Verifiably Truthful Mechanisms

Simina Brânzei
Computer Science Department
University of Aarhus, Denmark
simina@cs.au.dk

Ariel D. Procaccia
Computer Science Department
Carnegie Mellon University, USA
arielpro@cs.cmu.edu

ABSTRACT

It is typically expected that if a mechanism is truthful, then the agents would, indeed, truthfully report their private information. But why would an agent *believe* that the mechanism is truthful? We wish to design truthful mechanisms that are "simple", that is, whose truthfulness can be verified efficiently (in the computational sense). Our approach involves three steps: (i) specifying the structure of mechanisms, (ii) constructing a verification algorithm, and (iii) measuring the quality of verifiably truthful mechanisms. We demonstrate this approach using a case study: approximate mechanism design without money for facility location.

Categories and Subject Descriptors

F.m [**Theory of Computation**]: Miscellaneous; J.4 [**Social and Behavioral Sciences**]: Economics

Keywords

Mechanism Design, Decision Trees, Verification, Facility Location, Approximation

1. INTRODUCTION

The mechanism design literature includes a vast collection of clever schemes that, in most cases, provably give rise to a specified set of properties. Arguably, the most sought-after property is *truthfulness*, more formally known as *incentive compatibility* or *strategyproofness*: an agent must not be able to benefit from dishonestly revealing its private information. Truthfulness is, in a sense, a prerequisite for achieving other theoretical guarantees, because without it the mechanism may receive unpredictable input information that has little to do with reality. For example, if the designer's goal is to maximize utilitarian social welfare (the sum of agents' utilities for the outcome), but the mechanism is not truthful, the mechanism would indeed maximize social welfare — albeit, presumably, with respect to the wrong utility functions!

ITCS'15, January 11–13, 2015, Rehovot, Israel.

http://dx.doi.org/10.1145/2688073.2688098

An implicit assumption underlying the preceding (rather standard) reasoning is that when a truthful mechanism is used, (rational) agents would participate truthfully. This requires the agents to *believe* that the mechanism is actually truthful. Why would this be the case? Well, in principle the agents can look up the proof of truthfulness.[1] A more viable option is to directly verify truthfulness by examining the specification of the mechanism itself, but, from a computational complexity viewpoint, this problem would typically be extremely hard — even undecidable. This observation is related to the more general principle that the mechanism should be transparent and simple, so that bounded-rational economic agents can reason about it and take decisions efficiently.

Motivated by the preceding arguments, our goal in this paper is to design mechanisms that are *verifiably truthful*. Specifically, we would like the verification to be *efficient* — in the computational sense (i.e., polynomial time), not the economic sense. In other words, the mechanism must be truthful, and, moreover, each agent must be able to efficiently verify this fact.

1.1 Our Approach and Results

Our approach to the design of verifiably truthful mechanisms involves three steps:

(I) *Specifying the structure of mechanisms*: The verification algorithm will receive a mechanism as input — so we must rigorously specify which mechanisms are admissible as input, and what they look like.

(II) *Constructing a verification algorithm*: Given a mechanism in the specified format, the algorithm decides whether the mechanism is truthful.

(III) *Measuring the quality of verifiably truthful mechanisms*: The whole endeavor is worthwhile (if and) only if the family of mechanisms whose truthfulness can be verified efficiently (via the algorithm of Step 2) is rich enough to provide high-quality outcomes.

We instantiate this program in the context of a case study: *approximate mechanism design without money for facility location* [27]. The reason for choosing this specific domain is twofold. First, a slew of recent papers has brought about a good understanding of what quality guarantees are achievable via truthful facility location mechanisms [1, 21, 20, 25,

[1] A related, interesting question is: If we told human players that a non-truthful mechanism is provably truthful, would they play truthfully?

13, 14, 15, 29, 30, 8, 32]. Second, facility location has also served as a proof of concept for the approximate mechanism design without money agenda [27], whose principles were subsequently applied to a variety of other domains, including allocation problems [17, 16, 12, 9], approval voting [2], kidney exchange [3, 7], and scheduling [19]. Similarly, facility location serves as an effective proof of concept for the idea of verifiably truthful mechanisms, which, we believe, is widely applicable.

We present our results according to the three steps above:

(I) In §2, we put forward a representation of facility location mechanisms. In general, these are arbitrary functions mapping the reported locations of n agents on the real line to the facility location (also on the real line). We present *deterministic* mechanisms as *decision trees*, which branch on comparison queries in internal nodes, and return a function that is a convex combination of the reported locations in the leaves. Roughly speaking, randomized mechanisms are distributions over deterministic mechanisms, but we use a slightly more expressive model to enable a concise representation for certain randomized mechanisms that would otherwise need a huge representation.

(II) The *cost* of an agent is the distance between its (actual) location, which is its private information, and the facility location. A deterministic mechanism is truthful if an agent can never decrease its cost by reporting a false location. In §3, we show that the truthfulness of a deterministic mechanism can be verified in polynomial time in the size of its decision tree representation and number of agents. We also demonstrate that one cannot do much better: it is necessary to at least inspect all the tree's leaves. We establish that the efficient verification result extends to randomized mechanisms, as long as the notion of truthfulness is *universal truthfulness*: it must be impossible to gain from manipulating one's reported location, regardless of the mechanism's coin tosses.

(III) Building on the results of Step II, we focus on decision trees of polynomial size — if such mechanisms are truthful, their truthfulness can be efficiently verified. In §4, we study the quality of polynomial-size decision trees, via two measures of quality: the *social cost* (the sum of agents' cost functions) and the maximum cost (of any agent). Figure 1 summarizes our results. The table on the top shows tight bounds on the (multiplicative, worst-case) approximation ratio that can be achieved by truthful mechanisms [27] — deterministic in the first row, randomized in the second. The (lower) bound for the maximum cost of universally truthful mechanisms is new. The results for efficiently verifiable mechanisms are shown in the bottom table. Our main results pertain to the social cost (left column): while deterministic polynomial-size decision trees only achieve an approximation ratio of $\Theta(n/\log n)$, we construct (for any constant $\epsilon > 0$) a polynomial-size, randomized, universally truthful decision tree approximating the social cost to a factor of $1+\epsilon$.

General Mechanisms	Social Cost	Max Cost
Truthful	1	2
Univ. Truthful	1	2 $(*)$

Polynomial-size Decision Trees	Social Cost	Max Cost
Truthful	$\Theta\left(\frac{n}{\log n}\right)$	2
Univ. Truthful	$1+\epsilon$	2

Figure 1: The results of §4, outlined in §1.1. The lower bound $(*)$ for general mechanisms is also shown here

1.2 Related Work

Verification is a common theme in algorithmic mechanism design, but in the past it was always the agents' reports that were being verified, not the properties of the mechanism itself. In fact, in the eponymous paper by Nisan and Ronen [24], a class of mechanisms with verification (and money) for scheduling was proposed. These mechanisms are allowed to observe both the reported types and the actual types (based on the execution of jobs), and payments may depend on both. Verification of agents' reports has subsequently played a role in a number of papers; of special note is the work of Caragiannis et al. [6], who focused on different notions of verification. They distinguished between *partial verification*, which restricts agents to reporting a subset of types that is a function of their true type (e.g., in scheduling agents can only report that they are slower than they actually are, not faster), and *probabilistic verification*, which catches an agent red handed with probability that depends on its true type and reported type. There are also examples of this flavor of verification in approximate mechanism design without money [19].

A small body of work in multiagent systems [26, 5, 28] actually aims to verify properties of mechanisms and games. The work of Tadjouddine et al. [28] is perhaps closest to ours, as they verify the truthfulness of auction mechanisms. Focusing on the Vickrey Auction [31], they specify it using the Promela process modeling language, and then verify its truthfulness via model checking techniques. This basically amounts to checking all possible bid vectors and deviations in a discretized bid space. To improve the prohibitive running time, abstract model checking techniques are applied. While model checking approaches are quite natural, they inevitably rely on heuristic solutions to problems that are generally very hard. In contrast, we are interested in mechanisms whose truthfulness can be verified *in polynomial time*.

Mu'alem [23] considers a motivating scenario similar to ours and focuses on testing extended monotonicity, which is a property required for truthfulness in the single parameter domain studied therein. In particular, Mu'alem shows that if a function f is ϵ-close to extended monotonicity, then there *exists* an associated payment function p such that the mechanism given by the tuple (f,p) is $(1-2\epsilon)$-truthful. She also describes a shifting technique for obtaining almost truthful mechanisms and a monotonicity tester. While studying truthfulness in the context of property testing remains an interesting question for future work, we would like to obtain mechanisms whose truthfulness can be verified exactly and in polynomial time (independent of the size of the domain — in fact, our domain is continuous). On a technical level, we study a setting without payments, which does not admit a close connection between monotonicity and truthfulness.

