What to Verify for Optimal Truthful Mechanisms without Money

Diodato Ferraioli[*]
University of Salerno
dferraioli@unisa.it

Paolo Serafino
Teesside University
p.serafino@tees.ac.uk

Carmine Ventre[†]
Teesside University
c.ventre@tees.ac.uk

ABSTRACT

We aim at identifying a minimal set of conditions under which algorithms with good approximation guarantees are truthful without money. In line with recent literature, we wish to express such a set via verification assumptions, i.e., kind of agents' misbehavior that can be made impossible by the designer.

We initiate this research endeavour for the paradigmatic problem in approximate mechanism design without money, facility location. It is known how truthfulness imposes (even severe) losses and how certain notions of verification are unhelpful in this setting; one is thus left powerless to solve this problem satisfactorily in presence of selfish agents. We here address this issue and characterize the minimal set of verification assumptions needed for the truthfulness of optimal algorithms, for both social cost and max cost objective functions. En route, we give a host of novel conceptual and technical contributions ranging from topological notions of verification to a lower bounding technique for truthful mechanisms that connects methods to *test* truthfulness (i.e., cycle monotonicity) with *approximation guarantee*.

1. INTRODUCTION

How good an approximate solution can a truthful (or strategyproof (SP)) mechanism return for the optimization problem at hand? This question is an important line of investigation in mechanism design with monetary transfers [12, 1, 2] and without [15, 6, 23]. For the latter class of mechanisms, more appropriate to digital settings where there is no currency readily available, the tradeoff between incentives and approximation is at the heart of approximate mechanism design without money research agenda [21]. The paradigmatic problem in this area is K-facility location: n selfish agents are located on the real line; we want to place K facilities on input the n bids of the agents for their locations on the line. Each agent's objective is to minimize their connection cost, defined as the distance between their true location and the nearest facility. The designer's objective is to minimize the social cost (i.e., the sum of the connection costs of all the agents). It is known that the best deterministic SP mechanism can only return an $(n-2)$-approximation of the optimum, even for $K = 2$ [21, 6]. (A different version of the problem looks at the minimization of the maximum cost – the bound for $K = 2$ is, in this case, constant.)

This research leaves little hope to the mechanism designer facing this problem in presence of selfish agents. In fact, the designer cannot use computation time as a way out since practically all (few exceptions are known in the setting with money, e.g., [2]) the lower bounds to the approximation of SP mechanisms hold unconditionally, i.e., independently from the running time of algorithms. The designer could use randomization but with scarce results. For one, truthful in expectation mechanisms are vulnerable to different risk attitude of agents (indeed, SP is guaranteed only as long as agents are risk neutral). Secondly, mechanisms are suboptimal – a constant upper bound is known for $K = 2$ [15].

In this work, we want to propose a way forward to the mechanism designer in despair. On one hand, we want to focus on deterministic mechanisms, so as to avoid to make assumptions about agents' attitude to uncertainty. On the other hand, we wish to provide a small (ideally, minimum) set of conditions under which algorithms with "good" approximation guarantees are SP. This would inform the designer about the (minimum) investment in resources (e.g., policies, infrastructures, legislation) needed to prevent the lies that make the algorithms of interest not SP. The idea to restrict the way agents lie is well established in economics [10, 9] and computer science [20, 24, 13, 11, 3, 4, 7]. Therein, this assumption is dubbed *verification* to express that certain lies can be somehow verified and made impossible. Our novelty is the aim of "minimizing" the verification assumptions needed for the truthfulness of "good" algorithms, rather than showing that a particular verification leads to the truthfulness of a specific (class of) algorithm(s).

Our contribution

We initiate this line of enquiry and individuate a minimal set of verification assumptions for which optimum algorithms for K-facility location are SP.

The assumptions needed for a truthful optimum differ accordingly to the objective function of interest, social cost or maximum cost. The first ingredient, common to both scenarios, is a 'return to the origin' in verification literature. As in [17] the actual cost of the agents are bound to an insincere declaration (specifically, an agent overreporting their cost ends up paying this augmented cost), so here a lying agent

[*] This author was supported by the "GNCS – INdAM".

[†] This author was supported by EPSRC, through grant EP/M018113/1.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

is forced to use the facility closest to her reported location (rather than closest to her actual location). This notion, named *cluster imposing* and first used in [19], generalizes winner imposing mechanisms considered by [5] and is easy to implement by defining, e.g., 'catchment areas' for facilities (much like, the system in place for public schooling in many countries). The second ingredient, common to social and maximum cost, is a *no-underbidding* assumption whereby agents cannot say to be closer to the cluster-imposed facility than they actually are. This concept rephrases the main assumption made in related literature (see [20, 13] and references therein) for problems like combinatorial auctions and scheduling; it can be readily imposed by the designer whenever it is possible to measure/prove the distance the agent covers to reach the facility (in which case, it is, in fact, possible to simulate longer trips but not shorter ones).

The third ingredient for the result on social cost is a conceptual novelty. Verification is commonly defined only in relation to (true/reported) costs. We here define a topological restriction for the access to facilities: agents located to the left (right) of the facility are not allowed to access it from the right (left). This assumption, called *direction imposing*, further restricts the way agents can misbehave as an agent with true location t cannot declare b whenever the algorithm locates the facility closest to b in between t and b. Direction imposing can be realized whenever it is feasible to implement a 'left/right door' infrastructure for the facilities. For instance, when facilities are routers relaying voice/data and the area code of source address cannot be spoofed, the direction (subnetwork) voice/data have been transmitted from can be checked.

The third ingredient for the result on max cost (which, incidentally, holds only for $K = 2$) falls again in the class of cost-only verification. We here need to also prevent agents from reporting to be farther from the cluster-imposed facility than they actually are. Together with no-underbidding, this gives rise to a *no-cost forging* verification that is adequate in settings in which expense proofs must be provided.

We prove that relaxing any of the assumptions above leads to suboptimal outcomes (even when the other notions are strengthened) for both objective functions, already for $K = 2$. This shows that our (set of) verification(s) is *necessary* in the sense that it individuates the incentive-compatibility (IC) constraints that make optimal algorithms vulnerable to misreports. Furthermore, our guarantee about minimal sets of assumptions is, in a sense, the best one can hope for in this setting. The difficulty here is about "weighing" an assumption in the set. If all were equally heavy then our results would actually prove that ours is a minimum set of assumptions, as we prove that relaxing either of those leads to suboptimal outcomes already for $K = 2$. However, one could also weigh an assumption with the number of IC constraints that a verification assumption removes (i.e., by how much an assumption restricts the possible declarations available to agents). From this perspective, though, it becomes very hard (if possible, at all) to give a compact, useful-to-the-mechanism-designer characterization of maximum (or, even, maximal in fact) set of IC constraints according to which algorithm f is SP. Firstly, any such characterization would need to list somehow IC constraints that f satisfies. Secondly, the mechanism designer has no way to distinguish feasible IC constraints from infeasible ones. For example, the no-underbidding verification would not be needed for f and a pair of declarations t, b whenever an agent positioned at t would not gain by underbidding the distance from the location in which f places the facility closest to b. But since the mechanism only knows b, there is no way the designer can avoid verifying the pair t, b (there might in fact be a location c for which c would indeed gain by underbidding the distance from the facility closest to b).

Discussion on verification

Our verification notions are *ex-post* as in all – [10] being an exception – aforementioned related literature, i.e., the actual outcome of the algorithm (the location of the facilities) is used to define restricted misbehavior. A different approach is *ex-ante* verification, where the set of restricted strategies is defined upon the type (location) of each agent. So, for example, in the ϵ-verification of [8] an agent with true location t can only declare locations in $[t - \epsilon, t + \epsilon]$ – this 'symmetric' verification is, however, ineffective as any truthful mechanism with verification is truthful without [8]. Our contribution can be cast in that framework as the study of (the "minimal") 'asymmetric' verification that truthfully implements optimum algorithms.

