Fault-Tolerant Euclidean k-Centres*

Stephane Durocher† Sahar Lamey† Pak Ching Li†

Abstract

Given a set P of n points in $\mathbb{R}^d$, the Euclidean k-centre problem seeks to select a set F of k points in $\mathbb{R}^d$ such that the maximum distance between any point in P and its nearest neighbour in F is minimized, i.e., cover P with k balls of minimum radius. Upon introducing a parameter ℓ, the ℓ-fault-tolerant Euclidean k-centre problem seeks to minimize the maximum distance from any point in P to its ℓth nearest neighbour in F, i.e., select k balls of minimum radius such that each point in P is contained in ℓ balls. We give an $O(n \log n)$-time algorithm that solves the ℓ-fault-tolerant k-centre problem exactly in $\mathbb{R}$. We show the problem is NP-hard in $\mathbb{R}^2$, and give an $O(nk/\ell)$-time algorithm that finds a 2-approximate solution.

1 Introduction

1.1 Motivation

Given positions for a set of *clients*, a facility location problem seeks to identify positions for a multiset of *facilities* to serve the clients while optimizing a given *cost* function on the relative positions of clients to facilities. In the *k-centre problem*, the cost function is the maximum distance between any client and its nearest facility. E.g., given the locations of 1000 houses, select where to place 10 electric vehicle charging stations such that the maximum distance between any house and its nearest charger is minimized. Chargers are sometimes occupied or out of service, which suggests including *fault tolerance* in the cost function. E.g., select where to place 10 chargers such that the maximum distance between any house and its second-nearest charger is minimized.

1.2 Definitions

Let $\text{dist}(u, v)$ denote the Euclidean (ℓ_2) distance between the points $u, v \in \mathbb{R}^d$. Equivalently, $\text{dist}(u, v)$ is the radius of the smallest ball centred at u that contains v. Given a multiset S of points in $\mathbb{R}^d$ and a positive integer $\ell \leq |S|$, let $\text{dist}_\ell(u, S)$ denote the distance from u to its ℓth nearest neighbour in S. Equivalently, $\text{dist}_\ell(u, S)$ is the radius of the smallest ball centred at u that contains ℓ points of S. Let ℓ, k, and n denote integers such that $1 \leq \ell \leq k \leq n$, let P denote a set[1] of n points in $\mathbb{R}^d$, and let F denote a multiset of k points in $\mathbb{R}^d$. The *cost* of F serving P with fault tolerance ℓ is

$$\text{cost}_\ell(P, F) = \max_{p \in P} \text{dist}_\ell(p, F). \quad (1)$$

Expressed in the terminology of facility location, P is a set of *clients* served by a multiset F of *facilities*. Interpreted geometrically, each point in P is covered by at least ℓ balls in the multiset of k balls of *radius* $r = \text{cost}_\ell(P, F)$, centred on points in F. We examine the problem of finding a multiset F that minimizes (1) for given P, k, and ℓ.

Definition 1 (ℓ-fault-tolerant Euclidean k-centre) *An (optimal)* ℓ-fault-tolerant Euclidean k-centre *of P is a multiset*

$$F^* = \arg\min_{|F|=k} \text{cost}_\ell(P, F). \quad (2)$$

A multiset F' is an *α-approximate* ℓ-fault-tolerant Euclidean k-centre of P if $\text{cost}_\ell(P, F') \leq \alpha \cdot OPT$, where $OPT = \text{cost}_\ell(P, F^*)$. Interpreted geometrically, covering each point in P by ℓ balls centred on points in F' requires a radius at most α times larger than the radius of balls centred on points in an optimal solution F^*.

1.3 Contributions

In Section 4, we describe an $O(n \log n)$-time algorithm that solves the ℓ-fault-tolerant k-centre problem exactly in $\mathbb{R}$. In Section 5, we show the problem is NP-hard in $\mathbb{R}^2$. In Section 6, we describe an $O(nk/\ell)$-time algorithm that finds a 2-approximate ℓ-fault-tolerant k-centre in $\mathbb{R}^2$.

2 Related Work

When $\ell = 1$, Definition 1 corresponds to the *Euclidean k-centre* problem. I.e., the objective is to cover P with k balls of minimum radius. The Euclidean k-centre problem is NP-hard when k is an arbitrary input parameter [11] and $d \geq 2$, remains NP-hard to approximate

*This work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

†University of Manitoba, Canada,
stephane.durocher@umanitoba.ca, rahimzas@myumanitoba.ca, ben.li@cs.umanitoba.ca

[1] Whether P is a set or multiset of clients makes no difference to the cost of a solution (1). Due to the fault-tolerance parameter ℓ, however, an optimal solution to (2) may require collocating points in F; therefore, unlike P, F is a multiset.