Kang and Parkes [18] consider the scenario in which multiple entities (e.g. companies, people, network services) can deploy mechanisms in an open computational infrastructure. Like us, they are interested in verifying the truthfulness of mechanisms, but they sidestep the question of how mechanisms are represented by focusing on what they call *passive verification*: their verifier acts as an intermediary and monitors the sequence of inputs and outputs of the mechanism. The verifier is required to be sound and complete; in particular, if the mechanism is not strategyproof, the verifier is guaranteed to establish this fact after observing all the possible inputs and outputs.

Our work is also related to the line of work on *automated mechanism design* [10], which seeks to automatically design truthful mechanisms that maximize an objective function, given a prior distribution over agents' types. In an informal sense, this problem is much more difficult than our verification problem, and, indeed, in general it is computationally hard even when the mechanism is explicitly represented as a function whose domain is all possible type vectors. Automated mechanism design is tractable in special cases — such as when the number of agents is constant and the mechanism is randomized — but these results do not yield nontrivial insights on the design of verifiably truthful mechanisms.

2. STEP I: SPECIFYING THE STRUCTURE OF MECHANISMS

We consider the (game-theoretic) facility location problem [27]. An instance includes a set $N = \{1, \ldots, n\}$ of agents. Each agent $i \in N$ has a location x_i. The vector $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ represents the location profile. We relegate the presentation of the strategic aspects to Section 3.

2.1 Deterministic Mechanisms

A *deterministic mechanism* (for n agents) is a function $\mathcal{M} : \mathbb{R}^n \to \mathbb{R}$, which maps each location profile $\mathbf{x}$ to a facility location $y \in \mathbb{R}$. We put forward a simple, yet expressive, representation of deterministic mechanisms, via *decision trees*.

In more detail, given input $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, the mechanism is represented as a tree, with:

- *Internal nodes*: used to verify sets of constraints over the input variables. We focus on a comparison-based model of computation, in which each internal node verifies one constraint, of the form $(x_i \geq x_j)$, $(x_i \leq x_j)$, $(x_i > x_j)$, or $(x_i < x_j)$, for some $i, j \in N$. The node has two outgoing edges, that are taken depending on whether the condition is true or false.

- *Leaves*: store the outcome of the mechanism if the path to that leaf is taken, i.e. the facility location.We require that for each leaf $\mathcal{L}$, the location of the facility at $\mathcal{L}$, $y_{\mathcal{L}}(x)$, is a convex combination of the input locations: $y_{\mathcal{L}}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_{\mathcal{L},i} \cdot x_i$, where the $\lambda_{\mathcal{L},i}$ are constants with $\lambda_{\mathcal{L},i} \geq 0$ and $\sum_{i=1}^{n} \lambda_{\mathcal{L},i} = 1$.

The Average Mechanism *Dictatorship of Agent i*

For example, the left figure (above) shows the decision tree representation of the average mechanism, which returns the average of the reported locations. It is just a single leaf, with coefficients $\lambda_i = 1/n$ for all $i \in N$. The right figure shows a dictatorship of agent i — whatever location is reported by agent i is always selected. Figure 2 shows the median mechanism for $n = 3$, which returns the median of the three reported locations; this mechanism will play a key role later.

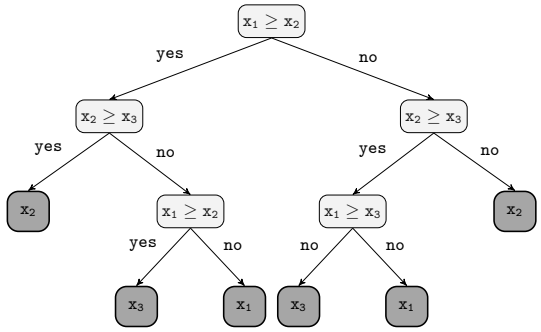

Figure 2: The median mechanism for 3 agents.

We remark that our positive results are based on mechanisms that have the so-called *peaks-only* property: they always select one of the reported locations. However, our more expressive definition of the leaves of the decision tree (as convex combinations of points in $\mathbf{x}$) is needed to compute optimal solutions under one of our two objectives (as we discuss below), and also strengthens our negative results.

2.2 Randomized Mechanisms

Intuitively, randomized mechanisms are allowed to make branching decisions based on coin tosses. Without loss of generality, we can just toss all possible coins in advance, so a randomized mechanism can be represented as a probability distribution over deterministic decision trees. However, this can lead to a large representation of simple mechanisms that consist of the same (fixed) subroutine executed with possibly different input variables. For example, the mechanism that selects a (not very small) subset of agents uniformly at random and computes the median of the subset can be seen as a median mechanism parameterized by the identities of the agents. In order to be able to represent such mechanisms concisely, we make the representation a bit more expressive.

Formally, a randomized mechanism is represented by a decision tree with a chance node of degree K as the root, such that the r'th edge selects a decision tree $\mathcal{T}_r$ and is taken with probability p_r, where $\sum_{i=1}^{K} p_i = 1$. Each tree $\mathcal{T}_r$ is defined as follows:

- There is a set of agents $N_r \subseteq N$, such that the locations x_i for $i \in N$ appear directly in the internal nodes and leaves of the tree.

- There is a set of parameters $Z_r = \{z_{r,1}, \ldots, z_{r,m_r}\}$, that also appear in the internal nodes and leaves of $\mathcal{T}_r$, where $0 \leq m_r \leq |N \setminus N_r|$.

- The description of $\mathcal{T}_r$ includes a probability distribution over tuples of m_r distinct agents from $N \setminus N_r$.

The semantics are as follows. At the beginning of the execution, a die is tossed to determine the index $r \in \{1, \ldots, K\}$ of the function (i.e. tree $\mathcal{T}_r$) to be implemented. Then, the parameters $z_{r,j}$ are bound to locations of agents from $N \setminus N_r$ according to the given probability distribution for $\mathcal{T}_r$; each

$z_{r,j}$ is bound to a different agent. At this point all the parameters in the nodes and leaves of $\mathcal{T}_r$ have been replaced by variables x_i, and we just have a deterministic decision tree, which is executed as described above.

For example, say we want to implement the mechanism that selects three agents uniformly at random from N and outputs the median of these three agents. This mechanism requires a randomized decision tree with a chance node of degree one, that selects with probability $p_1 = 1$ a single decision tree $\mathcal{T}_1$, which is the tree in Figure 2 with the x_i variables replaced by z_i. We set $N_1 = \emptyset$ (thus the tree $\mathcal{T}_1$ is completely parameterized), and the probability distribution over distinct subsets of size 3 from $N \setminus N_1 = N$ is just the uniform distribution over such subsets.

3. STEP II: CONSTRUCTING A VERIFICATION ALGORITHM

In Section 2 we focused on the non-strategic aspects of the facility location game: agents report their locations, which are mapped by a mechanism to a facility location. The potential for strategic behavior stems from the assumption that the agents' locations $\mathbf{x}$ are private information — x_i represents agent i's *ideal* location for the facility (also known as agent i's *peak*). Like Procaccia and Tennenholtz [27], and almost all subsequent papers, we assume that the *cost* of agent i for facility location y is simply the Euclidean distance between (the true) x_i and y,

$$\text{cost}(x_i, y) = |x_i - y|.$$

3.1 Deterministic Mechanisms

A deterministic mechanism $\mathcal{M} : \mathbb{R}^n \to \mathbb{R}$ is *truthful* if for every location profile $\mathbf{x} \in \mathbb{R}^n$, every agent $k \in N$, and every $x'_k \in \mathbb{R}$, $\text{cost}(x_k, \mathcal{M}(\mathbf{x})) \leq \text{cost}(x_k, \mathcal{M}(x'_k, \mathbf{x}_{-k})$, where $\mathbf{x}_{-k} = \langle x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n \rangle$. Our next goal is to construct an algorithm that receives as input a deterministic mechanism, represented as a decision tree, and verifies that it is truthful.

