Strategyproof facility location mechanisms on discrete trees

Alina Filimonov1 · Reshef Meir1

Accepted: 17 November 2022 / Published online: 8 December 2022
© Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
We address the problem of strategyproof (SP) facility location mechanisms on discrete trees. Our main result is a full characterization of onto and SP mechanisms. In particular, we prove that when a single agent significantly affects the outcome, the trajectory of the facility is almost contained in the trajectory of the agent, and both move in the same direction along the common edges. We show tight relations of our characterization to previous results on discrete lines and on continuous trees. We then derive further implications of the main result for infinite discrete lines.

Keywords Facility location · Mechanism design · Strategyproof

Mathematics Subject Classification MSC 91

1 Introduction
In facility location problems, a central planner has to determine the location of a public facility that needs to serve a set of agents. Once the facility is located, each agent incurs some cost. Importantly, in non-cooperative settings, agents may have an incentive to misreport their locations to decrease their costs. One key objective of the planner that received much attention in the multiagent systems literature is to design a mechanism that incentivizes agents to report their true locations, i.e., mechanisms that are strategyproof (SP).

The fundamental characterization result for strategyproof facility location was given by Moulin [27], who characterized the class of deterministic SP mechanisms on the real line when the preferences of the agents are single-peaked as “generalized median voter schemes” (g.m.v.s.’s). An agent with single-peaked preferences on a line prefers a closer location to her peak over a distant location on the same side of her peak.

Border and Jordan [5] proved that the characterization also applies for cases where the preferences are “quadratic” (i.e., symmetric and single-peaked)—the more common model...
in facility location used in this work as well. An agent with quadratic preferences on a line prefers a closer location to her peak over a distant location.

As quadratic preferences are a special case of single-peaked preferences, the class of SP mechanisms for quadratic preferences may be larger. This is indeed the case e.g. for mechanisms on discrete lines [13], but not on continuous lines [5]. Schummer and Vohra [32] generalized the result of Border and Jordan to prove that an SP mechanism on a continuous tree, under quadratic preferences, is a consistent collection of g.m.v.s.

As hinted above, the trigger for the current work is the observation by Dokow et al. [13] that results on continuous graphs do not necessarily carry over to discrete graphs. In particular, while g.m.v.s entails that the trajectory of the facility is contained in the trajectory of the moving agent on a line (we later observe this also applies for continuous trees), Dokow et al. show it is only “almost contained” when the line is made of discrete vertices. Similarly, while a strategyproof onto mechanism on a continuous circle must be dictatorial [32], it is only “almost dictatorial” when the circle is discrete [13]. These extensions may be subtle, but they help us understand what in the characterization is inherent to the topology of the graph.

Given these previous results, a natural question is whether a similar extension to the results of Schummer and Vohra can be applied to discrete trees.

As we will later show, a naïve extension of the properties defined in [13] fails. We therefore formulate similar properties to characterize the valid moves of the facility under onto, SP mechanisms on discrete trees. In particular, we provide a definition of “almost Pareto efficient” mechanisms that might be of independent interest.

Recent research by Peters et al. [29] provides a different characterization of randomized strategyproof voting mechanisms on trees and on other graphs (which, of course, include deterministic mechanisms), under general single-peaked preferences. However, the class of strategyproof mechanisms on discrete trees under single-peaked preferences is not equivalent to the one under quadratic preferences and therefore the characterization in [29] does not apply for our study.

1.1 Structure and contribution

After some preliminary notation in Sect. 3, we provide an alternative characterization of onto, SP mechanisms on continuous trees in Sect. 4, based on the work of Schummer and Vohra [32].

We then present our main result in Sect. 5— a full characterization of onto, SP mechanisms on discrete trees. We show that our characterization indeed generalizes the one given by Dokow et al. [13] for mechanisms on lines. In contrast to the work of Dokow et al., our proof also works for infinite trees with bounded degree and thus for infinite lines.

In Sect. 6, we derive a characterization of SP and shift-invariant mechanisms on infinite discrete lines.

A preliminary version of this work was published in the proceedings of AAMAS 2021 [17].

2 Related work

Following the early work of Black [4] and Moulin [27] on strategyproofness under single-peaked preferences, several researchers have developed characterizations of deterministic and probabilistic strategyproof facility location mechanisms in various scenarios.
Schummer and Vohra [32], beyond their work on trees, showed that any onto, SP mechanism on the continuous cycle must be a dictatorship and that any SP mechanism on a graph has a dictator in a subdomain. Their work was extended to discrete cycles in [13].

Additional variations of the problem include the multiple facility problem [7, 18, 21, 25, 33], the obnoxious facility location problem [9, 10, 23, 33], the heterogeneous facility location problem [2, 8, 12, 14, 24], and the activity scheduling problem [37]. A work by Feldman et al. [15] considered mechanisms in which the agents reported their preferences by voting for their most preferred location or ranking the candidate locations, instead of reporting their locations. A study by Roy and Storcken [31] characterizes the domains that admit non-dictatorial, unanimous, and strategyproof mechanisms.

Todo et al. [34] extended Moulin’s work for characterizing the class of false-name-proof mechanisms on the continuous line. Their work was extended to discrete structures in [28, 35]. The motivation for designing such mechanisms is to prevent agents from submitting multiple reports under different identities, e.g., in internet polls by creating different e-mail addresses. An example of an SP mechanism that is not false-name-proof is the median mechanism on the line, which selects the median of all reported locations. This mechanism is SP. However, if an agent submits multiple “true” reports, the median location will move closer to her location. A later work of Wada et al. [36] on variable and dynamic populations characterizes mechanisms that incentivize the agents to participate in the reporting process. Other works that deal with dynamic settings are provided in [11, 20]. They study multi-stage facility reallocation problems on the real line, where a facility is being moved based on the locations reports of a fixed number of agents.

Finally, concrete cost functions also allow us to measure the social cost (e.g., as the sum or max of agents’ costs). The research line of approximate mechanism design without money builds on characterizations such as those mentioned above, and seeks the mechanisms that minimize the social cost among all strategyproof mechanisms. Incidentally, the iconic domain for this line of work, as reflected in the fundamental paper of Procaccia and Tennenholtz [30], is the facility location problem. Their work was extended to the domain of continuous graphs by Alon et al. [1]. Additional research on approximation bounds for the facility location problem on lines and general graphs is presented in [7, 15, 19, 22, 26].

In the context of onto and strategyproof mechanisms on trees (either continuous or discrete), the question of minimizing the utilitarian social cost—defined as the sum of agents’ costs—is moot since there is a simple mechanism for trees (the median voter) that is both strategyproof and socially optimal.

More details on prior results can be found in a recent survey by Chan et al. [6].

3 Preliminaries

Consider an unweighted, undirected, bounded degree discrete tree \( T = (V, E) \) with a set \( V \) of vertices and a set \( E \) of edges. The sets \( V \) and \( E \) can be infinite. For any two vertices \( v_1, v_2 \in V \), \( d(v_1, v_2) \) is the length of the unique path between \( v_1 \) and \( v_2 \). The distance between two sets of vertices \( A \subseteq V \) and \( B \subseteq V \) is the length of the shortest path between any pair of vertices \( a, b \), where \( a \in A \) and \( b \in B \). We sometimes refer to a discrete tree as the set of its vertices. Consequently, the distance between two subtrees of a tree is the distance between the corresponding sets of vertices. For \( u, w \in V \) with \( u \neq w \), \([u, w]\) is the sequence of vertices \( v_0, \ldots, v_k \) on the unique path of length \( k \) between \( u \) and \( w \) s.t. \( v_0 = u, v_k = w \). We denote by \((u, w)\) the sequence \( v_1, \ldots, v_k \), and by \((u, w)\) the sequence \( v_1, \ldots, v_{k-1} \), where \( \{v_i, v_{i+1}\} \in E \)
for all \( i = 0, \ldots, k - 1 \). We say that \( e = \{u, w\} \in [a, b] \) if \( [u, w] \subseteq [a, b] \). A line-graph is a tree with a maximum degree of 2.

Let \( N = \{1, \ldots, n\} \) be the set of agents, and \( a = (a_1, \ldots, a_n) \in V^n \) be a location profile, where \( a_i \in V \) denotes the location of agent \( i \) for every \( i \in N \). The location profile of all agents excluding agent \( i \) is denoted by \( a_{-i} \in V^{n-1} \). A deterministic facility location mechanism on a discrete tree is a function \( f : V^n \rightarrow V \), that maps a given profile of the agents’ locations to a single facility location.

For any two vertices \( u, w \in V \), the notation \( u >_i w \) indicates that agent \( i \) prefers vertex \( u \) over vertex \( w \). The notation \( u \geq_i w \) indicates that agent \( i \) prefers vertex \( u \) over vertex \( w \), or is indifferent between the two.

In this research we assume that the agents’ costs are related to their distance from the chosen location. We refer to such cost functions as “quadratic” costs. The class of preferences induced by quadratic costs is single-peaked and symmetric. Formally, for every agent \( i \in N \) located at \( a_i \in V \),

\[
\forall u, w \in V : u >_i w \iff d(a_i, u) < d(a_i, w).
\]

Next, we give the standard definitions of mechanism properties:

**Definition 1** (Strategyproof) A mechanism \( f \) is **strategyproof** (SP) if no agent can benefit from reporting a false location. Formally, \( f \) is strategyproof if for every agent \( i \in N \), every profile \( a \in V^n \) and every alternative location \( a' \in V \), it holds that

\[
d(a_i, f(a, a_{-i})) \leq d(a_i, f(a', a_{-i})).
\]

**Definition 2** (Onto) A mechanism \( f \) is **onto**, if for every location \( x \in V \) there is a location profile \( a \in V^n \) s.t. \( f(a) = x \).

**Definition 3** (Unanimous) A mechanism \( f \) is **unanimous** if for every location \( x \in V \), \( f(x, \ldots, x) = x \).

Clearly, every unanimous mechanism is onto. The following lemma provides a necessary condition for an onto, SP mechanism on any domain.

