Constrained k-center and movement to independence

Adrian Dumitrescu* Minghui Jiang†

November 23, 2010

Abstract

We obtain hardness results and approximation algorithms for two related geometric problems involving movement. The first is a constrained variant of the k-center problem, arising from a geometric client-server problem. The second is the problem of moving points towards an independent set.

Keywords: Computational geometry, client-server problem, independent set, approximation algorithm, fixed parameter tractable algorithm, linear programming.

1 Introduction

Given a set S of n points in the plane, the k-center problem is to find k congruent disks of minimum radius r that cover S [1, p. 276]. We study the following constrained variant of the k-center problem:

CONSTRAINED k-CENTER

Instance: A set $P = \{p_1, \ldots, p_n\}$ of n black points and a set $Q = \{q_1, \ldots, q_k\}$ of k red points in the plane. P and Q are not necessarily disjoint.

Problem: Find a set $\mathcal{D} = \{D_1, \ldots, D_k\}$ of k disks *constrained* to the set $Q = \{q_1, \ldots, q_k\}$ of k red points (that is, for $1 \leq j \leq k$, the disk D_j contains the corresponding red point q_j) such that all points in P are covered by the union of the disks in $\mathcal{D}$, and the maximum radius of the disks in $\mathcal{D}$ is minimized.

Observe that $\mathcal{D}$ can consist of congruent disks, which we choose for simplicity and without any loss of generality. The problem CONSTRAINED k-CENTER is the geometric version of a movement problem originally proposed by Demaine et al. [3] in the graph-theoretical setting: Given a connected graph G in which some vertices are occupied by clients and some vertices are occupied by servers, the problem FACILITY-LOCATION MOVEMENT is that of moving both the clients and the servers in the graph until each client occupies the same vertex as some server, such that the maximum movement of a client or a server is minimized; here the distance is the path length in the graph. The authors [3] observed that a 2-approximation can be achieved simply by keeping each server at its original location and moving each client to its nearest server. Friggstad and

*Department of Computer Science, University of Wisconsin–Milwaukee, WI 53201-0784, USA. Email: dumitres@uwm.edu. Supported in part by NSF CAREER grant CCF-0444188.

†Department of Computer Science, Utah State University, Logan, UT 84322-4205, USA. Email: mjiang@cc.usu.edu. Supported in part by NSF grant DBI-0743670.

Salavatipour [9] showed that this simple 2-approximation is in fact best possible: Unless P = NP, Facility-location Movement is NP-hard to approximate within $2 - \varepsilon$ for any constant $\varepsilon > 0$.

Here we focus on the geometric version, where the clients and servers are points in the Euclidean plane (or more generally, in $\mathbb{R}^d$), and the movement is measured as the Euclidean distance, rather than the number of edges of a path in the graph. The task is to determine a movement of the clients and servers, so that in the end, each client coincides with some server, and the maximum movement is minimized.

Let P be the set of clients, and Q be the set of servers, where $|P| = n$ and $|Q| = k$. Usually k is much smaller than n. Let us first observe that our Constrained k-center problem is essentially the same as the Facility-location Movement problem. Indeed, consider an optimal solution to the Facility-location Movement problem with maximum movement λ. Then the disks of radius λ centered at the server locations after the movement cover all clients and servers at their original locations. Conversely, consider a set of disks, say of radius λ, in an optimal solution to Constrained k-center. Then by moving the clients and the server contained in each disk (with ties broken arbitrarily) to its center, gives a solution to the Facility-location Movement problem with the maximum movement at most λ.

