Fault Tolerance in Euclidean Committee Selection

Chinmay Sonar

Department of Computer Science, University of California, Santa Barbara, CA, USA

Subhash Suri

Department of Computer Science, University of California, Santa Barbara, CA, USA

Jie Xue

Department of Computer Science, New York University, Shanghai, China

Abstract

In the committee selection problem, the goal is to choose a subset of size k from a set of candidates C that collectively gives the best representation to a set of voters. We consider this problem in Euclidean d-space where each voter/candidate is a point and voters' preferences are implicitly represented by Euclidean distances to candidates. We explore *fault-tolerance* in committee selection and study the following three variants: (1) given a committee and a set of f failing candidates, find their optimal replacement; (2) compute the worst-case replacement score for a given committee under failure of f candidates; and (3) design a committee with the best replacement score under worst-case failures. The score of a committee is determined using the well-known (min-max) Chamberlin-Courant rule: minimize the maximum distance between any voter and its closest candidate in the committee. Our main results include the following: (1) in one dimension, all three problems can be solved in polynomial time; (2) in dimension $d \geq 2$, all three problems are NP-hard; and (3) all three problems admit a constant-factor approximation in any fixed dimension, and the optimal committee problem has an FPT bicriterion approximation.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Multiwinner elections, Fault tolerance, Geometric Hitting Set, EPTAS

Digital Object Identifier 10.4230/LIPIcs.ESA.2023.95

Related Version *Full Version*: `https://arxiv.org/abs/2308.07268`

1 Introduction and Problem Statement

We consider the computational complexity of adding fault tolerance into spatial voting. In spatial voting [7, 1, 29], the voters and the candidates are both modeled as points in some d-dimensional space, where each dimension represents an independent *policy issue* that is important for the election, and each voter's preference among the candidates is implicitly encoded by a distance function. For example, in the simplest 1-dimensional setting, voters and candidates are points on a line indicating their real-valued preference on a single issue. The specific setting for our work is *multiwinner* spatial elections, also called *committee selection*, in d dimensions where we have a set V of n voters, a set C of m candidates, and a committee size (integer) k. The goal is to choose a subset of k candidates, called the *winning committee*, that collectively best represents the preferences of all the voters [9, 11, 12].

One aspect of committee selection that appears not to have been investigated is *fault tolerance*, that is, how robust a chosen committee is against the possibility that some of the winning members may default. Committee selection problems model a number of applications in the social sciences and in computer science where such defaults are not uncommon, such as democratic elections, staff hiring, choosing public projects, locations of public facilities, jury selection, cache management, etc. [21, 14, 26, 2, 4, 23, 13]. In this paper, we are particularly interested in designing algorithms to address questions of the following kind: If some of the winning members default, how badly does this affect the overall score of the committee? Or,

31st Annual European Symposium on Algorithms (ESA 2023).
Editors: Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman; Article No. 95; pp. 95:1–95:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

how much does the committee score suffer if a *worst-case* subset of size f defaults? Finally, can we *proactively* choose a committee in such a way that it can tolerate up to f faults with the minimum possible score degradation? We begin by formalizing these problems more precisely and then describing our results.

Suppose $V = \{v_1, \ldots, v_n\}$ is a set of n voters and $C = \{c_1, \ldots, c_m\}$ is a set of m candidates, modeled as points in d-dimensional Euclidean space. (We occasionally call the tuple (V, C) an election E.) Given a positive integer k, we want to elect k candidates, called the *committee*, using the well-known Chamberlin-Courant voting rule [5]. This rule assigns a score to each committee as follows. Let $T \subseteq C$ be a committee. For each voter v, the score of T for v is defined as $\sigma(v, T) = \min_{c \in T} d(v, c)$, namely, the distance from v to its closest candidate in T.[1] The *score* of the committee T is defined as $\sigma(T) = \max_{v \in V} \sigma(v, T)$, namely, the largest distance between any voter and its closest neighbor in T. (In facility location parlance, this is the well-known k-center problem.)

The fault tolerance of a committee is parameterized by a positive integer f, which is the upper bound on the number of candidates that can fail.[2] Throughout the paper, we use the notation J to denote a failing set of candidates. We are allowed to replace the failing members of J with any set of at most $|T \cap J|$ candidates from $C \setminus J$. We often denote this set of *replacement* candidates by R. However, *we must keep all the non-failing members of T in the committee* – that is, the replacement committee is the set $(T \setminus J) \cup R$ – and throughout the paper our goal is to optimize this committee's score, namely $\sigma((T \setminus J) \cup R)$.

We consider the following three versions of fault-tolerant committee selection, presented in increasing order of complexity. The first problem is the simplest: given a committee and a failing set, find the best replacement committee.

Optimal Replacement Problem (ORP)
Input: An election $E = (V, C)$, a committee $T \subseteq C$ and a failing set $J \subseteq C$.
Goal: Find a replacement set $R \subseteq C \setminus J$ of size at most $|T \cap J|$ minimizing $\sigma((T \setminus J) \cup R)$.

Our second problem is to quantify the fault tolerance of a given committee T over worst-case faults. That is, what is the largest score of T's replacement when a worst-case subset of f faults occur? We introduce the following notation as T's measure of j-fault-tolerance, for any $0 \leq j \leq f$: $\sigma_j(T) = \max_{J \subseteq C \ s.t.\ |J| \leq j} \sigma((T \setminus J) \cup R)$, where R is an optimal replacement set with size at most $|T \cap J|$. We want to compute $\sigma_f(T)$. Occasionally, we also use the notation $\sigma_0(T)$ for the no-fault score of T, namely $\sigma(T)$.

Fault-Tolerance Score (FTS)
Input: An election $E = (V, C)$, a committee $T \subseteq C$ and a fault-tolerance parameter f.
Goal: Compute $\sigma_f(T)$.

Our third and final problem is to compute a committee with optimal fault-tolerance score.

Optimal Fault-Tolerant Committee (OFTC)
Input: An election $E = (V, C)$, a committee size k and a fault-tolerance parameter f.
Goal: Find $T \subseteq C$ of size at most k minimizing $\sigma_f(T)$.

---

[1] Originally, Chamberlin and Courant [5] defined a voting rule on Borda scores (also known as Borda-CC). In this paper, similarly to [2], we study a min-max version of this rule on a more general scoring function, which in our case is based on voter-candidate distances.

[2] In our work, we will allow any subset of size f from C to fail, so the faults can also include candidates not in the selected committee T. This only makes the problem harder because the adversary can always limit the faults to T, and elimination of candidates from $C \setminus T$ makes finding replacements for failing committee members more difficult.

1.1 Our Results

We first show that even in one dimension, fault-tolerant committee problems are nontrivial. In particular, while the Optimal Replacement Problem (ORP) is easily solved by a simple greedy algorithm, the other two problems, Fault-Tolerance Score (FTS) and Optimal Fault-Tolerant Committee (OFTC), do not appear to be easy. Our main result in one dimension is the design of efficient dynamic-programming-based algorithms for these two problems. Along the way, we solve a *fault-tolerant* Hitting Set problem for points and unit intervals, which may be of independent interest.