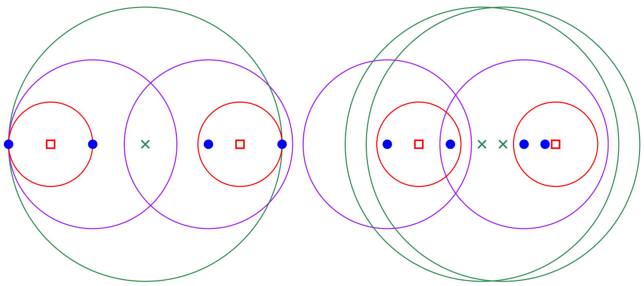

Figure 1: A set P of eight points (blue), a Euclidean 4-centre of P (red □), a discrete 4-centre of P (purple), and a 2-fault-tolerant Euclidean 4-centre of P (green ×) with two facilities collocated on the left ×.

within a factor of $(1+\sqrt{7})/2 \approx 1.8229$ [4], and has an $O(n \log k)$-time 2-approximation algorithm [4]. When $P \subseteq \mathbb{R}$ ($d = 1$), the Euclidean k-centre problem can be solved exactly in $O(n \log n)$ time [2].

The *discrete k-centre* problem examines the corresponding problem (when $\ell = 1$) in the graph setting, where the metric space is limited to the set of graph vertices. That is, facilities must be selected from the set of clients: $F \subseteq P$. Conversely, in the Euclidean k-centre problem, points in F may be positioned anywhere in $\mathbb{R}^d$. The corresponding radii of covering balls for these two versions of the k-centre problem can differ by a factor of two (see Figure 1).

The *ℓ-fault-tolerant discrete k-centre* problem for arbitrary $1 \leq \ell \leq k$ has been examined on graphs whose edge weights define a metric space, where the problem has been shown to be NP-hard, and for which no polynomial-time $(2-\epsilon)$-approximation algorithm is possible for any $\epsilon > 0$ unless P=NP [8]. Chaudhuri et al. [1] gave a polynomial-time 2-approximation algorithm.

As far as the authors are aware, nearly all previous work on fault-tolerant k-centre problems considers the *discrete* k-centre [1, 6, 8, 9]. The results in this paper consider the ℓ-fault-tolerant Euclidean k-centre problem, in which the facilities are not restricted to be a subset of the input point set, but can be any points in $\mathbb{R}^d$. Drezner [3] briefly examined the 2-fault-tolerant Euclidean 2-centre problem (k-centre for *unreliable facilities*) and analyzed the example in Figure 2.

3 Geometric Properties

As with k-centres, an ℓ-fault-tolerant Euclidean k-centre F^* is not unique in general, but $\text{cost}_\ell(P, F^*)$ (radius of the covering balls) is equal for all F^* that minimize (2).

When $\ell > 1$, an ℓ-fault-tolerant Euclidean k-centre is a multiset that sometimes requires collocating multiple facilities in F at a common point (see Figure 1). A natural strategy for identifying k points that approximate an ℓ-fault-tolerant k-centre of P is to place ℓ points on

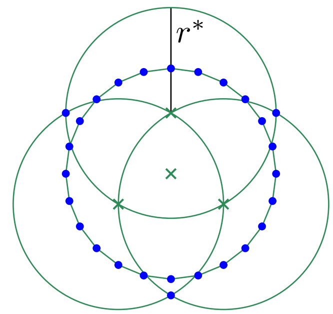

Figure 2: The set P of blue points has a 2-fault-tolerant Euclidean 4-centre (green ×) of radius r^*; all Euclidean 2-centres of P have radius strictly greater than r^* [3].

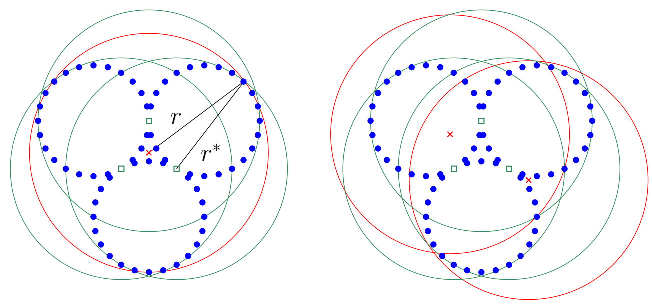

Figure 3: The set P of blue points has a 2-fault-tolerant Euclidean 6-centre (green □) of radius r^*, but all Euclidean 3-centres of P (e.g., red ×) have radius $r = (1/\sqrt{3} + 1/2)r^* \approx 1.0774r^*$.

each centre in a Euclidean $\lfloor k/\ell \rfloor$-centre of P. We examine this technique in Sections 4 and 6. As shown by Drezner [3], this strategy does not give an optimal ℓ-fault-tolerant Euclidean k-centre in $\mathbb{R}^2$ in general (see Figures 2 and 3).

Observation 1 (Drezner 1987 [3]) $\exists P \subseteq \mathbb{R}^2$, $k \in \mathbb{R}^+$, $\ell \in \mathbb{R}^+$ *such that no Euclidean* $\lfloor k/\ell \rfloor$*-centre of* P *is an* ℓ*-fault-tolerant Euclidean* k *centre of* P.