The afore-mentioned 2-approximation works in this setting as follows. Let d denote the maximum black-red (client-server) distance obtained by assigning each black point to its closest red point. Let OPT denote an optimal solution and ALG denote the solution returned by the algorithm. Then clearly

$$\text{OPT} \geq \frac{d}{2}, \text{ and ALG} = d, \tag{1}$$

and the ratio 2 immediately follows. It is worth observing that the algorithm which keeps fixed each red point achieves ratio 2 even on the line: place two red points at 0 and $2 + \varepsilon$, and two black points at $1 + \varepsilon$ and $3 + \varepsilon$. Then $\text{OPT} = (1 + \varepsilon)/2$, while $\text{ALG} = 1$ (this tight example can be easily extended for a larger number of points).

We first show that the approximation lower bound for the problem remains close to 2 already for the planar variant.

Theorem 1. Constrained k-center *in the plane is NP-hard to approximate within* 1.8279.

On the other hand, we have the following positive result showing that constant approximations for Constrained k-center can be obtained by a fixed parameter tractable algorithm [10] with k as the parameter. In particular, we have a polynomial-time approximation scheme if k is logarithmic in n, and a linear-time approximation scheme if k is a constant.

Theorem 2. *For any given $\varepsilon > 0$, there exists a $(1+\varepsilon)$-approximation algorithm for* Constrained k-center *in the plane that runs in $O(\varepsilon^{-2k} \cdot n)$ time. Moreover, there exist: a 1.87-approximation algorithm that runs in $O(3^k k \cdot n)$ time, a 1.71-approximation algorithm that runs in $O(4^k k \cdot n)$ time, and a 1.61-approximation algorithm that runs in $O(5^k k \cdot n)$ time.*

In the second part of the paper, we study another movement problem proposed by Demaine et al. [3]:

Movement to Independence

Instance: A set $P = \{p_1, \ldots, p_n\}$ of n points in $\mathbb{R}^d$, and a threshold distance Δ.

Problem: Find a set $Q = \{q_1, \ldots, q_n\}$ of n (target) points in $\mathbb{R}^d$, one point $q_i \in Q$ for each point $p_i \in P$, such that the minimum pairwise distance $\min_{i,j} |q_i q_j|$ among the points in Q is at least Δ, and that the maximum movement $\max_i |p_i q_i|$ from any point $p_i \in P$ to the corresponding target point $q_i \in Q$ is minimized.

There is a natural connection between Movement to Independence and the dispersion problem in a set of congruent disks. The problem of *dispersion* in a given set of disks is that of selecting n points, one in each disk, such that the minimum inter-point distance is maximized.

Dispersion in congruent disks

Instance: A set $\{D_1, \ldots, D_n\}$ of n congruent disks.

Problem: Find a set $Q = \{q_1, \ldots, q_n\}$ of n points, one point $q_i \in Q$ in each disk D_i, such that the minimum pairwise distance $\min_{i,j} |q_i q_j|$ among the points in Q is maximized.

The dispersion problem was introduced by Fiala et al. [8] in a more general setting as "systems of distant representatives", generalizing the classic problem "systems of distinct representatives". See also [2, 5, 6] for algorithmic results on the dispersion problem in disks. Fiala et al. [8] showed that dispersion in congruent disks is NP-hard. As a corollary we obtain

Theorem 3. Movement to Independence *in the plane (and in higher dimensions) is NP-hard.*

Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in $\mathbb{R}^2$. Denote by $\text{OPT}(x)$ the minimum maximum movement for the instance (P, x) of the problem Movement to Independence in $\mathbb{R}^2$ (that is, the value of the optimal solution to this instance). Demaine et al. [3] presented a polynomial-time algorithm for Movement to Independence on an instance $(P, 1)$ with maximum movement at most $\text{OPT}(1) + 1 + \frac{1}{\sqrt{3}}$. Their algorithm moves the points to the grid points of an equilateral triangular lattice of unit side. By a scaling argument, this algorithm can be turned into an algorithm for (P, x) for any $x > 0$, with maximum movement at most $\text{OPT}(x) + (1 + \frac{1}{\sqrt{3}})x$. We have the following complementary result:

Theorem 4. *There exists a polynomial-time approximation algorithm for* Movement to Independence *in the plane that moves any given set P of n points in $\mathbb{R}^2$ to another set Q of n points in $\mathbb{R}^2$, with a maximum movement no more than the optimal maximum movement necessary for a threshold distance of* 1*, and such that the minimum pairwise distance among the points in Q is at least $c = \frac{1}{3+2/\sqrt{3}} = 0.24\ldots$.*

2 The two problems on the line and on a closed curve

As a warm-up exercise, we first study the two problems Constrained k-center and Movement to Independence on the line and on a closed curve. The *distance* between two points on a closed curve is the length of the shorter subcurve determined by the two points. In these two settings, both problems can be solved exactly in polynomial time.

Proposition 1. *There exists an exact algorithm running in $O((n + k) \log(n + k))$ time for* Constrained k-center *on the line.*

Proof. Observe that there exists an optimal solution consisting of a set of k disjoint intervals $I_j = [u_j, v_j]$, $1 \le j \le k$, such that $u_j, v_j \in P \cup Q$ and $q_j \in I_j$ for $j = 1, \ldots, k$. We next show that such a solution can be computed in $O(n \log n)$ time by dynamic programming.

Order the $n + k$ points in $P \cup Q$ from left to right with indices $1, \ldots, n + k$; in case of ties, put the red points before black points. Let $s_1, \ldots, s_k$ be the indices of the k red points, $1 \le s_1 < \ldots < s_k \le n + k$. Partition the list $P \cup Q$ of $n + k$ points into $k + 1$ contiguous sublists $L_0, L_1, \ldots, L_k$ such that, for $1 \le j \le k$, the red point s_j is the first point in L_j (the sublist L_0 contains no red points).

For each point i in $P \cup Q$, $1 \le i \le n+k$, denote by $j[i]$ the index j, $0 \le j \le k$, such that the point i is in the sublist L_j. For each point i such that $1 \le j[i] \le k$, denote by $D[i]$ the minimum interval length of $j = j[i]$ intervals, constrained to the red points $s_1, \ldots, s_j$, that cover the points in $P \cup Q$ from 1 to i.

Denote by $\mathrm{d}ist(i_1, i_2)$ the distance between two points with indices i_1 and i_2 in $P \cup Q$. The dynamic programming algorithm has the following base case for each $i \in L_1$,

$$D[i] = \mathrm{d}ist(1, i),$$

and the following recurrence for each $i \in L_j$, $j = 2, \ldots, k$,

$$D[i] = \min_{t \in L_{j-1}} \max \big\{ D[t],\, \mathrm{d}ist(t+1, i) \big\}.$$

Note that $D[t]$ is an increasing function of t for $t \in L_{j-1}$, and that $\mathrm{d}ist(t+1, i)$ is a decreasing function of t for $1 \le t < i$. Thus, by a binary search, we can compute $D[i]$ for each $i \in L_j$ in $O\big(\log(|L_{j-1}| + 1)\big)$ time, for increasing values of j from 2 to k. The desired entry is $D[n+k]$. The overall running time is clearly $O((n+k)\log(n+k))$. □

Proposition 2. *There exists an exact algorithm running in $O(\frac{1}{k}(n+k)^2 \log(n+k))$ time for* Constrained k-center *on a closed curve.*

Proof. A closed curve containing n black points and k red points has a subcurve containing at most n/k black points between two red points. For each pair of consecutive points on this subcurve, we can cut the curve between the pair and obtain an instance of the problem Constrained k-center on a line. There are at most $(n+k)/k$ such instances on a line, and each of them can be solved exactly in $O((n+k)\log(n+k))$ time by Proposition 1. The overall optimal solution for these instances on a line is an optimal solution for the original instance on a closed curve. □

Proposition 3. *There exists a polynomial-time exact algorithm based on linear-programming for* Movement to Independence *on the line.*