In two dimensions and higher, OFTC is NP-hard because of its close connection to the k-center problem. However, we show that even the seemingly simpler problem of optimal replacement (ORP) is also NP-hard. Our main results include a constant-factor approximation for all three problems in any fixed dimension (in fact, in any metric space), as well as a novel bicriterion FPT approximation via an EPTAS whose running time has the form $f(\epsilon)n^{\mathcal{O}(1)}$. For ease of reference, we show these results in the following table.

Table 1 Summary of our results.

<table>
<tr><th rowspan="2"></th><th rowspan="2">One-dimensional instances</th><th colspan="3">Dimension $d \geq 2$</th></tr>
<tr><th>Complexity</th><th>Approximation</th><th>Bounded f</th></tr>
<tr><td>ORP</td><td>P
(Theorem 1)</td><td>NP-hard
(Theorem 11)</td><td>3-approx.
(Lemma 12)</td><td>P
(full version)</td></tr>
<tr><td>FTS</td><td>P
(Theorem 6)</td><td>NP-hard
(Theorem 11)</td><td>3-approx.
(Lemma 13)</td><td>P
(full version)</td></tr>
<tr><td>OFTC</td><td>P
(Theorem 9)</td><td>NP-hard
(Theorem 11)</td><td>5-approx.
(Lemma 17)
Bicriterion-EPTAS
(Theorem 20)</td><td>NP-hard
(full version)
3-approx.
(Theorem 16)</td></tr>
</table>

Due to limited space, proofs of some of the theorems/lemmas (marked with ($\star$)) are deferred to the full version of the paper.

1.2 Related Work

To the best of our knowledge, the issue of fault tolerance in committee selection has not been studied in voting literature – their primary focus is on protocols and algorithms for choosing candidates [11, 12, 25, 27, 28, 8, 3]. However, the following two lines of work consider some related issues. First, in the "unavailable candidate model" [22, 15] the goal is to choose a *single winner* with maximum expected score when candidates fail according to a given probability distribution; in contrast, we consider multiwinner elections under worst-case faults. In the second line of work, a set of election control problems are considered where candidates are added [10] or deleted [17] to change the outcome of the election. In this setting, the candidate set is modified to obtain a favorable election outcome, which is a rather different problem than ours.

In the facility-location research, there has been prior work on adding fault tolerance to k-center or k-median solutions [6, 20, 19, 30, 16], but the main approach there is to assign each user (voter) to multiple facilities (candidates). In particular, the "p-neighbor k-center" framework [6] minimizes the maximum distance between a user and its pth center as a way to protect against $p-1$ faults. This formulation, however, differs from our optimal fault-tolerant committee problem (OFTC) because in our setting the replacement candidates are chosen *after* failing candidates are announced. Therefore, in the OFTC problem, the designer does not have to *simultaneously* allocate p neighbors for all the voters. Furthermore, to the best of our knowledge, neither of our first two problems – Optimal Replacement (ORP) and Fault-Tolerance Score (FTS) – have been studied in the facility-location literature, and initiate a new research direction. We also formulate and solve a fault-tolerant hitting set problem in one dimension, which may be of independent interest.

2 Fault-Tolerant Committees in One Dimension

Even in one dimension, computing the fault-tolerance score of a given committee or finding a committee with minimum fault-tolerance score is nontrivial. The optimal replacement problem, however, is easy – a simple greedy algorithm works. Our main result in this section is to design efficient dynamic-programming algorithms for the former two problems. In doing so, we also solve the fault-tolerant version of Hitting Set for points and unit segments.

2.1 Optimal Replacement Problem

In the Optimal Replacement Problem (ORP), we are given a committee $T \subseteq C$ and a failing set $J \subseteq C$, and we must find a replacement set R minimizing the score $\sigma((T \setminus J) \cup R)$, where $|R| \leq |T \cap J|$. Since this score is always the distance between some voter-candidate pair, it suffices to solve the following decision problem: Is there a replacement set with score at most r? We can then try all possible $O(nm)$ distances to find the smallest feasible replacement score.

This decision problem is equivalent to the following hitting set problem: for each voter $v \in V$, let I_v be the interval of length $2r$ centered at v, and let $\mathcal{I} = \{I_v : v \in V\}$ be the set of these n (voter) intervals. A subset of candidates is a hitting set for $\mathcal{I}$ if each interval contains at least one of the candidates. In our problem, we are given a hitting set T and a failing subset of candidates J, and we must find the minimum-size replacement hitting set. Such a replacement is easily found using the standard greedy algorithm, as follows. We first remove all of the intervals from $\mathcal{I}$ that are already hit by a candidate in $T \setminus J$, and we also remove all the failing candidates J from C. For the leftmost remaining interval, we then choose the rightmost candidate c contained in it, add it to R, delete all intervals hit by c, and iterate until all remaining intervals are hit. If we ever encounter an interval containing no candidate, or if the size of the replacement set is larger than $|T \setminus J|$, the answer to the decision problem is no. Otherwise, the solution is R. The greedy algorithm is easily implemented to run in time $\mathcal{O}((m+n)\log(m+n))$. To find the optimal replacement set, we can do a binary search over $O(nm)$ values of r and find the smallest r for which $|(T \setminus J) \cup R| \leq k$.

▶ **Theorem 1.** *The Optimal Replacement Problem can be solved in time $\mathcal{O}((m+n)\log^2(m+n))$ for one-dimensional Euclidean elections.*

2.2 Computing the Fault-Tolerance Score (FTS) of a Committee

We now come to the more difficult problem of computing the fault-tolerance score $\sigma_f(T)$ of a committee T in one dimension, which is the worst case over all possible failing sets of T. Once again it suffices to solve the following decision problem: given a size-k committee T and a real number r, can we find a replacement with score at most r for *every* failing subset of size f? Using our hitting set formulation, $\sigma_f(T) \leq r$ if and only if T is an *f-tolerant hitting set* of $\mathcal{I}$, that is, for any failing set $J \subseteq C$ of size at most f, there exists a replacement set $R \subseteq C \setminus J$ such that $|(T \setminus J) \cup R| \leq |T|$ and $(T \setminus J) \cup R$ hits $\mathcal{I}$. (Recall that each member of $\mathcal{I}$ is an interval of length $2r$ centered at one of the voter positions.) We can then compute the fault-tolerance score of T by trying each of the $O(nm)$ voter-candidate distances to find the smallest r for which this decision problem has a positive answer.

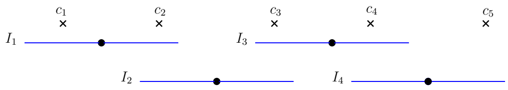

Figure 1 The figure shows an interval hitting set instance with four intervals and five points. The set $\{c_2, c_4\}$ is a feasible hitting set. For $X = \{c_2, c_3, c_5\}$, the intervals I_1, I_3, I_4 are X-disjoint.