The verification algorithm is quite intuitive, although its formal specification is somewhat elaborate. Consider a mechanism $\mathcal{M} : \mathbb{R}^n \to \mathbb{R}$ that is represented by a tree $\mathcal{T}$. For a leaf $\mathcal{L}$, denote the location chosen by $\mathcal{M}$ at this leaf by $y_{\mathcal{L}}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_{\mathcal{L},i} \cdot x_i$. In addition, let $\mathcal{C}(\mathcal{L})$ denote the set of constraints encountered on the path to $\mathcal{L}$. For example, the set of constraints corresponding to the leftmost leaf in Figure 2 is $\{(x_1 \geq x_2), (x_2 \geq x_3)\}$, while the second leaf from the left verifies: $\{(x_1 \geq x_2), (x_2 < x_3), (x_1 \geq x_3)\}$. We define a procedure, Build-Leaf-Constraints, that gathers these constraints (Algorithm 2). One subtlety is that the procedure "inflates" strict inequality constraints to constraints that require a difference of at least 1; we will explain shortly why this is without loss of generality.

The main procedure, Truthful (given as Algorithm 1), checks whether there exist location profiles $\mathbf{x}$ and $\mathbf{x}'$ that differ only in the k'th coordinate, such that $\mathbf{x}$ reaches leaf $\mathcal{L}$ (based on the constraints of the Build-Leaf-Constraints procedure, given as Algorithm 3), $\mathbf{x}'$ reaches leaf $\mathcal{L}'$, and

$$\text{cost}(x_k, y_{\mathcal{L}'}(\mathbf{x}')) + 1 \leq \text{cost}(x_k, y_{\mathcal{L}}(\mathbf{x}))$$

That is, the reduction in cost is at least 1.

So why can we "inflate" strict inequalities by requiring a difference of 1? Assume that we are given a mechanism $\mathcal{T}$ and an agent i such that for some strategy profiles $\mathbf{x}$ and $\mathbf{x}'$ with $\mathbf{x}_{-i} = \mathbf{x}'_{-i}$, agent i can strictly benefit by switching from $\mathbf{x}$ to $\mathbf{x}'$. Then there exists $\epsilon > 0$ such that agent i's improvement is at least ϵ, and for every strict inequality satisfied by $\mathbf{x}$ and $\mathbf{x}'$, the difference between the terms is at least ϵ; for example, if $x_k > x_l$, then it is the case that $x_k - x_l \geq \epsilon$. Since each facility location is a homogeneous linear function of the input $\mathbf{x}$, all variables can be multiplied by $\frac{1}{\epsilon}$ to obtain that $\mathbf{x}/\epsilon$ and $\mathbf{x}'/\epsilon$ satisfy the more stringent constraints (with a difference of 1) on agent locations and facility locations.

Algorithm 1: Truthful($\mathcal{T}$) // *Verifier for Deterministic Mechanisms*

```
Data: mechanism T
Result: true if T represents a truthful mechanism,
        false otherwise
Build-Leaf-Constraints(T)
foreach k ∈ N do
    foreach leaf L ∈ T do
        // y_L(x) is the symbolic expression for the
        facility at L on x and d_k(x) is agent k's
        distance from the facility
        foreach d_k(x) ∈ {x_k − y_L(x), −x_k + y_L(x)}
        do
            // two cases, for x_k to the left or right of the
            facility y_L(x)
            foreach leaf L' ∈ T do
                foreach
                d'_k(x') ∈ {x'_k − y_L'(x'), −x'_k + y_L'(x')}
                do
                    inc(x, x') ← {(d_k(x) − d'_k(x') ≥ 1),
                                  d_k(x) ≥ 0, d'_k(x') ≥ 0}
                    // utility increase from x to x',
                    distances are non-negative
                    if Exists-Solution(k, C_L, C_L', inc)
                    then
                        return False
return True
```

Finally, this algorithm works in polynomial time because the procedure Exists-Solution, which checks whether there is a solution to the different constraints (corresponding to a profitable manipulation), just solves a linear program using the procedure Solve.

We summarize the preceding discussion with the theorem:

Theorem 1. *Let $N = \{1, \ldots, n\}$. The truthfulness of a deterministic mechanism $\mathcal{M}$ represented as a decision tree $\mathcal{T}$ can be verified in polynomial time in n and $|\mathcal{T}|$.*

Algorithm 1 essentially carries out a brute force search over pairs of leaves to find a profitable manipulation. Under the decision tree representation, is it possible to verify truthfulness much more efficiently? Our next result answers this question in the negative.

Theorem 2. *Let $N = \{1, \ldots, n\}$ with $n \geq 2$, and $\ell \leq n!$. Then any algorithm that verifies truthfulness for every deterministic decision tree with ℓ leaves for n agents must inspect all the leaves in the worst case.*

Proof. Assume by contradiction there exists a verification algorithm that can check truthfulness for every tree

Algorithm 2: BUILD-LEAF-CONSTRAINTS($\mathcal{T}$)

Data: mechanism $\mathcal{T}$
Result: set of symbolic constraints $\mathcal{C}$; the location at leaf $\mathcal{L}$ is selected on input $\mathbf{x} \iff$ constraints $\mathcal{C}_{\mathcal{L}}(\mathbf{x})$ hold

$\mathcal{C} \leftarrow \emptyset$ // *Initialize the set of constraints*
foreach *leaf* $\mathcal{L} \in \mathcal{T}$ **do**
  $Q \leftarrow \mathcal{L}$
  while $Q \neq Null$ **do**
    // *Add the constraint that must hold for* Q *to be reached from* $parent(Q)$
    $c \leftarrow constraint(parent(Q).Next() = Q)$
    switch c **do**
      case $x_{i_c} \geq x_{j_c}$
        $\mathcal{C}_{\mathcal{L}}(\mathbf{x}) \leftarrow \mathcal{C}_{\mathcal{L}}(\mathbf{x}) \cup \{x_{i_c} - x_{j_c} \geq 0\}$
      case $x_{i_c} > x_{j_c}$
        $\mathcal{C}_{\mathcal{L}}(\mathbf{x}) \leftarrow \mathcal{C}_{\mathcal{L}}(\mathbf{x}) \cup \{x_{i_c} - x_{j_c} \geq 1\}$
      case $x_{i_c} \leq x_{j_c}$
        $\mathcal{C}_{\mathcal{L}}(\mathbf{x}) \leftarrow \mathcal{C}_{\mathcal{L}}(\mathbf{x}) \cup \{x_{j_c} - x_{i_c} \geq 0\}$
      case $x_{i_c} < x_{j_c}$
        $\mathcal{C}_{\mathcal{L}}(\mathbf{x}) \leftarrow \mathcal{C}_{\mathcal{L}}(\mathbf{x}) \cup \{x_{j_c} - x_{i_c} \geq 1\}$
    $Q \leftarrow parent(Q)$
return $\mathcal{C}$

Algorithm 3: EXISTS-SOLUTION($k, \mathcal{C}_{\mathcal{L}}, \mathcal{C}'_{\mathcal{L}'}, inc$)

Data: agent k and symbolic constraint sets $\mathcal{C}_{\mathcal{L}}, \mathcal{C}'_{\mathcal{L}'}$, inc
Result: $true \iff \exists\, x_1, \ldots, x_n, x'_k \in \mathbb{R}^+$ *subject to* $\mathcal{C}_{\mathcal{L}}(\mathbf{x})$ & $\mathcal{C}'_{\mathcal{L}}(x'_k, \mathbf{x}_{-k})$ & $inc(\mathbf{x}, (x'_k, \mathbf{x}_{-k}))$

$\mathbf{x}' \leftarrow (x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n)$
$W \leftarrow \{\mathcal{C}_{\mathcal{L}}(\mathbf{x}), \mathcal{C}'_{\mathcal{L}}(\mathbf{x}'), inc(\mathbf{x}, \mathbf{x}')\}$
$\mathbf{z} \leftarrow (x_1, \ldots, x_n, x'_i)$
return SOLVE($\mathbf{z}, W, \mathbf{z} \geq 0$) // *Linear program solver*

with ℓ leaves without inspecting all the leaves. Let $\mathcal{T}$ be a decision tree in which every internal node has the form $x_i < x_j$, for $i, j \in N$ such that $i < j$, and the location is set to x_1 in every leaf. Since there are $n!$ possible orders of the agent locations, we can generate such a tree with ℓ leaves. Clearly, $\mathcal{T}$ is truthful since it coincides with the mechanism in which agent 1 is a dictator.