Finally, collocating facilities of F on a k-centre of P can result in $\text{cost}_\ell(P, F)$ that is arbitrarily larger than $r^* = \text{cost}_\ell(P, F^*)$ for an ℓ-fault-tolerant Euclidean k-centre F^* of P. That is, a Euclidean k-centre cannot guarantee any approximation of an ℓ-fault-tolerant Euclidean k-centre when $\ell > 1$ (see Figure 4).

Observation 2 $\forall \alpha \in \mathbb{R}$, $\exists P \subseteq \mathbb{R}^2$ *such that* $|P| = 4$ *and* $\text{cost}_2(P, F) > \alpha \cdot OPT$*, where* F *is a Euclidean 4-centre of* P*,* F^* *is a 2-fault-tolerant Euclidean 4-centre of* P*, and* $OPT = \text{cost}_2(P, F^*)$.

4 1D Algorithm

In this section, we give an $O(n \log n)$-time algorithm for computing an optimal ℓ-fault-tolerant Euclidean k-centre when the set P of clients is in $\mathbb{R}$. The algorithm

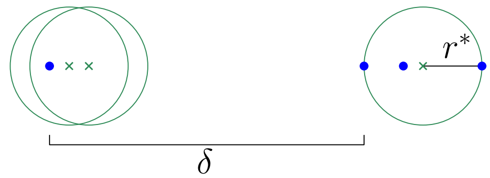

Figure 4: The set P of blue points has a 2-fault-tolerant Euclidean 4-centre of radius r^* (green $\times$) with two facilities collocated on the right $\times$. Setting F to a Euclidean 4-centre of P ($F = P$ in this case) gives $\mathrm{cost}_2(P, F) = \delta$, which can be arbitrarily larger than r^*.

computes an $\lfloor k/\ell \rfloor$-centre of P and outputs ℓ copies of this solution.

Lemma 1 *Given a set P of n points in $\mathbb{R}$, and integers k, ℓ such that $1 \le \ell \le k \le n$, if Z is an ℓ-fault-tolerant Euclidean k-centre of P with $d_Z = \mathrm{cost}_\ell(P, Z)$, then there exists a multiset Y that is a Euclidean k'-centre of P with $\mathrm{cost}_1(P, Y) \le d_Z$, where $k' = \lfloor k/\ell \rfloor$.*

Proof. Let $Z = \{\{z_1, z_2, ..., z_k\}\}$ be a multiset, where $z_1 \le z_2 \le \cdots \le z_k$. Define $Y = \{\{y_1, y_2, ..., y_{k'}\}\}$ where $y_i = z_{i\ell}$, for $i = 1$ to k'. Let $p \in P$. We will now show that $\mathrm{dist}_1(p, Y) \le d_Z$. Since Z is an ℓ-fault-tolerant k-centre of P with cost d_Z, there exist (at least) ℓ consecutive elements of Z that are within distance of d_Z from p. That is, there exists i such that $\mathrm{dist}(p, z_j) \le d_Z$ for $j = i, i+1, ..., i+\ell$. By definition of Y, one of these z_j is in Y. Therefore, $\mathrm{dist}_1(p, Y) \le d_Z$. □

Theorem 2 *Given a set P of n points in $\mathbb{R}$, and integers k, ℓ such that $1 \le \ell \le k \le n$, an optimal ℓ-fault-tolerant Euclidean k-centre can be computed in $O(n \log n)$ time.*

Proof. Let $k' = \lfloor k/\ell \rfloor$. The algorithm computes and returns ℓ copies of S, a Euclidean k'-centre of P.

Let ALG denote the cost of the solution returned by the algorithm and let OPT denote the cost of an optimal solution $\mathcal{O}$. Note that the cost of the Euclidean k'-centre solution S is ALG. By Lemma 1, we can obtain from $\mathcal{O}$ a k'-centre solution C with cost at most OPT. Since S is an optimal k'-centre with cost ALG, we see that $OPT \ge ALG$. Since $ALG \le OPT$ by definition, we conclude that $ALG = OPT$ and the solution computed by the algorithm is optimal.

The running time of the algorithm corresponds to the time required to compute a k'-centre of P. When $P \subseteq \mathbb{R}$, a Euclidean κ-centre of P can be computed in $O(n \log n)$ time for any κ, where $n = |P|$ [2]. Consequently, our algorithm computes an ℓ-fault-tolerant Euclidean k-centre of P in $O(n \log n)$ time. □

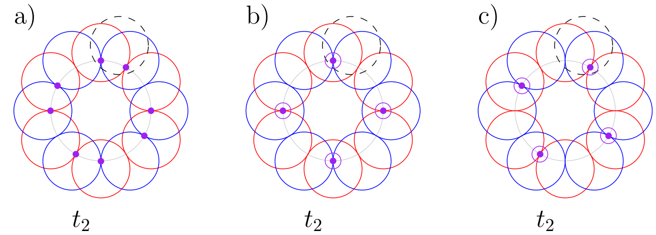

Figure 5: **Variable Gadget (true)**: These three configurations of eight purple points stab each red or blue disc with two points, as well as the dashed disc with two points. Pairs of purple points are collocated in the second and third configurations. The dashed disc is part of a clause gadget.