As discussed above, our assumptions are necessary: no optimal truthful algorithm for K-facility location exists without. When for the application of interest those assumptions cannot be implemented then one must content oneself with suboptimal solutions. Our results should then be read in the negative whenever our verification concepts cannot be enforced. Note, however, that in principle different definitions of verification could remove the exact same set of IC constraints we prove to break truthfulness. Nevertheless, the study of the best way to express those IC constraints depends on the setting at hand and is outside the scope of this work. (We stress that facility location is rather general and thus encodes many different real-life applications.)

More technical contributions

We believe that our results only scratch the surface as we conjecture that our assumptions are minimal not just for optimal algorithms but for all "simple" algorithms with constant approximation guarantee to the optimal social cost. By simple here we mean algorithms that only place the facilities at K of the locations declared by the agents. These algorithms are the most natural (e.g., no algorithm has better approximation guarantee) especially in the case of deterministic algorithms (cf. known upper bounds); our conjecture (if proved) suggests to look for 'unnatural' algorithms in order to get a good approximation truthfully.

We give some preliminary results towards settling this conjecture. Among the verification notions needed for a SP optimum, we drop direction-imposing (arguably, the most controversial and somewhat less practical of the concepts) and study the extent to which cost-only verification can be helpful in this context. We adopt the cycle-monotonicity technique to dig deeper into the structure of SP algorithms. This technique features a weighed graph encoding all the IC constraints. We begin by proving a surprising parallel between mechanisms with money and no verification, and mechanisms without money and no-underbidding verification (for any problem). A mechanism in the former category is SP iff all the cycles of the graph have non-negative weight [25]. We show that a mechanism in the latter category is SP iff all the cycles are comprised of edges weighing 0. We essentially complement this characterization by showing that there must be no 0-weight edges outside cycles for good approximations. Specifically, we prove that a class of truthful algorithms can have approximation better than roughly $0.29n$ if and only if they do not have 0-weight edges outside cycles of the IC graph, even if we equip the mechanism with cluster-imposing and no-cost-forging verification. This result showcases a promising and novel approach, being the first known connection between cycle monotonicity and approximation.

We complement this lower bound with a mechanism for $K = 2$, MedianFurthest, truthful with cluster-imposing no-underbidding verification and with approximation guarantee $0.75n$. We further observe that MedianFurthest can be seen as a composition of two "basic" algorithms (i.e., MedianLeftmost and MedianRightmost) and prove that no algorithm with better approximation guarantee exists unless more than two algorithms are composed.

2. MODEL AND PRELIMINARIES

In abstract, we have a set $\mathcal{O}$ of feasible solutions and n selfish agents, each of them having a *cost* (or *type*) $t_i \in D_i$, D_i being the *domain* of agent i. For $t_i \in D_i$, $t_i(X)$ is the cost paid by agent i to implement outcome $X \in \mathcal{O}$. The type t_i is *private knowledge* of agent i. A *mechanism* f takes in input the types *reported* by each agent, that is, the *bids* $\mathbf{b} = (b_1, \ldots, b_n)$, $b_i \in D_i$ being the type reported by agent i, and returns a feasible solution $f(\mathbf{b}) \in \mathcal{O}$. We interchangeably use below the term mechanism and algorithm.

Definition 1. *We say that f is a truthful mechanism if for any bidder i, $b_i \in D_i$ and $\mathbf{b}_{-i}$, the declarations of the bidders other than i, we have: $t_i(f(t_i, \mathbf{b}_{-i})) \le t_i(f(\mathbf{b}))$.*

In certain contexts, some $b_i \in D_i$ can be "forgotten" when defining truthfulness.

Definition 2. *A mechanism f with verification V defines a set of allowed lies $M_{f,V}(t_i, \mathbf{b}_{-i})$ for agent i of type t_i. Agent i can report b_i iff $b_i \in M_{f,V}(t_i, \mathbf{b}_{-i})$. If $b_i \notin M_{f,V}(t_i, \mathbf{b}_{-i})$ then i is* caught lying *and punished by f.*

We assume that being caught lying is a very undesirable behavior for the bidder (e.g., in such a case the bidder loses prestige and the possibility to participate in future mechanisms – for simplicity, we assume that in such a case the bidder will have to pay a fine of infinite value). This way truthfulness is satisfied directly when $b_i \notin M_{f,V}(t_i, \mathbf{b}_{-i})$.

Cycle monotonicity

We set up a weighted graph for each bidder i depending on f, D_i, verification paradigm V, and the declarations $\mathbf{b}_{-i}$. Nonexistence of negative-weight edges in this graph guarantees the truthfulness of f.

More formally, fix mechanism f, bidder i and declarations $\mathbf{b}_{-i}$. Let V denote the verification paradigm at hand. The *declaration graph* with verification V associated to f has a vertex for each possible declaration in the domain D_i and an arc between t_i and b_i in D_i whenever $b_i \in M_{f,V}(t_i, \mathbf{b}_{-i})$. The weight of the edge (t_i, b_i) is defined as $-t_i(f(t_i, \mathbf{b}_{-i})) + t_i(f(b_i, \mathbf{b}_{-i}))$ and thus encodes the loss that a bidder whose type is t_i incurs into by declaring b_i.

Proposition 1. *Each declaration graph with verification V associated to f does not have negative-weight edges iff f is a truthful mechanism with verification V.*

The proposition above is adapted from [22, 25] to the verification setting V as in [24]. A corollary of this proposition is the following algorithmic characterization of truthfulness.

Corollary 1. *Algorithm f is truthful with verification V iff for all $t_i, b_i \in D_i$ and $\mathbf{b}_{-i}$, $b_i \in M_{f,V}(t_i, \mathbf{b}_{-i})$ implies $t_i(f(t_i, \mathbf{b}_{-i})) \le t_i(f(b_i, \mathbf{b}_{-i}))$.*

K-facility location

In the K-facility location problem, the set of feasible solutions $\mathcal{O}$ is comprised of all the K-tuples of possible allocations of the facilities whilst the domain of each agent is the real line. For a given mechanism f and $t_i \in D_i$, $t_i(f(t_i, \mathbf{b}_{-i})) = |t_i - f_{t_i}(t_i, \mathbf{b}_{-i})|$, where $f_{t_i}(t_i, \mathbf{b}_{-i})$ denotes the location of the facility output by $f(t_i, \mathbf{b}_{-i})$ closer to location t_i (whenever, t_i is equidistant to two facilities, ties are broken arbitrarily). In other words, $t_i(f(t_i, \mathbf{b}_{-i}))$ denotes the distance between t_i and the location of $f_{t_i}(t_i, \mathbf{b}_{-i})$ also denoted $d(t_i, f_{t_i}(t_i, \mathbf{b}_{-i}))$ below.

We focus on mechanisms f^* optimizing either the *social cost*, i.e., $f^*(\mathbf{b}) \in \arg\min_{X \in \mathcal{O}} cost(X, \mathbf{b})$, $cost(X, \mathbf{b}) = \sum_{i=1}^{n} b_i(X)$ or the *max cost*, i.e., $f^*(\mathbf{b}) \in \arg\min_{X \in \mathcal{O}} mc(X, \mathbf{b})$, $mc(X, \mathbf{b}) = \max_{i=1,\ldots,n} b_i(X)$. We say that f is α-approximate for either objective if it returns a solution a factor α away from the corresponding optimum.

Our verification assumptions

We are now ready to formally define the verification assumptions discussed in the introduction in relation to K-facility location.

We say that a mechanism f is *cluster-imposing* if for every i of type t_i, for all $\mathbf{b}_{-i}$ and for all $b_i \in M_{V,f}(t_i, \mathbf{b}_{-i})$, $t_i(f(b_i, \mathbf{b}_{-i})) = |t_i - f_{b_i}(b_i, \mathbf{b}_{-i})| = d(t_i, f_{b_i}(b_i, \mathbf{b}_{-i}))$, that is, the facility *assigned* to i is the one computed by $f(b_i, \mathbf{b}_{-i})$ that is closer to her declaration b_i.

