

# Strategyproof Facility Location Mechanisms on Discrete Trees

Alina Filimonov

Technion - Israel Institute of Technology  
Haifa, Israel  
salin@campus.technion.ac.il

Reshef Meir

Technion - Israel Institute of Technology  
Haifa, Israel  
reshefm@ie.technion.ac.il

## 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

### ACM Reference Format:

Alina Filimonov and Reshef Meir. 2021. Strategyproof Facility Location Mechanisms on Discrete Trees. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3-7, 2021*, IFAAMAS, 9 pages.

## 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).

An  $n$ -agent facility location mechanism on a domain  $\mathbb{D}$  receives a profile of the agents' locations  $a = (a_1, \dots, a_n) \in \mathbb{D}^n$  and outputs a location in  $\mathbb{D}$  depending on the profile. We refer to the agent's location as her peak. We say that a mechanism is SP if it is a weakly dominant strategy of every agent to report truthfully.

The fundamental characterization result for strategyproof facility location was given by Moulin [12], 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 the discrete lines [9], but not on continuous lines [5].

Schummer and Vohra [16] 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. [9] that results on continuous graphs do not 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 [16], it is only "almost dictatorial" when the circle is discrete [9]. 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 can be applied to discrete trees.

As we will later show, a naïve extension of the properties defined in [9] fails. We therefore formulate similar properties to characterize the valid moves of the facility under SP, onto 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. [14] 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 [14] does not apply to our study.

### 1.1 Structure and Contribution

After some preliminary notation in Section 3, we provide an alternative characterization of onto, SP mechanisms on continuous trees in Section 4, based on the work of Schummer and Vohra [16].

We then present our main result in Section 5— a full characterization of onto, SP mechanisms on discrete trees. In contrast to the work of Dokow et al. [9], our proof also works for infinite trees (with bounded degree).

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

Some of the omitted proofs are available in the full version of this paper.<sup>1</sup>

*Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*, U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3-7, 2021, Online. © 2021 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

<sup>1</sup><https://arxiv.org/abs/2102.02610>.

## 2 RELATED WORK

Following the initial work of Black [4], several researchers have developed characterizations of deterministic and probabilistic strategyproof facility location mechanisms in various scenarios.

Schummer and Vohra [16], 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 [9].

Additional variations of the problem include the multiple facility problem [6, 11], the obnoxious facility location problem [7, 8], the heterogeneous facility location problem [2, 10], and the activity scheduling problem [20].

Todo et al. [17] 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 [13, 18]. 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. A later work of Wada et al. [19] on variable and dynamic populations characterizes mechanisms that incentivize the agents to participate in the reporting process.

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 [15], is the facility location problem. Their work was extended to the domain of continuous graphs by Alon et al. [1]. 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.

## 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  $v_0, \dots, v_k$  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  $v_1, \dots, v_k$ , and by  $(u, w)$  the sequence  $v_1, \dots, v_{k-1}$ , where  $\{v_i, v_{i+1}\} \in E$  for all  $i = 0, \dots, k-1$ . 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  $N = \{1, \dots, n\}$  be the set of agents, and  $a = (a_1, \dots, a_n) \in V^n$  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 location.

For any two vertices  $u, w \in V$ , the notation  $u \succ_i w$  indicates that agent  $i$  prefers vertex  $u$  over vertex  $w$ . The notation  $u \succeq_i w$  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 inversely 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  $a_i \in V$ ,

$$\forall u, w \in V : u \succ_i w \iff d(a_i, u) < d(a_i, w)$$

Next, we give the standard definitions of mechanism properties:

**Definition 3.1 (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  $a'_i \in V$ , it holds that

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

**Definition 3.2 (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 3.3 (Unanimous).** A mechanism  $f$  is **unanimous** if for every location  $x \in V$ ,  $f(x, \dots, x) = x$

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

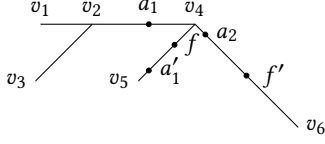
**LEMMA 3.4 (BARBERA AND PELEG [3]).** *Every mechanism that is both onto and SP, is unanimous.*

The definitions above apply also 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 [16]. 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  $s = \operatorname{argmin}_{s \in S} d(p, s)$ . 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 [16] provided a characterization of onto, SP mechanisms on continuous trees. They showed that when the agents' preferences are quadratic, every SP, onto mechanism on the continuous tree is based on a set of generalized median voter schemes, defined in [12], 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, 16].



