-2)}.$$

Proof: Let $\delta_i = \xi_{i+1} - \xi_i$. For $i = 1$ and $i = m-2$ we have

$$w_x(OPT_{i\bullet}) \leq \frac{k^2}{2m^2}\delta_i = \frac{k}{m}\frac{(m-2)k}{m}\delta_i\ \frac{1}{2(m-2)}.$$

For $1 < i < m-2$ we have

$$w_x(OPT_{i\bullet}) \leq \frac{k^2}{2m^2}\delta_i < \frac{ik}{m}\frac{(m-i-1)k}{m}\delta_i\ \frac{1}{2(m-2)}.$$

Since OPT has ik/m and $(m-i-1)k/m$ points to the left of ξ_i and right of ξ_{i+1} respectively, we have

$$w_x(OPT) \geq \sum_{0<i<m-1} \frac{ik}{m}\frac{(m-i-1)k}{m}\delta_i$$

so

$$\sum_{0<i<m-1} w_x(OPT_{i\bullet}) \leq \frac{1}{2(m-2)} w_x(OPT).$$

□

Lemma 4 *For $i = 0$ and $i = m-1$ we have*

$$w_x(OPT_{i\bullet}) \;\leq\; \frac{w_x(OPT)}{m-1}$$

Proof: Without loss of generality assume $i = 0$. Let $x_0, x_1, \cdots, x_{(k/m)-1}$ be the x-coordinates of the points $p_0, p_1, \ldots, p_{(k/m)-1}$ in $OPT_{0\bullet}$. So

$$\begin{aligned} w_x(OPT_{0\bullet}) &= (\frac{k}{m}-1)(x_{\frac{k}{m}-1}-x_0)+(\frac{k}{m}-3)(x_{\frac{k}{m}-2}-x_1)+\ \ldots \\ &\leq (\frac{k}{m}-1)(\xi_1-x_0)+(\frac{k}{m}-3)(\xi_1-x_1)+\ \ldots \\ &\leq \frac{k}{m}(\xi_1-x_0)+\frac{k}{m}(\xi_1-x_1)+\ \ldots \end{aligned}$$

Since $\xi_1 - x_j \leq x - x_j$ where $0 \leq j < k/m$ and x is the x-coordinate of any point in $\overline{OPT}_{0\bullet}$ and since there are $(m-1)k/m$ points in $\overline{OPT}_{0\bullet}$, we have

$$\xi_1 - x_j < \frac{m}{(m-1)k} w_x(p_j, \overline{OPT}_{0\bullet})$$

so

$$w_x(OPT_{0\bullet}) \leq \frac{k}{m}\frac{m}{(m-1)k}\sum_{0\leq i\leq \frac{k}{2m}-1} w_x(p_i, \overline{OPT}_{0\bullet})$$

$$\leq \frac{1}{m-1}\sum_{0\leq i<\frac{k}{m}} w_x(p_i, \overline{OPT}_{0\bullet}) = \frac{1}{m-1}w_x(OPT_{0\bullet}, \overline{OPT}_{0\bullet}) \leq \frac{1}{m-1}w_x(OPT).$$

□

Lemma 5

$$\begin{aligned} &\sum_{0\leq i<m} w_x(OPT_{i\bullet}, \overline{OPT}_{i\bullet}) + \sum_{0\leq j<m} w_y(OPT_{\bullet j}, \overline{OPT}_{\bullet j}) \\ &\qquad \geq (1-\frac{5m-9}{2(m-1)(m-2)})w(OPT)). \end{aligned}$$

Proof: From Lemmas 3 and 4 we derive that

$$\sum_{0\leq i<m} w_x(OPT_{i\bullet}) \ \leq\ \frac{5m-9}{2(m-1)(m-2)}w_x(OPT)$$

and similarly

$$\sum_{0\leq i<m} w_y(OPT_{\bullet j}) \ \leq\ \frac{5m-9}{2(m-1)(m-2)}w_y(OPT).$$