We solve this fault-tolerant hitting set decision problem by observing that the size of a *smallest* hitting set equals the size of a *maximum* independent set, defined with respect to candidate points and voter intervals in the following way. Suppose the intervals of $\mathcal{I} = \{I_1, \ldots, I_n\}$ are sorted left to right. First, we can assume without loss of generality that $|I_i \cap C| > f$ for all $i \in [n]$, since otherwise there is no f-tolerant hitting set for $\mathcal{I}$. Given a set of points X in $\mathbb{R}$, we say that a set of intervals is *X-disjoint* if each point in X is contained in at most one interval. (That is, X-disjoint intervals can be thought of as *independent* in that they contain disjoint sets of points in X). The following claim is easy to prove.

▶ **Lemma 2.** *Given a set of points X and a set of intervals $\mathcal{J}$ on the real line, the size of a minimum hitting set $X' \subseteq X$ of $\mathcal{J}$ equals the maximum size of an X-disjoint subset of $\mathcal{J}$.*

Thus, if $T \subseteq C$ is an f-tolerant hitting set for $\mathcal{I}$, then for any failing set $J \subseteq C$, the size of any $(C \setminus J)$-disjoint subset of $\mathcal{I}$ is at most $|T|$. One should note that the size of the maximum $(C \setminus J)$-disjoint subset in $\mathcal{I}$ is a monotonically increasing function of $|J|$ – as more candidates fail, more intervals can become disjoint. Our goal is to find the maximum size of such a disjoint interval family over all possible failure sets J of size at most f. We will do this using dynamic programming, by combining solutions of subproblems, where each subproblem corresponds to an index range $[i, j]$, over the set of candidate points $c_1, \ldots, c_m$. Assuming that the candidate points $C = \{c_1, \ldots, c_m\}$ are ordered from left to right, our subproblems are defined as follows, for $1 \leq i \leq j \leq m$:

- $C_{i,j} = \{c_i, \ldots, c_j\}$ is the set of candidates in the range $[c_i, c_j]$.
- $\mathcal{I}_{i,j} = \{I \in \mathcal{I} : I \cap C \subseteq C_{i,j}\}$ is the set of intervals that only contain points from $C_{i,j}$.
- For any $J \subseteq C_{i,j}$, $\delta_{i,j}(J)$ is the maximum size of a $(C_{i,j} \backslash J)$-disjoint subset of $\mathcal{I}_{i,j}$.
- The subproblems we want to solve are the values $\delta_{i,j}(f) = \max_{J \subseteq C_{i,j}, |J| \leq f} \delta_{i,j}(J)$.

The key technical lemma of this section is the following claim.

▶ **Lemma 3.** *$T \subseteq C$ is an f-tolerant hitting set of $\mathcal{I}$ if and only if $|T \cap C_{i,j}| \geq \delta_{i,j}(f)$, for all $1 \leq i \leq j \leq m$.*

Proof. We first show the "if" part of the lemma. Assume $|T \cap C_{i,j}| \geq \delta_{i,j}(f)$ for all $i, j \in [m]$ with $i \leq j$. To see that T is an f-tolerant hitting set of $\mathcal{I}$, consider a failing set $J \subseteq C$ of size at most f. We have to show the existence of a replacement set $R \subseteq C \backslash J$ such that $|(T \backslash J) \cup R| \leq |T|$ and $(T \backslash J) \cup R$ is a hitting set of $\mathcal{I}$. We write $T \backslash J = \{c_{i_1}, \ldots, c_{i_p}\}$, where $i_1 < \cdots < i_p$. For convenience, set $i_0 = 0$ and $i_{p+1} = m + 1$. By our assumption, every interval $I \in \mathcal{I}$ is hit by some point in C. Thus, either I is hit by $T \backslash J$ or I belongs to $\mathcal{I}_{i,j}$ where $i = i_{t-1} + 1$ and $j = i_t - 1$ for some index $t \in [p + 1]$. Now consider an index $t \in [p + 1]$. We write $T_t = T \cap C_{i,j}$ and define $R_t \subseteq C_{i,j} \backslash J$ as a minimum hitting set of $\mathcal{I}_{i,j}$. By Lemma 2, the size of R_t is equal to the maximum size of a $(C_{i,j} \backslash J)$-disjoint subset of $\mathcal{I}_{i,j}$, which is nothing but $\delta_{i,j}(J \cap C_{i,j})$. Also, by assumption, we have $|T_t| = |T \cap C_{i,j}| \geq \delta_{i,j}(f) \geq \delta_{i,j}(J \cap C_{i,j})$. Therefore, $|R_t| \leq |T_t|$. Finally, we define $R = \bigcup_{t=1}^{p+1} R_t$. Clearly, $(T \backslash J) \cup R$ hits $\mathcal{I}$. So it suffices to show that $|(T \backslash J) \cup R| \leq |T|$. Since $|R_t| \leq |T_t|$ for all $t \in [p + 1]$, we have

$$|(T\backslash J) \cup R| = |T\backslash J| + \sum_{t=1}^{p+1} |R_t| \leq |T\backslash J| + \sum_{t=1}^{p+1} |T_t| = |T|,$$

which completes the proof of the "if" part.

Next, we prove the "only if" part of the lemma. Assume $T \subseteq C$ is an f-tolerant hitting set of $\mathcal{I}$. Consider two indices $i, j \in [m]$ with $i \leq j$. To show $|T \cap C_{i,j}| \geq \delta_{i,j}(f)$, it suffices to show that $|T \cap C_{i,j}| \geq \delta_{i,j}(J)$ for all $J \subseteq C_{i,j}$ with $|J| \leq f$. Since T is an f-tolerant hitting set of $\mathcal{I}$, there exists $R \subseteq C\backslash J$ such that $|(T\backslash J) \cup R| \leq |T|$ and $(T\backslash J) \cup R$ is a hitting set of $\mathcal{I}$. For brevity, let $T' = (T\backslash J) \cup R$. By definition, the intervals in $\mathcal{I}_{i,j}$ can only be hit by the points in $C_{i,j}$. Thus, $T' \cap C_{i,j}$ is a hitting set of $\mathcal{I}_{i,j}$. As $T' \cap C_{i,j} \subseteq C_{i,j}\backslash J$, by Lemma 2, the size of $T' \cap C_{i,j}$ is at least the maximum size of a $(C_{i,j}\backslash J)$-disjoint subset of $\mathcal{I}_{i,j}$, i.e., $|T' \cap C_{i,j}| \geq \delta_{i,j}(J)$. Furthermore, because $J \subseteq C_{i,j}$, we have $(T\backslash J)\backslash C_{i,j} = T\backslash C_{i,j}$. It follows that $T\backslash C_{i,j} \subseteq T'\backslash C_{i,j}$ and thus $|T\backslash C_{i,j}| \leq |T'\backslash C_{i,j}|$. For a committee T, we can partition T into two parts: the part containing candidates in $C_{i,j}$ and the part containing candidates outside of $C_{i,j}$. Hence, $|T| = |T \cap C_{i,j}| + |T\backslash C_{i,j}|$ and $|T'| = |T' \cap C_{i,j}| + |T'\backslash C_{i,j}|$. Because $|T'| \leq |T|$ and $|T'\backslash C_{i,j}| \geq |T\backslash C_{i,j}|$, we have $|T' \cap C_{i,j}| \leq |T \cap C_{i,j}|$. Therefore, $|T \cap C_{i,j}| \geq \delta_{i,j}(J)$. This completes the proof of Lemma 3. ◀

In order to decide if $\sigma_f(T) \leq r$, therefore, we just have to compute $\delta_{i,j}(f)$, for all i, j, and check the condition $|T \cap C_{i,j}| \geq \delta_{i,j}(f)$. We now show how to do that efficiently.