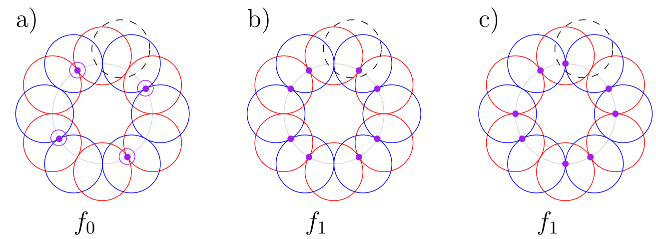

Figure 6: **Variable Gadget (false)**: These three configurations of eight purple points stab each red or blue disc with two points, but stab the dashed disc with only one (f_1) or zero (f_0) points. Pairs of purple points are collocated in the first configuration. The dashed disc is part of a clause gadget.

5 2D Hardness

We show that the 2-fault-tolerant Euclidean k-centre problem is NP-hard by describing a reduction from the Planar 3-SAT problem. An instance $\Phi = (U, E)$ of *3-SAT* consists of a set of variables $U = \{u_1, \ldots, u_n\}$ and a set of clauses $E = \{E_1, \ldots, E_s\}$. Each clause is the disjunction of three literals, each of which is either a variable in U or its negation, e.g., $E_1 = (u_1 \vee \neg u_2 \vee u_3)$. The objective is to determine if there exists an assignment of truth values to the variables in U such that all clauses in E are satisfied (evaluate to true). Every instance Φ of 3-SAT corresponds to a bipartite graph, G_Φ, with vertex set $U \cup E$, and an edge (u_i, E_j) if and only if the variable u_i is in the clause E_j. *Planar 3-SAT* is a restricted version of 3-SAT for which G_Φ is planar. Planar 3-SAT is NP-complete [10]. Furthermore, for every instance Φ of Planar 3-SAT, there exists a non-crossing rectilinear embedding of G_Φ in $\mathbb{R}^2$ [7].

We shall establish the NP-hardness of the Fault-tolerant Euclidean k-centre problem by proving that *Disc 2-Stabbing* is NP-hard. The input consists of a set P' of unit discs in $\mathbb{R}^2$ and an integer $k \ge 1$. The objective is to determine if there exists a multiset F of

k points such that each disc in P' contains (is stabbed by) at least two points of F. If so, we say F *2-stabs* P', which occurs if and only if F is a 2-fault-tolerant Euclidean k-centre of the set of centres of discs in P' with cost at most 1. That is, Disc 2-Stabbing reduces directly to ℓ-fault-tolerant Euclidean k-centre.

We now reduce Planar 3-SAT to Disc 2-Stabbing. Choose any instance $\Phi = (U, E)$ of Planar 3-SAT and a rectilinear planar embedding of G_Φ.

5.1 Variable Gadget

For each variable $u_i \in U$, we construct a cycle of $r_i = 3m_i$ overlapping discs that follows the edges adjacent to u_i in the drawing of G_Φ, where m_i is even (see Figures 5 and 6); let $C^i = \{C^i_0, C^i_1, ..., C^i_{r_{i-1}}\}$ denote[2] this set of discs. Position the discs in C^i such that $C^i_a \cap C^i_b \neq \varnothing$ if and only if $(a-b) \bmod r_i \in \{0,1,2\}$; that is, each disc in C^i intersects two discs ahead and two discs behind it in the cycle, but no other disc in C^i. Discs corresponding to u_i should not intersect discs corresponding to any other variable u_j (see Figure 11). That is,

$$\forall i,j,\ \forall C^i_x \in C^i,\ \forall C^j_y \in C^j,\ i \neq j \Rightarrow C^i_x \cap C^j_y = \varnothing.$$

A simple counting argument shows that if $|C^i| = 3m_i$, then at least $2m_i$ points are necessary to 2-stab C^i.

We begin by characterizing the possible 2-stabbing sets of C^i using $2m_i$ points and how these 2-stabbing sets correspond to truth values for variables in the reduction.

Observation 3 *If F' is a multiset of $2m_i$ points that 2-stabs C^i, then each $q \in F'$ must stab three discs in C_i; furthermore, $q \in C^i_j \cap C^i_{j+1} \cap C^i_{j+2}$ for some j.*

Lemma 3 *If F' is a multiset of $2m_i$ points that 2-stabs C^i, then $\forall j \in \{0,1,...,r_{i-1}\}$, $|C^i_j \cap F'| = 2$.*

Proof. By Observation 3, each point in F' stabs three discs in C^i. Since $|F'| = 2m_i$, there are $6m_i$ discs (counting multiplicities) stabbed by F'. Let D denote the multiset of discs stabbed by F'. Since F' is a 2-stabber of C^i, each disc of C^i must occur at least twice in D. As $|D|/|C^i| = 2$, we conclude that each disc of C^i appears exactly twice in D. That is, each disc of C^i is stabbed by exactly two points of F'. □

With Observation 3 and Lemma 3, Lemmas 4 and 5 below show that there are six combinatorially distinct configurations of $2m_i$ points that 2-stab C^i: three cases correspond to the truth value *True*, and the other three correspond to the truth value *False* (see Figures 5 and 6).