Since

$$\begin{aligned} &w(OPT) = w_x(OPT) + w_y(OPT) \\ &= \sum_{0\leq i<m} w_x(OPT_{i\bullet}, \overline{OPT}_{i\bullet}) + \sum_{0\leq i<m} w_x(OPT_{i\bullet}) \\ &\sum_{0\leq j<m} w_y(OPT_{\bullet j}, \overline{OPT}_{\bullet j}) + \sum_{0\leq j<m} w_y(OPT_{\bullet j}) \end{aligned}$$

the result follows. $\square$

Putting together Lemma 2 and the error estimate from Lemma 5, the approximation theorem can now be proven.

Theorem 2 *For any fixed m, HEU can be computed in polynomial time, and*

$$w(HEU) \geq \left(1 - \frac{5m-9}{2(m-1)(m-2)}\right) w(OPT).$$

Proof: The claim about the running time is clear. Using Lemmas 2 and 5 we derive

$$w(HEU) \geq \sum_{0 \leq i < m} w_x(HEU_{i\bullet}, \overline{HEU}_{i\bullet}) + \sum_{0 \leq j < m} w_y(HEU_{\bullet j}, \overline{HEU}_{\bullet j})$$

$$\geq \sum_{0 \leq i < m} w_x(OPT_{i\bullet}, \overline{OPT}_{i\bullet}) + \sum_{0 \leq j < m} w_y(OPT_{\bullet j}, \overline{OPT}_{\bullet j})$$

$$\geq \left(1 - \frac{5m-9}{2(m-1)(m-2)}\right) w(OPT).$$

$\square$

5 Implications

It is straightforward to modify our above arguments to point sets under L_1 distances in an arbitrary d-dimensional space, with fixed d.

Theorem 3 *Given a constant value for k and d, the maximal weight subset S of a set of n points in d-dimensional space, such that S has cardinality k, can be found in linear time. If d and ε are constants, but k is not fixed, then there is a polynomial time algorithm that finds a subset whose weight is within $(1+\varepsilon)$ of the optimum.*

Furthermore, we can use the approximation scheme from the previous section to get a $\sqrt{2}(1+\varepsilon)$ approximation factor for the case of Euclidean distances in two-dimensional space, for any $\varepsilon > 0$: In polynomial time, find a k-set S_1 such that $L_1(S)$ is within $(1+\varepsilon)$ of an optimal solution OPT_1 with respect to L_1 distances. Let OPT_2 be an optimal solution with respect to L_2 distances. Then

$$\begin{aligned} L_2(S) &\geq \frac{1}{\sqrt{2}} L_1(S) \geq \frac{1}{\sqrt{2}(1+\varepsilon)} L_1(OPT_1) \geq \frac{1}{\sqrt{2}(1+\varepsilon)} L_1(OPT_2) \\ &\geq \frac{1}{\sqrt{2}(1+\varepsilon)} L_2(OPT_2), \end{aligned}$$

and the claim follows.

6 Conclusions

We have presented algorithms for geometric instances of the maximum weighted k-clique problem. Our results give a dramatic improvement over the previous best approximation factor of 2 that was presented in [10] for the case of general metric spaces. This underlines the observation that geometry can help to get better algorithms for problems from combinatorial optimization.

Furthermore, the algorithms in [10] give better performance for Euclidean metric than for Manhattan distances. We correct this anomaly by showing that among problems involving geometric distances, the rectilinear metric may allow better algorithms than the Euclidean metric.

Acknowledgments

We would like to thank Katja Wolf and Magnús Halldórsson for helpful discussions.

References

[1] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for NP-hard problems. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 284–293, 1995.

[2] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense graph. In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 1097 of *Lecture Notes in Computer Science*, pages 136–148. Springer–Verlag, 1996.