Efficiently Computing $\delta_{i,j}(f)$

For ease of presentation, we show how to compute $\delta_{1,m}(f)$; computing other $\delta_{i,j}(f)$ is similar. We have $C_{1,m} = C$, $\mathcal{I}_{1,m} = \mathcal{I}$, and $\delta_{1,m}(f)$ is size of the largest subset of $\mathcal{I}$ that is $(C\backslash J)$-disjoint for any failing set $J \subseteq C$ with $|J| \leq f$. The intervals of $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ are in the left to right sorted order and, for each $i \in [n]$, let $C(I_i) = C \cap I_i$ be the set of points in C that hits I_i. Define $\mathit{\Gamma}[i][j]$ as the maximum size of an $(C \setminus X)$-disjoint subset $\mathcal{J} \subseteq \{I_1, \ldots, I_i\}$ such that $X \subseteq C$ and $|X| \leq j$.

► **Lemma 4.** *We have the following recurrence*

$$\mathit{\Gamma}[i][j] \;=\; \max \left\{ \begin{array}{l} \mathit{\Gamma}[i-1][j], \\ \max\limits_{0 \leq i' \leq i} 1 + \mathit{\Gamma}[i'][j - |C(I_i) \cap C(I_{i'})|] \end{array} \right\}$$

Clearly, $\delta_{1,m}(f) = \mathit{\Gamma}[n][f]$. The base case for our dynamic program is $\mathit{\Gamma}[0, j] = 0$ for all $j \in [f]$ and $\mathit{\Gamma}[i][j] = -\infty$ for $j < 0$ and all $i \in [n]$. Our dynamic program runs in time $\mathcal{O}(n^2 m f)$. In the same way, we can compute the values of $\delta_{i,j}(f)$ for all $i, j \in [m]$ with $i \leq j$.

► **Lemma 5.** *$\delta_{i,j}(f)$, for all $1 \leq i \leq j \leq m$, can be computed in time $\mathcal{O}(n^2 m^3 f)$.*

Given a hitting set $T \subseteq C$ and the values $\delta_{i,j}(f)$, we can verify the condition in Lemma 3 in time $\mathcal{O}(m^3)$. We can then use binary search to find the smallest value of r for which T is an f-tolerant hitting set. This establishes the following result.

► **Theorem 6.** *The fault-tolerance score of a 1-dimensional committee T can be computed in time $\mathcal{O}(n^2 m^3 f \log(nm))$.*

2.3 Optimal Fault-Tolerant Committee

We now address the problem of *designing* a fault-tolerant committee: select a committee T of size k whose fault-tolerance score $\sigma_f(T)$ is minimized. Thus, our goal is *not* to optimize the fault-free score of T, namely $\sigma_0(T)$, but rather the score that the best replacement will have after a worst-case set of f faults in T, namely $\sigma_f(T)$. Following the earlier approach, we again focus on the decision question: given some $r \geq 0$, is there a committee of size k with $\sigma_f(T) \leq r$? For a given value of r, we construct our hitting set instance with candidate-points and voter-intervals, and compute a minimum-sized f-tolerant hitting set $T \subseteq C$ as follows:

1. Compute the value of $\delta_{i,j}(f)$, for all $1 \leq i \leq j \leq m$.
2. Compute a minimum subset $T \subseteq C$ satisfying $|T \cap C_{i,j}| \geq \delta_{i,j}(f)$, for all $1 \leq i \leq j \leq m$.
3. If $|T| \leq k$, we have a solution; otherwise, the answer to the decision problem is no.

Step (1) is implemented using the dynamic program of the previous subsection, and so it suffices to explain how to implement step (2). We assume without loss of generality that $|C_{i,j}| \geq \delta_{i,j}(f)$ for all i, j, because otherwise there is no solution. We compute a set T using the following greedy algorithm.

- Initialize $T = \emptyset$.
- For each c_k for $k \in [m]$, if there exists $i, j \in [m]$ with $i \leq k \leq j \leq m$ such that $\delta_{i,j}(f) \geq |T \cap C_{i,j}| + (j - k + 1)$, then add c_k to T.

The algorithm runs in time $\mathcal{O}(m^3)$. To prove correctness, we first claim the following.

▶ **Lemma 7.** *$|T \cap C_{i,j}| \geq \delta_{i,j}(f)$, for all $1 \leq i \leq j \leq m$.*

Proof. Suppose not, so we have $|T \cap C_{i,j}| < \delta_{i,j}(f)$, for some $i \leq j$. We recall that for any interval $I_i \in \mathcal{I}$, $|I_i \cap C| > f$. Therefore, for any failing set J, $C_{i,j} \setminus J$ is a hitting set of $\mathcal{I}_{i,j}$, and $|C_{i,j}| \geq \delta_{i,j}(f)$. This implies that there exists some point among $c_i, \dots, c_j$ that is not in T. Let $k \in \{i, \dots, j\}$ be the largest index such that $c_k \notin T$. For convenience, we use T' to denote the set T in the iteration of our algorithm that considers c_k. Note that $T \cap C_{i,j} = (T' \cap C_{i,j}) \cup \{c_{k+1}, \dots, c_j\}$ and $(T' \cap C_{i,j}) \cap \{c_{k+1}, \dots, c_j\} = \emptyset$. Therefore, $|T' \cap C_{i,j}| = |T \cap C_{i,j}| - (j-k) < \delta_{i,j}(f) - (j-k)$. This implies $|T' \cap C_{i,j}| + (j-k+1) \leq \delta_{i,j}(f)$. By our algorithm, in this case we should include c_k in T, which contradicts the fact that $c_k \notin T$. ◀

We now argue that T has the minimum size among all subsets of C satisfying the property of Lemma 7. Let opt be the minimum size of a subset of C satisfying the desired property. We write $T = \{c_{k_1}, \dots, c_{k_r}\}$, where $k_1 < \dots < k_r$.

▶ **Lemma 8** ($\star$). *For any $t \in [r]$, there exists a subset $T^* \subseteq C$ such that (1) $|T^* \cap C_{i,j}| \geq \delta_{i,j}(f)$ for all $i, j \in [m]$ with $i \leq j$, (2) $|T^*| = \mathsf{opt}$, and (3) $\{c_{k_1}, \dots, c_{k_t}\} \subseteq T^*$.*

We use binary search to find the smallest r such that the reduced instance has an f-tolerant hitting set of size at most k. Therefore, the following theorem holds.