**Lemma 1** (Barbera and Peleg [3]) **Every mechanism that is both onto and SP, is unanimous.**

The definitions above also apply for continuous trees. A finite continuous tree \( G = (V, E) \) is a connected, acyclic collection of curves of finite length. \( E \) is the set of curves and \( V \) is the set of the extremities and intersections of the curves [32]. Let \( L \subseteq V \) denote the set of extremities only. For all \( p_1, p_2 \in G \), \( d(p_1, p_2) \) is the length of the unique path between \( p_1 \) and \( p_2 \). We denote by \( (p_1, p_2) \) the open segment between \( p_1 \) and \( p_2 \), and by \( [p_1, p_2] \) the closed segment between the two points. For any point \( p \) and a set \( S \subseteq G \) (which may itself be a segment), the notation \([p, S]\) stands for the segment \([p, s]\) where \( s = \arg\min_{s \in S} d(p, s)\). For a mechanism on a continuous tree, the agents and the facility can be placed on arbitrary points on the edges. A mechanism on a continuous tree is therefore a function \( f : G^n \rightarrow G \). The definitions are illustrated in Fig. 1.
4 SP mechanisms on continuous trees

Schummer and Vohra [32] provided a characterization of onto, SP mechanisms on continuous trees. They showed that when the agents’ preferences are quadratic, every onto, SP mechanism on the continuous tree is based on a set of generalized median voter schemes, defined in [27], satisfying a consistency condition.

In this section, we provide an alternative characterization of onto, SP mechanisms on continuous trees, that relies on previous works [5, 32]. We then show how it relates to the class of “boomerang” mechanisms defined in [16].

Previous Results on a Continuous Line

The following definition given in [32] describes onto and SP mechanisms on a continuous line. It is similar to the one introduced by Moulin [27] for single-peaked preferences and confirmed for quadratic preferences by Border and Jordan [5].

Definition 4 (Generalized Median Voter Scheme [32]) A function $g_{xy}$ is called a generalized median voter scheme (g.m.v.s.) on $[x, y]$ if there exist $\frac{2}{\sqrt{\varnothing}}$ points in $[x, y]$, $\{\alpha_{xy}S\}$ such that:

1. $S \subseteq R$ implies that $d(\alpha_{xy}^{-}, x) \leq d(\alpha_{xy}^{+}, x)$.
2. $\alpha_{xy}^{-} = x$ and $\alpha_{xy}^{+} = y$.
3. For all $a \in [x, y]^n$, $g_{xy}(a)$ is the unique point satisfying
   $$d(g_{xy}(a), x) = \max_{S \subseteq \varnothing} \min\{d(a_{i}, x)\}_{i \in S}, d(\alpha_{xy}^{+}, x)\}
   $$

The following is a key property of the class of g.m.v.s.’s defined in [5]. It implies that when an agent moves without crossing the mechanism outcome, the facility does not move.

Definition 5 (Uncompromising [5]) A mechanism $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is called uncompromising if for every $a \in \mathbb{R}^n$, $i \in \mathbb{N}$, $a'_i \in \mathbb{R}$, it holds that:

1. $a_i > f(a)$ implies that $f(a_{-i}, a'_i) = f(a)$ for all $a'_i \geq f(a)$
2. $a_i < f(a)$ implies that $f(a_{-i}, a'_i) = f(a)$ for all $a'_i \leq f(a)$

Lemma 2 (Border and Jordan [5]) Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is SP and unanimous. Then $f$ is uncompromising.

Fig. 1 In this tree, $V = \{v_1, \ldots, v_6\}$ is the set of vertices. $\{v_1, v_3, v_5, v_6\} \subseteq V$ is the set of leafs. The agents are located at $a_1$ and $a_2$ (i.e., not on any vertex). The facility is located at point $f$, which is closer to $a_2$. 
As shown in [32], this also applies for every mechanism \( f : [x, y]^n \rightarrow [x, y] \) where \([x, y]\) is a finite interval in \( \mathbb{R} \). Note that every SP and onto mechanism is unanimous by Lemma 1. Therefore we can rely on uncompromisingness for our characterization of onto and SP mechanisms.

*Previous Results on a Continuous Tree*

**Definition 6** (Graph Restriction [32]) For any subgraph \( G' \subset G \), the **graph restriction** of \( f : G^n \rightarrow G \) to \( G' \) is the function \( f|_{G'} : G'^n \rightarrow G \) s.t. for all profiles \( a \in G'^n \), \( f|_{G'}(a) = f(a) \).

By [32], if mechanism \( f \) is SP and onto, then for every \( a \in [x, y]^n \), \( f|_{xy}(a) \in [x, y] \).

The following property characterizes onto and SP mechanisms on continuous trees. For all \( x, y \in L \) and \( a_i \in G \), let the unique point in \([x, y]\) closest to \( a_i \) be denoted \( a_i|_{xy} = \text{argmin}_{z \in [x, y]} d(z, a_i) \).

**Definition 7** (Extended Generalized Median Voter Scheme [32]) A mechanism \( f \) is an **extended generalized median voter scheme** (e.m.v.s.) if

1. For all \( w, x, y, z \in G \), \( f|_{xy} \) and \( f|_{wz} \) are consistent g.m.v.s.’s.\(^1\)
2. For all \( a \in G^n \), \( f(a) \) is the unique point \( p \) such that for all \( x, y \in L \), \( p \in [x, y] \) implies \( f|_{xy}(a|_{xy}) = p \).

**Theorem 1** (Schummer and Vohra [32]) For any continuous tree \( G \), a rule \( f \) is SP and onto if and only if it is an e.m.v.s.

### 4.1 Our characterization

We rely on the characterization in [32] to formulate our characterization for SP and onto mechanisms on continuous trees, which is a conceptual step on the way to our main result on discrete trees. The following properties limit the effect of an agent’s move on the outcome of a mechanism on a continuous tree.

**Definition 8** (Tree Monotone) A mechanism \( f \) on the continuous tree is **tree monotone** (TMON) if for every profile \( a \in G^n \), every agent \( i \in N \), every location \( a_i \) and every segment \([x, y]\) s.t. \([x, y] \subseteq [a_i, a_i'] \cap [f(a), f(a_{-i}, a_i')] \), it holds that \( d(a_i, x) < d(a_i, y) \Leftrightarrow d(f(a), x) < d(f(a), y) \)

Intuitively, TMON means that the facility moves in the same direction as the moving agent (if it crosses the agent’s path at all).

**Definition 9** (Trajectory Contained) A mechanism \( f \) on the continuous tree is **trajectory contained** (TC) if for every profile \( a \in G^n \), every agent \( i \in N \) and every location \( a_i' \), it either holds that \([f(a), f(a_{-i}, a_i')] \subseteq [a_i, a_i'] \), or \( f(a) = f(a_{-i}, a_i') \).

---

\(^1\) Intuitively, consistency means that projecting on different pairs of vertices should not yield contradicting outcomes. We omit the exact definition since it is not relevant for our purpose.
In words, either the trajectory of the outcome is contained in the trajectory of the agent, or the facility does not move at all. In Fig. 1, when agent 1 moves from \(a_1\) to \(a'_1\), the facility moves from \(f\) to \(f'\). This violates TC since \([v_4, f'] \not\subseteq [a_1, a'_1]\). This also violates TMON since the facility and the agent move in opposite directions in the segment \([f, v_4]\).

**Lemma 3** Every onto and SP mechanism \(f\) on the continuous tree is TC.

**Proof** Consider an onto, SP mechanism \(f : G^n \rightarrow G\) on a continuous tree. Assume by contradiction that there exists an agent \(i\) and two profiles \(a, a' = (a'_1, a_{-1})\) s.t. \(f(a) \neq f(a')\) and w.l.o.g., that \(f(a) \not\in [a, a'_1]\). By Theorem 1, \(f\) is an e.m.v.s. and therefore it is a collection of g.m.v.s’s. Let \(g^{xy}\) denote the g.m.v.s. on \([x, y]\), where \(x, y \in L\) and \([f(a), f(a')] \subseteq [x, y]\). Note that \(g^{xy} = f|_{xy}\). By Lemma 2, \(g^{xy}\) is uncompromising. From the second property of the e.m.v.s., it holds that

\[
g^{xy}(a|_{xy}) = f(a) \quad \text{and} \quad g^{xy}(a'|_{xy}) = f(a')
\]

We divide into two cases, according to the locations \(a|_{xy}, a'|_{xy}\). Note that for any other agent \(j \neq i\), \(a'_j = a_j\) and thus \(a'_j|_{xy} = a_j|_{xy}\).

1. \([a_i, a|_{xy}] \cap [a'_i, a'|_{xy}] > 0\): In this case, \(a|_{xy} = a'|_{xy}\) and therefore \(g^{xy}(a|_{xy}) = g^{xy}(a'|_{xy})\) contradicting the assumption that \(f(a) \neq f(a')\).

2. \([a_i, a|_{xy}] \cap [a'_i, a'|_{xy}] = 0\): In this case, the path between \(a_i\) and \(a'_i\) must intersect the segment \([x, y]\) and therefore, \([x, y] \cap [a_i, a'_i] = [a|_{xy}, a'|_{xy}]\). By our initial assumption \(f(a) = g^{xy}(a|_{xy}) \not\in [a, a'_i]\), thus \(f(a) \not\in [a|_{xy}, a'_i|_{xy}]\), contradicting the uncompromisingness of \(g^{xy}\).

**Lemma 4** Every TC mechanism on the continuous tree is TMON.

**Proof** Consider a TC mechanism \(f : G^n \rightarrow G\) on a continuous tree. Assume by contradiction that there exists an agent \(i\) and two profiles \(a, a' = (a'_1, a_{-1})\) s.t. \(f(a), f(a') \in [a, a'_i]\) and \(d(a_i, f(a)) > d(a_i, f(a'))\). Let \(x\) denote \(f(a)\) and \(x'\) denote \(f(a')\). Consider a point \(y \in (x, x')\). On the one hand, \(f(y) = x\), otherwise the pair of profiles \(a, (y, a_{-1})\) violates TC. On the other hand, \(f(y) = x'\), otherwise the pair of profiles \(a', (y, a_{-1})\) violates TC. Thus a contradiction is reached.

As we later show, this is not true for mechanisms on discrete trees. Below is an example of a mechanism that demonstrates that not every TMON mechanism is TC. Consider the following single-agent mechanism \(g\) on the tree in Fig. 1.

\[
g(v \in G) = \begin{cases} 
    v_5 & v \neq v_4 \\
    v_4 & v = v_4
\end{cases}
\]

The possible deviations of the agent can be divided into the following cases:

1. \(v \Leftrightarrow v_4\) for \(v \in (v_4, v_5)\): The facility move satisfies TMON and violates TC.
2. \( v \leftrightarrow v_4 \) for \( v \notin \{v_4, v_5\} \): This move violates TC and satisfies TMON, since the intersection of the trajectories is the point \( v_4 \).

3. \( v_4 \leftrightarrow v_5 \): The facility moves between \( v_4 \) and \( v_5 \) in the same direction as the agent. Both TC and TMON are satisfied.