[3] B. Chandra and M. M. Halldórsson. Facility dispersion and remote subgraphs. In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 1097 of *Lecture Notes in Computer Science*, pages 53–65. Springer–Verlag, 1996.

[4] E. Erkut. The discrete p-dispersion problem. *European Journal of Operations Research*, 46:48–60, 1990.

[5] U. Feige and M. Seltser. On the densest k-subgraph problems. Technical Report CS97-16, `http://www.wisdom.weizmann.ac.il`, 1997.

[6] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Operations Research Letters*, 21:133–137, 1997.

[7] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proceedings of the 37th IEEE Annual Symposium on Foundations of Computer Science*, pages 627–636, 1996.

[8] G. Kortsarz and D. Peleg. On choosing a dense subgraph. In *Proceedings of the 34th IEEE Annual Symposium on Foundations of Computer Science*, pages 692–701, Palo Alto, CA, 1993.

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

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

[11] A. Srivastav and K. Wolf. Finding dense subgraphs with semidefinite programming. In K. Jansen and J. Rolim, editors, *Approximation Algorithms for Combinatorial Optimization (APPROX '98)*, volume 1444 of *Lecture Notes in Computer Science*, pages 181–191, Aalborg, Denmark, 1998. Springer–Verlag.

[12] A. Tamir. Obnoxious facility location in graphs. *SIAM Journal on Discrete Mathematics*, 4:550–567, 1991.

[13] A. Tamir. Comments on the paper: 'Heuristic and special case algorithms for dispersion problems' by , S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. *Operations Research*, 46:157–158, 1998.

Reports from the group

"Combinatorial Optimization and Graph Algorithms"

of the Department of Mathematics, TU Berlin

674/2000 *Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Sándor P. Fekete, Joseph S. B. Mitchell, and Saurabh Sethia:* Optimal Covering Tours with Turn Costs

669/2000 *Michael Naatz:* A Note On a Question of C. D. Savage

667/2000 *Sándor P. Fekete and Henk Meijer:* On Geometric Maximum Weight Cliques

666/2000 *Sándor P. Fekete, Joseph S. B. Mitchell, and Karin Weinbrecht:* On the Continuous Weber and k-Median Problems

664/2000 *Rolf H. Möhring, Andreas S. Schulz, Frederik Stork, and Marc Uetz:* A Note on Scheduling Problems with Irregular Starting Time Costs

661/2000 *Frederik Stork and Marc Uetz:* Resource-Constrained Project Scheduling: From a Lagrangian Relaxation to Competitive Solutions

658/1999 *Olaf Jahn, Rolf H. Möhring, and Andreas S. Schulz:* Optimal Routing of Traffic Flows with Length Restrictions in Networks with Congestion

655/1999 *Michel X. Goemans and Martin Skutella:* Cooperative facility location games

654/1999 *Michel X. Goemans, Maurice Queyranne, Andreas S. Schulz, Martin Skutella, and Yaoguang Wang:* Single Machine Scheduling with Release Dates

653/1999 *Andreas S. Schulz and Martin Skutella:* Scheduling unrelated machines by randomized rounding

646/1999 *Rolf H. Möhring, Martin Skutella, and Frederik Stork:* Forcing Relations for AND/OR Precedence Constraints

640/1999 *Foto Afrati, Evripidis Bampis, Chandra Chekuri, David Karger, Claire Kenyon, Sanjeev Khanna, Ioannis Milis, Maurice Queyranne, Martin Skutella, Cliff Stein, and Maxim Sviridenko:* Approximation Schemes for Minimizing Average Weighted Completion Time with Release Dates

639/1999 *Andreas S. Schulz and Martin Skutella:* The Power of α-Points in Preemptive Single Machine Scheduling

634/1999 *Karsten Weihe, Ulrik Brandes, Annegret Liebers, Matthias Müller–Hannemann, Dorothea Wagner and Thomas Willhalm:* Empirical Design of Geometric Algorithms