▶ **Theorem 9.** *Optimal Fault-Tolerant Committee can be solved in time $\mathcal{O}(n^2 m^3 f \log(nm))$ for one-dimensional Euclidean elections.*

▶ Remark 10. Our dynamic programming algorithm works as long as either the set V or the set C is embedded in $\mathbb{R}$ (i.e., has a linear ordering), while the other set can have an arbitrary d-dimensional embedding. Moreover, we can also extend our algorithms to ordinal elections with (widely studied) single-peaked preferences [2, 24] to compute an optimal fault-tolerant Chamberlin-Courant committee.

3 Fault-Tolerant Committees in Multidimensional Space

We now consider fault tolerance in multidimensional elections. Unsurprisingly, the optimal committee design problem is intractable – it is similar to facility location – but it turns out that the seemingly simpler variants ORP and FTS are also intractable. In particular, we have the following (refer to the full version of the paper for details).

▶ **Theorem 11** (⋆). *All three problems (Optimal Replacement, Fault-Tolerance Score, and Optimal Fault-Tolerant Committee) are NP-hard, in any dimension $d \geq 2$ under the Euclidean norm, where size of the committee k and the failure parameter f are part of the input.*

3.1 Optimal Replacement Problem

A simple greedy algorithm achieves a 3-approximation for the Optimal Replacement Problem in any fixed dimension d as well as in any metric space.

▶ **Lemma 12.** *We can find a 3-approximation for ORP in time $O(k(nk + m))$.*

Proof. Let $T \subseteq C$ be the given committee and let $J \subseteq T$ be the failing set. In order to find the replacement set R, we initialize $\hat{T} = T \setminus J$, and then repeat the following two steps $|T \cap J|$ times: (1) Choose the farthest voter from $\hat{T}$, namely, choose $\hat{v} = \arg\max_{v \in V} d(v, \hat{T})$, and (2) Add to $\hat{T}$ the candidate $\hat{c} \notin \hat{T}$ that is closest to $\hat{v}$. Upon termination, we clearly have $|\hat{T}| = |T|$. Due to limited space, the proof of the approximation ratio is deferred to the full version of our paper. ◀

3.2 Computing the Fault-Tolerance Score

We can also approximate the optimal fault-tolerance score of a committee within a factor of 3. Specifically, if the optimal fault-tolerance score of T is $\sigma_f(T) = \sigma^*$, then our algorithm returns a real number σ' such that $\sigma^* \leq \sigma' \leq 3\sigma^*$.

For each voter v, let $d_f(v)$ be v's distance to its $(f+1)^{th}$ closest candidate, and let $d' = \max_{v \in V} d_f(v)$ be the maximum of these values over all voters. The basic idea behind our approximation is simple and uses the following two facts: (1) $\sigma^* \geq d'$, and (2) $\sigma^* \geq \sigma(T)$. The first one holds because d' is the best score possible if some voter's f nearest candidates fail, and the second one holds because a failure can only worsen the score (that is, $\sigma_f(T) \geq \sigma(T)$ for any $f > 0$). Therefore, the distance $\sigma' = d' + 2\sigma(T)$ is clearly within a factor of 3 of the optimal σ^*. We claim that for any failing set $J \subseteq C$, there exists a replacement $R \subseteq C \setminus J$ of size at most $|T \cap J|$ such that $\sigma((T \setminus J) \cup R) \leq \sigma'$. Due to limited space, we omit the proof from the extended abstract; it is included in the full version of our paper.

▶ **Lemma 13.** *The fault-tolerance score of a committee can be approximated within a factor of 3 in time $\mathcal{O}(nm \log(f))$.*

3.3 Optimal Fault-Tolerant Committee

We now discuss how to design approximately optimal fault-tolerant committees in multiwinner elections. Specifically, given a set of voters V and a set of candidates C in d-space, along with parameters k (committee size) and f (number of faults), we want to compute a size k committee $T \subseteq C$ with the minimum fault-tolerance score $\sigma_f(T)$. We prove two approximation results for this problem: (1) We can solve this problem within an approximation factor of 3 in polynomial time if the parameter f is treated as a constant (while k remains possibly unbounded). If f is not assumed to be a constant, we can solve the problem within an approximation factor of 5. (2) We give an EPTAS with running time $(1/\varepsilon)^{O(1/\varepsilon^{2d})}(m+n)^{O(1)}$ which is a *bicriterion* approximation, where the output committee T is fault-tolerant for at least $(1-\varepsilon)n$ voters with $\sigma_f(T) \leq (1+\varepsilon)\sigma^*$. The next two subsections discuss these results.

3.3.1 3-Approximation for Bounded f

Let σ^* be the optimal f-tolerant score of a committee of size k. We compute the approximation solution via an approximate decision algorithm, which takes as input a number $\sigma \geq \sigma_f(C)$ and returns a committee $T \subseteq C$ of size at most k with $\sigma_f(T) \leq 3\sigma$ if $\sigma \geq \sigma^*$. (We slightly abuse notation to introduce a convenient quantity $\sigma_f(C)$, which is the f-fault-tolerance score of a committee with all the input candidates. This is clearly a lower bound on any size k committee's score.)

For a committee $T \subseteq C$ and a failing set $J \subseteq C$, let $\delta(T, J)$ denote the score obtained after finding an optimal replacement K. That is,

$$\delta(T, J) = \min_{K \in C \setminus J, |K| = |T \cap J|} \sigma_0((T \setminus J) \cup K).$$

Thus, $\sigma_f(T) = \max_{J \subseteq C, |J| \leq f} \delta(T, J)$. Our approximation algorithm is shown in Algorithm 1. It begins with an empty committee T (line 1), and as long as there exists a failing set J of size at most f for which $\delta(T, J) > 3\sigma$, [3] we do the following.

First, we remove all candidates in J from T (line 3). Then, whenever there exists a voter $v \in V$ with $d(v, T) > 3\sigma$, we add to T a candidate $c \in C \setminus J$ whose distance to v is at most σ (lines 5-6). Such a c always exists because σ is at least the distance to the $(f+1)^{th}$ closest neighbor to v.

We call this voter v the *witness* of c, denoted by $\mathsf{wit}[c]$ (line 7). Adding c to T guarantees that $d(v, T) \leq \sigma$. We repeat this procedure (the inner while loop) until $d(v, T) \leq 3\sigma$ for all $v \in V$. Finally, the outer while loop terminates when $\delta(T, J) \leq 3\sigma$ for all $J \subseteq C$ of size at most f, i.e., $\sigma_f(T) \leq 3\sigma$. At this point, we return the committee T.

Algorithm 1 Approximate decision algorithm.