**Figure 1:** In this tree,  $\{v_i | 1 \leq i \leq 6\}$  is the set of vertices.  $\{v_1, v_3, v_5, v_6\}$  is the set of leaves. 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  $a_2$ .

*Previous Results on a Continuous Line.* The following definition given in [16] describes onto and SP mechanisms on a continuous line. It is similar to the one introduced by Moulin [12] for single-peaked preferences and confirmed for quadratic preferences by Border and Jordan [5].

*Definition 4.1 (Generalized Median Voter Scheme [16]).* A function  $g^{xy}$  is called a **generalized median voter scheme** (g.m.v.s.) on  $[x, y]$  if there exist  $2^{|N|}$  points in  $[x, y]$ ,  $\{\alpha_S^{xy}\}_{S \subseteq N}$  such that:

- (1)  $S \subset R$  implies that  $d(\alpha_S^{xy}, x) \leq d(\alpha_R^{xy}, x)$ .
- (2)  $\alpha_\emptyset^{xy} = x$  and  $\alpha_N^{xy} = y$ .
- (3) For all  $a \in [x, y]^n$ ,  $g^{xy}(a)$  is the unique point satisfying

$$d(g^{xy}(a), x) = \max_{S \subseteq N} \min\{d(a_i, x)_{i \in S}, d(\alpha_S^{xy}, x)\}$$

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 4.2 (Uncompromising [5]).* A mechanism  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is called **uncompromising** if for every  $a \in \mathbb{R}^n$ ,  $i \in N$ ,  $a'_i \in \mathbb{R}$ , 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 4.3 (BORDER AND JORDAN [5]).** Suppose that  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is SP and unanimous. Then  $f$  is uncompromising.

As shown in [16], 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 3.4. Therefore we can rely on uncompromisingness for our characterization of onto and SP mechanisms.

*Previous Results on a Continuous Tree.*

*Definition 4.4 (Graph Restriction [16]).* 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 [16], 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  $a_i|_{xy} = \operatorname{argmin}_{z \in [x, y]} d(z, a_i)$ .

*Definition 4.5 (Extended Generalized Median Voter Scheme [16]).* 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.

- (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 4.6 (SCHUMMER AND VOHRA [16]).** For any continuous tree  $G$ , a rule  $f$  is SP and onto if and only if it is an e.m.v.s.

We omit the definition of consistency since it is not relevant for our purpose.

## 4.1 Our Characterization

We rely on the characterization in [16] 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 4.7 (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  $a'_i \in G$  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 4.8 (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  $a'_i \in G$ , it either holds that  $[f(a), f(a_{-i}, a'_i)] \subseteq [a_i, a'_i]$ , or  $f(a) = f(a_{-i}, a'_i)$ .

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 4.9.** Every onto and SP mechanism  $f$  on the continuous tree is TC.

**PROOF.** Consider an SP, onto mechanism  $f : G^n \rightarrow G$  on a continuous tree. Assume by contradiction that there exists an agent  $i$  and two profiles  $a, a' = (a'_i, a_{-i})$  s.t.  $f(a) \neq f(a')$  and w.l.o.g., that  $f(a) \notin [a_i, a'_i]$ . By Theorem 4.6,  $f$  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 4.3,  $g^{xy}$  is uncompromising. From the second property of the e.m.v.s., it holds that

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

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

- (1)  $|[a_i, a_i|_{xy}] \cap [a'_i, a'_i|_{xy}]| > 0$ : In this case,  $a_i|_{xy} = a'_i|_{xy}$  and therefore  $g^{xy}(a|_{xy}) = g^{xy}(a'|_{xy})$  contradicting the assumption that  $f(a) \neq f(a')$ .
- (2)  $|[a_i, a_i|_{xy}] \cap [a'_i, a'_i|_{xy}]| = 0$ : In this case, the path between  $a_i$  and  $a'_i$  must intersect the segment  $[x, y]$  and therefore,  $[x, y] \cap [a_i, a'_i] = [a_i|_{xy}, a'_i|_{xy}]$ . By our initial assumption  $f(a) = g^{xy}(a|_{xy}) \notin [a_i, a'_i]$ , thus  $f(a) \notin [a_i|_{xy}, a'_i|_{xy}]$ , contradicting the uncompromisingness of  $g^{xy}$ .  $\square$

LEMMA 4.10. *Every SP mechanism on the continuous tree is TMON.*

LEMMA 4.11. *Every TC and TMON mechanism on the continuous tree is SP.*

The proofs of these lemmas are omitted since they are straightforward from the definition of strategyproofness and Lemma 4.9.