Proof. Sort the points in P by increasing x-coordinates $a_1 \le a_2 \le \ldots \le a_n$. Observe that in an optimal solution, no two points in P need to swap their order. Let $x_1 \le x_2 \le \ldots \le x_n$ be the new x-coordinates of the points after the move (the ith point moves from a_i to x_i). Computing an optimal solution amounts to solving the following linear program with the n variables x_i and $3n-1$ constraints:

$$\begin{aligned} &\text{minimize} \quad z && \text{(LP1)} \\ &\text{subject to} \quad \begin{cases} x_{i+1} - x_i \ge \Delta, & 1 \le i \le n-1 \\ x_i - a_i \le z, & 1 \le i \le n \\ -x_i + a_i \le z, & 1 \le i \le n \end{cases} \end{aligned}$$

□

While Movement to Independence on the line is always feasible, this is not the case for the new variant on a closed curve. Let γ be a closed curve of length $L = |\gamma|$. Obviously, Movement to Independence on γ admits a solution if and only if $L \ge n\Delta$. We show next that an exact solution can still be found via linear-programming.

Proposition 4. *There exists a polynomial-time exact algorithm based on linear-programming for* Movement to Independence *on a closed curve.*

Proof. For simplicity, we can assume that γ is drawn in the plane as a circle centered at the origin, and that the input points (in P) are numbered counterclockwise, as $1, \ldots, n$ on γ, and their initial positions (γ-coordinates) are $0 \leq a_1 \leq a_2 \leq \ldots \leq a_n < L$. Refer to the point whose γ-coordinate is 0 as the *origin* of γ. As in the proof of Proposition 3, it is crucial to observe that there exists an optimal solution such that the circular order of the points in P on γ remains the same. Moreover, OPT $\leq L/2$, since the distance on γ from any (input) point to any other point is at most $L/2$.

Let $x_i \in [0, L]$, $i = 1, \ldots, n$, be the new γ-coordinates of the points after an optimal move (the ith point moves from a_i to x_i). Note that these coordinates uniquely identify the movement of the points, since OPT $\leq L/2$. Observe also that in an optimal solution, *not all* the points move in the same direction on γ (clockwise or counterclockwise): indeed, assuming such a move, the smallest of the moves can be canceled out from each move, with a strict decrease in the optimal solution, which would be a contradiction. So there must exist two adjacent points on γ that move away from each other (or stay put) in an optimal solution. We pick a new origin of γ between these two points (or coincident with one of them), and find an optimal movement for the points, subject to the constraint that no point crosses the origin of γ. We do this for all n pairs of adjacent points on γ.

It remains to show that these n cases can be implemented as n linear programs with the n variables x_i, where each LP has $O(n)$ constraints. Fix a pair of adjacent points as described above. Assume that $0 \leq a_1 \leq a_2 \leq \ldots \leq a_n < L$ are the new γ-coordinates of the n points on γ. These coordinates can be computed in $O(n)$ time as they implement a simple circular shift. An optimal movement for the points amounts to solving the following linear program:

$$\text{minimize} \quad z \tag{LP2}$$
$$\text{subject to} \quad \begin{cases} x_1 \geq 0, & \\ x_n \leq L, & \\ x_{i+1} - x_i \geq \Delta, & 1 \leq i \leq n-1 \\ x_1 + L - x_n \geq \Delta, & \\ x_i - a_i \leq z, & 1 \leq i \leq n \\ -x_i + a_i \leq z, & 1 \leq i \leq n \end{cases}$$

The algorithm first checks whether the feasibility condition is met, and assuming it is, it solves the n linear programs (one for each adjacent pair of points on γ) and then selects the LP whose solution gives the overall minimum z. □

3 NP-hardness of Constrained k-center

Proof of Theorem 1. We show that Constrained k-center is NP-hard by a reduction from the NP-hard problem Planar-3SAT [11]. A reduction for k-center based on similar ideas, however from another problem, Planar-Vertex-Cover appears in [7]. Let (V, C) be a 3SAT instance consisting of a set V of n boolean variables and a set C of m clauses that are disjunctions of three literals. The instance (V, C) is planar if the corresponding bipartite graph G, with a vertex for each variable in V and each clause in C, and with an edge connecting a variable to a clause if and only if a literal of the variable occurs in the clause, is planar. We will construct a Constrained k-center instance I that has a feasible solution if and only if the Planar-3SAT instance (V, C) is satisfiable.