4. \( v \leftrightarrow u \) for \( v, u \neq v_4 \): In this case \( g(u) = g(v) \). Both TC and TMON are satisfied.

The proof of the following lemma is omitted since it is straightforward from the definition of strategyproofness.

**Corollary 1** Every TC mechanism on the continuous tree is SP.

Since a TC mechanism is also TMON, an agent cannot benefit from misreporting her location.

**Theorem 2** An onto mechanism on the continuous tree is SP if and only if it is TC.

The proof follows from Lemma 3 and Corollary 1.

### 4.2 Boomerang mechanisms

Feldman and Wilf [16] proposed a family of SP location mechanisms on continuous trees, the “parameterized boomerang mechanisms” (PB). This family generalizes a significant number of efficient mechanisms that are SP under quadratic preferences (e.g., the median mechanism, the random dictator mechanism). A deterministic PB mechanism is referred to as a “boomerang mechanism”.

**Definition 10** (Boomerang Mechanism [16]) A deterministic mechanism \( f \) is said to be a boomerang mechanism if for every location profile \( a \), agent \( i \), and point \( a_i' \),

\[
d(f(a'), a_i) - d(f(a), a_i) = d(f(a'), f(a)),
\]

where \( a' = (a_i', a_{-i}) \).

In this subsection we show that the class of boomerang mechanisms is equivalent to the class of TC mechanisms.

**Lemma 5** A mechanism on the continuous tree is a boomerang mechanism if and only if it is TC.

**Proof** First we show that every boomerang mechanism is TC. Let \( f \) be a boomerang mechanism on a continuous tree. Let \( a, a' = (a_{-i}, a'_i) \in G^n \) be a pair of profiles that differ only by the location of a single agent \( i \in N \). Let \( x \) denote \( f(a) \) and \( x' \) denote \( f(a') \). Let \( y, y' \in [x, x'] \) denote the closest points to \( a_i \) and \( a_i' \) respectively. Formally:

\[
y = \arg\min_{z \in [x, x']} d(z, a_i); \quad y' = \arg\min_{z \in [x, x']} d(z, a'_i).
\]

It holds that

\[
d(x, a_i) = d(a_i, y) + d(y, x) \quad \text{and} \quad d(x', a_i) = d(a_i, y') + d(y', x')
\]

and therefore

\[
\]
Since \( y \in [x, x'] \) and \( G \) is a tree, it holds that

\[
d(x, x') = d(x, y) + d(y, x')
\]  

(3)

From the definition of boomerang mechanisms and Eqs. (2), (3) above,

\[
cost(x', a_i) - cost(x, a_i) = d(y, x') - d(y, x) = d(x, x') = d(y, x') + d(y, x)
\]  

(4)

Therefore, \( d(y, x) = 0 \) and symmetrically, \( d(y', x') = 0 \), which implies that \( f \) is TC.

Clearly, every TC mechanism is a boomerang mechanism. This can be obtained by a similar calculation to the above. \(\square\)

In this chapter we have presented several properties and characterizations of SP mechanisms on continuous trees. Having different characterizations allows different mechanism representations and developing further extensions. In addition, it helps to understand certain properties of SP mechanisms better. For example, Theorem 2 added to the characterization in [32] by showing how an agent’s deviation may affect the location of the facility.

5 SP mechanisms on discrete trees

In this section we provide a complete characterization of onto, SP mechanisms on discrete trees, generalizing the result of Dokow et al. for discrete lines [13]. We then show that when the discrete tree is a line, the characterization of Dokow et al. follows from our results.

Before presenting the main result, we show that a naïve extension of the properties defined for mechanisms on discrete lines in [13] fails for trees. Their result implies that an agent can affect the outcome of the mechanism only in a way in which its trajectory intersects the trajectory of the facility in at least two consecutive points.

The mechanism described in Fig. 2 is an example of an onto, SP mechanism that violates a naïve extension of this property. Agent 1 is located at vertex 0. Agent 2 is initially at 3 and moves to 4. As a result, the facility moves from vertex 1 to 0 without intersecting the segment [3, 4].

**Quadratic vs. Single-Peaked Preferences**

**Definition 11** (Single-Peaked [29]) A preference of an agent \( i \) is single-peaked on a graph \( G \) if there is a spanning tree \( T = (V, E) \) of \( G \) such that for all distinct \( x, y \in V \) with \( a_i \neq y \),

\[
x \in [a_i, y) \Rightarrow x >_i y.
\]

The following example shows that under single-peaked preferences, the mechanism in Fig. 2 is not SP. Assume the preferences of the agents are as follows:

1. Agent 1: \( 0 >_1 2 >_1 3 \geq_1 1 >_1 4 \)
2. Agent 2: \( 3 >_2 4 \geq_2 2 >_2 0 >_2 1 \)

Both agents have single-peaked preferences according to the definition in [29]. In particular, the preference of the first agent is quadratic. The preference of the second agent is not,
since she strictly prefers vertex 0 over vertex 1. If both agents report truthfully, the facility will be located at vertex 1. However, if the second agent reports vertex 4 as her peak, the facility will be located at vertex 0 and the agent will benefit.

We conclude that similar to the case of the line-graph, quadratic preferences on trees allow more SP mechanisms than single-peaked preferences, and therefore the characterization of probabilistic SP mechanisms under single-peaked preferences in [29] does not apply for quadratic preferences.

5.1 Basic mechanism properties

Here we define several new terms which are specific for mechanisms on discrete trees.

Definition 12 (Tree) \( \forall i \in N, \forall a \in V, \forall b \in V \), \( \text{tree}_k(a \rightarrow b) \) is the sub-tree with the root \( r \) s.t. \( r \) is the \( k \)th vertex in the path from \( a \) to \( b \), which includes only \( r \) and vertices which are accessible from \( r \) via the edges that are not in \([a, b]\).

Definition 13 (Tree) \( \forall v \in V, \forall a \in V, \forall b \in V \), \( \text{tree}(a \rightarrow b, v) \) is the sub-tree which includes only \( v \) and vertices which are accessible from \( v \), via the edges that are not in \([a, b]\).

Definition 14 (Depth) \( \forall v \in V, \forall a \in V, \forall b \in V \), \( \text{depth}(a \rightarrow b, v) \) is the distance of vertex \( v \) from \([a, b]\).

We demonstrate the above definitions in Fig. 3: \( \text{tree}(a_1 \rightarrow a_1', v_1) \) contains \( v_1 \) (at depth 0) and another node at depth 1. \( \text{depth}(a_1 \rightarrow a_1', v_2) = 2 \) in \( \text{tree}_0(a_1 \rightarrow a_1') \), which is the sub-tree rooted by \( a_1 \).

Our next definitions are intended to generalize the properties defined in [13]. The next definition is inspired by 1-SI as a necessary condition for onto, SP mechanisms on discrete lines. As we saw in Fig. 2, a naïve extension of this property does not hold for discrete trees. We, therefore, formulate a relaxed definition of \( m \)-SI and later prove that 1-TSI is a necessary condition for onto, SP mechanisms on discrete trees.

Definition 15 (m-tree step independent) A mechanism \( f \) is \( m \)-tree step independent (\( m \)-TSI) if for every \( a \in V^n, i \in N, a_i' \in V \) s.t. \( d([a_i, a_i'], f(a)) > m \), it holds that

\[
\text{tree}(a_i \rightarrow a_i', f(a)) = \text{tree}(a_i \rightarrow a_i', f(a_{-i}, a_i'))
\]

For \( m = 1 \), the definition states that for every \( a \in V^n, i \in N, a_i' \in V \) s.t. \([f(a), f(a_{-i}, a_i')] \cap [a_i, a_i'] \geq 2 \) it holds that
Figure 3 illustrates a violation of the property. Mechanism $g$ violates 1-TSI since \( \text{tree}(a_1 \rightarrow a'_1, g(a)) \neq \text{tree}(a_1 \rightarrow a'_1, g(a')) \) and \( d([a_1, a'_1], g(a)) = 3 \).

**Definition 16** (Depth Balanced) A mechanism $f$ is **depth balanced** (DB) if for every $a \in V^n, i \in N, a'_i \in V$, it holds that

\[
d(f(a), [a_i, a'_i]) \leq 1.
\]

Figure 3 illustrates a violation of the property. Mechanism $g$ violates 1-TSI since \( \text{tree}(a_1 \rightarrow a'_1, g(a)) \neq \text{tree}(a_1 \rightarrow a'_1, g(a')) \) and \( d([a_1, a'_1], g(a)) = 3 \).

**Definition 17** (Tree Pareto Location) Let $\text{Int}(a)$ be the set of interior vertices of the subtree defined by profile $a$:

\[
\text{Int}(a) = \{ v \in V | \exists a_i, a_j \in a \text{ s.t. } v \in (a_i, a_j) \}.
\]

A location $x \in V$ is **tree Pareto** w.r.t. $a$ if $d(x, \text{Int}(a)) \leq 1$ or $x = a_i$ for some $i \in N$.

This definition generalizes the definition in [13] of a Pareto location on the discrete line. Note that it is weaker than the standard definition of Pareto.

**Definition 18** (Tree Pareto Mechanism) A mechanism $f$ is **tree Pareto** (TPAR) if for every profile $a \in V^n, f(a)$ is a tree Pareto location w.r.t. $a$.

Mechanism $f$ in Fig. 3 violates TPAR w.r.t. profile $a'$ since

\[
d(f(a'), \text{Int}(a')) = d(f(a'), [v_1, v_2]) = 2.
\]

The purpose of the next definition is to complete the 1-TSI property of onto, SP mechanisms. A 1-TSI mechanism allows the facility to move even when its original position is...
more than one step away from the deviating agent’s trajectory. Nevertheless, we assume
that such a move would be negligible. That is, an agent cannot significantly affect the facil-
ity location by a deviation that is far from the facility. We prove that if such a movement of
the facility occurs, it can only be to a “sibling” node, i.e. one step up towards the moving
agent trajectory and one step down.

**Definition 19** (Almost Depth Restricted) A mechanism \( f \) is **almost depth restricted**
(ADR) if for every \( a \in V^n, i \in N, a'_i \in V \) s.t. \( f(a) \neq f(a'_i, a_{-i}) \) and

\[
\text{tree}(a_i \rightarrow a'_i, f(a)) = \text{tree}(a_i \rightarrow a'_i, f(a'_i, a_{-i})),
\]

the following holds: Let \( z \in V \) be the unique point s.t.

\[
z := \min_{v \in [f(a), f(a'_i, a_{-i})]} d(v, a_i).
\]

Then,

1. \( d(f(a), z) = d(f(a'_i, a_{-i}), z) \)
2. \( d(f(a), z) = 1 \)

Informally, ADR means that when the facility moves as a result of a single agent’s
deviation, without intersecting the trajectory of the agent in at least two points, the new
outcome has the same parent as the original outcome in the tree induced by the deviation.
That is, either the facility does not move, or it moves to a sibling node. We can think of this
property as “approximate uncompromising” (replacing ‘1’ with ‘0’ in the definition would
yield exact uncompromising).