Consider the execution of the verification algorithm on input $\mathcal{T}$ and let $\mathcal{L}$ be a leaf that is not inspected by the algorithm. Construct a tree $\mathcal{T}'$ that is identical to $\mathcal{T}$, with the exception of leaf $\mathcal{L}$, where the selected location is $y_{\mathcal{L}}(\mathbf{x}) = \frac{x_1 + \ldots x_n}{n}$. First note that mechanism $\mathcal{T}'$ is not truthful. For every leaf of $\mathcal{T}'$, the mechanism cannot enforce that two variables are equal, since that would require comparing $x_i < x_j$ and $x_j < x_i$ (similarly if weak inequalities are used). However, if $i < j$ then $\mathcal{T}'$ can only check if $x_i < x_j$; similarly, if $j < i$, then $\mathcal{T}'$ can only check if $x_j < x_i$. Thus the leaf $\mathcal{L}$ can be reached when the input $\mathbf{x}$ is consistent with some strict ordering π on n elements.

Define $\mathbf{x} \in \mathbb{R}^n$ such that $x_{\pi_1} < x_{\pi_2} < \ldots < x_{\pi_n}$. Then $y_{\mathcal{L}}(\mathbf{x}) = \frac{x_1 + \ldots x_n}{n}$ and the cost of agent π_n is $\text{cost}(x_{\pi_n}, y_{\mathcal{L}}(\mathbf{x})) = x_{\pi_n} - y_{\mathcal{L}}(\mathbf{x})$. There exists $\delta > 0$ such that by reporting $x'_{\pi_n} = x_{\pi_n} + \delta$, agent π_n ensures that leaf $\mathcal{L}$ is still reached and the new cost is lower:

$$\begin{aligned} \text{cost}(x_{\pi_n}, y_{\mathcal{L}}(x'_{\pi_n}, \mathbf{x}_{-\pi_n})) &= x_{\pi_n} - \frac{\left(\sum_{i \neq \pi_n} x_i\right) + (x_{\pi_n} + \delta)}{n} \\ &< x_{\pi_n} - \frac{x_1 + \ldots + x_n}{n} \\ &= \text{cost}(x_{\pi_n}, y_{\mathcal{L}}(\mathbf{x})). \end{aligned}$$

However, since the verification algorithm does not inspect leaf $\mathcal{L}$, it cannot distinguish between $\mathcal{T}$ and $\mathcal{T}'$, and so it decides that $\mathcal{T}'$ is also truthful. This contradicts the correctness of the verification algorithm. □

Crucially, our decision trees are binary trees, so the number of leaves is exactly the number of internal nodes plus one. Theorem 2 therefore implies:

COROLLARY 1. *Let $N = \{1, \ldots, n\}$, $n \geq 2$. Any verification algorithm requires superpolynomial time in n (in the worst-case) to verify the truthfulness of trees of superpolynomial size in n.*

3.2 Randomized Mechanisms

In the context of randomized mechanisms, there are two common options for defining truthfulness: *truthfulness in expectation* and *universal truthfulness*. In our context, truthfulness in expectation means that an agent cannot decrease its expected distance to the facility by deviating; universal truthfulness means that the randomized mechanism is a probability distribution over truthful deterministic mechanisms, i.e., an agent cannot benefit from manipulation regardless of the mechanism's random coin tosses. Clearly, the former notion of truthfulness is weaker than the latter. In some settings, truthful-in-expectation mechanisms are known to achieve guarantees that cannot be obtained through universally truthful mechanisms [11].

We focus on universal truthfulness, in part because we do not know whether truthful-in-expectation mechanisms can be efficiently verified (as we discuss in §5). Using Theorem 1, universal truthfulness is easy to verify, because it is sufficient and necessary to verify the truthfulness of each of the decision trees in the mechanism's support. One subtlety is the binding of agents in $N \setminus N_r$ to the $z_{r,j}$ parameters. However, for the purpose of verifying truthfulness, any binding will do by symmetry between the agents in $N \setminus N_r$. We therefore have the following result:

THEOREM 3. *Let $N = \{1, \ldots, n\}$. The universal truthfulness of a randomized mechanism $\mathcal{M}$ represented as a distribution over K decision trees $\mathcal{T}_1, \ldots, \mathcal{T}_K$ can be verified in polynomial time in n and its representation size, $\sum_{r=1}^{K} |\mathcal{T}_r|$.*

4. STEP III: MEASURING THE QUALITY OF VERIFIABLY TRUTHFUL MECHANISMS

We have shown that the truthfulness of mechanisms represented by decision trees of polynomial size can be verified in polynomial time. This result is encouraging, but it is only truly meaningful if decision trees of polynomial size can describe mechanisms that provide good guarantees with respect to the quality of the solution.

Like Procaccia and Tennenholtz [27], and subsequent papers, we measure solution quality in the facility location domain via two measures. The *social cost* of a facility location

$y \in \mathcal{R}$ for a location profile $\mathbf{x} \in \mathbb{R}^n$ is

$$\mathrm{sc}(\mathbf{x}, y) = \sum_{i=1}^{n} \mathrm{cost}(x_i, y),$$

and the *maximum cost* is

$$\mathrm{mc}(\mathbf{x}, y) = \max_{i \in N} \mathrm{cost}(x_i, y).$$

We denote the optimal solutions with respect to the social cost and maximum cost by $\mathrm{sc}^*(\mathbf{x}) = \min_{y \in \mathbb{R}} \sum_{i=1}^{n} \mathrm{cost}(x_i, y)$, and $\mathrm{mc}^*(\mathbf{x}) = \min_{y \in \mathbb{R}} \max_{i \in N} \mathrm{cost}(x_i, y)$, respectively.

4.1 Deterministic Mechanisms

Let us first review what can be done with deterministic mechanisms represented by decision trees of arbitrary size, without necessarily worrying about verification.

For the maximum cost, the optimal solution is clearly the midpoint between the leftmost and rightmost reported locations. It is interesting to note that the midpoint may not be one of the agents' reported locations — so, to compute the optimal solution, our expressive representation of the leaves as convex combinations of points in $\mathbf{x}$ is required. Procaccia and Tennenholtz [27] have shown that any truthful mechanism cannot achieve an approximation ratio smaller than 2 for the maximum cost. A ratio of 2 is achieved by any solution that places the facility between the leftmost and rightmost reported locations. It follows that the optimal ratio is trivial to obtain truthfully, e.g., by always selecting the location x_1 reported by agent 1. This mechanism is representable via a tiny decision tree with one leaf.

We conclude that, in the context of deterministic mechanisms and the maximum cost objective, truthful mechanisms that are efficiently verifiable can do just as well as any truthful mechanism.

Let us therefore focus on the social cost. For any number of agents n, it is easy to see that selecting the median of the reported locations is the optimal solution. Indeed, if the facility moves right or left, the facility would get further away from a majority of locations, and closer to a minority of locations. The median mechanism was observed by Moulin [22] to be truthful. Intuitively, this is because the only way an agent can manipulate the median's location is by reporting a location that is on the other side of the median — but that only pushes the median away from the agent's actual location. Moreover, the median can be computed by a decision tree in which every internal node contains comparisons between the input locations, and each leaf $\mathcal{L}$ outputs the location of the facility (the median) when $\mathcal{L}$ is reached.

In contrast to the maximum cost, though, the optimal mechanism for the social cost — the median — requires a huge decision tree representation. The number of comparisons required to compute the median has been formally studied (see, e.g., Blum et al. [4]), but, in our case, simple intuition suffices: if there is an odd number of agents with distinct locations, the median cannot be determined when nothing is known about the location of one of the agents, so $(n-1)/2$ comparisons are required *in the best case*, leading to a tall binary tree of exponential size.

Our next result strengthens this insight by giving a lower bound on the approximation ratio achievable by polynomial size decision trees (i.e., trees efficiently verifiable by our algorithm of §3).

THEOREM 4. *For every constant $k \in \mathbb{N}$, every truthful deterministic decision tree for n agents of size at most n^k has an approximation ratio of $\Omega\left(\frac{n}{\log n}\right)$ for the social cost.*

PROOF. Let $\mathcal{M}$ be a deterministic mechanism represented by some decision tree $\mathcal{T}$ of size at most n^k. Recall that every internal node in $\mathcal{T}$ checks the order of two input variables with one of the following inequalities: $\{x_i \geq x_j,\ x_i \leq x_j,\ x_i < x_j,\ x_i > x_j\}$.

Since $\mathcal{T}$ is binary and $|\mathcal{T}| \leq n^k$, there exists at least one leaf $\mathcal{L} \in \mathcal{T}$ of depth

$$d < 2 \cdot \log(|\mathcal{T}|) \leq 2\log(n^k) = 2k\log(n).$$

Let $S_{\mathcal{L}} = \{i_1, \ldots, i_m\}$ be the set of agents whose locations are inspected on the path to $\mathcal{L}$. It holds that $|S_{\mathcal{L}}| = m \leq 2 \cdot d \leq 4k \cdot \log(n)$, since $\mathcal{L}$ has depth d and each node on the path to $\mathcal{L}$ inspects two locations. Note that if $S_{\mathcal{L}} = \emptyset$, then $\mathcal{M}$ is a dictatorship, and so its approximation ratio is no better than $n-1$. Thus we can assume that $S_{\mathcal{L}} \neq \emptyset$.