We now describe our construction of the Constrained k-center instance I, which consists of a gadget for each variable, clause, and literal. The gadget for each clause is a single black point.

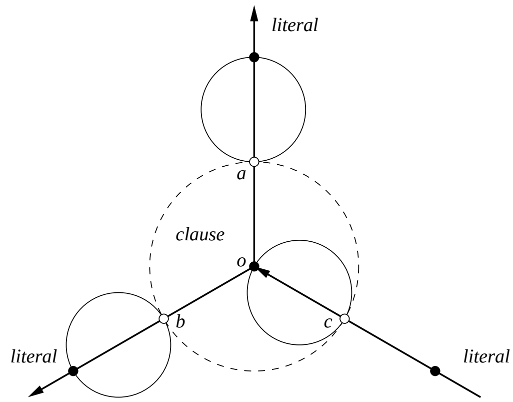

Figure 1: Connection between three literals in a clause. The three red points a, b, c (drawn as empty circles) come from three different literals, and are placed at the vertices of an equilateral triangle inscribed in a circle centered at the shared black point o; $|oa| = |ob| = |oc| = 2$. The black point o in the clause gadget is covered only if at least one of the three literals is true. In this example, the literal of c is true, and the literals of a and b are false.

The gadget for each variable is a closed chain of alternating black and red points. The gadget for each literal is an open chain of alternating black and red points, with a black point at one end and a red point at the other end. The literal gadgets model the incidence relation between the clauses and the variables: each literal gadget is connected to the corresponding clause gadget at the end with a red point, and to the corresponding variable gadget at the end with a black point.

We illustrate in Figure 1 the connection between the gadgets of a clause and its three literals, and in Figure 2 the connection between the gadgets of a literal and its variable. The distance between consecutive black and red points in each variable or clause gadget is exactly 2 (which is the diameter of a unit-radius disk) except at the junctions where a literal gadget is connected to a variable gadget (between c and e in Figure 2).

Write $|bc| = 2x$ for the configuration in Figure 2. Then

$$|ac| = |bd| = 2 + 2x,$$

and

$$|bf| = |cf| = \sqrt{\left(\sqrt{1 - x^2} + 3\right)^2 + x^2}.$$

Let x be the positive real solution to the equation

$$2 + 2x = \sqrt{\left(\sqrt{1 - x^2} + 3\right)^2 + x^2},$$

and let $y = 1 + x > 1$. Then $|ac| = |bd| = |bf| = |cf| = 2y$, and y is a solution to the following quartic equation

$$4y^4 - 11y^2 - 18y + 25 = 0.$$

A calculation shows that $y = 1.8279\ldots$. We arrange the gadgets in the plane such that (i) the connections between the gadgets follow precisely the configurations in Figure 1 and Figure 2, and (ii) each remaining point (except the single black point of each clause gadget, the two endpoints of each literal gadget, and the two points c and e of at each junction between a literal gadget and a variable gadget) has exactly two neighbors at distance 2 and no other neighbors within distance less than $2y$.