Input: a set V of voters, a set C of candidates, the committee size k, the fault-tolerance parameter f, and a number $\sigma \geq \sigma_f(C)$

1: $T \leftarrow \emptyset$
2: **while** $\exists\, J \subseteq C$ such that $|J| \leq f$ and $\delta(T, J) > 3\sigma$ **do**
3:     $T \leftarrow T \setminus J$
4:     **while** $\exists\, v \in V$ such that $d(v, T) > 3\sigma$ **do**
5:         $c \leftarrow$ a candidate in $C \setminus J$ satisfying $d(v, c) \leq \sigma$
6:         $T \leftarrow T \cup \{c\}$
7:         $\mathsf{wit}[c] \leftarrow v$
8: **return** T

▶ **Lemma 14** ($\star$). *Let T be the committee computed by Algorithm 1. Then $d(\mathsf{wit}[c], \mathsf{wit}[c']) > 2\sigma$ for any two distinct $c, c' \in T$.*

▶ **Lemma 15.** *If $\sigma \geq \sigma^*$, then Algorithm 1 outputs a size k committee T with $\sigma_f(T) \leq 3\sigma$.*

---

[3] We can check this condition by iterating over all failing sets of size f and computing an optimal replacement set in each case.

Proof. The condition of the outer while loop of Algorithm 1 guarantees that $\delta(T, J) \leq 3\sigma$ for all $J \subseteq C$ of size at most f, which implies $\sigma_f(T) \leq 3\sigma$. To prove $|T| \leq k$, suppose $T = \{c_1, \ldots, c_r\}$. By Lemma 14, the pairwise distances between the voters $\mathsf{wit}[c_1], \ldots, \mathsf{wit}[c_r]$ are all larger than 2σ and thus larger than $2\sigma^*$ (as $\sigma \geq \sigma^*$ by our assumption). Now consider a committee $T^* \subseteq C$ of size k satisfying $\sigma_f(T^*) = \sigma^*$. For each $\mathsf{wit}[c_i]$, there exists $c_i^* \in T^*$ such that $d(\mathsf{wit}[c_i], c_i^*) \leq \sigma^*$. Observe that $c_1^*, \ldots, c_r^*$ are all distinct. Indeed, if $c_i^* = c_j^*$ and $i \neq j$, then by the triangle inequality,

$$d(\mathsf{wit}[c_i], \mathsf{wit}[c_j]) \leq d(\mathsf{wit}[c_i], c_i^*) + d(\mathsf{wit}[c_j], c_j^*) \leq 2\sigma^*,$$

contradicting the fact that $d(\mathsf{wit}[c_i], \mathsf{wit}[c_j]) > 2\sigma^*$. Since $|T^*| = k$ and $c_1^*, \ldots, c_r^* \in T^*$, we have $r \leq k$, which completes the proof. ◀

Using these two lemmas, we can compute a 3-approximate solution using Algorithm 1 as follows. First, we compute $\sigma_f(C)$ in $O(nm^{f+1})$ time by enumerating all failing sets $J \subseteq C$ of size at most f. For every voter $v \in V$ and every candidate $c \in C$ such that $d(v, c) \geq \sigma_f(C)$, we run Algorithm 1 with $\sigma = d(v, c)$. Among all the committees returned of size at most k, we pick the one, say T^*, that minimizes $\sigma_f(T^*)$. To see that $\sigma_f(T^*) \leq 3\sigma^*$, note that σ^* must be the distance between a voter and a candidate. Thus, there is one call of Algorithm 1 with $\sigma = \sigma^*$, which returns a committee $T \subseteq C$ of size at most k such that $\sigma_f(T) \leq 3\sigma = 3\sigma^*$, by Lemma 15. We have $\sigma_f(T^*) \leq \sigma_f(T)$ by construction, which implies $\sigma_f(T^*) \leq 3\sigma^*$. In the full version, we show that each call of Algorithm 1 takes $O(nm^{2f+1})$ time. We need to call the algorithm $O(nm)$ times. Thus, we have the following result.

▶ **Theorem 16.** *We can find a 3-approximation for Optimal Fault-tolerant Committee in time $O(n^2 m^{2f+2})$, assuming the fault-tolerance parameter f is a constant.*

If we do not assume f to be a constant, then the well-known "farthest first" greedy rule for adding candidates achieves a factor 5 approximation. Due to limited space, we describe the algorithm and its analysis in the appendix.

▶ **Lemma 17** ($\star$)**.** *We can find a 5-approximation for Optimal Fault-Tolerant Committee in time $\mathcal{O}(mnk)$.*

All of the above approximations hold not just for d-dimensional Euclidean space, for any fixed d, but also for any metric space.

3.3.2 A bicriterion EPTAS

Finally, we design a bicriterion FPT approximation scheme with running time $f(\varepsilon) \cdot n^{\mathcal{O}(1)}$, which finds a size-$k$ committee whose fault-tolerance score for at least a $(1 - \varepsilon)$ fraction of the voters is within a factor of $(1 + \varepsilon)$ of the optimum. Formally, we say a committee T is (r, ρ)-*good* if there exists a subset $V' \subseteq V$ of size at least ρn such that the f-tolerant score of T with respect to only the voters in V' is at most r. Then our approximation scheme can output a size-k committee which is $((1 + \varepsilon)\sigma^*, 1 - \varepsilon)$-good. The core of our approximation scheme is the following (approximation) decision algorithm. The decision algorithm takes the problem instance and an additional number $r > 0$ as input. The output of the algorithm has two possibilities: it either **(i)** returns YES and gives a size-k committee that is $((1 + \varepsilon)r, 1 - \varepsilon)$-good or **(ii)** simply returns NO. Importantly, the algorithm is guaranteed to give output **(i)** as long as $r \geq \sigma^*$. Note that this decision algorithm directly gives us the desired approximation scheme. Indeed, we can apply it with $r = d(v, c)$ for all $v \in V$ and $c \in C$. Let r^* be the smallest r that makes the algorithm give output **(i)**. The size-k

committee T^* obtained when applying the algorithm with r^* is $((1+\varepsilon)r^*, 1-\varepsilon)$-good. We have $r^* \leq \sigma^*$ because the algorithm must be applied with $r = \sigma^*$ at some point and it is guaranteed to give output **(i)** at that time. Thus, T^* is $((1+\varepsilon)\sigma^*, 1-\varepsilon)$-good, as desired.

For simplicity of exposition, we describe our decision algorithm in two dimensions. By scaling, we may assume that the given number is $r = 1$. To solve the decision problem, our algorithm uses the shifting technique [18]. Let h be an integer parameter to be determined later. For a pair of integers $i, j \in \mathbb{Z}$, let $\Box_{i,j}$ denote the $h \times h$ square $[i, i+h] \times [j, j+h]$. A square $\Box_{i,j}$ is *nonempty* if it contains at least one voter or candidate. We first compute the index set $\widetilde{I} = \{(i,j) : \Box_{i,j} \text{ is nonempty}\}$. This can be easily done in time $\mathcal{O}((n+m)h^2)$.

Consider a pair $(x, y) \in \{0, \ldots, h-1\}^2$. Let $L_{x,y}$ be the set of all integer pairs (i, j) such that $i \pmod h \equiv x$ and $j \pmod h \equiv y$. We write $\widetilde{I}_{x,y} = \widetilde{I} \cap L_{x,y}$. For a voter $v \in V$ and a square $\Box_{i,j}$, we say v is a *boundary voter* for $\Box_{i,j}$ if $v \notin [i+2, i+h-2] \times [j+2, j+h-2]$. Furthermore, we say v *conflicts* with (x, y) if v is a boundary voter in $\Box_{i,j}$ for some $(i, j) \in \widetilde{I}_{x,y}$.