Recall that the facility at $\mathcal{L}$ is a convex combination of the input locations; that is, $y_{\mathcal{L}}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_{\mathcal{L},i} \cdot x_i$, where $\lambda_{\mathcal{L},i} \in [0,1], \forall i \in N$ and $\sum_{i=1}^{n} \lambda_{\mathcal{L},i} = 1$. Let π be a weak ordering consistent with the leaf $\mathcal{L}$ and $D_{\mathcal{L}} = \{i_1, \ldots, i_l\}$ a "deduplicated" version of $S_{\mathcal{L}}$, such that $D_{\mathcal{L}}$ contains one representative agent i for each maximal subset $W \subseteq S_{\mathcal{L}}$ with the property that under π, $x_j = x_i$, $\forall j \in W$. Note that $D_{\mathcal{L}}$ is consistent with some strict ordering σ on l elements.

We distinguish among three cases:

Case 1: The facility at $\mathcal{L}$ is a convex combination of agents in $S_{\mathcal{L}}$ only (i.e., $\lambda_{\mathcal{L},i} = 0$, $\forall i \notin S_{\mathcal{L}}$).

Let ϵ be fixed such that $0 < \epsilon < \frac{|S_{\mathcal{L}}|}{n}$ and define the following input $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$:

- For each $i \in D_{\mathcal{L}}$, let r_i be the number of agents in $D_{\mathcal{L}}$ strictly to the left of i according to σ; set $x_i \leftarrow \epsilon \cdot \left(\frac{r_i}{n}\right)$.
- For each $j \in S_{\mathcal{L}} \setminus D_{\mathcal{L}}$, set $x_j \leftarrow x_i$, where $i \in D_{\mathcal{L}}$ and $x_i = x_j$ according to π.
- For each $j \notin S_{\mathcal{L}}$, set $x_j \leftarrow 1$.

The optimal location of the facility given $\mathbf{x}$ is $y^* = 1$, since most agents are situated at 1 (except the agents in $S_{\mathcal{L}}$, of which there are at most: $4k\log(n) \ll n/2$). The optimal social cost is:

$$\mathrm{sc}^*(\mathbf{x}) = \sum_{i=1}^{n} \mathrm{cost}(x_i, y^*) = \sum_{i \in S_{\mathcal{L}}} (1 - x_i) \leq 1 \cdot |S_{\mathcal{L}}|.$$

On the other hand, the output of the mechanism is $y_{\mathcal{L}}(\mathbf{x}) = \sum_{i \in S_{\mathcal{L}}} \lambda_{\mathcal{L},i} \cdot x_i \leq \epsilon$; the social cost incurred by $\mathcal{M}$ on $\mathbf{x}$ is:

$$\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x})) = \sum_{i=1}^{n} \mathrm{cost}(x_i, y_{\mathcal{L}}(\mathbf{x})) \geq (n - |S_{\mathcal{L}}|) \cdot (1 - \epsilon).$$

Choosing $\epsilon \leq 1/n$, the approximation ratio of $\mathcal{M}$ is no better than:

$$\begin{aligned} \frac{\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x}))}{\mathrm{sc}^*(\mathbf{x})} &\geq \frac{(n - |S_{\mathcal{L}}|) \cdot (1 - \epsilon)}{|S_{\mathcal{L}}|} \\ &= \frac{n}{|S_{\mathcal{L}}|} - \frac{n\epsilon}{|S_{\mathcal{L}}|} - 1 + \epsilon \\ &> \frac{n}{4k\log(n)} - 2 \in \Omega\left(\frac{n}{\log(n)}\right). \end{aligned}$$

Case 2: The facility coincides with the location of some agent $t \notin S_{\mathcal{L}}$ (i.e. $y_{\mathcal{L}}(\mathbf{x}) = x_t$).

Similarly to Case 1, let ϵ be fixed such that $0 < \epsilon < \frac{|S_{\mathcal{L}}|+1}{n}$ and define $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ as follows:

- For each $i \in D_{\mathcal{L}}$, let r_i be the number of agents in $D_{\mathcal{L}}$ strictly to the left of i according to σ; set $x_i \leftarrow \epsilon \cdot \left(\frac{r_i}{n}\right)$.
- For each $j \in S_{\mathcal{L}} \setminus D_{\mathcal{L}}$, set $x_j \leftarrow x_i$, where $i \in D_{\mathcal{L}}$ and $x_i = x_j$ according to π.
- Set $x_t = 0$.
- For each $j \notin S_{\mathcal{L}}, j \neq t$, set $x_j \leftarrow 1$.

The optimal location on $\mathbf{x}$ is $y^* = 1$, since most agents are located at 1 (except agent t and the agents in $S_{\mathcal{L}}$). As in Case 1, by also taking agent t into account, we get:

$$\frac{\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x}))}{\mathrm{sc}^*(\mathbf{x})} \geq \frac{(n - |S_{\mathcal{L}}| - 1) \cdot (1 - \epsilon)}{|S_{\mathcal{L}}| + 1} \in \Omega\left(\frac{n}{\log(n)}\right).$$

Case 3: The facility is a weighted sum with at least two terms, one of which is an agent $t \notin S_{\mathcal{L}}$. We claim that no mechanism that is truthful on the full domain (i.e. the line) can have such an output at any leaf. Let $\epsilon, \delta > 0$ be such that

$$\delta = \frac{1}{2}\left(\frac{1}{\lambda_{\mathcal{L},t}} - 1\right) \text{ and } \epsilon = \frac{1 - \lambda_{\mathcal{L},t}(1+\delta)}{n-1}.$$

Consider an input $\mathbf{x}$ consistent with the ordering π such that $x_t = 1$ and $x_i \in (0, \epsilon), \forall i \neq t$. Then:

$$y_{\mathcal{L}}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_{\mathcal{L},i} \cdot x_i = \left(\sum_{i \neq t} \lambda_{\mathcal{L},i} \cdot x_i\right) + \lambda_{\mathcal{L},t} \cdot 1.$$

If agent t reports instead $x'_t = 1 + \delta$, the output of $\mathcal{M}$ on $\mathbf{x}' = (x'_t, \mathbf{x}_{-t})$ is:

$$y_{\mathcal{L}}(\mathbf{x}') = \left(\sum_{i \neq t} \lambda_{\mathcal{L},i} \cdot x_i\right) + \lambda_{\mathcal{L},t} \cdot (1 + \delta).$$

It can be verified that $0 < y_{\mathcal{L}}(\mathbf{x}) < y_{\mathcal{L}}(\mathbf{x}') < 1$, and so $\mathrm{cost}(x_t, y_{\mathcal{L}}(\mathbf{x}')) < \mathrm{cost}(x_t, y_{\mathcal{L}}(\mathbf{x}))$, which contradicts the truthfulness of $\mathcal{M}$. Thus Case 3 never occurs.

By the cases above, there exists at least one input on which the approximation ratio of $\mathcal{M}$ is $\Omega\left(\frac{n}{\log(n)}\right)$, which completes the proof. □

On the positive side, we show that the lower bound of Theorem 4 is asymptotically tight.

THEOREM 5. *For every $n \in \mathbb{N}$ there is a truthful deterministic decision tree of size $O(n^6)$ that approximates the social cost within a factor of $O\left(\frac{n}{\log(n)}\right)$.*

PROOF. First, we claim that for every $k \in \{1, \ldots, n/2\}$, there exists a truthful, deterministic decision tree of size $O(2^{6k})$ that approximates the social cost within a factor of $O\left(\frac{n-k}{k}\right)$. Given a fixed k, let $\mathcal{M}$ be the mechanism:

- Given $\mathbf{x} = (x_1, \ldots, x_n)$, output median$(\{x_1, \ldots, x_k\})$.

That is, $\mathcal{M}$ always outputs the median of the fixed set of agents $\{1, \ldots, k\}$. Computing the median on an input vector of size k requires fewer than $6k$ comparisons [4], and since the decision tree for $\mathcal{M}$ is binary, its size is $O(2^{6k})$.