For a mechanism f with *no-underbidding* (*no-overbidding*, resp.) verification, $t_i(f(\mathbf{b})) \le b_i(f(\mathbf{b}))$ ($t_i(f(\mathbf{b})) \ge b_i(f(\mathbf{b}))$, resp.) for every i of type t_i, for all $\mathbf{b}_{-i}$ and for all $b_i \in M_{V,f}(t_i, \mathbf{b}_{-i})$, i.e., agent i of type t cannot underreport (overreport, resp.) her distance from $f_{b_i}(b_i, \mathbf{b}_{-i})$. A mechanism f with *no-cost-forging verification* is a mechanism with no-underbidding *and* no-overbidding verification, i.e., for all $b_i \in M_{V,f}(t_i, \mathbf{b}_{-i})$, $t_i(f(\mathbf{b})) = b_i(f(\mathbf{b}))$.

We say that a mechanism has *direction-imposing* verification if $t_i, b_i < f_{b_i}(b_i, \mathbf{b}_{-i})$ or $t_i, b_i > f_{b_i}(b_i, \mathbf{b}_{-i})$ or $b_i = f_{b_i}(b_i, \mathbf{b}_{-i})$ or $t_i = f_{b_i}(b_i, \mathbf{b}_{-i})$, that is, t_i and b_i are on the same side of $f_{b_i}(b_i, \mathbf{b}_{-i})$.

2.1 Strengthening Proposition 1

Given that, like for K-facility location, it may be difficult to work with the algorithmic characterization of Corollary 1, we next give a more detailed graph-theoretic characterization of truthfulness with no-underbidding verification. Such a characterization holds not only for the facility location problem, but for any general setting that uses this notion of verification (and thus applies to all the aforementioned papers on ex-post verification).

Theorem 1. *A mechanism f is truthful with no-underbidding verification iff in each declaration graph associated* *to f the cycles are comprised of 0-weight edges while the edges not belonging to any cycle have non-negative weight.*

Proof. One direction easily follows from Proposition 1.

For the opposite direction, fix i and $\mathbf{b}_{-i}$ and consider a cycle $C = t_i^0 \to \cdots \to t_i^k = t_i^0$ in the declaration graph with no-underbidding verification associated to f. By definition, the weight of the cycle is

$$\sum_{j=0}^{k-1} -t_i^j(f(t_i^j, \mathbf{b}_{-i})) + t_i^j(f(t_i^{j+1}, \mathbf{b}_{-i})).$$

The existence of edge (t_i^j, t_i^{j+1}) yields

$$t_i^j(f(t_i^{j+1}, \mathbf{b}_{-i})) \le t_i^{j+1}(f(t_i^{j+1}, \mathbf{b}_{-i})).$$

Since f is truthful, then $t_i^j(f(t_i^j, \mathbf{b}_{-i})) \le t_i^j(f(t_i^{j+1}, \mathbf{b}_{-i}))$, for all j. Summing these inequalities, we have

$$t_i^j(f(t_i^j, \mathbf{b}_{-i})) = t_i^j(f(t_i^{j+1}, \mathbf{b}_{-i}))$$

for all j thus proving the theorem. □

3. SOCIAL COST

We call f^* the optimal algorithm for K-facility location that uses a *fixed tie-breaking rule*, i.e., for every i, $\mathbf{b}_{-i}$ and $t_i, b_i \in D_i$, if

$$cost(f^*(\mathbf{t}), \mathbf{b}) = cost(f^*(\mathbf{b}), \mathbf{b}) \tag{1}$$

and

$$cost(f^*(\mathbf{t}), \mathbf{t}) = cost(f^*(\mathbf{b}), \mathbf{t}), \tag{2}$$

then $f^*(\mathbf{b}) = f^*(t_i, \mathbf{b}_{-i})$. (Handling ties consistently is a rather standard assumption in mechanism design, see, e.g., [14].) It is easy to check that an optimal algorithm with fixed tie-breaking always exists. For example, the rule that chooses the lexicographically minimal allocation among all optimal allocations satisfies the property above.

Theorem 2. *f^* is a truthful mechanism with cluster-imposing, no-underbidding and direction-imposing verification.*

Proof. Suppose, by contradiction, that there is an agent i of type t_i, a declaration $b_i \neq t_i$ and $\mathbf{b}_{-i}$ such that

$$d(t_i, f^*_{t_i}) > d(t_i, f^*_{b_i}), \tag{3}$$

where $f^*_{t_i} = f^*_{t_i}(\mathbf{t})$ and $f^*_{b_i} = f^*_{b_i}(\mathbf{b})$, with $\mathbf{t} = (t_i, \mathbf{b}_{-i})$. Since the mechanism has no-underbidding verification

$$d(t_i, f^*_{b_i}) \le d(b_i, f^*_{b_i}). \tag{4}$$

By direction-imposing verification instead we have that,

$$\begin{array}{ll} t_i > f^*_{b_i} \Rightarrow b_i \ge f^*_{b_i}; & t_i < f^*_{b_i} \Rightarrow b_i \le f^*_{b_i}; \\ b_i > f^*_{b_i} \Rightarrow t_i \ge f^*_{b_i}; & b_i < f^*_{b_i} \Rightarrow t_i \le f^*_{b_i}. \end{array} \tag{5}$$

We then distinguish four possible cases.
Case 1. if $t_i \ge f^*_{t_i}$ and $t_i > f^*_{b_i}$ or $b_i \ge f^*_{b_i} = t_i$, then, from (3), it follows $t_i - f^*_{t_i} > t_i - f^*_{b_i}$ and thus $f^*_{t_i} < f^*_{b_i}$, and, from (5), we get $b_i \ge f^*_{b_i}$. The latter implies, along with (4), $t_i - f^*_{b_i} \le b_i - f^*_{b_i}$ and thus $t_i < b_i$. We conclude $f^*_{t_i} < f^*_{b_i} \le t_i < b_i$.
Case 2. if $t_i \le f^*_{t_i}$ and $t_i < f^*_{b_i}$ or $b_i \le f^*_{b_i} = t_i$, then, by the same arguments as above, we have that $b_i < t_i \le f^*_{b_i} < f^*_{t_i}$.
Case 3. if $t_i \ge f^*_{t_i}$ and $t_i < f^*_{b_i}$ or $b_i \le f^*_{b_i} = t_i$, then from (5), it follows that $b_i \le f^*_{b_i}$. The latter implies, along with (4), that $f^*_{b_i} - t_i \le f^*_{b_i} - b_i$ and thus $t_i > b_i$. Thus we have that $b_i < t_i \le f^*_{b_i}$, $f^*_{t_i} < t_i$ and $d(t_i, f^*_{t_i}) > d(t_i, f^*_{b_i})$.
Case 4. if $t_i \le f^*_{t_i}$ and $t_i > f^*_{b_i}$ or $b_i \ge f^*_{b_i} = t_i$, then, by the same arguments, we have $f^*_{b_i} \le t_i < b_i$, $f^*_{t_i} > t_i$ and $d(t_i, f^*_{t_i}) > d(t_i, f^*_{b_i})$.

We will show that these cases can never arise if the facilities are placed by f^*. Consider Case 1: since $b_i > t_i \ge f^*_{b_i}$, then $d(b_i, f^*_{b_i}) = d(t_i, f^*_{b_i}) + d(b_i, t_i)$. Hence,

$$\begin{aligned} cost(f^*(\mathbf{b}), \mathbf{b}) &= \sum_{j \neq i} b_j(f^*(\mathbf{b})) + d(t_i, f^*_{b_i}) + d(b_i, t_i) \\ &\ge \sum_{j \neq i} b_j(f^*(\mathbf{t})) + d(t_i, f^*_{t_i}) + d(b_i, t_i) \\ &= cost(f^*(\mathbf{t}), \mathbf{b}), \end{aligned}$$

where the inequality follows from $f^*(\mathbf{t})$ being the optimal facility location on input $\mathbf{t}$ and from the fact that $b_i > t_i > f^*_{t_i}$, so that $d(b_i, f^*_{t_i}) = d(t_i, f^*_{t_i}) + d(b_i, t_i)$. However, since $f^*(\mathbf{b})$ is optimal for $\mathbf{b}$ then $cost(f^*(\mathbf{b}), \mathbf{b}) \le cost(f^*(\mathbf{t}), \mathbf{b})$. Thus (1) holds. The same argument can be adopted for proving that (2) also holds. Hence, since f^* has a fixed tie-breaking, $f^*(\mathbf{b}) = f^*(\mathbf{t})$, contradicting the hypothesis that $f^*_{b_i} \neq f^*_{t_i}$. Observe that the Case 2 is symmetrical and thus the exact same arguments can be used.