THEOREM 4.12. *An onto mechanism on the continuous tree is SP if and only if it is TMON and TC.*

PROOF. The proof follows from Lemmas 4.9, 4.10 and 4.11.  $\square$

## 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 [9].

Before presenting the main result, we show that a naïve extension of the properties defined for mechanisms on discrete lines in [9] 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 SP, onto 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 5.1 (Single-Peaked [14]). 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  $a_i \neq y$ ,

$$x \in [a_i, y) \Rightarrow x \succ_i y$$

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:  $0 \succ_1 2 \succ_1 3 \succeq_1 1 \succ_1 4$
- (2) Agent 2:  $3 \succ_2 4 \succeq_2 2 \succ_2 0 \succ_2 1$

Both agents have single-peaked preferences according to the definition in [14]. 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 similarly to the case of the line-graph, quadratic preferences allow more SP mechanisms than single-peaked preferences, and therefore the characterization of probabilistic SP mechanisms under single-peaked preferences in [14] 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.

```

if  $\exists i : a_i = 0$  then
   $f(a) = a_{-i} \bmod 2$ 
else if  $\exists i : a_i = 2$  then
   $f(a) = \min\{a_{-i}, 3\}$ 
else  $f(a) = \min\{a_i, a_{-i}\}$ 
end if

```

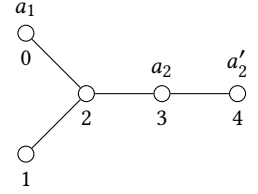


Figure 2: An example of a two-agent mechanism that is SP and onto on a discrete tree, that violates the properties defined by Dokow et al. [9].

Definition 5.2 (Tree).  $tree(a \rightarrow b, v)$  is the subtree which includes only  $v$  and vertices which are accessible from  $v$ , via the edges that are not in  $[a, b]$ .

Definition 5.3 (Depth).  $depth(a \rightarrow b, v)$  is the distance of vertex  $v$  from  $[a, b]$ .

We demonstrate the above definitions in Fig. 3:  $tree(a_1 \rightarrow a'_1, v_1)$  contains  $v_1$  (at depth 0) and another node at depth 1.  $depth(a_1 \rightarrow a'_1, v_2) = 2$  in the subtree rooted by  $a_1$ .

Our next definitions are intended to generalize the properties defined in [9].

Definition 5.4 ( $m$ -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

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

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

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

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

Definition 5.5 (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

$$d(tree(a_i \rightarrow a'_i, f(a)), tree(a_i \rightarrow a'_i, f(a_{-i}, a'_i))) \geq |depth(a_i \rightarrow a'_i, f(a)) - depth(a_i \rightarrow a'_i, f(a_{-i}, a'_i))|$$

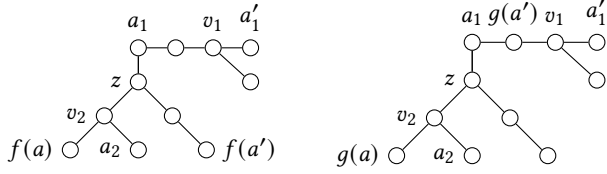
Informally, DB means that when the facility moves as a result of a single agent's deviation, the distance between the tree of the original outcome and the tree of the new outcome is bigger than the difference between the depths of the outcomes. Fig. 3 illustrates a violation of the property by mechanism  $g$ .  $g$  violates DB since

$$d(tree(a_1 \rightarrow a'_1, g(a)), tree(a_1 \rightarrow a'_1, g(a'_1, a_2))) = 1 < |depth(a_1 \rightarrow a'_1, g(a)) - depth(a_1 \rightarrow a'_1, g(a'_1, a_2))| = 3$$