In Fig. 3, ADR is violated by mechanism \( f \) for the pair of profiles \((a, a_i)\), which differ by
the location of agent 1, since \( d(f(a), z) = 2 \).

**Definition 20** (Almost Trajectory Contained) A mechanism \( f \) is **almost trajectory con-
tained** (ATC) if it is ADR and 1-TSI.

**Definition 21** (Tree Monotone) A mechanism \( f \) on the discrete tree is **tree mono-
tone** (TMON) if for every \( a \in V^n, i \in N, a'_i \in V \) s.t. \( f(a) \in \text{tree}_j(a_i \rightarrow a'_i) \) and
\( f(a_{-i}, a'_i) \in \text{tree}_k(a_i \rightarrow a'_i) \), it holds that \( k \geq j \).

Note that on discrete trees, the class of ATC mechanisms is not contained in the class
of TMON mechanisms. Consider the following single-agent mechanism \( g \) on the tree in
Fig. 2.

\[
g(v \in V) = \begin{cases} 
v & v \in \{0, 1, 4\} \\ 3 & v = 2 \\ 2 & v = 3 \\
\end{cases}
\]

\[\text{This is different from the tree monotonicity property in [29].}\]
This mechanism satisfies ATC but violates TMON for the pair \( a_1 = 2, a_{\bar{1}} = 3 \).

Analogously to the case of continuous trees, the class of TMON mechanisms on discrete trees is not contained in the class of ATC mechanisms on discrete trees. Consider the following single-agent mechanism \( h \) on the tree in Fig. 2.

\[
h(v \in V) = \begin{cases} 
  v & v \in \{0, 1, 4\} \\
  4 & v \in \{2, 3\}
\end{cases}
\]

This mechanism satisfies TMON but violates ATC for the pair \( a_1 = 1, a_{\bar{1}} = 2 \).

We now go on to characterize SP and onto mechanisms on discrete trees.

**Theorem 3** An onto mechanism \( f \) on the discrete tree is SP if and only if it is TMON and ATC.

The intuition for Theorem 3 stems from the characterizations of onto, SP mechanisms on continuous trees and discrete lines. As we have shown in Sect. 4, in the case of onto, SP mechanisms on continuous trees, the trajectory of the facility is contained in the trajectory of the deviating agent, and they move in the same direction. According to [13], in the case of the discrete line, this property is relaxed in the sense that SP onto mechanisms are “almost-TC”, i.e., the facility can move only if it is at most one step away from the trajectory of the deviating agent.

This relaxation is not sufficient to characterize all onto, SP mechanisms on discrete trees, as shown in Fig. 2, therefore the challenge is to define the property of “almost TC” in a way that will capture the whole class of onto, SP mechanisms on discrete trees.

First we present a weak characterization. We then use it to prove the Tree Pareto property and the main property of “almost trajectory containment”, which consists of two properties, ADR and 1-TSI. To prove ADR, we first show that when an agent moves towards the facility, either the facility remains in place (as in the continuous case), or it moves exactly one step towards the agent and one step away. To prove 1-TSI, we first show that when an agent moves to a neighboring vertex on edge \( e \), the facility can intersect \( e \) only if it is at most one step away from \( e \).

**DB and TMON**

**Lemma 6** A pair of profiles violates SP if and only if it violates DB or TMON.

The idea is that if a pair of profiles \( a, a' = (a_{-i}, a'_i) \) violates one of the properties, agent \( i \) will benefit from the move \( a_i \to a'_i \) or the move \( a'_i \to a_i \). This is true for infinite trees as well.

**Proof** “\( \Rightarrow \)” Every pair of profiles \( a, a' = (a_{-i}, a'_i) \) that violates SP, violates DB or TMON. Consider a mechanism \( f \) and a pair of profiles \( a, a' = (a_{-i}, a'_i) \) s.t.

\[
x := f(a); x' := f(a_{-i}, a'_i); \text{ and } d(a_i, x) > d(a_i, x').
\]

For any mechanism \( f \) and profile \( a \) it holds that

\[
d(a_i, x) = d(a_i, \text{tree}(a_i \to a'_i, x)) + \text{depth}(a_i \to a'_i, x)
\]

Assume that \( f \) is TMON. Then it follows that:
\[
\begin{align*}
\text{d}(a_i, x') &= \text{d}(a_i, \text{tree}(a_i \rightarrow a_i', x)) \\
&\quad + \text{d}(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) \\
&\quad + \text{depth}(a_i \rightarrow a_i', x') \quad (6)
\end{align*}
\]

And thus from Eqs. (5), (6) above,
\[
\begin{align*}
\text{d}(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) + \text{depth}(a_i \rightarrow a_i', x') < \\
\text{depth}(a_i \rightarrow a_i', x)
\end{align*}
\]

And therefore,
\[
\begin{align*}
\text{d}(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) < \\
\text{depth}(a_i \rightarrow a_i', x) - \text{depth}(a_i \rightarrow a_i', x'),
\end{align*}
\]

contradicting DB.

“⇐” Every pair of profiles \( a, a' = (a_{\_i}, a'_i) \) that violates TMON or DB, violates SP.

TMON. Consider a mechanism \( f \) and a pair of profiles \( a, a' = (a_{\_i}, a'_i) \) that violates TMON, i.e., \( \exists i \in \mathbb{N}, a, a'_i \) s.t.
\[
x : = f(a) \in \text{tree}_k(a_i \rightarrow a'_i); x' : = f(a_{\_i}, a'_i) \in \text{tree}_j(a_i \rightarrow a'_i); \text{ and } j < k.
\]

It follows that
\[
\begin{align*}
\text{d}(a_i, x') &= \text{d}(a_i, \text{tree}(a_i \rightarrow a'_i, x')) + \text{depth}(a_i \rightarrow a'_i, x') \\
\text{d}(a_i, x) &= \text{d}(a_i, \text{tree}(a_i \rightarrow a'_i, x')) \\
&\quad + \text{d}(\text{tree}(a_i \rightarrow a'_i, x'), \text{tree}(a_i \rightarrow a'_i, x)) \\
&\quad + \text{depth}(a_i \rightarrow a'_i, x).
\end{align*}
\]

Assume by contradiction that \( f \) is SP. Then it follows that \( \forall i, a, a'_i, \text{d}(a_i, x) \leq \text{d}(a_i, x') \), and thus from Eqs. (7), (8) above,
\[
\begin{align*}
\text{d}(\text{tree}(a_i \rightarrow a'_i, x'), \text{tree}(a_i \rightarrow a'_i, x)) + \text{depth}(a_i \rightarrow a'_i, x) \leq \\
\text{depth}(a_i \rightarrow a'_i, x').
\end{align*}
\]

From the contradiction assumption \( x \) and \( x' \) are in different trees. Therefore,
\[
\text{depth}(a_i \rightarrow a'_i, x) < \text{depth}(a_i \rightarrow a'_i, x').
\]

Symmetrically, \( \forall i, a, a'_i : \text{d}(a'_i, x) \geq \text{d}(a'_i, x') \) from which we derive
\[
\text{depth}(a_i \rightarrow a'_i, x) > \text{depth}(a_i \rightarrow a'_i, x'),
\]
contradicting Eq. (9) above. Therefore the pair \( a, a' \) violates SP.

DB. Consider an SP mechanism \( f \) and a pair of profiles \( a, a' = (a_{\_i}, a'_i) \). For any mechanism \( f \) and profile \( a \) it holds that
\[
\text{d}(a_i, x) = \text{d}(a_i, \text{tree}(a_i \rightarrow a'_i, x)) + \text{depth}(a_i \rightarrow a'_i, x).
\]
From TMON:
\[
\begin{align*}
    d(a_i, x') &= d(a_i, \text{tree}(a_i \rightarrow a_i', x)) \\
    &\quad + d(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) \\
    &\quad + \text{depth}(a_i \rightarrow a_i', x').
\end{align*}
\]
(11)

From strategyproofness, \(\forall i, a, a_i', d(a_i, x) \leq d(a_i, x')\), and thus from Eqs. (10),(11) above,
\[
depth(a_i \rightarrow a_i', x) \leq d(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) + \text{depth}(a_i \rightarrow a_i', x')
\]
Therefore,
\[
d(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) \geq \text{depth}(a_i \rightarrow a_i', x) - \text{depth}(a_i \rightarrow a_i', x')
\]
(12)

Symmetrically, from strategyproofness,
\[
\forall i, a, a_i' : d(a_i', x) \geq d(a_i', x')
\]
from which we derive
\[
d(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) \geq \text{depth}(a_i \rightarrow a_i', x') - \text{depth}(a_i \rightarrow a_i', x)
\]
(13)

And thus, from Eqs. (12),(13) above,
\[
d(\text{tree}(a_i \rightarrow a_i', x), \text{tree}(a_i \rightarrow a_i', x')) \geq |\text{depth}(a_i \rightarrow a_i', x') - \text{depth}(a_i \rightarrow a_i', x)|
\]
Therefore, any pair of profiles that violates DB and satisfies TMON, violates SP.

Lemma 6 characterizes all pairs of profiles that violate SP. We later show a stronger characterization that describes the pairs that indicate that an onto mechanism is not SP. These pairs differ only in a single agent’s location, who does not necessarily benefit from misreporting, contrary to the characterization in Lemma 6.

TPAR

Lemma 7 Every SP and onto mechanism \(f\) on the discrete tree is TPAR.

Proof We show that the unanimity property of SP mechanisms is violated. Assume by contradiction that an onto, SP mechanism \(f\) is not TPAR. Then there exists a profile \(a\) s.t. \(d(f(a), \text{Int}(a)) > 1\). Let \(v\) and \(u\) denote the second and the third vertex on the path from \(f(a)\) to all other locations of \(a\). Let \(z' := (a_1, \ldots, a_i, v, \ldots, v)\). Note that \(z''\) is the profile \(a\) and \(z''\) is the profile \((v, \ldots, v)\). We use induction on \(i\) to show that for every profile \(z' \in V^n\) and it holds that

1. \(f(z') \in \text{tree}(u \rightarrow v, v)\)
2. \(\text{depth}(u \rightarrow v, f(z')) = 1\)
For the base case we consider \( i = n \).