Let us now consider Case 3. If $b_i \le f^*_{t_i} < t_i$, then $d(b_i, f^*_{t_i}) < d(b_i, f^*_{b_i})$. Hence, $cost(f^*(\mathbf{b}), \mathbf{b})$ equals

$$\begin{aligned} &\sum_{j \neq i} b_j(f^*(\mathbf{b})) + d(t_i, f^*_{b_i}) + d(b_i, f^*_{b_i}) - d(t_i, f^*_{b_i}) \\ &> \sum_{j \neq i} b_j(f^*(\mathbf{t})) + d(t_i, f^*_{t_i}) + d(b_i, f^*_{t_i}) - d(t_i, f^*_{t_i}) \end{aligned}$$

where the inequality uses that $f^*(\mathbf{t})$ is optimal for $\mathbf{t}$. Observe that the latter quantity is $cost(f^*(\mathbf{t}), \mathbf{b})$, and so we get a contradiction with the optimality of $f^*(\mathbf{b})$. If instead $f^*_{t_i} < b_i < t_i$, then $d(b_i, f^*_{t_i}) = d(t_i, f^*_{t_i}) - d(b_i, t_i)$. Moreover, $b_i < t_i \le f^*_{b_i}$ yields $d(b_i, f^*_{b_i}) = d(t_i, f^*_{b_i}) + d(b_i, t_i)$. Hence,

$$\begin{aligned} cost(f^*(\mathbf{b}), \mathbf{b}) &= \sum_{j \neq i} b_j(f^*(\mathbf{b})) + d(t_i, f^*_{b_i}) + d(b_i, t_i) \\ &> \sum_{j \neq i} b_j(f^*(\mathbf{t})) + d(t_i, f^*_{t_i}) - d(b_i, t_i) \\ &= cost(f^*(\mathbf{t}), \mathbf{b}), \end{aligned}$$

where the inequality uses optimality of $f^*(\mathbf{t})$ and $d(b_i, t_i) > 0$. However, this contradicts the optimality of f^*. Finally, Case 4 is symmetrical to Case 3 and the same arguments prove that also this case is impossible. □

We next show that it is not possible to prove Theorem 2 by relaxing some verification notions or its hypothesis.

Theorem 3. *The assumptions of Theorem 2 are necessary, even for $K = 2$.*

Proof. We begin by proving that if the mechanism is not cluster-imposing, then the optimal algorithm is not truthful even if the mechanism uses no-cost-forging and direction-imposing verification. Note that the definition of direction-imposing verification given above makes sense only if we assume that the mechanism is also cluster-imposing. However, we can still define some weaker (and somewhat less natural) forms of direction-imposing verification in its absence. We say that the mechanism has a *weak direction-imposing* verification if $t_i \ge f_{t_i}(b_i, \mathbf{b}_{-i})$ iff $b_i \ge f_{b_i}(b_i, \mathbf{b}_{-i})$

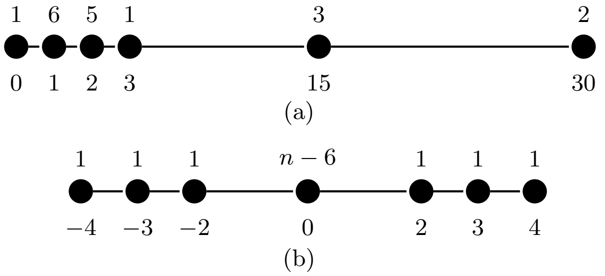

Figure 1: Instances used in the proof of Theorem 3

and $t_i \leq f_{t_i}(b_i, \mathbf{b}_{-i})$ iff $b_i \leq f_{b_i}(b_i, \mathbf{b}_{-i})$ (that is, both the real and the declared position of agent i are on the same side of their closest facilities); instead, we say that the mechanism has *ex-post direction-imposing* verification if $t_i \geq f_{t_i}(b_i, \mathbf{b}_{-i})$ iff $b_i \geq f_{t_i}(b_i, \mathbf{b}_{-i})$ and $t_i \leq f_{t_i}(b_i, \mathbf{b}_{-i})$ iff $b_i \leq f_{t_i}(b_i, \mathbf{b}_{-i})$ (i.e., both the real and the declared position of agent i are on the same side of the facility closest to t_i).

LEMMA 1. *Every optimal algorithm for K-facility location is not truthful, even if $K = 2$, the mechanism has no-cost-forging, weak direction-imposing and ex-post direction-imposing verification.*

PROOF. Consider the truthful instance described in Figure 1(a) where numbers below (above) the vertex represent its location (number of players on it). By inspection, the optimal algorithm places the facilities at 1 and 15. Let i be the agent at 3; her cost when she is truthful is 2. If she declares 31, then the optimal algorithm places a facility at 2 and the other at 30. Thus, she decreases her cost from 2 to 1, and is not caught lying by any kind of verification.

The instance can be generalized for any n, by having $\lceil (n-7)/2 \rceil$ in 1, $\lfloor (n-7)/2 \rfloor$ in 2 and moving the two leftmost positions from 15 and 30 to l and r, respectively where $r/2 + 2/3 > l > \min\{2r/5 + 4/5, (n+102)/10\}$. □

We now prove that if the mechanism has not the direction-imposing verification, then the optimum is not truthful even if the mechanism is cluster-imposing and has no-cost-forging verification. Consider the truthful instance described in Figure 1(b). The optimal algorithm places a facility in 0 and the second facility either in -3 or in 3. Suppose, w.l.o.g., that the optimum chooses -3 and consider agent i with truthful position 2. The cost of i when she truthfully declares her position is 2. If she declares location 4, then the optimal algorithm places a facility in 0 and the second in 3. Regardless from the mechanism being cluster-imposing or not, agent i is assigned to the facility in position 3, decreases her cost from 2 to 1, and is not caught lying by the no-cost-forging verification, since the distance from her real position and the facility is exactly the same as the distance from her declared position and the facility.

It is not too hard to use the instance in Figure 1(b) to prove the necessities of no-underbidding verification, and fixed tie-breaking rule. We omit the details. □

3.1 Simplicity and cost-only verification

Let us now focus on *simple algorithms* f, i.e., algorithms such that $f(\mathbf{b}) \subseteq \{b_1, \ldots, b_n\}$ for all $\mathbf{b}$. Moreover, let us consider mechanisms that do not use direction imposing, but only cluster-imposing, and no-underbidding or no-cost-forging verification. Next results highlight how hard it is to design a truthful deterministic mechanism with sub-linear approximation that uses only non-topological notions of verification.

A linear lower bound

We first prove a lower bound that holds for a large class of mechanisms even if we equip them with no-cost-forging verification along with cluster imposing.

DEFINITION 3. *A cluster-imposing algorithm f is* 0-edged *if for all $\mathbf{b}_{-i}$, the declaration graph with no-cost forging verification associated to f has a 0-edge that is not in a cycle.*

Roughly speaking, truthful 0-edged algorithms are not strictly truthful (i.e., truthtelling is not the only dominant strategy) for all the instances. These algorithms have a technical interest since they permit an "easy" connection between cycle monotonicity and approximation lower bounds (see Section 5 for obstacles and possible approaches to deal with more general classes of algorithms).

THEOREM 4. *No* 0*-edged simple algorithm f has approximation guarantee better than $2n/7$.*

PROOF. Let $\mathbf{b}_{-i}$ be the vector in which $\frac{5}{7}n - 1$ players are located at -1, $\frac{2}{7}n - 1$ at 0 and one at 1. Since f is simple, $f(\mathbf{b}) \subset \{-1, 0, 1, b_i\}$. The proof has two steps. We first show that no matter where player i is located, if f is 0-edged and is better than $2n/7$-approximate (absurdum hypothesis) then no facility can be placed at 0. We then observe that there is a bid b_i of player i for which $f(\mathbf{b})$ cannot return better than $2n/7$-approximate solutions – a contradiction.