We next claim that the approximation ratio of $\mathcal{M}$ is $O\left(\frac{n-k}{k}\right)$. Indeed, given any instance $\mathbf{x} \in \mathbb{R}^n$, denote $\tilde{m} = \mathcal{M}(\mathbf{x})$ and $m^* = \mathrm{argmin}_{y \in \mathbb{R}} \mathrm{sc}(\mathbf{x}, y)$. Without loss of generality, assume that $\tilde{m} < m^*$ and let $\Delta = |\tilde{m} - m^*|$. Let $S_l = \{x_i \mid x_i \leq \tilde{m}\}$, $S_r = \{x_i \mid x_i \geq m^*\}$, and $S_m = \{x_i \mid \tilde{m} < x_i < m^*\}$ be the sets of points to the left of $\tilde{m}$, to the right of m^*, and strictly between $\tilde{m}$ and m^*, respectively. Denote the sizes of the sets by $n_l = |S_l|$, $n_r = |S_r|$, and $n_m = |S_m|$, where $n_l + n_m + n_r = n$.

We compute the upper bound by comparing the social cost of $\mathcal{M}$ on $\mathbf{x}$, $\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x})) = \sum_{i=1}^{n} \mathrm{cost}(x_i, \tilde{m})$, with $\mathrm{sc}^*(\mathbf{x}) = \sum_{i=1}^{n} \mathrm{cost}(x_i, m^*)$. Observe that for all the points in S_r, the cost increases by exactly Δ when moving the location from m^* to $\tilde{m}$. On the other hand, the change from m^* to $\tilde{m}$ results in a decrease by exactly Δ for the points in S_l. Thus $\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x}))$ can be expressed as follows:

$$\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x})) = \mathrm{sc}^*(\mathbf{x}) + n_r \cdot \Delta + \sum_{j \in S_m} [\mathrm{cost}(x_j, \tilde{m}) - \mathrm{cost}(x_j, m^*)] - \Delta n_l$$

The ratio of the costs is:

$$\begin{aligned} \frac{\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x}))}{\mathrm{sc}^*(\mathbf{x})} &= \frac{\mathrm{sc}^*(\mathbf{x}) + n_r \cdot \Delta}{\mathrm{sc}^*(\mathbf{x})} + \\ &+ \frac{\sum_{j \in S_m} [\mathrm{cost}(x_j, \tilde{m}) - \mathrm{cost}(x_j, m^*)] - n_l \cdot \Delta}{\mathrm{sc}^*(\mathbf{x})}. \end{aligned}$$

We claim that

$$\frac{\mathrm{sc}(\mathbf{x}, \mathcal{M}(\mathbf{x}))}{\mathrm{sc}^*(\mathbf{x})} \leq \frac{3(n-k)}{k}. \tag{1}$$

Inequality (1) is equivalent to:

$$k \cdot n_r \cdot \Delta + k \cdot \sum_{j \in S_m} [\mathrm{cost}(x_j, \tilde{m}) - \mathrm{cost}(x_j, m^*)] - k \cdot n_l \cdot \Delta \leq (3n - 4k)\mathrm{sc}^*(\mathbf{x})$$

Note that for all $j \in S_m$, $\mathrm{cost}(x_j, \tilde{m}) - \mathrm{cost}(x_j, m^*) \leq \Delta$, and so if Inequality (1) holds when $\mathrm{cost}(x_j, \tilde{m}) - \mathrm{cost}(x_j, m^*) = \Delta$, then it also holds for all other instances where the change in cost is smaller for some agents $j \in S_m$. Formally, if:

$$k \cdot n_r \cdot \Delta + k \cdot n_m \cdot \Delta - k \cdot n_l \cdot \Delta \leq (3n - 4k)\mathrm{sc}^*(\mathbf{x}), \tag{2}$$

then Inequality (1) holds. Inequality (2) is equivalent to:

$$\begin{aligned} \mathrm{sc}^*(\mathbf{x}) &\geq \frac{k \cdot n_r \cdot \Delta + k \cdot n_m \cdot \Delta - k \cdot n_l \cdot \Delta}{3n - 4k} \\ &= \frac{k \cdot (n_r + (n - n_l - n_r) - n_l) \cdot \Delta}{3n - 4k} \\ &= \frac{k \cdot (n - 2n_l) \cdot \Delta}{3n - 4k}. \end{aligned} \tag{3}$$

Each of the agents in S_l pays a cost of at least Δ under $\mathrm{sc}^*(\mathbf{x})$, and so $\mathrm{sc}^*(\mathbf{x}) \geq n_l \cdot \Delta$. Moreover, since $\tilde{m}$ is the median of $\{x_1, \ldots, x_k\}$, it follows that $n_l \geq \frac{k}{2}$. We first show that $n_l \cdot \Delta \geq \frac{k \cdot (n-2n_l) \cdot \Delta}{3n-4k}$:

$$\begin{aligned} & n_l \cdot \Delta \geq \frac{k \cdot (n - 2n_l) \cdot \Delta}{3n - 4k} \\ \iff & n_l(3n - 4k) \geq k(n - 2n_l) \\ \iff & n_l(3n - 2k) \geq kn \\ \iff & n_l \geq \frac{kn}{3n - 2k} \end{aligned} \tag{4}$$

In addition, we have that

$$\frac{k}{2} \geq \frac{kn}{3n - 2k} \iff 3kn - 2k^2 \geq 2kn \iff n \geq 2k. \tag{5}$$

Inequality (5) holds by the choice of k; combining it with $n_l \geq \frac{k}{2}$, we obtain:

$$n_l \geq \frac{k}{2} \geq \frac{kn}{3n-2k}. \quad (6)$$

By Inequality (3), it follows that $n_l \cdot \Delta \geq \frac{k\cdot(n-2n_l)\cdot\Delta}{3n-4k}$. In addition, $\mathrm{sc}^*(\mathbf{x}) \geq n_l \cdot \Delta$, thus:

$$\mathrm{sc}^*(\mathbf{x}) \geq n_l \cdot \Delta \geq \frac{k\cdot(n-2n_l)\cdot\Delta}{3n-4k}.$$

Equivalently, Inequality (2) holds, which gives the worst case bound required for Inequality (1) to always hold. Thus $\frac{\mathrm{sc}(\mathbf{x},\mathcal{M}(\mathbf{x}))}{\mathrm{sc}^*(\mathbf{x})} \leq \frac{3(n-k)}{k}$, for every input $\mathbf{x}$.

Let $k = \log n$. Then $\mathcal{M}$ can be implemented using a decision tree of size $O(n^6)$ and has an approximation ratio bounded by

$$\frac{\mathrm{sc}(\mathbf{x},\mathcal{M}(\mathbf{x}))}{\mathrm{sc}^*(\mathbf{x})} \leq \frac{3(n-k)}{k} \in O\left(\frac{n}{\log(n)}\right)$$

This completes the proof of the theorem. □

In summary, polynomial-size decision trees can achieve the best possible approximation ratio (among all truthful deterministic mechanisms) with respect to the maximum cost objective and an approximation ratio of $\Theta(n/\log n)$ with respect to the social cost.

4.2 Randomized Mechanisms

We next turn to randomized mechanisms. In this context, we are interested in the expected social cost, or the expected maximum cost. The latter measure is somewhat subtle, so let us state specifically that, like Procaccia and Tennenholtz [27], we are interested in

$$\mathbb{E}\Big[\mathrm{mc}(\mathbf{x},\mathcal{M}(\mathbf{x}))\Big] = \mathbb{E}\left[\max_{i\in N} \mathrm{cost}(x_i,\mathcal{M}(\mathbf{x}))\right].$$

A less stringent alternative would be to take the maximum over agents of the agent's expected cost.

It is immediately apparent that universally truthful, randomized, small decision trees can easily beat the lower bound of Theorem 4 for social cost. To see this, consider the random dictator mechanism, that selects an agent $i \in N$ uniformly at random, and returns the location x_i. This mechanism is clearly universally truthful (it is a uniform distribution over dictatorships), and it is easy to verify that its approximation ratio is $2 - 2/n$.

Our next theorem, which we view as the main result of this section, shows that randomization allows us to get arbitrarily close to 1 using universally truthful, efficiently-verifiable mechanisms.

THEOREM 6. *For every $0 < \epsilon < \frac{1}{10}$ and $n \in \mathbb{N}$, there exists a universally truthful randomized decision tree of size $O(poly(n))$, approximates the social cost to a factor of $1+\epsilon$.*

PROOF. The idea is the following: we sample a subset of agents of logarithmic size – more exactly $O\left(\frac{\ln(n/\epsilon)}{\epsilon^2}\right)$ – and select the median among their reported locations. To reason about this mechanism, we define the rank of an element x in a set S ordered by $\succ$ to be $\mathrm{rank}(x) = |\{y \in S \mid y \succ x \vee y = x\}|$, and the *$\epsilon$-median* of S to be $x \in S$ such that $(1/2-\epsilon)|S| < \mathrm{rank}(x) < (1/2+\epsilon)|S|$. The following lemma is a folklore result when sampling is done with replacement; we include its proof because we must sample without replacement.