1. \( f(z^n) \in \text{tree}(u \to v, v) \)
2. \( \text{depth}(u \to v, f(z^n)) = 1 \)

This holds by the definition of \( u \) and \( v \). For the induction step, suppose the claim is true for \( i + 1 \). We need to show that

1. \( f(z^i) \in \text{tree}(u \to v, v) \)
2. \( \text{depth}(u \to v, f(z^i)) = 1 \)

The first claim follows from TMON. Since TPAR is violated, \( u \) is on the path from any agent \( i \) from \( a_i \) to \( v \). Therefore the facility cannot move towards \( u \) as a result of the deviation of agent \( i \). The second claim follows from DB. Since

\[ \text{depth}(u \to v, f(z^i)) = \text{depth}(u \to v, f(z^{i+1})) \]

the distance between the trees is 0, and therefore, from DB:

\[ \text{depth}(u \to v, f(z^i)) = \text{depth}(u \to v, f(z^{i+1})) = 1. \]

We have that \( \text{depth}(a_1 \to v, f(z^0)) = 1 \neq 0 \) and therefore \( f(v, \ldots, v) \neq v \) in contradiction to unanimity of SP and onto mechanisms. \( \square \)

**ADR**

Here we prove that ADR is a necessary condition for an onto, SP mechanism. We first show that if a pair of profiles violates the property, there exists a pair of profiles that violates the property in which the agent moves to a vertex on the path between the two outcomes.

**Lemma 8** If an onto, SP mechanism \( f \) violates ADR, there exists a pair of violating profiles \( a, a' = (a_i, a'_i) \) s.t.

\[ z = \arg\min_{v \in [f(a), f(a')]} d(v, a_i) = a'_i \]

and

\[ d(f(a), z) = d(f(a'), z). \]

**Proof** Assume by contradiction that there is a violating pair \((a, a')\) for which \( f(a) \) and \( f(a') \) are different vertices in the same tree w.r.t. the move \( a_i \to a'_i \).

1. \( d(f(a), z) = d(f(a'), z) \): From the definition of \( z \) it holds that

\[ z \in [a_i, f(a)] \cap [a_i, f(a')]. \quad (14) \]

Since \( f(a) \) and \( f(a') \) are in the same tree w.r.t. the move \( a_i \to a'_i \), it also holds that

\[ z \in [a'_i, f(a)] \cap [a'_i, f(a')], \quad (15) \]

Otherwise, let \( u := \arg\min_{v \in [f(a), f(a')]} d(v, a'_i) \) s.t. \( u \neq z \). In this case
\[ |[a_i, a'_i] \cap [f(a) \cap f(a')]| = |[z, a]| > 1 \]

which implies that

\[ \text{tree}(a_i \to a'_i, z) = \text{tree}(a_i \to a'_i, f(a)) \neq \text{tree}(a_i \to a'_i, f(a')) = \text{tree}(a_i \to a'_i, u), \]

contradicting the initial assumption. From Eq. (14), it holds that

\[ d(a_i, f(a)) = d(a_i, z) + d(z, f(a)) \quad (16) \]

and

\[ d(a_i, f(a')) = d(a_i, z) + d(z, f(a')). \quad (17) \]

From Eq. (15), it holds that

\[ d(a'_i, f(a)) = d(a'_i, z) + d(z, f(a)) \quad (18) \]

and

\[ d(a'_i, f(a')) = d(a'_i, z) + d(z, f(a')) \quad (19) \]

If \( d(f(a), z) < d(f(a'), z) \), from Eqs. (18), (19), agent \( i \) benefits for the deviation \( a'_i \to a_i \). Otherwise, if \( d(f(a), z) > d(f(a'), z) \), from Eqs. (16), (17), agent \( i \) benefits for the deviation \( a_i \to a'_i \). We conclude that \( d(f(a), z) = d(f(a'), z) \) for every violating pair \((a, a')\).

2. There exists a violating profile s.t. \( a'_i = z \): Consider the profile \( c = (a_i = z, a_{-i}) \). This profile is the result of the deviation \( a_i \to a'_i \) in profile \( a \) or the deviation \( a'_i \to a_i \) in profile \( a' \). Therefore it follows from TMON that

\[ f(c) \in \text{tree}(a_i \to a'_i, f(a)) = \text{tree}(a_i \to a'_i, f(a')) \]

Otherwise, if \( d(\text{tree}(a_i \to a'_i, f(c)), a_i) < d(\text{tree}(a_i \to a'_i, f(a)), a_i) \), mechanism \( f \) is not TMON w.r.t. the pair of profiles \((a, c)\). If \( d(\text{tree}(a_i \to a'_i, f(c)), a'_i) < d(\text{tree}(a_i \to a'_i, f(a')), a'_i) \), mechanism \( f \) is not TMON w.r.t. the pair of profiles \((a', c)\). For the same reason,

\[ f(c) \in \text{tree}(a_i \to z, f(a)) = \text{tree}(a'_i \to z, f(a')) \]

Let \( y \) denote the unique point s.t. \( y = \text{argmin}_{v \in [f(a), f(c)]} d(v, a_i) \) and \( y' \) the unique point s.t. \( y' = \text{argmin}_{v \in [f(a'), f(c)]} d(v, a_i) \) (see example in Fig. 4). From DB, we have that

\[ d(f(a), y) = d(f(c), y) \text{ and } d(f(a'), y') = d(f(c), y'). \]

We divide into the following cases:

(a) \( d(f(c), y) = d(f(a), y) = 0 \): This means that \( f(a) = f(c) \). In this case the pair \( a', c \) violates ADR and satisfies that the deviating agent moves from \( a'_i \) to \( c_i = z \) and the proof follows.

(b) \( d(f(a'), y') = d(f(c), y') = 0 \): Similar to the previous case, the proof follows for the pair \( a, c \).

(c) \( d(f(c), y) = d(f(a), y) = 1 = d(f(c), y') = d(f(a'), y') = 1 \): In this case, \( y = y' \), in contradiction to the assumption that the pair \((a, a')\) violates ADR.

(d) \( d(f(a), y) > 1 \): In this case, the proof follows for the pair \( a, (y, a_{-i}) \), which violates ADR.
(e) $d(f(a'), y') > 1$: In this case, the proof follows for the pair $a', (y', a_{-i})$, which violates ADR.

The following lemma proves the necessity of ADR by showing that there is no violating pair of profiles in which the agent moves to a location on the path between the outcomes.

The proof technique is based on the proof of 1-SI in [13]. They base the proof on a pair of maximum profiles on the line that violate the 1-SI property, and show that if there is a violating pair, then there is also a pair that maximizes the sum of all agents’ locations. In a similar way, we consider a pair of profiles that violates ADR and minimizes the sum of the distances of the agents’ locations from the trajectory of the deviating agent. We define an additional pair of profiles that exists due to the TPAR property. Each one of the four profiles can be achieved by a single agent’s deviation from another profile of the four. We show that there is a violation of one of the defined properties of onto, SP mechanisms, or there must be a different pair of profiles that achieves a smaller sum of distances. This way we reach a contradiction.

**Lemma 9** Every onto, SP mechanism on the discrete tree is ADR.

**Proof** We prove the lemma in three steps.

1. Definition of 4 profiles that differ by a single agent’s deviation: Assume that there exists an agent $i$ and two profiles $a, a' = (a'_i, a_{-i})$ s.t. $x := f(a), x' := f(a')$ and

   \[
   \text{tree}(a_i \rightarrow a'_i, x) = \text{tree}(a_i \rightarrow a'_i, x').
   \]

   From DB, we have that $d(x, z) = d(x', z)$, where $z$ is the unique point s.t. $z = \arg\min_{v \in [x, x']} d(v, a_i)$. We show that ADR holds when $z = a'_i$. Assume by contradiction that

   \[
   z = a'_i \text{ and } d(x, z) = d(x', z) > 1 \tag{20}
   \]

   Among the violating pairs, let $(a, a')$ be the one that minimizes $\Sigma_{k \neq i} d(a_k, [a_i, a'_i])$. We let $v_1$ denote the second vertex on the path from $x$ to $z$ and $v_2$ denote the third vertex on the path from $x$ to $z$ (see Fig. 5.1 and 2). Note that $v_2$ and $z$ is the same vertex if $d(x, z) = d(x', z) = 2$. Since $f$ is TPAR, there must be some other agent $j$ s.t. $x$ is sufficiently close to the path from agent $j$ to agent $i$. Formally,

   $\exists j : a_j \in \text{tree}(a_i \rightarrow v_1, v_1) \setminus \{v_1\}$

\[\square\]
(see locations \(a_j, a_j', a_j''\) in Fig. 5.1). We define two profiles \(b = (a_{-j}, b_j = v_1)\) and \(b' = (a'_{-j}, b'_j = v_1)\), which differ from \(a\) and \(a'\) only by the location of agent \(j\). Let \(y\) denote \(f(b)\) and \(y'\) denote \(f(b')\).

2. Showing that

For the pair of profiles \((a, b)\): If \(x\) is on the path from \(a_j\) to \(v_1\) (location \(a_j\) in Fig. 5.1), it follows from DB and TMON that the facility will stay in the same tree w.r.t. the move \(a_j \rightarrow v_1\) at depth 0 (location \(y_3\) in Fig. 5.3) or move to \(tree(a_j \rightarrow v_1, v_1)\) and be located at depth 0 (locations \(y_1, y_4\) in Fig. 5.3). Otherwise, if \(x\) is not on the path from \(a_j\) to \(v_1\) (locations \(a_j, a_j'\) in Fig. 5.1), \(x\) will stay in the same tree at the same depth w.r.t. the move \(a_j \rightarrow v_1\) (locations \(y_1, y_3\) in Fig. 5.3). Overall, the possible locations of \(y\) are \(x, v_1, v_2\) and the direct children of \(v_1\).

3. Any location \(y'\) violates SP or the profile minimality condition. Location \(y'\) satisfies: \(d(y', v_1) \leq d(x', v_1)\). Otherwise, SP is violated for the pair of profiles \((a', b')\). The location \(y'\) also satisfies

\[
y' \in tree(a_i \rightarrow a'_i, v_1); \text{ and } \quad depth(a_i \rightarrow a'_i, y') = depth(a_i \rightarrow a'_i, y) \tag{21}
\]

Otherwise, SP is violated for the pair of profiles \((b, b')\) (therefore \(y'\) in Fig. 5.4 is not a valid location). We divide into two cases by the possible locations of \(y'\):

Fig. 5 An illustration of the possible locations of agent \(j\) and the facility locations for the profiles \(a, a', b, b'\) in the proof of Lemma 9
(a) \( y' \) is a child of \( v_1 \) in \( \text{tree}(a_i \rightarrow a'_i, v_1) \): In this case the pair \((a', b')\) is a violation of SP, since
\[
d(a_j, y') \leq d(a_j, v_1) + d(v_1, y') \tag{22}
\]
\[
d(a_j, v_1) + 1 \tag{23}
\]
\[
d(a_j, v_1) + 3 \tag{24}
\]
Equation (22) follows from the triangle inequality, Eq. (23) follows from the case condition and Eq. (24) follows from Eq. (20).