<u>First step.</u> Since f is 0-edged then the declaration graph associated to f for the given $\mathbf{b}_{-i}$ must have a 0-weight edge that does not belong to a cycle. We next show this is not the case (i.e., the graph does not have such a 0-weight edge) when one of the facility is located at 0 and f is better than $2n/7$-approximate. Below, we drop $\mathbf{b}_{-i}$ from the notation.

For the declaration graph associated to f to have a 0-weight edge that is not part of a cycle, we need to provide two declarations of agent i, t_i and b_i, such that for some $\delta > 0$ one of the following must be true

$$f_{t_i}(t_i) = t_i - \delta < t_i < t_i + \delta = f_{b_i}(b_i) = b_i - \delta < b_i; \quad (6)$$

$$b_i < f_{b_i}(b_i) = b_i + \delta = t_i - \delta < t_i < f_{t_i}(t_i) = t_i + \delta, \quad (7)$$

while for $x, y \in \{t_i, b_i\}$, $x \neq y$, and $\delta > 0$, it must *not* be

$$x < f_x(x) = x + \delta = f_y(y) = y - \delta < y. \quad (8)$$

Indeed, since the graph has a 0-edge, then we need that there is an agent i, and two different declarations t_i and b_i, such that the following two properties occur: (i) agent i has the same cost when she declares her true type t_i and when she declares b_i, i.e., $t_i(f^*(\mathbf{b})) = t_i(f^*(\mathbf{t})) = d(t_i, f^*_{t_i})$ (let us denote with δ this value); (ii) agent i is not captured lying by the verification, i.e., by cluster-imposing verification, $t_i(f^*(\mathbf{b})) = d(t_i, f^*_{b_i})$, and, by no-cost-forging verification we require that $t_i(f^*(\mathbf{b})) = b_i(f^*(\mathbf{b})) = d(b_i, f^*_{b_i})$. Hence, we have that $d(t_i, f^*_{t_i}) = d(t_i, f^*_{b_i}) = d(b_i, f^*_{b_i}) = \delta$. There are only three ways of placing these four points (t_i, $f^*_{t_i}$, b_i, $f^*_{b_i}$) on a line so that they satisfy this condition, and they are exactly the ones described by equations (6)–(8). However, by definition, the 0-edge must not belong to a cycle. But in (8), if agent i, whose true type is b_i, declares t_i then she is not caught lying and therefore (b_i, t_i) is an edge of the graph. In particular, $t_i \to b_i \to t_i$ is a cycle. It is easy to check that this does not happen in (6) and (7).

We differentiate a number of cases according to the value of t_i and show that neither (6) nor (7) are possible without (8). Before, however, assume by contradiction that f is better than $2n/7$-approximate and note that $f(t_i, \mathbf{b}_{-i}) \subset \{-1, 0, t_i\}$ for each value of t_i.

Let us first consider the case in which $t_i \leq -1$ ($t_i > 0$, resp.). Since $f_{t_i}(t_i) \neq t_i$ for either (6) or (7) to be true, we have that f places the second facility at -1 and, consequently, that $f_{t_i}(t_i) = -1$ ($f_{t_i}(t_i) = 0$, resp.). Thus, we cannot have a situation like (6) ((7), resp.) since $f_{t_i}(t_i)$ is at the right (left, resp.) of t_i. We cannot have situation (7) ((6), resp.) either since when agent i goes to left (right, resp.) of t_i when declaring b_i there are not two locations on which $f_{b_i}(b_i)$ and b_i can be (recall that $f_{b_i}(b_i) \neq b_i$ and $1 \notin f(b_i, \mathbf{b}_{-i})$, since the approximation is better than $2n/7$).

We now deal with the case $t_i \in (-1, -1/2)$. Using the same argument as above we conclude that $f_{t_i}(t_i) = -1$. Here, we cannot have a situation like (7) given the relative order of $f_{t_i}(t_i)$ and t_i. Concerning situation (6), we note that when agent i goes to the right by declaring b_i the only possible locations for $f_{b_i}(b_i)$ are either 0 or 1: in both cases $d(t_i, f_{t_i}(t_i)) \neq d(t_i, f_{b_i}(b_i))$ as we would instead need.

Consider now the case $t_i = -1/2$. Here, t_i is equidistant from the locations of the two facilities (i.e., -1 and 0 – again, $f_{t_i}(t_i)$ must be different from t_i for otherwise no 0-weight edge (t_i, b_i) exists). We show that no matter the value of $f_{t_i}(t_i)$ if a 0-weight edge (t_i, b_i) exists according to either (6) or (7) then also (b_i, t_i) belongs to the graph; this shows the existence of a cycle, a contradiction with the fact that f is 0-edged. Specifically, if $f_{t_i}(t_i)$ is defined as -1 then $b_i = 1/2$ is a 0-weight edge as from (6) whilst if $f_{t_i}(t_i) = 0$ then $b_i = -3/2$ realizes the situation (7). However, in both scenarios the edge (b_i, t_i) also belongs to the graph.

Finally, take $t_i \in (-1/2, 0]$. Here, $f_{t_i}(t_i) = 0$ and we cannot have situation (6). As for (7), we note that when agent i goes to the left by declaring b_i the only possible location for $f_{b_i}(b_i)$ is -1, but then $d(t_i, f_{t_i}(t_i)) \neq d(t_i, f_{b_i}(b_i))$.
Second step. Let $b_i = 0$. By the argument above we know that $f(\mathbf{b})$ will locate one facility at -1 and the other at 1 for a cost of $\frac{2n}{7}$. The optimum, however, would only cost 1 by placing one facility at -1 and the other at 0. □

A linear upper bound

For $K = 2$, MedianFurthest locates one facility at the median location of the instance and the other at the furthest point from the median. Formally, given an instance $\mathbf{b}$, let b_M be the median location of $\mathbf{b}$. If $|\mathbf{b}|$ is even we take b_M to be the lower of the two middle values of $\mathbf{b}$. Let $\Delta_L = b_M - b_L$, where $b_L = \min_i b_i$, and $\Delta_R = b_R - b_M$, with $b_R = \max_i b_i$, be the distance of b_M from the leftmost and rightmost location of $\mathbf{b}$, respectively. Algorithm MedianFurthest on input $\mathbf{b}$ returns $\mathcal{F} = (b_M, b_L)$ if $\Delta_L > \Delta_R$, whereas it returns $\mathcal{F} = (b_M, b_R)$ if $\Delta_L \leq \Delta_R$. First we prove that this algorithm does not require a very demanding set of assumptions to be truthful (in particular, Theorem 2 does not apply since MedianFurthest is not optimal).

Theorem 5. MedianFurthest *is truthful with cluster-imposing and no-underbidding verification.*

Proof. Let us assume w.l.o.g. that the output of MedianFurthest on input $\mathbf{b}$ is $\mathcal{F} = (b_M, b_R)$ (the case when $\mathcal{F} = (b_M, b_L)$ is symmetric). Let i be the agent misreporting her location. It is easy to check that $t_i \notin \{b_M, b_R\}$, as in this case $d(\mathcal{F}, t_i) = 0$ and agent i cannot lower her cost any further. We will denote as b'_M and b'_R, respectively, the median and rightmost location of the instance $(b'_i, \mathbf{b}_{-i})$, and the output of MedianFurthest on such instance as $\mathcal{F}'$.

Let us first suppose that $t_i \in (b_M, b_R)$. If $b'_i < b_M$ and $b'_M - b'_i \leq b_R - b'_M$, then MedianFurthest would return allocation $\mathcal{F}' = (b'_M, b_R)$, with $b'_M < b_M$, and then $t_i(\mathcal{F}) \leq t_i(\mathcal{F}')$. If $b'_M - b'_i > b_R - b'_M$ the algorithm returns allocation $\mathcal{F}' = (b'_M, b'_i)$ and the misreport is detected by the verification step as $b'_i \in \mathcal{F}'$. If $b'_i > b_R$ then $\mathcal{F}' = (b_M, b'_i)$ and therefore $t_i(\mathcal{F}) \leq t_i(\mathcal{F}')$. Finally, if $b'_i \in (b_M, b_R)$, then the facility location does not change.