633/1999 *Matthias Müller–Hannemann and Karsten Weihe:* On the Discrete Core of Quadrilateral Mesh Refinement

632/1999 *Matthias Müller–Hannemann:* Shelling Hexahedral Complexes for Mesh Generation in CAD

631/1999 *Matthias Müller–Hannemann and Alexander Schwartz:* Implementing Weighted b-Matching Algorithms: Insights from a Computational Study

629/1999 *Martin Skutella:* Convex Quadratic Programming Relaxations for Network Scheduling Problems

628/1999 *Martin Skutella and Gerhard J. Woeginger:* A PTAS for minimizing the total weighted completion time on identical parallel machines

627/1998 *Jens Gustedt:* Specifying Characteristics of Digital Filters with FilterPro

620/1998 *Rolf H. Möhring, Andreas S. Schulz, Frederik Stork, and Marc Uetz:* Resource Constrained Project Scheduling: Computing Lower Bounds by Solving Minimum Cut Problems

619/1998 *Rolf H. Möhring, Martin Oellrich, and Andreas S. Schulz:* Efficient Algorithms for the Minimum-Cost Embedding of Reliable Virtual Private Networks into Telecommunication Networks

618/1998 *Friedrich Eisenbrand and Andreas S. Schulz:* Bounds on the Chvátal Rank of Polytopes in the 0/1-Cube

617/1998 *Andreas S. Schulz and Robert Weismantel:* An Oracle-Polynomial Time Augmentation Algorithm for Integer Proramming

616/1998 *Alexander Bockmayr, Friedrich Eisenbrand, Mark Hartmann, and Andreas S. Schulz:* On the Chvátal Rank of Polytopes in the 0/1 Cube

615/1998 *Ekkehard Köhler and Matthias Kriesell:* Edge-Dominating Trails in AT-free Graphs

613/1998 *Frederik Stork:* A branch and bound algorithm for minimizing expected makespan in stochastic project networks with resource constraints

612/1998 *Rolf H. Möhring and Frederik Stork:* Linear preselective policies for stochastic project scheduling

609/1998 *Arfst Ludwig, Rolf H. Möhring, and Frederik Stork:* A computational study on bounding the makespan distribution in stochastic project networks

605/1998 *Friedrich Eisenbrand:* A Note on the Membership Problem for the Elementary Closure of a Polyhedron

596/1998 *Andreas Fest, Rolf H. Möhring, Frederik Stork, and Marc Uetz:* Resource Constrained Project Scheduling with Time Windows: A Branching Scheme Based on Dynamic Release Dates

595/1998 *Rolf H. Möhring Andreas S. Schulz, and Marc Uetz:* Approximation in Stochastic Scheduling: The Power of LP-based Priority Policies

591/1998 *Matthias Müller–Hannemann and Alexander Schwartz:* Implementing Weighted b-Matching Algorithms: Towards a Flexible Software Design

590/1998 *Stefan Felsner and Jens Gustedt and Michel Morvan:* Interval Reductions and Extensions of Orders: Bijections to Chains in Lattices

584/1998 *Alix Munier, Maurice Queyranne, and Andreas S. Schulz:* Approximation Bounds for a General Class of Precedence Constrained Parallel Machine Scheduling Problems

577/1998 *Martin Skutella:* Semidefinite Relaxations for Parallel Machine Scheduling

Reports may be requested from:

Hannelore Vogt-Möller
Fachbereich Mathematik, MA 6–1
TU Berlin
Straße des 17. Juni 136
D-10623 Berlin – Germany

e-mail: moeller@math.TU-Berlin.DE

Reports are also available in various formats from

`http://www.math.tu-berlin.de/coga/publications/techreports/`

and via anonymous ftp as

`ftp://ftp.math.tu-berlin.de/pub/Preprints/combi/Report-`*number*`-`*year*`.ps`