Definition 5.6 (Tree Pareto Location). Let  $Int(a)$  be the set of interior vertices of the subtree defined by profile  $a$ :

$$Int(a) = \{v \in V \mid \exists a_i, a_j \in a \text{ s.t. } v \in (a_i, a_j)\}.$$

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



**Figure 3: Mechanism  $f$  violates ADR and TPAR w.r.t. profile  $a' = (a'_1, a_2)$ , but satisfies TPAR w.r.t. profile  $a$ . Mechanism  $g$  violates 1-TSI and DB.**

This definition generalizes the definition in [9] of a Pareto location on the discrete line. Note that it is weaker than the standard definition of Pareto.

*Definition 5.7 (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

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

*Definition 5.8 (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  $z$  be the unique point s.t.

$$z = [a_i, f(a)] \cap [a_i, f(a'_i, a_{-i})] \cap [f(a), f(a'_i, a_{-i})]$$

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  $(a, a')$ , which differ by the location of agent 1, since  $d(f(a), z) = 2$ .

*Definition 5.9 (Almost Trajectory Contained).* A mechanism  $f$  is **almost trajectory contained** (ATC) if it is ADR and 1-TSI.

Finally, we use the term TMON defined in Section 4 to describe mechanisms on discrete trees as well.

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

**THEOREM 5.10.** *An onto mechanism  $f$  on the discrete tree is SP if and only if it is TMON and ATC.*

First we show 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 5.11.** *A pair of profiles violates SP if and only if it violates DB or TMON.*

The proof is straightforward from the definition of SP. 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  $a_i \rightarrow a'_i$  or the move  $a'_i \rightarrow a_i$ . This is true for infinite trees as well.

Lemma 5.11 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 5.11.

*TPAR.*

**LEMMA 5.12.** *Every SP and onto mechanism  $f$  on the discrete tree is TPAR.*

**PROOF.** 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  $v$  denote the first vertex on the path from  $f(a)$  to all other locations of  $a$ . Let  $z^i := (a_1, \dots, a_i, v, \dots, v)$ . Note that  $z^n$  is the profile  $a$  and  $z^0$  is the profile  $(v, \dots, v)$ . By definition of  $v$ , we have that  $f(a) = f(z^n) \in \text{tree}(a_i \rightarrow v, v)$ . Since TPAR is violated,  $v$  has exactly one neighbor  $u$  on the path to any other agent location  $a_i \neq v$ . When moving from profile  $z^{i+1}$  to  $z^i$ , it follows from TMON that  $\forall 0 \leq i \leq n-1$ ,

$$f(z^i) \in \text{tree}(a_i \rightarrow v, f(z^{i+1})) = \text{tree}(a_i \rightarrow v, v) \quad (1)$$

From DB and (1) we have that

$$\text{depth}(a_i \rightarrow v, f(z^i)) = 1 \neq 0 = \text{depth}(a_i \rightarrow v, v) \quad (2)$$

It follows from (2) that

$$f(z^0) = f(v, \dots, v) \neq v$$

in contradiction to unanimity.  $\square$

*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 5.13.** *If an onto, SP mechanism  $f$  violates ADR, there exists a pair of violating profiles  $a, a' = (a_{-i}, a'_i)$  s.t.*

$$z = [a_i, f(a)] \cap [a_i, f(a')] \cap [f(a), f(a')] = a'_i$$

and

$$d(f(a), z) = d(f(a'), z)$$

The proof is omitted since it can easily be shown that if an agent's move violates ADR, a second move to the closest vertex on the path between the outcomes violates ADR w.r.t. one of the outcomes.

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.

LEMMA 5.14. *Every onto SP mechanism on the discrete tree is ADR.*

PROOF. 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.  $z = [a_i, x] \cap [a_i, x'] \cap [x, x']$ . We show that ADR holds when  $z = a'_i$ . Assume by contradiction that

$$z = a'_i \text{ and } d(x, z) = d(x', z) > 1 \quad (3)$$

Among the violating pairs, let  $(a, a')$  be one of the pairs that minimize  $\sum_{k \neq i} d(a_k, [a_i, a'_i])$ . We let  $v_1$  denote the first vertex on the path from  $x$  to  $z$  and  $v_2$  denote the second vertex on the path from  $x$  to  $z$  (see Figs. 4.1 and 4.2). 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,

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

(see locations  $a_{j_1}, a_{j_2}, a_{j_3}$  in Fig. 4.1). 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  $f(b')$ .