Let us now suppose that $t_i \in [b_L, b_M)$. Agent i can alter the output of the algorithm only if either (i) $b'_i > b_M$ or (ii) $b'_i < b_L$ and $b_M - b'_i > \Delta_R$ (note that in this case b_M does not change). If case (i) occurs, then $b'_i \geq b'_M > b_M$. Two sub-cases can occur: $b'_R - b'_M \geq b'_M - b_L$ and $b'_R - b'_M < b'_M - b_L$. If $b'_R - b'_M \geq b'_M - b_L$, then $\mathcal{F}' = (b'_M, b'_R)$ and $t_i(\mathcal{F}') > t_i(\mathcal{F})$ (in particular, if $b'_R \neq b_R$ it must be that $b'_R = b'_i$ and the misreport is detected by the verification step as $b'_i \in \mathcal{F}'$). If $b'_R - b'_M < b'_M - b_L$, then $\mathcal{F}' = (b'_M, b_L)$. We observe that since $b_L \leq b'_M \leq b'_i$, agent i must connect to the facility located at b'_M and since $b'_M \geq b_M > t_i$ then $d(t_i, b'_M) \geq d(b'_i, b_M)$. In case (ii) $b'_i < b_L$ and $b_M - b'_i > \Delta_R$, then $\mathcal{F}' = (b_M, b'_i)$. (Note that if $b_M - b'_i \leq \Delta_R$, MedianFurthest returns $\mathcal{F} = (b_M, b_R)$.) We note that in this case the verification step is capable of detecting the misreport by agent i, since $t_i(\mathcal{F}') = d(t_i, b'_i) > 0 = d(b'_i, b'_i) = b'_i(\mathcal{F}')$. □

This algorithm is not 0-edged (this is not too hard to see) yet it has approximation ratio $\frac{3}{4}n$. (The proof is only sketched due to lack of space.)

Theorem 6. MedianFurthest *is* $\frac{3}{4}n$*–approximate.*

Proof Idea. Let us denote by $\mathcal{F} = (b_M, b_R)$ the output of MedianFurthest on input $\mathbf{b}$ (i.e., we assume w.l.o.g. that $\Delta_R \geq \Delta_L$, the case $\Delta_R < \Delta_L$ being symmetric). Let $L_{\mathbf{b}} = \{i : b_i \leq b_M, M \neq i\}$ and $R_{\mathbf{b}} = \{i : b_i \geq b_M, M \neq i\}$ denote, respectively, the set of agents to the left and to the right of b_M. We can express the cost of allocation $\mathcal{F}$ as $cost(\mathcal{F}, \mathbf{b}) = cost(\mathcal{F}, L_{\mathbf{b}}) + cost(\mathcal{F}, R_{\mathbf{b}})$ where $cost(\mathcal{F}, L_{\mathbf{b}}) = \sum_{i \in L_{\mathbf{b}}} (b_M - b_i)$ denotes the cost incurred by the agents to the left of b_M whereas $cost(\mathcal{F}, R_{\mathbf{b}}) = \sum_{i \in R_{\mathbf{b}}} \min\{b_R - b_i, b_i - b_M\}$ denotes the cost incurred by the agents to the right of b_M. By definition of b_M it follows that $|L_{\mathbf{b}}| \leq \frac{n-1}{2}$ and $|R_{\mathbf{b}}| \leq \frac{n}{2}$. The maximum cost incurred by an agent in $L_{\mathbf{b}}$ is Δ_L, hence it follows that $cost(\mathcal{F}, L_{\mathbf{b}}) \leq \Delta_L \cdot \frac{n-1}{2}$. Likewise, the maximum cost incurred by an agent in $R_{\mathbf{b}}$ is $\frac{\Delta_R}{2}$, so $cost(\mathcal{F}, R_{\mathbf{b}}) \leq \frac{\Delta_R}{2} \cdot \frac{n}{2}$. Let us first notice that we need to consider instances where agents are located at more than 2 different locations, as otherwise MedianFurthest is optimal. Since, by hypothesis, $\Delta_R \geq \Delta_L$, two cases can occur: $\Delta_L = 0$ and $\Delta_L > 0$. We prove that in both cases the social cost of the facility location returned by MedianFurthest $\frac{3n}{4}$-approximates the optimal allocation. □

Composition of basic algorithms

For $1 \leq k < \ell \leq n$, a (k, ℓ)-algorithm f, in input $\mathbf{b}$, places the facilities at the k-th and ℓ-th smallest positions in $\mathbf{b}$. TwoExtremes [21] is a $(1, n)$-algorithm and has approximation $n-2$. MedianFurthest can be seen as a composition of two such basic algorithms: $(1, \lfloor n/2 \rfloor)$ and $(\lfloor n/2 \rfloor, n)$. Here, by *composition* we mean that the two basic algorithms are run and a test chooses which one of their outcomes is returned (e.g., the better). (Note that MedianFurthest does not choose the best of the outcomes computed by MedianLeftmost and MedianRightmost in terms of social cost.) This way of composing algorithms is widely considered in literature [16] and it is a quite natural and successful algorithmic techniques. For example, the optimal algorithm itself can be seen as a composition of a linear number of (k, ℓ)-algorithms. Hence, by composing sufficiently many of these basic algorithms we can achieve a good approximation of the social welfare. But can a ratio better than linear be achieved with two algorithms?

We next show that MedianFurthest is asymptotically optimal in this class no matter the test used to choose between the outcomes the two algorithms and the truthfulness of the composition. Let the two algorithms be (k, ℓ) and (k', ℓ'). We distinguish several cases. First assume that $\ell^+ = \max\{\ell, \ell'\} < n$; consider the instance wherein ℓ^+ agents are in position 0 and $n - \ell^+$ are in 1. Both algorithms place both facilities in 0 and, hence, the composition does to. The approximation ratio of the composition is then unbounded. Suppose now that $k^- = \min\{k, k'\} > 1$; consider the instance in which $k^- - 1$ agents are in 0 and $n + 1 - k^-$ in 1. Here the composition is unbounded as well. Consider now that $\ell^+ = n$ and $k^- = 1$. We set $\ell^- = \min\{\ell, \ell'\}$ and $k^+ = \max\{k, k'\}$. Let us first assume that $q = \max\{k^+, \ell^-\} \leq n/2$. Then consider the following instance: q agents in 0, $n - q - 1$ in 1, and 1 agent in 2. One algorithm will place a facility in 0 and the second in 2, whereas the other algorithm will place both facilities in 0. Hence, the composition has a cost of at least $\frac{n}{2} - 1$ while the optimum costs 1. The approximation ratio of the composition is then $\frac{n}{2} - 1$. If $p = \min\{k^+, \ell^-\} \geq n/2 + 1$, then similar arguments hold for the instance in which 1 agent is in 0, $p - 2$ in 1, and $n - p + 1$ in 2. We are left with the case that $p \leq n/2 < q$. We can distinguishing each of the possible realizations of k, k', ℓ, ℓ' satisfying these conditions and give the instance establishing the lower bound (in what follows we will assume w.l.o.g. that $k^- = k$).

Case $k = 1, \ell = p, k' = q, \ell' = n$: $n/2$ agents in 0 and $n/2$ in 1. No algorithm (and then their composition) places the facilities in two different locations. The approximation ratio of the composition is then unbounded.

Case $k = 1, \ell = q, k' = p, \ell' = n$: 1 agent in 0, $n/2 - 1$ in 1, $n/2 - 1$ in 2, and 1 in 3. Here, one algorithm locates the facilities in 0 and 2, whereas the other places them in 1 and 3. So, the composition costs $n/2$ while the optimum costs 2. The approximation ratio of the composition is then $n/4$.

Case $k = 1, \ell = n, k' = p, \ell' = q$: 1 agent in 0, $n - 2$ in $1/n$, and 1 in 1. Here, one algorithm locates the facilities in 0 and 1, whereas the other places them both in $1/n$. The composition costs at least $\frac{n-2}{n}$ while the optimum costs $1/n$.

It is not hard to see that the proof above can be adapted to work even if we assume that no algorithm places two facilities in the same position. In fact, we can assume that any set of $c > 1$ agents assigned to the same position actually corresponds to c agents assigned to different positions that are very close to each other (e.g., at distance at most $1/2^n$ from each other). Moreover, the bound is purely algorithmic and holds regardless of the truthfulness of the composition.