Lemma 4 *Suppose F' 2-stabs C^i and $|F'| = 2m_i$. Then for every j, $q_j \cap F' = \varnothing$, $q_{j+1} \cap F' = \varnothing$, or $q_{j+2} \cap F' = \varnothing$, where $q_x \in C^i_x \cap C^i_{x+1} \cap C^i_{x+2}$.*

[2] Indices are taken modulo r_i. E.g., C^i_j denotes $C^i_{j \bmod r_i}$.

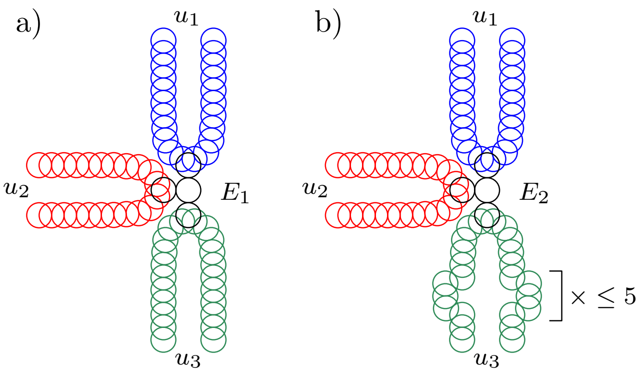

Figure 7: **Clause Gadget.** Each clause E_i is represented by a central disc tangent to three adjacent discs (black), each of which intersects the chain of discs corresponding to one of the three variables in the clause. This example illustrates two clauses: $E_1 = u_1 \vee u_2 \vee u_3$ and $E_2 = u_1 \vee u_2 \vee \neg u_3$. To negate the truth value of a variable, additional discs are added to the variable chain on either side of the clause gadget to switch the truth value of the variable where it intersects the black disc. In this example, one additional disc is added to each side of the green chain to negate u_3 in E_2. Since the number of discs in each variable disc chain must be a multiple of 6, additional discs may be required on the other side of the intersection with the black disc.

Lemma 4 implies that F' cannot contain three points that lie on consecutive intersections of three discs.

Proof. Suppose there is a j such that $\{q_j, q_{j+1}, q_{j+2}\} \subseteq F'$. Without loss of generality, assume $j = 0$. In this case, $|C^i_2 \cap F'| = 3$, contradicting Lemma 3. □

Lemma 5 *Suppose F' 2-stabs C^i, $|F'| = 2m_i$, and at least one point occurs twice in F'. Then every point occurs twice in F'.*

Proof. Suppose some point q_j occurs twice in F' and some point $q_{j'}$ occurs only once in F'. Without loss of generality, suppose $j = 0$, $j' > 0$, and $j' - j$ is minimum among all such q_j and $q_{j'}$, where $q_x \in C^i_x \cap C^i_{x+1} \cap C^i_{x+2}$. By Lemma 3, $j' \notin \{1,2\}$. In order to stab the disc C^i_3, F' must contain q_3. Therefore, $j' = 3$ and, by assumption, q_3 occurs only once in F'. Consequently, F' cannot 2-stab the disc C^i_3, contradicting the assumption that F' 2-stabs C^i. □

We now characterize all possible configurations F' of $2m_i$ points that 2-stab C^i. If F' contains some point twice, then by Lemma 5 F' has m distinct points, each of which appears twice in F'. By Lemma 3, F' must be the multiset $\{\{q_{0+j}, q_{0+j}, q_{3+j}, q_{3+j}, ..., q_{r_i-3+j}, q_{r_i-3+j}\}\}$, for some $j \in \{0,1,2\}$, corresponding to Figures 5(b), 5(c), and 6(a), respectively. In our reduction, we associate

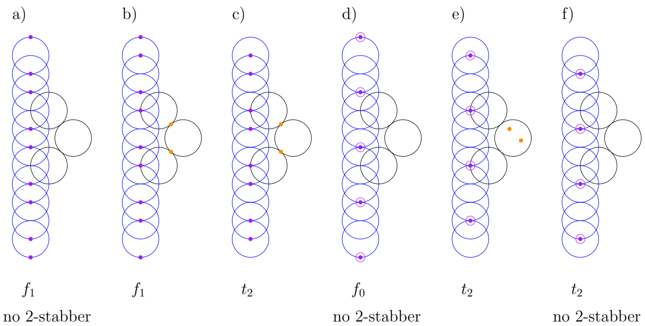

Figure 8: Adding three constraint discs (black) to each variable gadget reduces the number of possible configurations from six to three; the configurations b, c, and e can each be 2-stabbed with two additional points (orange), but a, d, and f cannot.

the 2-stabbing set for $j = 0, 1$ with *True* and the 2-stabbing set for $j = 2$ with *False*.