LEMMA 1. *Consider the algorithm that samples t elements without replacement from a set S of size n, and returns the median of the sampled points. For all $\epsilon, \delta < 1/10$, if*

$$\frac{100\ln\frac{1}{\delta}}{\epsilon^2} \leq t \leq \epsilon n,$$

then the algorithm returns an ϵ-median with probability $1-\delta$.

PROOF. We partition S into three subsets:

$$S_1 = \{x \in S \mid \mathrm{rank}(x) \leq n/2 - \epsilon n\},$$

$$S_2 = \{x \in S \mid n/2 - \epsilon n < \mathrm{rank}(x) < n/2 + \epsilon n\},$$

and

$$S_3 = \{x \in S \mid \mathrm{rank}(x) \geq n/2 - \epsilon n\}.$$

Suppose that t elements are sampled without replacement from S. If less than $t/2$ are sampled from S_1, and less than $t/2$ are sampled from S_3, then the median of the sampled elements will belong to S_2 — implying that it is an ϵ-approximate median.

Let us, therefore, focus on the probability of sampling *at least* $t/2$ samples from S_1. Define a Bernoulli random variable X_i for all $i = 1,\ldots,t$, which takes the value 1 if and only if the i'th sample is in S_1.

Note that $X_1,\ldots,X_t$ are not independent (because we are sampling with replacement), but for all i it holds that

$$\begin{aligned}\Pr\left[X_i = 1 \mid X_1 = x_1,\ldots,X_{i-1} = x_{i-1}\right] &\leq \frac{\frac{n}{2}-\epsilon n}{n-(i-1)} \\ &\leq \frac{\frac{n}{2}-\epsilon n}{n-\epsilon n} \leq \frac{1}{2}-\frac{\epsilon}{3}\end{aligned}$$

for any $(x_1,\ldots,x_{i-1}) \in \{0,1\}^{i-1}$, where the second inequality follows from $i \leq t \leq \epsilon n$.

Let $Y_1,\ldots,Y_t$ be i.i.d. Bernoulli random variables such that $Y_i = 1$ with probability $1/2 - \epsilon/3$. Then for all x,

$$\Pr\left[\sum_{i=1}^{t} X_i \geq x\right] \leq \Pr\left[\sum_{i=1}^{t} Y_i \geq x\right].$$

Using Chernoff's inequality, we conclude that

$$\begin{aligned}\Pr\left[\sum_{i=1}^{t} X_i \geq \frac{t}{2}\right] &\leq \Pr\left[\sum_{i=1}^{t} Y_i \geq \frac{t}{2}\right] \\ &= \Pr\left[\sum_{i=1}^{t} Y_i \geq \left(1+\frac{\epsilon}{\frac{3}{2}-\epsilon}\right)\mathbb{E}\left[\sum_{i=1}^{t} Y_i\right]\right] \\ &\leq \Pr\left[\sum_{i=1}^{t} Y_i \geq \left(1+\frac{\epsilon}{2}\right)\mathbb{E}\left[\sum_{i=1}^{t} Y_i\right]\right] \\ &\leq \exp\left(-\frac{\left(\frac{\epsilon}{2}\right)^2\left(\frac{1}{2}-\frac{\epsilon}{3}\right)t}{3}\right) \leq \frac{\delta}{2}\end{aligned}$$

The last inequality follows from the assumption that $t \geq \frac{100\ln(1/\delta)}{\epsilon^2}$. The proof of the lemma is completed by applying symmetric arguments to S_3, and using the union bound. □

Let $\mathbf{x} = \langle x_1,\ldots,x_n\rangle$ be the set of inputs. For every $k \in N$, define the mechanism $\mathcal{M}_{n,k}$ as follows:

- Select uniformly at random a subset $S_k \subseteq N$, where $|S_k| = k$.

- Output the median of S_k.

Note that $\mathcal{M}_{n,1}$ coincides with random dictator, while $\mathcal{M}_{n,n}$ is the median mechanism. Recall that random dictator, $\mathcal{M}_{n,1}$, has an approximation ratio of $2 - 2/n$ for the social cost, while the median, $\mathcal{M}_{n,n}$, is optimal. The approximation ratio of $\mathcal{M}_{n,k}$ improves as k grows from 1 to n and the mechanism is universally truthful for every k; in particular, we show there exists a choice of k that achieves a good tradeoff between the size of the mechanism and its approximation ratio.

First, we describe the implementation of $\mathcal{M}_{n,k}$ as a randomized decision tree. The root has outgoing degree one and selects a function $\mathcal{F}$ that takes k arguments $Z = \{z_1, \ldots, z_k\}$ and computes the median of $z_1, \ldots, z_k$. At execution time, $z_1, \ldots, z_k$ are instantiated using the locations $x_{i_1}, \ldots, x_{i_k}$ of k distinct agents, chosen uniformly at random from k-subsets of N. Note that $\mathcal{F}$ can be implemented with a decision tree of size $O(2^{6k})$.

Let $\epsilon', \delta > 0$ be fixed such that $\epsilon', \delta < \frac{1}{10}$. By Lemma 1, the algorithm that samples without replacement $t = \lceil \frac{100 \ln \frac{1}{\delta}}{(\epsilon')^2} \rceil$ elements from a set of n elements returns an ϵ'-median with probability $1 - \delta$, as long as $t \leq \epsilon' n$.

Let $\mathbf{x} \in \mathbb{R}^n$; without loss of generality $x_1 \leq \cdots \leq x_n$. We wish to compare $\mathbb{E}[\mathrm{sc}(\mathbf{x}, \mathcal{M}_{n,t}(\mathbf{x}))]$ and $\mathrm{sc}^*(\mathbf{x})$. Let us suppose that $\mathcal{M}_{n,t}$ returns an ϵ'-median, call it x_l. Since x_l is an ϵ'-median, we have that $\frac{n}{2} - \epsilon' n < l < \frac{n}{2} + \epsilon' n$. Take the case where $l < \frac{n}{2}$ (the other case, where $l > \frac{n}{2}$, is similar) and let $\Delta = |x_l - x_m|$, where $x_m = \mathrm{median}(\mathbf{x})$. Then by moving the facility from x_m to x_l, the costs of the agents change as follows:

(i) Each agent to the left of x_l (including agent l) has the cost decreased by exactly Δ.

(ii) Each agent strictly between x_l and x_m incurs an increase in cost of at most Δ.

(iii) Each agent to the right of x_m (including agent m) has the cost increased by Δ.

It follows that

$$\mathrm{sc}(\mathbf{x}, x_l) \leq \mathrm{sc}^*(\mathbf{x}) - l \cdot \Delta + (n - l) \cdot \Delta = \mathrm{sc}^*(\mathbf{x}) + (n - 2l) \cdot \Delta.$$

On those instances where $\mathcal{M}_{n,t}$ does not return the median, the social cost is at most $(n-1) \cdot \mathrm{diam}(\mathbf{x})$, where $\mathrm{diam}(\mathbf{x}) = \max_{i,j \in N} |x_i - x_j|$. On the other hand, the optimal cost satisfies: $\mathrm{sc}^*(\mathbf{x}) \geq \mathrm{diam}(\mathbf{x})$ and $\mathrm{sc}^*(\mathbf{x}) \geq l \cdot \Delta$.

Since $\mathcal{M}_{n,t}$ returns an ϵ'-median with probability $1 - \delta$, the ratio of the costs can be bounded by:

$$\begin{aligned}
\frac{sc_{\mathcal{M}_{n,t}}(\mathbf{x})}{\mathrm{sc}^*(\mathbf{x})} &\leq \frac{(1-\delta)\mathrm{sc}^*(\mathbf{x}) + \Delta(1-\delta)(n-2l) + \delta(n-1)\mathrm{diam}(\mathbf{x})}{\mathrm{sc}^*(\mathbf{x})} \\
&\leq (1-\delta) + \frac{\Delta(1-\delta)(n-2l)}{\Delta \cdot l} + \frac{\delta(n-1)\mathrm{diam}(\mathbf{x})}{\mathrm{diam}(\mathbf{x})} \\
&= 1 - \delta + (1-\delta)\frac{n}{l} - 2(1-\delta) + \delta(n-1) \\
&\leq \delta \cdot n - 1 + (1-\delta)\frac{2}{1-2\epsilon'} \leq 1 + \delta \cdot n + 5\epsilon'.
\end{aligned}$$

Given $\epsilon < 1/10$, let $\epsilon' = \epsilon/10$ and $\delta = \epsilon/(2n)$, and set $t = \lceil \frac{100 \ln \frac{1}{\delta}}{(\epsilon')^2} \rceil$. Then $\mathcal{M}_{n,t}$ can be represented as a randomized decision tree of size $O(2^{6t})$, which is polynomial in n. Moreover, for this choice of ϵ', δ, the approximation ratio of $\mathcal{M}_{n,t}$ is bounded by

$$1 + \delta \cdot n + 5\epsilon' = 1 = \frac{\epsilon}{2} + \frac{\epsilon}{2} = 1 + \epsilon.$$

In stark contrast, universal truthfulness does not help obtain a better bound than the trivial approximation ratio of 2 for the maximum cost — even in the case of general mechanisms. The proof is included in the full version of the paper.