4. MAX COST

We now consider $K = 2$. The algorithm OptMinMax returns an optimal allocation (f^0, f^1), $f^0 < f^1$, minimizing the maximum cost with a particular tie-breaking rule that we are going to define next.

Given an allocation (f^0, f^1), let $S_j \subseteq N$ be the set of agents that are closer to facility f^j than to facility $f^{|j-1|}$. Let $\Delta(S_j) = \max_{l_1, l_2 \in S_j} |b_{l_1} - b_{l_2}|$ denote the maximum distance between two elements of S_j. Whenever there is more than one solution minimizing the max cost, OptMinMax will choose the solution that minimizes $\Delta(S_0)$ and $\Delta(S_1)$, breaking any further tie in favor of minimizing $\Delta(S_0)$. It is easy to see that in this last case, i.e., there is a tie and $\Delta(S_0) = \Delta(S_1)$, one of the following two assertions must be true: either (i) $\Delta(S_j) = 2 \cdot mc(f(\mathbf{b}), \mathbf{b})$ $\forall j \in \{0, 1\}$, or (ii) $\Delta(S_0) < 2 \cdot mc(f(\mathbf{b}), \mathbf{b})$. Hereinafter, we will denote as $L(S_j) = \min S_j$ and $R(S_j) = \max S_j$. The tie breaking rule is such that in all cases the facility is allocated at the central point of the interval $[L(S_j), R(S_j)]$, namely, $f^j = \frac{L(S_j) + R(S_j)}{2}$, $j \in \{0, 1\}$. Fixed $\mathbf{b}_{-i}$, we let S'_j be the set S_j when agent i reports b_i instead of her true type t_i; we also let, as above, f_{t_i} and f_{b_i} be shorthands for $\text{OptMinMax}_{t_i}(t_i, \mathbf{b}_{-i})$ and $\text{OptMinMax}_{b_i}(b_i, \mathbf{b}_{-i})$, respectively. Next lemma assumes that the mechanism uses cluster-imposing no-cost-forging verification.

Lemma 2. *Let b_i be a misreport by agent i located at t_i. Let $j \in \{0, 1\}$ be such that $f^j = f_{b_i}$. If $f_{t_i} \neq f_{b_i}$, then either $b_i \in \{L(S'_j), R(S'_j)\}$ or $t_i \notin [L(S'_j), R(S'_j)]$.*

Proof. Two cases can occur: (i) $t_i \in (L(S_\ell), R(S_\ell))$ or (ii) $t_i \in \{L(S_\ell), R(S_\ell)\}$ for some $\ell \in \{0, 1\}$.

Let us consider case (i) first. We notice that if either $b_i \in [L(S_0), R(S_0)]$ or $b_i \in [L(S_1), R(S_1)]$, then $\Delta(S_0) = \Delta(S'_0)$ and $\Delta(S_1) = \Delta(S'_1)$ and hence $f_{t_i} = f_{b_i}$. Let us assume then that $b_i \notin [L(S_0), R(S_0)]$ and $b_i \notin [L(S_1), R(S_1)]$. Three cases can occur: $b_i < L(S_0)$, $R(S_0) < b_i < L(S_1)$ or $b_i > R(S_1)$. In all three cases it is immediately evident that $b_i \in \{L(S'_j), R(S'_j)\}$.

As for case (ii), we can assume that there is no $s \neq i$ such that $t_s = t_i$, as otherwise the same argument as case (i) applies (i.e., intervals cannot shrink). It is easy to check that if $b_i > R(S_1)$ or $b_i < L(S_0)$ then $b_i \in \{R(S'_1), L(S'_0)\}$. Let us consider the case when $t_i \in \{L(S_0), R(S_0)\}$ (the case when $t_i \in \{L(S_1), R(S_1)\}$ is symmetric). If $t_i = L(S_0)$ and $L(S_0) < b_i \leq R(S_0)$, the thesis holds. Likewise, if $t_i = R(S_0)$ and $L(S_0) \leq b_i < R(S_0)$ the thesis holds. If $R(S_0) < b_i < L(S_1)$, either $b_i \in \{R(S'_0), L(S'_1)\}$ (if $t_i = R(S_0)$) or $t_i \notin S'_0$ (if $t_i = L(S_0)$). If $L(S_1) \leq b_i \leq R(S_1)$ and $t_i = L(S_0)$ then $t_i \notin [L(S'_j), R(S'_j)]$. If $L(S_1) \leq b_i \leq R(S_1)$ and $t_i = R(S_0)$, it can either be $b_i = R(S'_0)$, for which the thesis holds, or $L(S'_1) \leq b_i$, in which case the thesis holds since $t_i \notin [L(S'_j), R(S'_j)]$. □

Theorem 7. OptMinMax *is SP with cluster-imposing no-cost-forging verification.*

Proof. Let us consider the case when agent i lies declaring b_i instead of her true type t_i. For the sake of contradiction, let us assume that OptMinMax is not SP with cluster-imposing no-cost-forging verification, i.e., $d(t_i, f_{t_i}) > d(t_i, f_{b_i})$. Let us suppose w.l.o.g. that $b_i \in S'_j$. We note that the misreport by agent i is not detected by the verification step only if $|f_{b_i} - b_i| = |f_{b_i} - t_i|$, which can happen only if $f_{b_i} - b_i = -(f_{b_i} - t_i)$, which implies that $f_{b_i} = \frac{t_i+b_i}{2}$. By Lemma 2, either (i) $t_i \notin [L(S'_j), R(S'_j)]$ or (ii) $b_i \in \{L(S'_j), R(S'_j)\}$. In case (i) we notice that the misreport is always detected by the verification step as $f_{b_i} = \frac{R(S'_j)+L(S'_j)}{2} \neq \frac{t_i+b_i}{2}$, since $t_i \notin [L(S'_j), R(S'_j)]$ and $b_i \in [L(S'_j), R(S'_j)]$. Hence we can assume $b_i \in \{L(S'_j), R(S'_j)\}$. By construction of the algorithm and by the verification step, this also implies that $t_i \in \{L(S'_j), R(S'_j)\}$, i.e. t_i and b_i are the extremal point of S'_j. We need to consider 4 cases.

$t_i \in S_j \wedge \Delta(S'_j) > \Delta(S_j)$. By construction of the algorithm we also have that $t_i \in \{L(S'_j), R(S'_j)\}$ and then

$$d(t_i, f_{b_i}) = \Delta(S'_j)/2 > \Delta(S_j)/2 \geq d(t_i, f_{t_i})$$

where: $t_i \in \{L(S'_j), R(S'_j)\}$ implies $d(t_i, f_{b_i}) = \Delta(S'_j)/2$, $\Delta(S'_j)/2 > \Delta(S_j)/2$ by hypothesis and $\Delta(S_j)/2 \geq d(t_i, f_{t_i})$ by construction.

$t_i \in S_j \wedge \Delta(S'_j) < \Delta(S_j)$. Here $t_i \in \{L(S_j),\ R(S_j)\}$. As above, agent i is not caught by verification only if $f_{b_i} = \frac{b_i+t_i}{2}$ and $S'_j = [\min\{t_i, b_i\}, \max\{t_i, b_i\}]$. We note that agent i is indeed always caught by the verification step as $\Delta(S'_j) < \Delta(S_j)$ implies that $t_i \notin S'_j$.

$t_i \in S_{|j-1|} \wedge \Delta(S'_j) > \Delta(S_j)$. By construction, $\Delta(S'_j) = |t_i - b_i|$ and $\frac{\Delta(S_{|j-1|})}{2} \geq |t_i - f^{|j-1|}| = d(t_i, f_{t_i})$. The following holds by the hypothesis that OptMinMax is not SP: $\frac{\Delta(S_{|j-1|})}{2} \geq |t_i - f^{|j-1|}| = d(t_i, f_{t_i}) > d(t_i, f_{b_i}) = \frac{|t_i-b_i|}{2}$. Let us define intervals $T_j = S'_j$, and $T_{|j-1|} = S_{|j-1|} \backslash S'_j$. We will first prove that the max cost of allocation (f^0, f^1) is $\frac{\Delta(S_{|j-1|})}{2}$. To this aim, we need to prove that $\Delta(S_{|j-1|}) \geq \Delta(S_j)$. By contradiction, let us suppose $\Delta(S_j) > \Delta(S_{|j-1|})$. From the latter, we get

$$\begin{aligned} d(t_i, f_{b_i}) = \quad & \frac{|t_i - b_i|}{2} = \frac{\Delta(S'_j)}{2} > \frac{\Delta(S_j)}{2} > \\ > \quad & \frac{\Delta(S_{|j-1|})}{2} \geq |t_i - f^{|j-1|}| = \quad d(t_i, f_{t_i}) \end{aligned}$$

where $\Delta(S'_j)/2 > \Delta(S_j)/2$ follows by hypothesis, thus contradicting the hypothesis that OptMinMax is not SP.