(b) \( y' \) is not a child of \( v_1 \) in \( \text{tree}(a_i \rightarrow a'_i, v_1) \): It follows from Eq. (21) and the case condition that
\[
y' \notin \text{tree}(a_i \rightarrow a'_i, v_1) \setminus \{v_1\} \tag{25}
\]
Therefore, \( v_1 \in [a_j, y'] \). From Eq. (25),
\[
d(v_1, y') = d(v_1, x') \tag{26}
\]
Otherwise, the pair \((a', b')\) violates SP, since the nearest location to \( v_1 \) between \( y' \) and \( x' \) is strictly closer to agent \( j \) whether agent \( j \) is located at \( a_j \) or at \( v_1 \). From Eq. (21),
\[
depth(a_i \rightarrow a'_i, y') \in \{\text{depth}(a_i \rightarrow a'_i, x), \text{depth}(a_i \rightarrow a'_i, v_1), \text{depth}(a_i \rightarrow a'_i, v_2)\}
\]

We divide into two cases:

(i) \( \text{depth}(a_i \rightarrow a'_i, y') \in \{\text{depth}(a_i \rightarrow a'_i, v_1), \text{depth}(a_i \rightarrow a'_i, v_2)\} \) (locations \( y'_4, y'_5 \) in Fig. 5.4):
\[
d(v_1, y') \leq d(v_1, z) + d(z, y') \tag{27}
\]
\[
d(v_1, z) + d(z, x') \tag{28}
\]
contradicting Eq. (26). Equation (27) follows from the triangle inequality and Eq. (28) follows from the case condition.

(ii) \( \text{depth}(a_i \rightarrow a'_i, y') = \text{depth}(a_i \rightarrow a'_i, x) \) (locations \( y'_2, y'_3 \) in Fig. 5.4): Since \( y' \) is not a direct child of \( v_1 \), the pair \((b, b')\) violates ADR in contradiction to the minimality of \( \sum_{k \neq i} d(a_k, [a_i, a'_i]) \), since agent \( j \) is closer to the trajectory of agent \( i \) in profiles \( b, b' \) than in profiles \( a, a' \).

We have shown that there is no valid location for \( y' \), and therefore ADR is not violated for the case in which \( a'_i = z \). From Lemma 8, every onto, SP mechanism is ADR. \( \square \)

1-TSI
Here we prove that 1-TSI is a necessary condition for an onto, SP mechanism. First we show that if there is a violation of 1-TSI, there must exist a single-step deviation that violates 1-TSI. We then show that a violating one-step deviation violates ADR.

Lemma 10  If an onto, SP mechanism $f$ violates 1-TSI, there exists a pair of violating profiles $a, a' = (a_i, a'_i)$ s.t. $d(a_i, a'_i) = 1$.

Proof  Assume that $d(a_i, a'_i) > 1$ and assume w.l.o.g. that $d(f(a), [a_i, a'_i]) > 1$.

We denote by $x, y$ the two adjacent vertices on the path from $a_i$ to $a'_i$ s.t. $x' := (a_{-i}, x), y' := (a_{-i}, y)$ and

$$\text{tree}(a_i \rightarrow a'_i, f(x')) \neq \text{tree}(a_i \rightarrow a'_i, f(y'))$$  \hspace{1cm} (29)

If there is more than one such pair then we select the pair $x, y$ closest to $a_i$ (see Fig. 6). From Lemma 6, every onto, SP mechanism satisfies DB. Therefore, it holds that for every move $a_i \rightarrow z$ where $z \in [a_i, x]$, the facility stays in the same depth w.r.t. its initial tree, i.e.,

$$\text{depth}(a_i \rightarrow z, f(a)) = \text{depth}(a_i \rightarrow z, f(a))$$

Assume by contradiction that

$$\text{tree}(x \rightarrow y, f(x')) = \text{tree}(x \rightarrow y, f(y'))$$  \hspace{1cm} (30)

From Eq. (29), when agent $i$ moves from $x$ to $y$, the trajectory of the facility intersects the segment $[a_i, a'_i]$ in two points, and since $d(f(a), [a_i, a'_i]) \geq 2$, it holds that

$$d(f(a), f(y')) \geq 3$$  \hspace{1cm} (31)

On the other hand, from Eq. (30) and ADR we have that

$$d(f(a), f(y')) \in \{0, 2\}$$

contradicting Eq. (31) (see locations $f(x'), f(y')$ in Fig. 6). Thus, it follows that $\text{tree}(x \rightarrow y, f(x')) \neq \text{tree}(x \rightarrow y, f(y'))$. Therefore, the pair $(x', y')$ violates 1-TSI by a one-step deviation, since $d(f(x'), [a_i, a'_i]) > 1$, and in particular, $d(f(a), [x, y]) > 1$ as required.

The following lemma proves the necessity of 1-TSI by showing that there is no violating pair of profiles in which the agent moves to a neighboring vertex. Similar to the proof of ADR, we prove this by analyzing the deviation that violates the property and also satisfies a minimality condition, i.e., that the sum of the other agents’ distances from the deviating agent’s trajectory is the minimal possible for the mechanism.
Lemma 11  Every onto, SP mechanism on the discrete tree is 1-TSI.

Proof  Assume by contradiction that there exists a violating pair of profiles \( a, a' = (a_{-j}, a'_j) \) s.t.

\[
\text{tree}(a_i \rightarrow a', f(a)) \neq \text{tree}(a_i \rightarrow a', f(a_{-j}, a'_j))
\]

Assume w.l.o.g. that \( d(f(a'), [a_i, a'_j]) > 1 \). We can assume that \( d(a_i, a'_j) = 1 \) from Lemma 10. Among these pairs, let \( (a, a') \) be the pair that minimizes \( \sum_{k \neq i} d(a_k, [a_i, a'_j]) \). From TMON, we have that

\[
f(a) \in \text{tree}(a_i \rightarrow a'_j, a_i) \text{ and } f(a') \in \text{tree}(a_i \rightarrow a'_j, a'_j)
\]

Since \( d(a_i, a'_j) = 1 \),

\[
d(\text{tree}(a_i \rightarrow a'_j, f(a)), \text{tree}(a_i \rightarrow a'_j, f(a'))) = 1.
\]

In order to satisfy DB, the difference between the depths of \( f(a) \) and \( f(a') \) has to be at most 1. Therefore, \( \text{depth}(a_i \rightarrow a'_j, f(a)) \geq 1 \).

Let \( p \) denote the second vertex on the path from \( f(a) \) to \( a_i \) (see Fig. 7.1). From TPAR, there exists an agent \( j \) s.t.

\[
a_j \in \text{tree}(a'_j \rightarrow p, p) \setminus \{p\}
\]

(see locations \( a_{j_1}, a_{j_2}, a_{j_3} \) in Fig. 7.1). We define two profiles:

\[
b := (a_{-j}, b_j = p) \text{ and } b' := (a'_{-j}, b'_j = p)
\]

and their outcomes: \( y := f(b) \) and \( y' := f(b') \).

For the pair \( (a, b) \), when agent \( j \) moves to \( p \), it follows from DB that the facility can only move to \( p \), or to a vertex \( z \) s.t.
\[ d(p, z) = 1 \]  

(32)

For the pair \((a, a')\), when agent \(j\) moves to \(p\), it follows from ADR that the facility can only move from \(f(a')\) to a different child of \(p\) in the tree induced by the move \(a_i \rightarrow a'_i\) (see Fig. 7.2, 4). Therefore it holds that \(y' \in \text{tree}(a \rightarrow a', a'_i)\) and

\[ \text{depth}(a \rightarrow a', y') = \text{depth}(a \rightarrow a', f(a')) \geq 2 \]  

(33)

(see Figs. 7.2 and 4). We show that every possible location of the facility for profile \(b\) violates SP or the minimality condition:

1. \(y \in \text{tree}(a_i \rightarrow a'_i, a'_j)\) (location \(y_2\) in Fig. 7.3): DB is violated for the pair \((b, b')\) since \(y, y'\) are in the same tree w.r.t. the move \(a_i \rightarrow a'_i\), but from Eq. (32), \(\text{depth}(a_i \rightarrow a'_i, y) = 0\) while \(\text{depth}(a_i \rightarrow a'_i, y') \geq 2\) from Eq. (33).

2. \(y \in \text{tree}(a_i \rightarrow a'_i, a_j)\) (locations \(y_1, y_3, y_4\) in Fig. 7.3): The pair \((b, b')\) violates 1-TSI. This contradicts the minimality of \(\Sigma_{k \neq i}d(a_k, [a_i, a'_i])\), since agent \(j\) is closer to the trajectory of agent \(i\) in profiles \(b, b'\) than in profiles \(a, a'\).

\[ \square \]

5.2 Our characterization

We now complete our characterization of onto, SP mechanisms.

Lemma 12 Every TMON and ATC mechanism \(f\) on the discrete tree is SP.

Proof Suppose \(f\) is TMON and ATC. By definition it is 1-TSI and ADR. Consider an arbitrary pair of profiles \((a, a')\) s.t. \(a' = (a_{-i}, a'_i)\) for some \(i \in N, a'_i \in V\). If \(\text{tree}(a_i \rightarrow a'_i, f(a)) = \text{tree}(a_i \rightarrow a'_i, f(a'))\), it follows from ADR that

\[ \text{depth}(a_i \rightarrow a'_i, f(a)) - \text{depth}(a_i \rightarrow a'_i, f(a')) = 0 \]

\[ = d(\text{tree}(a_i \rightarrow a'_i, f(a)), \text{tree}(a_i \rightarrow a'_i, f(a'))) \]  

(34)

If \(\text{tree}(a_i \rightarrow a'_i, f(a)) \neq \text{tree}(a_i \rightarrow a'_i, f(a'))\), it follows from 1-TSI that

\[ d(f(a), [a_i, a'_i]) \leq 1; \text{ and } d(f(a'), [a_i, a'_i]) \leq 1 \]  

(35)

Eq. (35) implies that

\[ |\text{depth}(a_i \rightarrow a'_i, f(a)) - \text{depth}(a_i \rightarrow a'_i, f(a_{-i}, a'_i))| \leq 1 \]

\[ \leq d(\text{tree}(a_i \rightarrow a'_i, f(a)), \text{tree}(a_i \rightarrow a'_i, f(a'))) \]

From Eqs. (34), (35), \(f\) satisfies DB, and therefore \(f\) is SP from Lemma 6. \[ \square \]

We conclude that every onto mechanism \(f\) on the discrete tree is strategyproof if and only if it is TMON and ATC. This follows from Lemmas 6, 9, 11 and 12.
5.3 Generalization of previous results on the line

Here we show that when the discrete tree is a line, the mechanism properties MON, m-SI and DI, which were defined in [13], are easily derived from the properties TMON, 1-TSI and ADR. Note that on a line, all vertices \( V \) are ordered.