For the pair of profiles  $(a, b)$ : If  $x$  is on the path from  $a_j$  to  $v_1$  (location  $a_{j_3}$  in Fig. 4.1), 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. 4.3) or move to  $\text{tree}(a_j \rightarrow v_1, v_1)$  and be located at depth 0 (location  $y_2$  in Fig. 4.3) or 1 (locations  $y_1, y_4$  in Fig. 4.3). Otherwise, if  $x$  is not on the path from  $a_j$  to  $v_1$  (locations  $a_{j_1}, a_{j_2}$  in Fig. 4.1),  $x$  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. 4.3). Overall, the possible locations of  $y$  are  $x, v_1, v_2$  and the direct children of  $v_1$ .

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

$$y' \in \text{tree}(a_i \rightarrow a'_i, a'_i); \text{ and} \\ \text{depth}(a_i \rightarrow a'_i, y') = \text{depth}(a_i \rightarrow a'_i, y) \quad (4)$$

Otherwise, SP is violated for the pair of profiles  $(b, b')$  (therefore  $y'_1$  in Fig. 4.4 is not a valid location). We divide into two cases by the possible locations of  $y'$ :

- (1)  $y'$  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

$$d(a_j, y') \leq d(a_j, v_1) + d(v_1, y') \quad (5)$$

$$= d(a_j, v_1) + 1 \quad (6)$$

$$< d(a_j, v_1) + 3 \quad (7)$$

$$\leq d(a_j, x') \quad (7)$$

Eq. (5) follows from the triangle inequality, Eq. (6) follows from the case condition and Eq. (7) follows from Eq. (3).

- (2)  $y'$  is not a child of  $v_1$  in  $\text{tree}(a_i \rightarrow a'_i, v_1)$ : It follows from Eq. (4) and the case condition that

$$y' \notin \text{tree}(a_i \rightarrow a'_i, v_1) \setminus \{v_1\} \quad (8)$$

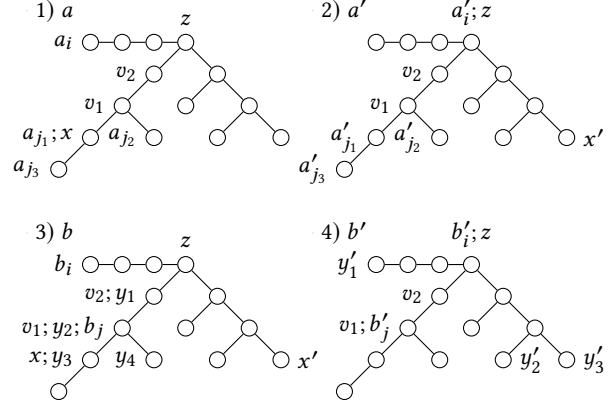


Figure 4: An illustration of the possible locations of agent  $j$  and the facility locations for the profiles  $a, a', b, b'$  in the proof of Lemma 5.14.

Therefore,  $v_1 \in [a_j, y']$ . From Eq. (8),

$$d(v_1, y') = d(v_1, x') \quad (9)$$

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. (4),

$$\text{depth}(a_i \rightarrow a'_i, y') \in \{\text{depth}(a_i \rightarrow a'_i, x), \\ \text{depth}(a_i \rightarrow a'_i, v_1), \text{depth}(a_i \rightarrow a'_i, v_2)\} \quad (10)$$

We divide into two cases:

- (a)  $\text{depth}(a_i \rightarrow a'_i, y') \in \{\text{depth}(a_i \rightarrow a'_i, v_1), \\ \text{depth}(a_i \rightarrow a'_i, v_2)\}$  (locations  $y'_4, y'_5$  in Fig. 4.4):

$$d(v_1, y') \leq d(v_1, z) + d(z, y') \quad (11)$$

$$< d(v_1, z) + d(z, x') \quad (12)$$

$$= d(v_1, x')$$

contradicting Eq. (9). Eq. (11) follows from the triangle inequality and Eq. (12) follows from the case condition.