▶ **Lemma 18** ($\star$)**.** *There exists a pair $(x, y) \in \{0, \ldots, h-1\}^2$ such that at most $\frac{4h-4}{h^2} \cdot |V|$ voters conflict with (x, y).*

We fix a pair $(x, y) \in \{0, \ldots, h-1\}^2$ that conflicts with the minimum number of voters. For $(i, j) \in \widetilde{I}_{x,y}$, we define the set of (non-boundary) voters $V_{i,j} = \{v \in \Box_{i,j} : v \in [i+2, i+h-2] \times [j+2, j+h-2]\}$, and the set of candidates $C_{i,j} = \{c \in C : c \in \Box_{i,j}\}$. Note that for $(i, j) \in \widetilde{I}_{x,y}$, the $C_{i,j}$'s are disjoint and form a partition of C. Next, we show an important lemma which allows our algorithm to divide our problem into smaller subproblems, solve them individually, and combine the solutions to solve the overall problem.

▶ **Lemma 19** ($\star$)**.** *Let $V_1, V_2, \ldots, V_s$ be subsets of V and let $T_1, T_2, \ldots, T_s$ be pairwise disjoint subsets of C such that T_i is a fault-tolerant committee for V_i with $\sigma_f(T_i) = \sigma$. Then, $T = \bigcup_{i=1}^{s} T_i$ is a fault-tolerant committee of $\bigcup_{i=1}^{s} V_i$ with $\sigma_f(T) = \sigma$.*

Consider a pair $(i, j) \in \widetilde{I}_{x,y}$. Let $\overline{T}_{i,j}$ be a smallest fault-tolerant committee for $V_{i,j}$ with $\sigma_f(\overline{T}_{i,j}) \leq 1$. We observe that any inclusion-minimal fault-tolerant committee $T_{i,j}$ for $V_{i,j}$ satisfies $T_{i,j} \subseteq C_{i,j}$. This is because any candidate outside $C_{i,j}$ has distance more than $1 + 6/h$ to any voter in $V_{i,j}$ (for a large enough value of h). In the next section we will show how to compute a fault-tolerant committee $T_{i,j} \subseteq C_{i,j}$ for $V_{i,j}$ such that $|T_{i,j}| \leq |\overline{T}_{i,j}|$ and $\sigma_f(T_{i,j}) \leq 1 + 6/h$ in $h^{O(h^4)} n^{O(1)}$ time. Assuming we can compute the above-mentioned committee $T_{i,j}$, our overall algorithm is as follows:

1. Fix a pair $(x, y) \in \{0, \ldots, h-1\}^2$ conflicting with the minimum number of voters, and set h to be the smallest integer such that $(4h-4)/h^2 \leq \varepsilon$ and $6/h \leq \varepsilon$.
2. For each pair $(i, j) \in \widetilde{I}_{x,y}$, compute $T_{i,j} \subseteq C_{i,j}$.
3. Let $T = \bigcup_{(i,j) \in \widetilde{I}_{x,y}} T_{i,j}$. If $|T| \leq k$, return YES (along with T); otherwise, return NO.

Let $V' = \bigcup_{(i,j) \in \widetilde{I}_{x,y}} V_{i,j}$. Since the $C_{i,j}$'s are disjoint, using Lemma 19, we conclude that T is a fault-tolerant committee for V'. Furthermore, from our choice of (x, y), we have $|V'| \geq (1-\varepsilon)n$. It is easy to show that the f-tolerant score of T with respect to the voters in V' is at most $1+\varepsilon$, and in addition, if $\sigma^* \geq 1$, we have $|T| \leq k$; we give a formal argument in the full version of our paper. This proves correctness of our decision algorithm. The overall algorithm takes $(1/\varepsilon)^{O(1/\varepsilon^4)}(m+n)^{O(1)}$ time. We note that the algorithm can be directly generalized to the d-dimensional case with running time $(1/\varepsilon)^{O(1/\varepsilon^{2d})}(m+n)^{O(1)}$. Therefore, we have the following result.

▶ **Theorem 20.** *Given a d-dimensional Fault-Tolerant Committee Selection instance, we can compute a size-k committee T such that the f-tolerant score of T with respect to at least $(1-\varepsilon)n$ voters is at most $(1+\varepsilon)\sigma^*$, where σ^* is the optimal f-tolerant score of a size-k committee (with respect to the entire set V). This algorithm runs in time $(1/\varepsilon)^{O(1/\varepsilon^{2d})}(m+n)^{O(1)}$.*

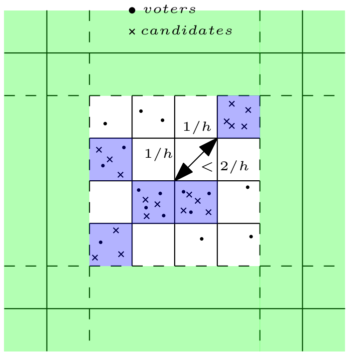

Figure 2 The figure shows a cell in the shifted grid. The solid lines around the sides are the grid lines (and the region inside them is a cell). The shaded (green) region is the boundary region. Inside the boundary region, we divide the cell into $1/h \times 1/h$ smaller cells. The distance between any two points in a smaller cell is $< 2/h$. All candidates in smaller cells are identical (i.e., candidates in blue regions). In this example, since only five cells are nonempty, we have at most 2^5 distinct failing sets.

Algorithm to Compute $T_{i,j}$

We now present the most challenging piece of our algorithm: the computation of the $T_{i,j}$'s. Consider a box $\square_{i,j}$. Suppose there exists a fault-tolerant committee $T \subseteq C$ for $V_{i,j}$ with $\sigma_f(T) \leq 1$. Our task is to compute a fault-tolerant committee $T_{i,j} \subseteq C$ for $V_{i,j}$ such that $|T_{i,j}| \leq |T|$ and $\sigma_f(T_{i,j}) \leq 1 + 6/h$.

We divide $\square_{i,j}$ into h^4 smaller cells each with size $\frac{1}{h} \times \frac{1}{h}$, and we denote the set of these cells by $L = \{l_1, \ldots, l_{h^4}\}$. (See Figure 2.) Our algorithm is based on two key observations:
(i) A committee with a candidate in every nonempty cell has f-tolerant score within a difference of at most $2/h$ from the optimum score. Since the number of cells is h^4, this implies that the size of a smallest approximately optimal committee is bounded by h^4 (formally shown in Lemma 21).
(ii) All candidates in a cell can be treated as identical, causing only a loss of $2/h$ in the score. This implies that for any $T_{i,j}$, to approximately compute the f-tolerant score of $T_{i,j}$, we *only* need to consider the failing sets where either all or none of the candidates in a cell fail. Note that the number of such failing sets is at most $2^{O(h^4)}$ (formally shown in Lemma 22).

Using these two observations, at a high level, our algorithm goes through all committees of size at most h^4 (there are $h^{O(h^4)}$ of these as we can assume that each cell has at most h^4 candidates), approximately computes the f-tolerant score of each of these committees in time $2^{O(h^4)}$, and returns the smallest one with the desired score.