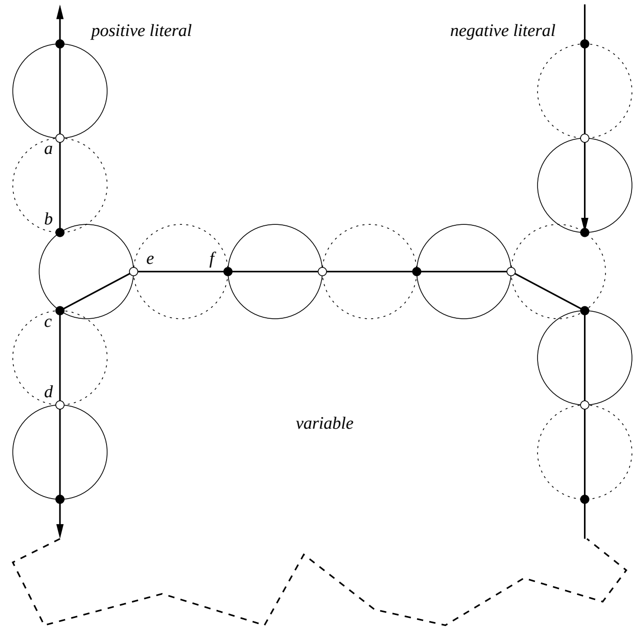

Figure 2: Connection between a variable and its literals. The four points d, c, e, f are part of a variable gadget; the two points a, b are part of a literal gadget; a, b, c, d are collinear; b, c, e are on a circle of unit radius; $ef \perp ad$; $|ab| = |cd| = |ef| = 2$, $|ac| = |bd| = |bf| = |cf|$. Red points are drawn as small empty circles. Large solid and dotted circles (of unit radius) correspond to true and false assignments, respectively.

We next show that the arrangement of the gadgets as specified above is indeed possible. Construct an expanded graph G' from the planar graph G by splitting each variable vertex to a cycle of literal vertices, one for each literal of the variable in the clauses. Each edge between a clause vertex and a variable vertex in G becomes an edge between the clause vertex and a distinct literal vertex in G', following the same circular order. The expanded graph G' is still planar and has maximum degree 3. It is known [13, 4] that for any planar graph of N vertices with maximum degree 4, there exists a polynomial-time algorithm that computes a rectilinear embedding of the graph in an integer grid, where each vertex of the graph is represented by a grid point, and each edge of the graph is represented by a rectilinear curve along the grid edges, such that the curves are non-crossing and the area of the rectilinear bounding box of the embedding is $O(N^2)$. We obtain such an embedding E of G', then scale it to a larger embedding $E' = \lambda E$. With a suitably large constant scaling factor λ, we can arrange the gadgets to satisfy the specified distances following the embedding E' of the planar graph G': The single black point in each clause gadget is placed at the grid point in E' for the corresponding clause vertex in G'. The three points b, c, e at each junction between a literal gadget and a variable gadget are placed near the grid point in E' for the corresponding literal vertex in G'. The chains in the literal and variable gadgets are aligned approximately with the rectilinear curves in E' for the corresponding edges in G'.

Assume that $1 \leq r < y$ in the following. It is easy to check that for any such r, a disk of radius r that contains a red point can contain at most one black point in our construction, except at the junction between each literal and its variable, where a disk may contain the red point e and the two black points b and c as in Figure 2. Now set the parameter k to the number of red points in the construction. Then the Planar-3SAT instance (V, C) is satisfiable if and only if the Constrained k-center instance I has a feasible solution with k disks of radius r.

To ensure that the reduction is polynomial, for any constant $\varepsilon > 0$, we approximate each coordinate of a point in our construction by a rational number with an encoding-length polynomial in n, m, and $1/\varepsilon$, such that each point is shifted by the encoding imprecision for a distance less than ε. Then the slackness in the disk radius r in our construction implies that CONSTRAINED k-CENTER is NP-hard to approximate within $y - \varepsilon$ for any constant $\varepsilon > 0$. $\square$

4 Approximation algorithm for Constrained k-center

Proof of Theorem 2. The idea of our approximation algorithm is very simple, namely to enumerate the approximate positions of an optimal constrained disk cover. Fix an optimal solution $O = \{\Omega_1, \ldots, \Omega_k\}$. Suppose that the red point q_j is covered by a disk Ω_j of radius r^* in the optimal solution O. Then the center c_j of Ω_j is contained in a disk D_j of radius r^* centered at q_j.