**Definition 22** (monotone [13]) A mechanism \( f \) on the discrete line is **monotone** (MON) if for every profile \( a, j \in N \) and \( b_j > a_j \), it holds that \( f(a_{-j}, b_j) \geq f(a) \).

**Definition 23** (m-step independent [13]) A mechanism \( f \) on the discrete line is **m-step independent** (m-SI) if the two following properties hold for every profile \( a \):

(a) For every \( j \in N, a_j' > a_j \), if \( d([a_j, a_j'], f(a)) > m \), then \( f(a_j', a_{-j}) = f(a) \).

(b) For every \( j \in N, a_j' \leq a_j \), if \( d([a_j, a_j'], f(a)) > m \), then \( f(a_j', a_{-j}) = f(a) \).

**Definition 24** (disjoint independent [13]) A mechanism \( f \) on the discrete line is **disjoint independent** (DI) if for every profile \( a, j \in N \) and \( a_j' \), it holds that if \( f(a) = x \neq x' = f(a_j', a_{-j}) \), then \( |A \cap X| \geq 2 \), where \( A = [\min(a_j, a_j'), \max(a_j, a_j')] \) and \( X = [\min\{x, x'\}, \max\{x, x'\}] \).

Since every line is also a tree, our previous definitions of TMON, m-TSI and ADR apply for lines. We show that for line graphs they coincide with the properties above.

**Lemma 13** A mechanism \( f \) on the discrete line-tree is DI if and only if it is ADR.

**Proof** A mechanism \( f \) on the discrete line is DI if and only if it is ADR.

”\( \Rightarrow \)” If \( f \) is DI, for every \( a \in V^n, i \in N, a_i' \in V \) s.t. \( f(a) \neq f(a_{-i}, a_i') \), it holds that

\[
|[a_i, a_i'] \cap [f(a), f(a_{-i}, a_i')]| \geq 2,
\]

which is equivalent to stating that

\[
\text{tree}(a_i \rightarrow a_i', f(a)) \neq \text{tree}(a_i \rightarrow a_i', f(a_{-i}, a_i')).
\]

Therefore it follows from DI that

\[
\text{tree}(a_i \rightarrow a_i', f(a)) = \text{tree}(a_i \rightarrow a_i', f(a_{-i}, a_i')) \Rightarrow f(a) = f(a_{-i}, a_i'),
\]

which satisfies a condition that is stronger than the condition for ADR.

”\( \Leftarrow \)” If \( f \) is ADR, for every \( a \in V^n, i \in N, a_i' \in V \) s.t.

\[
\text{tree}(a_i \rightarrow a_i', f(a)) = \text{tree}(a_i \rightarrow a_i', f(a_{-i}, a_i'))
\]

(36)

It holds that

\[
\text{depth}(a_i \rightarrow a_i', f(a)) = \text{depth}(a_i \rightarrow a_i', f(a_{-i}, a_i')),
\]

which, in case of a line-tree, is equivalent to stating that \( f(a) = f(a_{-i}, a_i') \). Eq. (36) is equivalent to the condition
Therefore, it follows that
$$\| [a_i, a'_i] \cap [f(a), f(a_{-i}, a'_i)] \| < 2.$$  
which satisfies the condition for DI.

\[ \square \]

**Lemma 14** For every \( m \in \mathbb{N}_0 \), a mechanism \( f \) on the discrete line-tree is \( m\text{-SI} \) if it is ADR and \( m\text{-TSI} \).

**Proof** For every \( m \in \mathbb{N}_0 \), a mechanism \( f \) on the discrete line-tree is \( m\text{-SI} \) if it is ADR and \( m\text{-TSI} \).

For every \( a \in V^n, i \in N, a'_i \in V \) s.t. \( f(a) \neq f(a_{-i}, a'_i) \), we separate into the following two cases:

1. \( \text{tree}(a_i \rightarrow a'_i, f(a)) = \text{tree}(a_i \rightarrow a'_i, f(a_{-i}, a'_i)) \): It follows from ADR that
   $$\text{depth}(a_i \rightarrow a'_i, f(a)) = \text{depth}(a_i \rightarrow a'_i, f(a_{-i}, a'_i)),$$
   which implies that \( f(a) = f(a_{-i}, a'_i) \) on lines.

2. \( \text{tree}(a_i \rightarrow a'_i, f(a)) \neq \text{tree}(a_i \rightarrow a'_i, f(a_{-i}, a'_i)) \): It follows from \( m\text{-TSI} \) that
   $$d([a_i, a'_i], f(a)) \leq m.$$

In any case, if \( f(a) \neq f(a_{-i}, a'_i) \) then
$$d([a_i, a'_i], f(a)) \leq m,$$
which satisfies the condition for \( m\text{-SI} \). \[ \square \]

**Lemma 15** For every \( m \in \mathbb{N}_0 \), a mechanism \( f \) on the discrete line-tree is \( m\text{-TSI} \) if it is \( m\text{-SI} \).

**Proof** For every \( m \in \mathbb{N}_0 \), a mechanism \( f \) on the discrete line-tree is \( m\text{-TSI} \) if it is \( m\text{-SI} \). If \( f \) is \( m\text{-SI} \), then for every \( a \in V^n, i \in N, a'_i \in V \) s.t. \( d([a_i, a'_i], f(a)) > m \), it follows that

$$f(a) = f(a_{-i}, a'_i),$$
which is stronger than the condition for \( m\text{-TSI} \). \[ \square \]

**Lemma 16** A mechanism \( f \) on the discrete line-tree is \( \text{TMON} \) if it is \( \text{MON} \).

**Proof** Consider a \( \text{MON} \) mechanism \( f \) on a discrete line, a profile \( a \in V^n \), an agent \( i \in N \) and an alternative location \( a'_i \in V \). Assume w.l.o.g. that \( a_i < a'_i \). Let

$$f = f(a) \text{ and } f' = f(a_{-i}, a'_i),$$
$$f \in \text{tree}_{k}(a_i \rightarrow a'_i) \text{ and } f' \in \text{tree}_{k}(a_i \rightarrow a'_i).$$

If \( f \leq a_i \) and \( f' \leq a_i \), or \( f \geq a'_i \) and \( f' \geq a'_i \) or \( f = f' \), it holds that \( j = k \). Otherwise, \( \text{MON} \) implies that \( j < k \). The case \( a_i > a'_i \) is symmetric. It follows that every pair of profiles satisfies \( \text{TMON} \). \[ \square \]
Lemma 17 A mechanism \( f \) on the discrete line-tree is MON if it is TMON and ADR.

**Proof** Consider a TMON and ADR mechanism \( f \) on a discrete line, a profile \( a \in \mathbb{V}^n \), an agent \( i \in N \) and an alternative location \( a'_i \in V \). Assume w.l.o.g. that \( a_i < a'_i \). Let
\[
\begin{align*}
f &= f(a) \quad \text{and} \quad f' = f(a_{i-1}, a'_i), \\
f &\in \text{tree}_j(a_i \to a'_i) \quad \text{and} \quad f' \in \text{tree}_k(a_i \to a'_i).
\end{align*}
\]

if \( j \neq k \), TMON implies that \( j > k \) and therefore \( f' > f \). Otherwise, ADR implies that \( f = f' \). It follows that every pair of profiles satisfies MON. \( \square \)

**Corollary 2** A mechanism \( f \) on a discrete line is MON, 1-SI and DI if and only if it is TMON, 1-TSI and ADR.

Corollary 2 together with our Theorem 3 immediately implies Dokow et al. [13] characterization for discrete lines, including the infinite discrete line. We should mention that their proof cannot be extended to infinite lines due to its use of a “maximal” profile.

### 6 Shift-invariant mechanisms

In this section, we show that on infinite discrete lines, the only anonymous, shift-invariant SP mechanisms are order statistics mechanisms—as on continuous lines [27]. Many characterization results have a “neutral” variant. Here shift-invariance is the corresponding property to neutrality. The result itself is interesting since it shows neutrality kills the difference between discrete and continuous domains. That is, all additional SP mechanisms in the discrete domain must single out specific locations to satisfy neutrality.

**Definition 25** (Shift-Invariant) A mechanism \( f \) on an infinite discrete line (w.l.o.g. \( \mathbb{Z} \)) is shift-invariant if for every location profile \( a \in \mathbb{Z}^n \) and \( d \in \mathbb{Z} \), it holds that \( f(a_1 + d, \ldots, a_n + d) = f(a) + d \).

**Definition 26** (Anonymous) A mechanism \( f \) is anonymous if for every location profile \( a \in \mathbb{Z}^n \) and every permutation of agents \( \pi : N \to N \), it holds that \( f(a_1, \ldots, a_n) = f(a_{\pi_1}, \ldots, a_{\pi_n}) \).

Due to anonymity, we assume w.l.o.g. that the agents are ordered by their location in profile \( a \), i.e., if \( a_i < a_j \), \( i < j \).

**Definition 27** (kth order statistic) A mechanism \( f \) is the kth order statistic mechanism for some \( k \leq n \), if for every profile \( a = (a_1, \ldots, a_n) \), it holds that \( f(a) = a_k \).

**Lemma 18** If a mechanism \( f \) is onto, SP and shift-invariant, then for every profile \( a = (a_1, \ldots, a_n) \), it holds that \( f(a) = a_i \) for some \( i \in [1, n] \).

**Proof** Consider a pair of profiles \( a, a' = (a_1 + 1, \ldots, a_n + 1) \). From shift-invariance, \( f(a') = f(a) + 1 \). The profile \( a' \) can be achieved by \( n \) one-step moves, one per agent. Let

\[\footnote{This property is sometimes called “tops-only” or “peaks-only”}.\]
$z' := (a_1, \ldots, a_i, a_{i+1} + 1, \ldots, a_n + 1)$. Note that $z'' = a$ and $z^0 = a'$. For every such move of agent $j \in N$, resulting in profile $a^{j+1}$, the following hold:

1. $f(a') \geq f(a^{j+1})$ from TMON.
2. If $f(a') \neq f(a^{j+1})$, from shift-invariance and the previous statement, it holds that $f(a') = f(a^{j+1}) + 1$.