The only other possibility is that F' contains $2m$ distinct points. In this case, Lemmas 3 and 4 imply that F' is the set $\{q_{0+j}, q_{1+j}, q_{3+j}, q_{4+j}, ..., q_{r_i-3+j}, q_{r_i-2+j}\}$, for some $j \in \{0, 1, 2\}$, corresponding to Figures 5(a), 6(b), and 6(c), respectively. In our reduction, we associate the 2-stabbing set for $j = 0$ with *True* and the 2-stabbing set for $j = 1, 2$ with *False*.

Consequently, exactly six configurations are possible for F'. We further restrict the possible configurations by adding three *constraint discs* to each literal's chain of discs, as illustrated in Figure 8. In three cases, two of the constraint discs contain one or two points that stab C^i, and two additional points are necessary and sufficient to 2-stab all three constraint discs. In the remaining three cases, all three constraint discs are empty and cannot be 2-stabbed by any two additional points. Therefore, this reduces the number of possible configurations to three. Position the three constraint discs such that Figures 8c, e, and b correspond to Figures 5a, 5b, and 6c, respectively. This allows negating a variable before it meets a clause gadget (if the variable is negated in that clause) by adding discs (see Figure 7).

5.2 Clause Gadget and Reduction

We now describe the clause gadgets in our reduction. Each clause $E_h \in E$ is represented by four discs: one central disc that intersects three other discs, each in a point (see Figures 7 and 10). If the clause contains the literal u_i, then the corresponding clause gadget intersect a disc C^i_j such that $j \bmod 3 \in \{0, 1\}$; if the clause contains the literal $\neg u_i$, then $j \bmod 3 = 2$. The index j of the intersecting disc can be controlled by adding discs (see Figure 7). If the assignment of a truth value to u_i implies that some point $q \in C^i_{j-1} \cap C^i_j \cap C^i_{j+1}$ is selected for the solution F, then q may be selected so as to belong to the black disc as well. These correspond exactly to configurations of points for true literals (see Figure 5 and 6).

Observation 4 *A clause gadget can be 2-stabbed by two points if and only if at least one of its literals is true. Furthermore, at least two points are required to 2-stab the clause gadget for every combination of truth assignments to its literals.*

Let Ψ denote the set $\bigcup_{i=1}^{n} C^i$ of discs in variable gadgets, along with the $4s$ discs for the s clauses in E, and the $3n$ constraint discs for the n variables in U. Let

$$k = 2s + 2n + 2\sum_{i=1}^{n} m_i. \qquad (3)$$

We now prove that E is satisfiable if and only if there exists a set F of k points that 2-stabs Ψ. First, assume that E is satisfied by a truth assignment Γ to U. For $i \in \{1, \ldots, n\}$, if $\Gamma(u_i) = $ *True*, then add to F $2m_i$ points that 2-stab C^i as in Figure 5a. Otherwise, $\Gamma(u_i) = $ *False*; add to F $2m_i$ points as in Figure 6b. So far, $|F| = 2\sum_{i=1}^{n} m_i$, and each set C^i of discs is 2-stabbed by F. By our assumption, each clause $E_h = u_x \vee u_y \vee u_z$ is satisfied; add to F two points per clause as in Figure 10 such that each clause's set of four discs is 2-stabbed by F. Finally, add to F two points per variable as in Figure 8, such that each variable's constraint discs are 2-stabbed by F. Therefore, $|F| = k$ and F 2-stabs Ψ.

To prove the converse, assume F is a set of k points that 2-stabs Ψ. We must show there exists a truth assignment Γ for U that satisfies E. By Observation 4, at least $2s$ points of F are required to 2-stab the clause gadgets and at least $2n$ points of F are required to 2-stab the constraint discs for variable gadgets (excluding points that 2-stab $\bigcup_{i=1}^{n} C^i$) by (3); this leaves at most $2\sum_{i=1}^{n} m_i$ points in F to 2-stab $\bigcup_{i=1}^{n} C^i$. By Observation 3 and Lemmas 4 and 5, $2m_i$ points are necessary in F to 2-stab each C^i; furthermore, these correspond to one of the configurations in Figures 5 or 6. That is, for each C^i, F uniquely determines the truth assignment $\Gamma(u_i)$ associated with that point configuration. By (3), this leaves $k - 2\sum_{i=1}^{n} m_i = 2s + 2n$ points with which to 2-stab the s clause gadgets (Figures 7 and 10) and the n sets of variable constraint discs (Figure 8). Since each clause gadget is 2-stabbed by F, by Observation 4, each clause in E is satisfied. Since a rectilinear embedding of G_Φ exists on a grid of size $3s \times 3s$, where $s = |E|$ denotes the number of clauses in Φ [10], and each unit length of each edge in the embedding can be represented by a chain of $O(1)$ unit discs, therefore the reduction has polynomial size. See Figure 11 for an example reduction. Given any set P' of unit discs and any set F of points in $\mathbb{R}^2$, it is straightforward to verify in polynomial time whether F 2-stabs P'.