▶ **Lemma 21** ($\star$). *Let $T, T^* \subseteq C$ be fault-tolerant committees for $V_{i,j}$. If $|T^* \cap l_a| = 1$ for all $a \in [h^4]$ such that $C \cap l_a \neq \emptyset$, then $\sigma_f(T^*) - \sigma_f(T) \leq 2/h$.*

Based on the above observation, we solve the problem as follows. We enumerate all maps $\chi \colon L \to \{0, 1, \ldots, h^4\}$ where $\chi(l_a)$ is the number of candidates from l_a in the committee. The total number of such maps is $h^{O(h^4)}$. For each feasible map, i.e., χ satisfying $\chi(l_a) \leq |C \cap l_a|$ for all $a \in [h^4]$, we construct a fault-tolerant committee T^*_χ for $V_{i,j}$ by picking (arbitrarily) $\chi(l_a)$ candidates in $C \cap l_a$ for all $a \in [h^4]$ and including them in T^*_χ. For each constructed T^*_χ, we compute a number $\widetilde{\sigma_f}(T^*_\chi)$ that approximates $\sigma_f(T^*_\chi)$ using the following lemma.

▶ **Lemma 22** ($\star$). *Given T^*_χ, one can compute a number $\widetilde{\sigma_f}(T^*_\chi)$ in $2^{O(h^4)} n^{O(1)}$ time such that $|\widetilde{\sigma_f}(T^*_\chi) - \sigma_f(T^*_\chi)| \leq 2/h$.*

Finally, we let $T_{i,j}$ be the smallest among all committees T_χ^* satisfying $\widetilde{\sigma_f}(T_\chi^*) \leq 1 + 4/h$, and we return it as our solution. The running time of our algorithm is clearly $h^{O(h^4)} n^{O(1)}$. The following lemma shows that our algorithm is correct.

▶ **Lemma 23** ($\star$). *We have $\sigma_f(T_{i,j}) \leq 1+6/h$. Furthermore, $|T_{i,j}| \leq |T|$ for any fault-tolerant committee T for $V_{i,j}$ with $\sigma_f(T) \leq 1$.*

References

1 Kenneth Arrow. *Advances in the spatial theory of voting.* Cambridge University Press, 1990.

2 Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.

3 Robert Bredereck, Piotr Faliszewski, Andrzej Kaczmarczyk, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. Robustness among multiwinner voting rules. *Artificial Intelligence*, 290:103403, 2021.

4 Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier. On coalitional manipulation for multiwinner elections: Shortlisting. *Autonomous Agents and Multi-Agent Systems*, 35(2):38, 2021.

5 John R Chamberlin and Paul N Courant. Representative deliberations and representative decisions: Proportional representation and the borda rule. *American Political Science Review*, 77(3):718–733, 1983.

6 Shiva Chaudhuri, Naveen Garg, and Ramamoorthi Ravi. The p-neighbor k-center problem. *Information Processing Letters*, 65(3):131–134, 1998.

7 Otto A Davis, Melvin J Hinich, and Peter C Ordeshook. An expository development of a mathematical model of the electoral process. *American political science review*, 64(2):426–448, 1970.

8 Edith Elkind, Piotr Faliszewski, Jean-François Laslier, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. What do multiwinner voting rules do? an experiment over the two-dimensional euclidean domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31 of *AAAI'17*, pages 494–501, 2017.

9 Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of multiwinner voting rules. *Social Choice and Welfare*, 48(3):599–632, 2017.

10 Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Llull and copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res.*, 35:275–341, 2009. `doi:10.1613/jair.2697`.

11 Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner voting: A new challenge for social choice theory. *Trends in computational social choice*, 74(2017):27–47, 2017.

12 Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Committee scoring rules: Axiomatic characterization and hierarchy. *ACM Transactions on Economics and Computation (TEAC)*, 7(1):1–39, 2019.

13 Eric J Friedman, Vasilis Gkatzelis, Christos-Alexandros Psomas, and Scott Shenker. Fair and efficient memory sharing: Confronting free riders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1965–1972, July 2019.

14 Ashish Goel, Anilesh K Krishnaswamy, Sukolsak Sakshuwong, and Tanja Aitamurto. Knapsack voting for participatory budgeting. *ACM Transactions on Economics and Computation (TEAC)*, 7:1–27, 2019.

15 Arnaud Grivet Sébert, Nicolas Maudet, Patrice Perny, and Paolo Viappiani. Preference aggregation in the generalised unavailable candidate model. In *International Conference on Algorithmic Decision Theory*, pages 35–50. Springer, 2021.

16 Mohammadtaghi Hajiaghayi, Wei Hu, Jian Li, Shi Li, and Barna Saha. A constant factor approximation algorithm for fault-tolerant k-median. *ACM Transactions on Algorithms (TALG)*, 12(3):1–19, 2016.

17 Edith Hemaspaandra, Lane A Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5-6):255–285, 2007.
18 Dorit S Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM (JACM)*, pages 130–136, 1985.
19 Samir Khuller, Robert Pless, and Yoram J Sussmann. Fault tolerant k-center problems. *Theoretical Computer Science*, 242(1-2):237–245, 2000.
20 Sven Oliver Krumke. On a generalization of the p-center problem. *Information processing letters*, 56(2):67–71, 1995.
21 Tyler Lu and Craig Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Twenty-Second International Joint Conference on Artificial Intelligence*, pages 280–286, 2011.
22 Tyler Lu and Craig E Boutilier. The unavailable candidate model: a decision-theoretic view of social choice. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 263–274, 2010.
23 Sujaya Maiyya, Victor Zakhary, Divyakant Agrawal, and Amr El Abbadi. Database and distributed computing fundamentals for scalable, fault-tolerant, and consistent maintenance of blockchains. *Proceedings of the VLDB Endowment*, 11(12), 2018.
24 Neeldhara Misra and Chinmay Sonar. Robustness radius for chamberlin-courant on restricted domains. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 341–353. Springer, 2019.
25 Kamesh Munagala, Zeyu Shen, and Kangning Wang. Optimal algorithms for multiwinner elections and the chamberlin-courant rule. *EC '21: Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 697–717, 2021.
26 Piotr Skowron, Piotr Faliszewski, and Jérôme Lang. Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artificial Intelligence*, 241:191–216, 2016.
27 Piotr Skowron, Piotr Faliszewski, and Arkadii Slinko. Achieving fully proportional representation: Approximability results. *Artificial Intelligence*, 222:67–103, 2015.
28 Chinmay Sonar, Palash Dey, and Neeldhara Misra. On the complexity of winner verification and candidate winner for multiwinner voting rules. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 89–95, 2021.
29 Chinmay Sonar, Subhash Suri, and Jie Xue. Multiwinner elections under minimax chamberlin-courant rule in euclidean space. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 475–481. International Joint Conferences on Artificial Intelligence Organization, July 2022.
30 Chaitanya Swamy and David B Shmoys. Fault-tolerant facility location. *ACM Transactions on Algorithms (TALG)*, 4(4):1–27, 2008.