- (b)  $\text{depth}(a_i \rightarrow a'_i, y') = \text{depth}(a_i \rightarrow a'_i, x)$  (locations  $y'_2, y'_3$  in Fig. 4.4): 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  $a, a'$ .

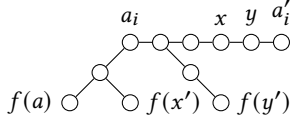
We have shown that there is no valid location for  $y'$ , and therefore ADR is not violated for the case in which  $a'_i = z$ . From Lemma 5.13, every SP onto mechanism is ADR.  $\square$

1-TSI. Here we prove that 1-TSI is a necessary condition for an onto, SP mechanism. We first show that if there is a violation of the 1-TSI property, then w.l.o.g. it occurs when an agent moves a single step.

LEMMA 5.15. *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

$$d(f(a), [a_i, a'_i]) > 1$$



**Figure 5: An illustration of the proof of Lemma 5.15. The pair  $(x' = (a_{-i}, x), y' = (a_{-i}, y))$  violates ADR.**

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

$$tree(a_i \rightarrow a'_i, f(x')) \neq tree(a_i \rightarrow a'_i, f(y')) \quad (13)$$

If there is more than one such pair then we select the pair  $x, y$  closest to  $a_i$  (see Fig. 5). From Lemma 5.11, every SP, onto 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.,

$$depth(a_i \rightarrow z, f(a_{-i}, z)) = depth(a_i \rightarrow z, f(a)) \quad (14)$$

Assume by contradiction that

$$tree(x \rightarrow y, f(x')) = tree(x \rightarrow y, f(y')) \quad (15)$$

From Eq. (13), 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

$$d(f(a), f(y')) \geq 3 \quad (16)$$

On the other hand, from Eq. (15) and ADR we have that

$$d(f(a), f(y')) \in \{0, 2\} \quad (17)$$

contradicting Eq. (16) (see locations  $f(x'), f(y')$  in Fig. 5). Thus, it follows that  $tree(x \rightarrow y, f(x')) \neq tree(x \rightarrow y, f(y'))$ . 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.  $\square$

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.

**LEMMA 5.16.** *Every onto, SP mechanism on the discrete tree is 1-TSI.*

**PROOF.** Assume by contradiction that there exists a violating pair of profiles  $a, a' = (a_{-i}, a'_i)$  s.t.

$$tree(a_i \rightarrow a', f(a)) \neq tree(a_i \rightarrow a', f(a_{-i}, a'_i)) \quad (18)$$

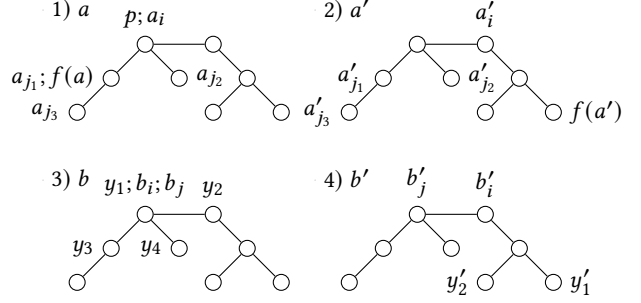
Assume w.l.o.g. that  $d(f(a'), [a_i, a'_i]) > 1$ . We can assume that  $d(a_i, a'_i) = 1$  from lemma 5.15. Among these pairs, let  $(a, a')$  be the pair that minimizes  $\sum_{k \neq i} d(a_k, [a_i, a'_i])$ . From TMON, we have that

$$f(a) \in tree(a_i \rightarrow a'_i, a_i) \text{ and } f(a') \in tree(a_i \rightarrow a'_i, a'_i)$$

Since  $d(a_i, a'_i) = 1$ ,

$$d(tree(a_i \rightarrow a'_i, f(a)), tree(a_i \rightarrow a'_i, f(a'))) = 1$$

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



**Figure 6: An illustration of the possible locations of agent  $j$  and the facility locations  $y, y'$  for the profiles  $a, a', b, b'$ .**

Let  $p$  denote the first vertex on the path from  $f(a)$  to  $a_i$  (see Fig. 6.1). From TPAR, there exists an agent  $j$  s.t.