It is well-known that a disk of radius 1 can be covered by three smaller disks of radii $\frac{\sqrt{3}}{2}$, whose centers form an equilateral triangle, as shown in Figure 3. Now place three points around the red point q_j in an equilateral triangle formation (in some arbitrary orientation) such that the distance from q_j to each point is $\frac{1}{2}r^*$. Hence the disk D_j is covered by three smaller disks, E_{j1}, E_{j2}, E_{j3} of radius $\frac{\sqrt{3}}{2}r^*$ centered at the three points. Recall that c_j is contained in D_j, so it is covered by one of the disks E_{j1}, E_{j2}, E_{j3}. Let $F_{j1} \supset E_{j1}$, $F_{j2} \supset E_{j2}$, $F_{j3} \supset E_{j3}$, be three larger concentric disks of radius $(\frac{\sqrt{3}}{2} + 1)r^*$. Since Ω_j has radius r^*, it is covered by one of the larger disks F_{j1}, F_{j2}, F_{j3} of radius $(\frac{\sqrt{3}}{2} + 1)r^*$. So all black points covered by Ω_j are also covered by one of these three larger disks.

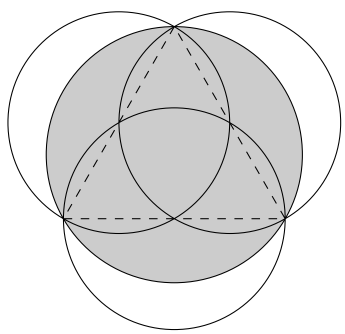

Figure 3: Covering a disk of radius 1 by three smaller disks of radius $\sqrt{3}/2$. The three sides of the equilateral triangle inscribed in the unit-radius disk are the diameters of the three smaller disks. The distance from the center of the unit-radius disk to the center of each smaller disk is 1/2.

By the preceding observation, given any candidate radius r, we can either find a feasible solution of k disks of radii $(\frac{\sqrt{3}}{2}+1)r$ by enumerating one of three possible disks for each red point and testing the black points for containment, all in $O(3^k k \cdot n)$ time, or decide (correctly) that there is no feasible solution with radius r. By (1), we can find a radius $\bar{r}$ such that $\frac{1}{2}r^* \leq \bar{r} \leq r^*$ in $O(kn)$ time. Then, by a binary search in the range $[\bar{r}, 2\bar{r}]$, we can obtain a $(\frac{\sqrt{3}}{2}+1+\varepsilon)$-approximation in $O(3^k k \cdot \log \frac{1}{\varepsilon} \cdot n)$ time, which is linear in n for any constants k and ε. In particular, since $\frac{\sqrt{3}}{2} + 1 = 1.8660\ldots$, we have a 1.87-approximation algorithm that runs in $O(3^k k \cdot n)$ time.

Similarly, a disk of unit radius can be covered by four disks of radius $\sqrt{2}/2 = 0.707\ldots$, and we get a 1.71-approximation in $O(4^k k \cdot n)$ time. By an old result of Neville [12], a disk of unit radius can be covered by five disks of radius $0.609383\ldots$, and we get a 1.61-approximation in $O(5^k k \cdot n)$ time. To obtain a finer approximation, note that a disk of radius 1 can be covered by $O(\varepsilon^{-2})$

disks of radius ε. Our algorithm can be obviously generalized to obtain a $(1+\varepsilon)$-approximation in $O(\varepsilon^{-2k} \cdot n)$ time. $\square$

Remark. Our algorithm can be obviously generalized to obtain a $(1+\varepsilon)$-approximation to Constrained k-center in $\mathbb{R}^d$ in $O(\varepsilon^{-dk} \cdot n)$ time.