It follows that the facility moves as a result of the move of a single agent $k$ and $f(a) = a_k$. Otherwise,

$$\text{tree}(a_k \to a_k + 1, f(a')) = \text{tree}(a_k \to a_k + 1, f(a))$$

and

$$\text{depth}(a_k \to a_k + 1, f(a')) \neq \text{depth}(a_k \to a_k + 1, f(a))$$

contradicting DB.

Lemma 19 An onto, SP, anonymous, shift-invariant mechanism $f$ for the discrete line is a $k$th order statistic mechanism.

Proof Assume by contradiction that $f$ is not a $k$th order statistic mechanism, i.e. there exists a pair of profiles $(a, b)$ s.t. $f(a)$ is the $j$th agent location and $f(b)$ is the $k$th agent location. Assume that $j < k$. Let $d$ denote $b_n - a_1 + 1$. Consider a profile $c$ where $c_i = b_i - d$ if $d > 0$ and $c_i = b_i$ otherwise. Note that from shift-invariance, $f(c) = c_k$.

We now iteratively move each agent in profile $c$ to a location in profile $a$. In the $l$th iteration we move the agent with index $n + 1 - l$ to the location $a_{n+1-l}$, getting a sequence of profiles $(c^l)_{l=1}^n$. From TMON, the output in every iteration is the location of an agent with an index higher then or equal to $k$. After the $n$th iteration we reach a profile $c^n$ that is identical to $a$, up to a permutation of agents. Thus by anonymity $f(a) = f(c^n)$.

On the other hand, $f(c^n) \geq (c^n)_k = a_k > a_j = f(a)$, i.e. a contradiction. The proof is similar for the case $j > k$.

Theorem 4 An onto, $n$-agent mechanism $f$ is anonymous, shift-invariant and SP if and only if it is the $k$th order statistic for some $k \leq n$.

Proof The first direction follows from Lemmas 18 and 19.

We prove the second direction. Clearly, a $k$th order statistic mechanism is anonymous and shift-invariant. The only way for agent $i$ to change the chosen location is by reporting a location $a_i > a_k$ if $i < k$ or a location $a_i < a_k$ if $i > k$. In both cases, the distance of the agent from the facility will increase.

7 Conclusion and open questions

In this research, we provide a complete characterization of onto and strategyproof facility location mechanisms on discrete trees—under quadratic preferences. Interestingly, while a characterization for continuous trees exists due to [32], these are not easily compared,
as the latter uses a collection of median-like rules rather than axiomatic properties. The key property that allows comparison of continuous and discrete mechanisms is trajectory containment (TC): while this property characterizes exactly the strategyproof onto mechanisms on continuous trees, it needs to be relaxed in a particular way to apply for discrete trees.

A different characterization for discrete trees and general single-peaked preferences [29] uses a third type of properties that map agents to leafs of the tree. Thus a better understanding of how properties from the different models map onto one another is important.

One possible direction for further research is a characterization of strategyproof mechanisms on discrete weighted graphs. Contrary to continuous graphs, there is a finite set of possible locations for each agent and outcome. Unlike the discrete case, the distances between such two neighboring locations vary along the tree.

Additionally, characterizations of SP mechanisms can promote the study of optimal SP approximation mechanisms, for example for minimizing the Egalitarian cost.

Acknowledgement This research was supported by the Israel Science Foundation (ISF; Grant No. 2539/20).

References

1. Alon, N., Feldman, M., Procaccia, A. D., & Tennenholtz, M. (2010). Strategyproof approximation of the minimax on networks. Mathematics of Operations Research, 35(3), 513–526.
2. Anastasiadis, E., & Deligkas, A. (2018). Heterogeneous facility location games. In: The 17th International conference on autonomous agents and multiagent systems (AAMAS’18).
3. Barbera, S., & Peleg, B. (1990). Strategy-proof voting schemes with continuous preferences. Social Choice and Welfare, 7(1), 31–38.
4. Black, D. (1948). On the rationale of group decision-making. Journal of Political Economy, 56(1), 23–34.
5. Border, K. C., & Jordan, J. S. (1983). Straightforward elections, unanimity and phantom voters. The Review of Economic Studies, 50(1), 153–170.
6. Chan, H., Filos-Ratsikas, A., Li, B., Li, M., & Wang. C. (2021). Mechanism design for facility location problems: A survey. arXiv preprint arXiv:2106.03457.
7. Chen, X., Fang, Q., Liu, W., & Ding, Y. (2020). Strategyproof mechanisms for 2-facility location games with minimax envy. In: International conference on algorithmic applications in management, pp. 260–272. Springer.
8. Chen, Z., Fong, K. C., Li, M., Wang, K., Yuan, H., & Zhang, Y. (2020). Facility location games with optional preference. Theoretical Computer Science, 847, 185–197.
9. Cheng, Y., Yu, W., & Zhang, G.(2011). Mechanisms for obnoxious facility game on a path. In: International conference on combinatorial optimization and applications, pp. 262–271. Springer.
10. Cheng, Y., Yu, W., & Zhang, G. (2013). Strategy-proof approximation mechanisms for an obnoxious facility game on networks. Theoretical Computer Science, 497, 154–163.
11. De Keijzer, B., & Wojtczak, D. (2022). Facility reallocation on the line. Algorithmica, 84(10), 2898–2925.
12. Deligkas, A., Filos-Ratsikas, A., & Voudouris, A. A. (2022). Heterogeneous facility location with limited resources. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 5, pp. 4966–4974.
13. Dokow, E., Feldman, M., Meir, R., & Nehama, I. (2012). Mechanism design on discrete lines and cycles. In: Proceedings of the 13th ACM conference on electronic commerce, pp. 423–440.
14. Duan, L., Li, B., Li, M., & Xu, X. (2019) Heterogeneous two-facility location games with minimum distance requirement. In: The 18th international conference on autonomous agents and multiagent systems (AAMAS’19), pp. 1461–1469.
15. Feldman, M., Fiat, A., & Golomb, I. (2016). On voting and facility location. In: Proceedings of the 2016 ACM conference on economics and computation, pp. 269–286. ACM.
16. Feldman, M., & Wilf, Y. (2013). Strategyproof facility location and the least squares objective. In: Proceedings of the fourteenth ACM conference on Electronic commerce, pp. 873–890.
17. Filimonov, A., & Meir, R. (2021). Strategyproof facility location mechanisms on discrete trees. In: Proceedings of the 20th international conference on autonomous agents and multiagent systems, AAMAS ’21, pp. 510–518.
18. Filos-Ratsikas, A., Voudouris, & A.A. (2021). Approximate mechanism design for distributed facility location. In: International symposium on algorithmic game theory, pp. 49–63. Springer.
19. Fong, K.C., Li, M., Lu, P., Todo, T., & Yokoo, M. (2018). Facility location games with fractional preferences. In: 32nd AAAI conference on artificial intelligence, AAAI 2018, pp. 1039–1046. AAAI Press.
20. Fotakis, D., Kavouras, L., Kostopanagiotis, P., Lazos, P., Skoulakis, S., & Zarifis, N. (2021). Reallocating multiple facilities on the line. Theoretical Computer Science, 858, 13–34.
21. Fotakis, D., & Patsilinakos, P. (2021) Strategyproof facility location in perturbation stable instances. In: International conference on web and internet economics, pp. 95–112. Springer.
22. Fotakis, D., & Tzamos, C. (2014). On the power of deterministic mechanisms for facility location games. ACM Transactions on Economics and Computation (TEAC), 2(4), 1–37.
23. Ibara, K., & Nagamochi, H. (2012). Characterizing mechanisms in obnoxious facility game. In: International conference on combinatorial optimization and applications, pp. 301–311. Springer.
24. Kyropoulou, M., Ventre, C., & Zhang, X. (2019). Mechanism design for constrained heterogeneous facility location. In: International symposium on algorithmic game theory, pp. 63–76. Springer.
25. Liu, W., Ding, Y., Chen, X., Fang, Q., & Nong, Q. (2020). Multiple facility location games with envy ratio. In: International conference on algorithmic applications in management, pp. 248–259. Springer.
26. Meir, R. (2019). Strategyproof facility location with three agents on a circle. In: International symposium on algorithmic game theory, pp. 18–33. Springer.
27. Moulin, H. (1980). On strategy-proofness and single peakedness. Public Choice, 35(4), 437–455.
28. Nehama, I., Todo, T., & Yokoo, M. (2019). Manipulations-resistant facility location mechanisms for zv-line graphs. In: Proceedings of the 18th international conference on autonomous agents and multiagent systems, pp. 1452–1460.
29. Peters, H., Roy, S., & Sadhukhan, S. (2019) Unanimous and strategy-proof probabilistic rules for single-peaked preference profiles on graphs. Tech. rep., Working Paper.
30. Procaccia, A.D., & Tennenholtz, M. (2009) Approximate mechanism design without money. In: Proceedings of the 10th ACM conference on Electronic commerce, pp. 177–186.
31. Roy, S., & Storecken, T. (2019). A characterization of possibility domains in strategic voting. Journal of Mathematical Economics, 84, 46–55.
32. Schummer, J., & Vohra, R. V. (2002). Strategy-proof location on a network. Journal of Economic Theory, 104(2), 405–428.
33. Tang, Z.Z., Wang, C.H., Zhang, M.Q., & Zhao, Y.C. (2022) Strategyproof facility location with limited locations. Journal of the Operations Research Society of China pp. 1–15.
34. Todo, T., Iwasaki, A., & Yokoo, M. (2011) False-name-proof mechanism design without money. In: The 10th international conference on autonomous agents and multiagent systems (AAMAS’11), pp. 651–658.
35. Todo, T., Okada, N., & Yokoo, M. (2019) False-name-proof facility location on discrete structures. arXiv preprint arXiv:1907.08914.
36. Wada, Y., Ono, T., Todo, T., & Yokoo, M. (2018) Facility location with variable and dynamic populations. In: The 17th International conference on autonomous agents and multiagent systems (AAMAS’18), pp. 336–344.
37. Xu, X., Li, M., & Duan, L. (2020) Strategyproof mechanisms for activity scheduling. In: The 19th international conference on autonomous agents and multiagent systems (AAMAS’20), pp. 1539–1547.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.