THEOREM 7. *For each $\epsilon > 0$, there exists no universally truthful mechanism given by a distribution over countably many deterministic mechanisms that can approximate the maximum cost within a factor of $2 - \epsilon$.*

We have the following corollary for universally truthful decision trees.

COROLLARY 2. *For each $\epsilon > 0$, there exists no universally truthful decision tree mechanism given by a distribution over countably many deterministic decision trees that can approximate the maximum cost within a factor of $2 - \epsilon$.*

5. DISCUSSION

Theorem 7 shows that universally truthful decision trees cannot achieve a nontrivial (better than 2) approximation for the maximum cost. In contrast, Procaccia and Tennenholtz [27] designed a *truthful-in-expectation* mechanism that approximates the maximum cost to a factor of 3/2. This motivates the study of truthful-in-expectation randomized decision trees, as an alternative to universal truthfulness. However, we do not know whether truthfulness in expectation can be efficiently verified (and we believe that it cannot). Intuitively, the main difficulty is that, for every selection of one leaf from each tree in the support of the randomized mechanism, a naïve verification algorithm would need to reason about whether a certain location profile $\mathbf{x}$ can reach this collection of leaves under the constraints imposed by the different trees.

Our work focuses on the case of locating one facility on the line, which is quite simple from the approximate-mechanism-design-without-money viewpoint. Researchers have investigated approximate mechanism design in generalized facility location settings, involving multiple facilities [27, 21, 20, 25, 13, 14, 15], different cost functions [32, 15], metric spaces and graphs [1, 20], and so on. Of these generalizations and extensions, all but one only require a rethinking of our results of §4 — that is, mechanisms can still be represented as polynomial-size decision trees. But moving from the real line to a more general metric space requires a revision of the way mechanisms are represented in our framework.

We conclude by re-emphasizing the main message of our paper. In our view, our main contribution is the three-step approach to the design of verifiably truthful mechanisms. Our technical results provide a proof of concept by instantiating this approach in the context of a well-studied facility location setting, and constructing verifiably truthful mechanisms that achieve good quality guarantees. We firmly believe, though, that the same approach is widely applicable. For example, is there a class of mechanisms for combinatorial auctions that gives rise to verifiably truthful mechanisms providing a good approximation to social welfare? One can ask similar questions in the context of every problem studied

in algorithmic mechanism design (with or without money). More generally, how should economic systems be designed so that players can reason efficiently about their decisions?

6. ACKNOWLEDGMENTS

We would like to thank Aris Filos-Ratsikas for a helpful discussion on characterizations of mechanisms for single peaked preferences and Joan Feigenbaum, Kevin Leyton-Brown, Peter Bro Miltersen, and Tuomas Sandholm for useful feedback.

Simina Brânzei acknowledges support from the Danish National Research Foundation and the National Science Foundation of China (under the grant 61361136003) for the Sino-Danish Center for the Theory of Interactive Computation and from the Center for Research in Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council. Simina also acknowledges support from an IBM Ph.D. fellowship.

Ariel D. Procaccia was partially supported by the NSF under grants CCF-1215883 and IIS-1350598.

7. REFERENCES

[1] N. Alon, M. Feldman, A. D. Procaccia, and M. Tennenholtz. Strategyproof approximation of the minimax on networks. *Mathematics of Operations Research*, 35(3):513–526, 2010.

[2] N. Alon, F. Fischer, A. D. Procaccia, and M. Tennenholtz. Sum of us: Strategyproof selection from the selectors. In *TARK*, pages 101–110, 2011.

[3] I. Ashlagi, F. Fischer, I. Kash, and A. D. Procaccia. Mix and match. *Game. Econ. Behav.*, 2014. Forthcoming.

[4] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, 1973.

[5] R. H. Bordini, M. Fisher, W. Visser, and M. Wooldridge. Verifying multi-agent programs by model checking. *JAAMAS*, 12:239–256, 2006.

[6] I. Caragiannis, E. Elkind, M. Szegedy, and L. Yu. Mechanism design: from partial to probabilistic verification. In *EC*, pages 266–283, 2012.

[7] I. Caragiannis, A. Filos-Ratsikas, and A. D. Procaccia. An improved 2-agent kidney exchange mechanism. In *WINE*, pages 37–48, 2011.

[8] Y. Cheng, W. Yu, and G. Zhang. Strategy-proof approximation mechanisms for an obnoxious facility game on networks. *Theoretical Computer Science*, 497:154–163, 2013.

[9] R. Cole, V. Gkatzelis, and G. Goel. Mechanism design for fair division: Allocating divisible items without payments. In *EC*, pages 251–268, 2013.

[10] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *UAI*, pages 103–110, 2002.

[11] S. Dobzinski and S. Dughmi. On the power of randomization in algorithmic mechanism design. *SIAM Journal on Computing*, 42(6):2287–2304, 2013.

[12] S. Dughmi and A. Ghosh. Truthful assignment without money. In *EC*, pages 325–334, 2010.

[13] D. Fotakis and C. Tzamos. Winner-imposing strategyproof mechanisms for multiple facility location games. In *WINE*, pages 234–245, 2010.

[14] D. Fotakis and C. Tzamos. On the power of deterministic mechanisms for facility location games. In *ICALP*, pages 449–460, 2013.

[15] D. Fotakis and C. Tzamos. Strategyproof facility location for concave cost functions. In *EC*, pages 435–452, 2013.

[16] M. Guo and V. Conitzer. Strategy-proof allocation of multiple items between two agents without payments or priors. In *AAMAS*, pages 881–888, 2010.

[17] M. Guo, V. Conitzer, and D. Reeves. Competitive repeated allocation without payments. In *WINE*, pages 244–255, 2009.

[18] Laura Kang and David C. Parkes. Passive verification of the strategyproofness of mechanisms in open environments. In *ICEC*, 2006.

[19] E. Koutsoupias. Scheduling without payments. In *SAGT*, pages 143–153, 2011.

[20] P. Lu, X. Sun, Y. Wang, and Z. A. Zhu. Asymptotically optimal strategy-proof mechanisms for two-facility games. In *EC*, pages 315–324, 2010.

[21] P. Lu, Y. Wang, and Y. Zhou. Tighter bounds for facility games. In *WINE*, pages 137–148, 2009.

[22] H. Moulin. On strategy-proofness and single-peakedness. *Public Choice*, 35:437–455, 1980.

[23] Ahuva Mu'alem. A note on testing truthfulness. *ECCC, Report No. 130*, 2005.

[24] N. Nisan and A. Ronen. Algorithmic mechanism design. *Game. Econ. Behav.*, 35(1–2):166–196, 2001.

[25] K. Nissim, R. Smorodinsky, and M. Tennenholtz. Approximately optimal mechanism design via differential privacy. In *ITCS*, pages 203–213, 2012.

[26] M. Pauly and M. Wooldridge. Logic for mechanism design—a manifesto. In *GTDT*, 2003.

[27] A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. *ACM Transactions on Economics and Computation*, 2013. Forthcoming; preliminary version in EC'09.

[28] E. M. Tadjouddine, F. Guerin, and W. Vasconcelos. Abstracting and verifying strategy-proofness for auction mechanisms. In *DALT*, pages 197–214, 2009.

[29] N. K. Thang. On (group) strategy-proof mechanisms without payment for facility location games. In *WINE*, pages 531–538, 2010.

[30] T. Todo, A. Iwasaki, and M. Yokoo. False-name-proof mechanism design without money. In *AAMAS*, pages 651–658, 2011.

[31] W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *J. Financ.*, 16(1):8–37, 1961.

[32] Y. Wilf and M. Feldman. Strategyproof facility location and the least squares objective. In *EC*, pages 873–890, 2013.