Theorem 6 *Disc 2-Stabbing is NP-complete.*

Since every P' and F in $\mathbb{R}^2$ can be embedded in a flat in $\mathbb{R}^d$ for any $d \geq 2$, this gives the following theorem:

Theorem 7 *For all $d \geq 2$, the 2-fault-tolerant k-centre problem is NP-complete in $\mathbb{R}^d$.*

6 2D Approximation Algorithm

In this section, we give a simple 2-approximation algorithm based on the greedy k-centre approximation algorithm of Gonzalez [5].

Algorithm 1: Computing an approximate ℓ-fault-tolerant Euclidean k-centre solution

Input : A set $P \subset \mathbb{R}^2$ of n points, integers k, ℓ satisfying $0 < \ell \leq k < n$.
Output: A multiset of (at most) k points from P.

1 $k' \leftarrow \lfloor k/\ell \rfloor$
2 Select an arbitrary point $s_1 \in P$.
3 $S \leftarrow \{s_1\}$
4 $i \leftarrow 1$
5 **while** $|S| < k'$ **do**
6 $\quad s_{i+1} \leftarrow \arg\max_{p \in P \setminus S} \text{dist}_1(p, S)$
7 $\quad r_i \leftarrow \text{dist}_1(s_{i+1}, S)$
8 $\quad S \leftarrow S \cup \{s_{i+1}\}$
9 $\quad i \leftarrow i + 1$
10 **end**
11 **return** ℓ copies of S

When $\ell = 1$, Algorithm 1 is identical to the greedy 2-approximation algorithm of Gonzalez [5], which in each iteration locates a facility that is farthest from the facilities already selected. Line 7 of the algorithm is used only for the analysis of the algorithm.

Theorem 8 *Algorithm 1 is a 2-approximation for the ℓ-fault-tolerant Euclidean k-centre problem.*

Proof. Let $k' \leftarrow \lfloor k/\ell \rfloor$ and $S = \{s_1, s_2, ..., s_{k'}\}$ be the k' centres selected by Lines 1–10 of Algorithm 1, where s_i is the i^{th} facility selected. The algorithm returns ℓ copies of S and, therefore, the cost of the solution returned is equal to the cost of the Euclidean k'-centre solution S.

Let $r_{k'} = \text{cost}_1(P, S)$ and let $s_{k'+1}$ be a point in P that realizes $r_{k'}$. The value $r_{k'}$ is the cost of the solution returned by (Line 11) of the algorithm as each point in S appears ℓ times in the solution. Since $k < n$, we have $k' < n$ and, therefore, the point $s_{k'+1}$ is distinct from the points in S.

The values $r_1, \ldots, r_{k'-1}$ computed in Line 7 and $r_{k'}$ defined above satisfy $r_1 \geq \cdots \geq r_{k'-1} \geq r_{k'}$. These inequalities imply that any two points in $S \cup \{s_{k'+1}\}$ are at least a distance of $r_{k'}$ apart.

Let OPT denote the cost of an optimal solution for the ℓ-fault-tolerant k-centre instance. It suffices to show that $r_{k'} \leq 2 \cdot OPT$, which we show by contradiction.

Suppose $r_{k'} > 2 \cdot OPT$ and let $\mathcal{O}$ be an optimal solution, where $|\mathcal{O}| = k$. Consider any point $p \in S \cup \{s_{k'+1}\}$. As $p \in P$, there must be at least ℓ facilities in $\mathcal{O}$ that are within distance OPT from p. Since $r_{k'} > 2 \cdot OPT$ and by the triangle inequality, none of these facilities are within a distance of OPT from any member of $S \cup \{s_{k'+1}\} \setminus \{p\}$. This implies that there must be at least $(k' + 1)\ell > k$ facilities in $\mathcal{O}$, which is a contradiction. □

The running time of Algorithm 1 is dominated by the nested loops on Lines 5 and 6 that iterate k' and n times, respectively, giving a worst-case running time of $\Theta(nk') = \Theta(nk/\ell)$. Figure 9 gives an example showing that Algorithm 1 cannot guarantee any approximation factor better than 2 in general.

Theorem 8 holds for both the discrete and continuous versions of the ℓ-fault-tolerant k-centre problem over any metric space.

7 Discussion and Directions for Future Research

7.1 Approximation by a $\lfloor k/\ell \rfloor$-Centre

Our example in Figure 3 shows that placing ℓ points on each facility of a Euclidean $\lfloor k/\ell \rfloor$-centre of a set P of points cannot guarantee an approximation for the ℓ-fault-tolerant Euclidean k-centre better than $(1/\sqrt{3} + 1/2) \approx 1.0774$ in general. Theorem 8 shows that a 2-approximation is possible. It remains open to determine what approximation factors are possible in the range $[1/\sqrt{3} + 1/2, 2)$. Our 2-approximation algorithm takes $O(nk/\ell)$ time; it may be possible to apply techniques similar to those used by Feder and Greene [4] to reduce the running time to $O(n \log k)$.