5 Movement to Independence: NP-hardness and approximation

Proof of Theorem 3. To verify the NP-hardness, we make a reduction from the problem Dispersion in congruent disks in a set $\mathcal{D}$ of disks of radius λ via the following claim, whose proof is immediate from the definitions of the two problems.

Claim. Dispersion in congruent disks in a set $\mathcal{D}$ of disks of radius λ is feasible if and only if Movement to Independence for the center points of the disks in $\mathcal{D}$ can be attained with a maximum movement at most λ. $\square$

Proof of Theorem 4. Let δ be the minimum pairwise distance $\min_{i,j} |p_i p_j|$ of the points in P. We claim that if $\delta \leq x \leq 1$, then

$$\mathrm{OPT}(1) \geq \mathrm{OPT}(x) + (1-x)/2.$$

To see that the claim is true, imagine all points move from the start configuration to a target configuration with the same speed as in an optimal solution for OPT(1). Pause the points as soon as their minimum pairwise distance is x. Then the movement is at least OPT(x) before the pause and at least $(1-x)/2$ after the pause.

Let $c = \frac{1}{3+2/\sqrt{3}} = 0.24\ldots$. We now give an algorithm that moves the points to minimum pairwise distance at least c using maximum movement at most OPT(1). Consider two cases:

(1) $\delta \geq c$ Stay put.

(2) $\delta < c$ Use the algorithm of Demaine et al. [3] with a smaller grid of size $x = c$. Then the maximum movement is at most

$$\mathrm{OPT}(c) + \left(1 + \frac{1}{\sqrt{3}}\right) c \leq \mathrm{OPT}(1) - \frac{(1-c)}{2} + \left(1 + \frac{1}{\sqrt{3}}\right) c = \mathrm{OPT}(1). \qquad \square$$

References

[1] F. Aurenhammer and R. Klein, Voronoi diagrams, in *Handbook of Computational Geometry (J-R. Sack and J. Urrutia, eds.)*, Chap. 5, Elsevier, 2000, pp. 201–290.

[2] S. Cabello: Approximation algorithms for spreading points, *Journal of Algorithms*, **62** (2007), 49–73.

[3] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, and M. Zadimoghaddam, Minimizing movement, *ACM Transactions on Algorithms*, **5** (2009), article 30.

[4] D. Dolev, T. Leighton, and H. Trickey, Planar embedding of planar graphs, *Advances in Computing Research*, **2** (1984), 147–161.

[5] A. Dumitrescu and M. Jiang, Dispersion in unit disks, *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science* (STACS 2010), Nancy, France, March 2010, pp. 311–322. Also at `http://arxiv.org/abs/0912.2125v2`.

[6] A. Dumitrescu and M. Jiang, Dispersion in disks, manuscript 2010, `http://arxiv.org/abs/0912.2125v3`.

[7] T. Feder and D. H. Greene, Optimal algorithms for approximate clustering, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, (STOC 1988), pp. 434–444.

[8] J. Fiala, J. Kratochvíl, and A. Proskurowski, Systems of distant representatives, *Discrete Applied Mathematics*, **145** (2005), 306–316.

[9] Z. Friggstad and M. R. Salavatipour, Minimizing movement in mobile facility location problems, in *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2009, pp. 357–366.

[10] P. Giannopoulos, C. Knauer, and S. Whitesides, Parameterized complexity of geometric problems, *Computer Journal*, **51(3)** (2008), 372–384.

[11] D. Lichtenstein, Planar formulae and their uses, *SIAM Journal on Computing*, **11** (1982), 329–343.

[12] E. H. Neville, On the solution of numerical functional equations, illustrated by an account of a popular puzzle and of its solution, *Proceedings of the London Mathematical Society*, **14** (1915), 308–326.

[13] L. G. Valiant, Universality considerations in VLSI circuits, *IEEE Transactions on Computers*, **30** (1981), 135–140.