$$a_j \in tree(a'_i \rightarrow p, p) \setminus p$$

(see locations  $a_{j_1}, a_{j_2}, a_{j_3}$  in Fig. 6.1). 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.

$$d(p, z) = 1 \quad (19)$$

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. 6.2, Fig. 6.4). Therefore it holds that  $y' \in tree(a \rightarrow a'_i, a'_i)$  and

$$depth(a \rightarrow a', y') = depth(a \rightarrow a', f(a')) \geq 2 \quad (20)$$

(see Figs. 6.2 and 6.4). We show that every possible location of the facility for profile  $b$  violates SP or the minimality condition:

- (1)  $y \in tree(a_i \rightarrow a'_i, a'_i)$  (location  $y_2$  in Fig. 6.3): 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. (19),  $depth(a_i \rightarrow a'_i, y) = 0$  while  $depth(a_i \rightarrow a'_i, y') \geq 2$  from Eq. (20).
- (2)  $y \in tree(a_i \rightarrow a'_i, a_i)$  (locations  $y_1, y_3, y_4$  in Fig. 6.3): The pair  $(b, b')$  violates 1-TSI. This contradicts 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  $a, a'$ .  $\square$

## 5.2 Our Characterization

We now complete our characterization of onto, SP mechanisms.

**LEMMA 5.17.** *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  $tree(a_i \rightarrow a'_i, f(a)) = tree(a_i \rightarrow a'_i, f(a'))$ , it follows from ADR that

$$\begin{aligned} & depth(a_i \rightarrow a'_i, f(a)) - depth(a_i \rightarrow a'_i, f(a')) = 0 \\ & = d(tree(a_i \rightarrow a'_i, f(a)), tree(a_i \rightarrow a'_i, f(a'))) \end{aligned} \quad (21)$$

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

$$d(f(a), [a_i, a'_i]) \leq 1; \text{ and } d(f(a'), [a_i, a'_i]) \leq 1 \quad (22)$$

Eq. (22) implies that

$$\begin{aligned} & |depth(a_i \rightarrow a'_i, f(a)) - depth(a_i \rightarrow a'_i, f(a_{-i}, a'_i))| \leq 1 \\ & \leq d(tree(a_i \rightarrow a'_i, f(a)), tree(a_i \rightarrow a'_i, f(a'))) \end{aligned}$$

From Eqs. (21), (22),  $f$  satisfies DB, and therefore  $f$  is SP from Lemma 5.11.  $\square$

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 5.11, 5.14, 5.16 and 5.17.

## 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 [12]. 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 6.1 (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  $f(a_1+d, \dots, a_n+d) = f(a)+d$ .

*Definition 6.2 (Anonymous).* A mechanism  $f$  is **anonymous** if for every location profile  $a \in \mathbb{Z}^n$  and every permutation of agents  $\pi : N \rightarrow N$ , it holds that  $f(a_1, \dots, a_n) = f(a_{\pi_1}, \dots, a_{\pi_n})$ .

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 6.3 ( $k$ th order statistic).* A mechanism  $f$  is the  **$k$ th order statistic** mechanism for some  $k \leq n$ , if for every profile  $a = (a_1, \dots, a_n)$ , it holds that  $f(a) = a_k$ .

LEMMA 6.4. *If a mechanism  $f$  is onto, SP and shift-invariant, then for every profile  $a = (a_1, \dots, a_n)$ , it holds that  $f(a) = a_i$  for some  $i \in [1, n]$ .*<sup>2</sup>

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

Let  $z^i := (a_1, \dots, a_i, a_{i+1} + 1, \dots, a_n + 1)$ . Note that  $z^n = a$  and  $z^0 = a'$ . For every such move of agent  $j \in N$ , resulting in profile  $a^{j-1}$ , the following hold:

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

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

$$tree(a_k \rightarrow a_k + 1, f(a')) = tree(a_k \rightarrow a_k + 1, f(a))$$

<sup>2</sup>This property is sometimes called "tops-only" or "peaks-only".

and

$$depth(a_k \rightarrow a_k + 1, f(a')) \neq depth(a_k \rightarrow a_k + 1, f(a))$$

contradicting DB.  $\square$

LEMMA 6.5. *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 or equal to  $k$ . After the  $n$ th iteration we reach a profile  $c^n$  that is identical to  $a$ , up to 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  $j > k$ .  $\square$

THEOREM 6.6. *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 6.4 and 6.5.