7.2 Hardness of Approximation

We showed that the ℓ-fault-tolerant Euclidean k-centre problem is NP-hard in two or more dimensions. Feder and Greene [4] showed that the Euclidean k-centre problem is NP-hard to approximate within a factor of $(1+\sqrt{7})/2 \approx 1.8229$ in two or more dimensions. A similar hardness of approximation result likely holds for the ℓ-fault-tolerant Euclidean k-centre, but remains open at present. In fact, even without fault tolerance, it remains open whether any polynomial-time algorithm can guarantee an α-approximation of the Euclidean k-centre problem for any $\alpha \in ((1 + \sqrt{7})/2, 2)$ in $\mathbb{R}^2$. No such gap exists for the discrete k-centre problem, for which a polynomial-time 2-approximation exists, and it is NP-hard to find a $(2 - \epsilon)$-approximation for any $\epsilon > 0$ [5].

References

[1] S. Chaudhuri, N. Garg, and R. Ravi. The p-neighbor k-center problem. *Information Processing Letters*, 65:131–134, 1998.

[2] D. Z. Chen, J. Li, and H. Wang. Efficient algorithms for the one-dimensional k-center problem. *Theoretical Computer Science*, 592:135–142, 2015.

[3] Z. Drezner. Heuristic solution methods for two location problems with unreliable facilities. *Journal of the Operational Research Society*, 38(6):509–514, 1987.

[4] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *Proc. Symposium on Theory of Computing (STOC)*, 1988.

[5] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[6] S. Khuller, R. Pless, and Y. J. Sussmann. Fault tolerant K-center problems. *Theoretical Computer Science*, 242:237–245, 2000.

[7] D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5:422–427, 1992.

[8] S. Krumke. On a generalization of the p-center problem. *Information Processing Letters*, 56:67–71, 1995.

[9] N. Kumar and B. Raichel. Fault tolerant clustering revisited. In *Proc. Canadian Conference on Computational Geometry (CGGG)*, pages 103–108, 2013.

[10] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[11] N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

A Appendix: Figures

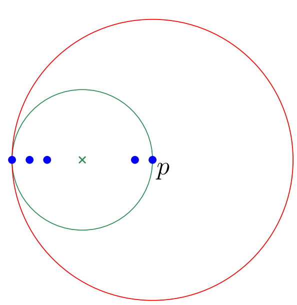

Figure 9: In this example, the unique optimal 2-fault-tolerant 2-centre has two facilities on $\times$, whereas Algorithm 1 could place two facilities on p.

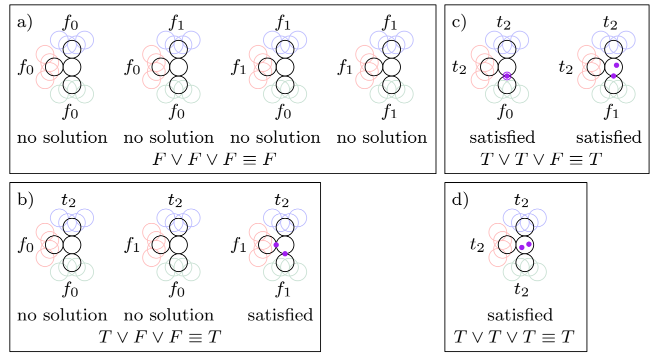

Figure 10: **Clause gadget configurations.** The black discs labelled f_0 and f_1 corresponds to false variables, and contain zero or one points, respectively. The black discs labelled t_2 correspond to true variables, and contain two points. Two additional points (purple) suffice to 2-stab clause gadgets that have one or more true literals (b–d). No configuration of two points can 2-stab a clause gadget with three false literals (a). Two purple points are collocated in the left configuration of (c).

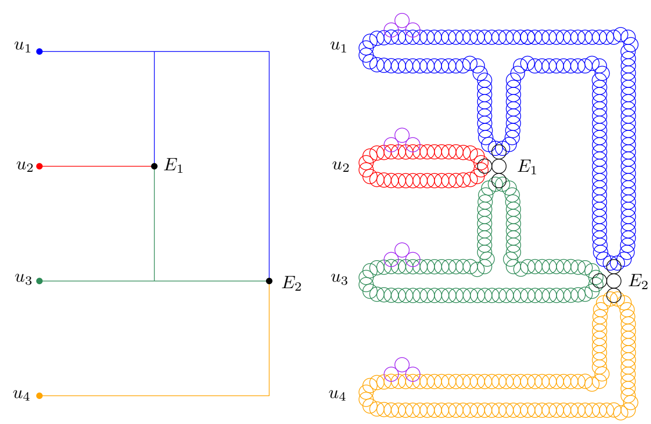

Figure 11: Given a rectilinear planar embedding of G_ϕ (left), we replace the edges adjacent to each variable in U with a cycle of overlapping discs (right) and each clause in E with four discs (black) that overlap the three cycles of discs corresponding to variables in that clause. Each variable also has one set of three constraint discs (purple). This example illustrates the reduction for an instance consisting of the clauses $E_1 = u_1 \vee u_2 \vee u_3$ and $E_2 = u_1 \vee u_3 \vee u_4$.