We prove the second direction. Clearly, a  $k$ th order statistic mechanism is anonymous and shift-invariant. The only way for an 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.  $\square$

## 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 [16], 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 [14] 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.



## REFERENCES

- [1] Noga Alon, Michal Feldman, Ariel D Procaccia, and Moshe Tennenholtz. 2010. Strategyproof approximation of the minimax on networks. *Mathematics of Operations Research* 35, 3 (2010), 513–526.
- [2] Eleftherios Anastasiadis and Argyrios Deligkas. 2018. Heterogeneous Facility Location Games. In *The 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'18)*.
- [3] Salvador Barbera and Bezalel Peleg. 1990. Strategy-proof voting schemes with continuous preferences. *Social choice and welfare* 7, 1 (1990), 31–38.
- [4] Duncan Black. 1948. On the rationale of group decision-making. *Journal of political economy* 56, 1 (1948), 23–34.
- [5] Kim C Border and James S Jordan. 1983. Straightforward elections, unanimity and phantom voters. *The Review of Economic Studies* 50, 1 (1983), 153–170.
- [6] Xin Chen, Qizhi Fang, Wenjing Liu, and Yuan Ding. 2020. Strategyproof Mechanisms for 2-Facility Location Games with Minimax Envy. In *International Conference on Algorithmic Applications in Management*. Springer, 260–272.
- [7] Yukun Cheng, Wei Yu, and Guochuan Zhang. 2011. Mechanisms for obnoxious facility game on a path. In *International Conference on Combinatorial Optimization and Applications*. Springer, 262–271.
- [8] Yukun Cheng, Wei Yu, and Guochuan Zhang. 2013. Strategy-proof approximation mechanisms for an obnoxious facility game on networks. *Theoretical Computer Science* 497 (2013), 154–163.
- [9] Elad Dokow, Michal Feldman, Reshef Meir, and Ilan Nehama. 2012. Mechanism design on discrete lines and cycles. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 423–440.
- [10] Lingjie Duan, Bo Li, Minming Li, and Xinpeng Xu. 2019. Heterogeneous Two-facility Location Games with Minimum Distance Requirement. In *The 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'19)*. 1461–1469.
- [11] Wenjing Liu, Yuan Ding, Xin Chen, Qizhi Fang, and Qingqin Nong. 2020. Multiple Facility Location Games with Envy Ratio. In *International Conference on Algorithmic Applications in Management*. Springer, 248–259.
- [12] Hervé Moulin. 1980. On strategy-proofness and single peakedness. *Public Choice* 35, 4 (1980), 437–455.
- [13] Ilan Nehama, Taiki Todo, and Makoto Yokoo. 2019. Manipulations-resistant Facility Location Mechanisms for ZV-line Graphs. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 1452–1460.
- [14] Hans Peters, Souvik Roy, and Soumyarup Sadhukhan. 2019. *Unanimous and strategy-proof probabilistic rules for single-peaked preference profiles on graphs*. Technical Report. Working Paper.
- [15] Ariel D Procaccia and Moshe Tennenholtz. 2009. Approximate mechanism design without money. In *Proceedings of the 10th ACM conference on Electronic commerce*. 177–186.
- [16] James Schummer and Rakesh V Vohra. 2002. Strategy-proof location on a network. *Journal of Economic Theory* 104, 2 (2002), 405–428.
- [17] Taiki Todo, Atsushi Iwasaki, and Makoto Yokoo. 2011. False-name-proof mechanism design without money. In *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*. 651–658.
- [18] Taiki Todo, Nodoka Okada, and Makoto Yokoo. 2019. False-Name-Proof Facility Location on Discrete Structures. *arXiv preprint arXiv:1907.08914* (2019).
- [19] Yuho Wada, Tomohiro Ono, Taiki Todo, and Makoto Yokoo. 2018. Facility Location with Variable and Dynamic Populations. In *The 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'18)*. 336–344.
- [20] Xinpeng Xu, Minming Li, and Lingjie Duan. 2020. Strategyproof Mechanisms for Activity Scheduling. In *The 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'20)*. 1539–1547.