We will now prove that allocating the facilities in the middle points of $T_{|j-1|}$ and T_j has a lower cost than $\frac{\Delta(S_{|j-1|})}{2}$, contradicting the optimality of (f^0, f^1). By construction $\Delta(T_{|j-1|}) < \Delta(S_{|j-1|})$. Furthermore, by hypothesis, it follows:

$$\frac{\Delta(T_j)}{2} = \frac{|t_i - b_i|}{2} < |t_i - f^{|j-1|}| \leq \frac{\Delta(S_{|j-1|})}{2}$$

which implies $\Delta(T_j) < \Delta(S_{|j-1|})$.

$t_i \in S_{|j-1|} \wedge \Delta(S'_j) < \Delta(S_j)$. The misreport of agent i is not detected by the verification step if and only if $f_{b_i} = \frac{t_i+b_i}{2}$, which means that $S'_j \subseteq [\min\{t_i, b_i\}, \max\{t_i, b_i\}]$. Let us define intervals $T_j = S'_j$ and $T_{|j-1|} = S_{|j-1|} \backslash S'_j$. Since $\Delta(T_{|j-1|}) < \Delta(S_{|j-1|})$ (by construction, as $t_i \in S'_j \cap S_{|j-1|}$) and $\Delta(T_j) = \Delta(S'_j) < \Delta(S_j)$ (by hypothesis), we obtain the absurd that allocating the facilities in the middle nodes of intervals T_j and $T_{|j-1|}$ has a lower max cost. □

Theorem 8. *The assumptions of Theorem 7 are necessary.*

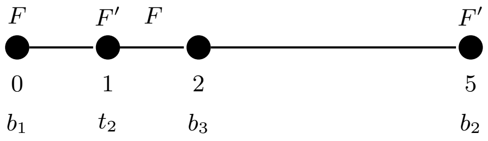

Figure 2: Instance used in Theorem 8

Proof. Let us drop cluster-imposition and consider a 3-agent instance, where $b_1 = 0$, $t_2 = 1$ and $b_3 = 2$. The instance is depicted in Figure 2. The output of OptMinMax is $(f^0 = 0, f^1 = 1.5)$, and the cost for agent 2 is 0.5. When agent 2 lies declaring $b_2 = 5$ instead of her true type, then the allocation is $(f^0 = 1, f^1 = 5)$, and her cost becomes 0. The lie is not detected by the verification step as $d(b_2, f(b_2, \mathbf{b}_{-2})) = d(t_2, f(b_2, \mathbf{b}_{-2})) = 0$.

Let us now maintain cluster-imposition and relax no-cost-forging to no-underbidding. Consider the instance with 4 agents: $b_1 = 0$, $b_2 = 1$, $b_3 = \Delta + 1$, $t_4 = \Delta + 2$, for $\Delta \geq 2$. OptMinMax allocates one facility on 0.5 and the other on $\Delta + 1.5$; the cost for agent 4 is 0.5. If agent 4 declares $b_4 = \Delta + 2 + \epsilon$, $0 < \epsilon < 0.5$, OptMinMax allocates the second facility on $\Delta + 1.5 + \frac{\epsilon}{2}$, yielding a lower cost for agent 4. It is easy to check that this misreport is caught neither by no-underbidding nor cluster-imposing verification.

It is not hard to use similar ideas to find a counterexample for cluster-imposition and no-overbidding verification. □

5. CONCLUSIONS

We have shown what lies make the computation of optimal solutions for facility location vulnerable to selfish misreports. The set of verification assumptions are shown to be minimal and hence necessary. The parameters of the instances given in the proofs of Theorems 3 and 8 to prove this minimality can in fact be optimized to prove constant lower bounds. However, we conjecture that minimality can be extended to cover not just optimal algorithms, but all (simple) algorithms with constant (or even sublinear) approximation guarantees. We have focused on cost-only verification and given some results towards proving the conjecture, including the first use of cycle-monotonicity to establish approximation lower bounds, cf. Theorem 4. To strengthen this claim, one would need a property for $\mathbf{b}_{-i}$ which allows the construction of the right instance for a larger class of algorithms. This could be complemented by a better understanding of constant-approximation algorithms for the social cost (via, e.g., a deeper look at compositions of basic algorithms).

We leave a number of interesting, challenging open questions starting from the conjecture above. A first step could be to focus on Maximal-In-Range (MIR) algorithms [18] – indeed, it is not too difficult to check that Theorem 2 holds for any MIR algorithm.

Generalizing our results to different problems or settings without single-peaked preferences is clearly an interesting open problem and a further step in the research agenda we suggest. A first issue one would run into is represented by the "translation" of direction-imposing verification to diverse settings. This topological notion exploits the structure of facility location on the line and might be difficult to apply to different scenarios. A possible approach could be to look into different ways to express such a notion (see "Discussion on verification" paragraph in Section 2).

REFERENCES

[1] G. Christodoulou, E. Koutsoupias, and A. Kovács. Mechanism design for fractional scheduling on unrelated machines. In *ICALP*, pages 40–52, 2007.

[2] A. Daniely, M. Schapira, and S. Gal. Inapproximability of truthful mechanisms via generalizations of the VC dimension. In *STOC*, 2015.

[3] D. Fotakis, P. Krysta, and C. Ventre. Combinatorial auctions without money. In *AAMAS*, pages 1029–1036, 2014.

[4] D. Fotakis, P. Krysta, and C. Ventre. The power of verification for greedy mechanism design. In *AAMAS*, pages 307–315, 2015.

[5] D. Fotakis and C. Tzamos. Winner-imposing strategyproof mechanisms for multiple Facility Location games. *Theoretical Computer Science*, 472:90–103, 2013.

[6] D. Fotakis and C. Tzamos. On the power of deterministic mechanisms for facility location games. *ACM Trans. Economics and Comput.*, 2(4):15:1–15:37, 2014.

[7] D. Fotakis, C. Tzamos, and E. Zampetakis. Who to trust for truthfully maximizing welfare? *CoRR*, abs/1507.02301, 2015.

[8] D. Fotakis and E. Zampetakis. Truthfulness flooded domains and the power of verification for mechanism design. *ACM Trans. Economics and Comput.*, 3(4):20, 2015.

[9] C. Gorkem. Mechanism design with weaker incentive compatibility constraints. *Games and Economic Behavior*, 56(1):37–44, 2006.

[10] J. R. Green and J. Laffont. Partially Verifiable Information and Mechanism Design. *The Review of Economic Studies*, 53:447–456, 1986.

[11] E. Koutsoupias. Scheduling without payments. *Theory Comput. Syst.*, 54(3):375–387, 2014.

[12] E. Koutsoupias and A. Vidali. A lower bound of 1+Φ for truthful scheduling mechanisms. *Algorithmica*, 66(1):211–223, 2013.

[13] P. Krysta and C. Ventre. Combinatorial auctions with verification are tractable. *Theoretical Computer Science*, 571:21–35, 2015.

[14] D. J. Lehmann, L. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *J. ACM*, 49(5):577–602, 2002.

[15] P. Lu, X. Sun, Y. Wang, and Z. Zhu. Asymptotically Optimal Strategy-Proof Mechanisms for Two-Facility Games. In *EC*, pages 315–324, 2010.

[16] A. Mu'alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. *Games and Economic Behavior*, 64(2):612–631, 2008.

[17] N. Nisan and A. Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35:166–196, 2001.

[18] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. *J. Artif. Intell. Res. (JAIR)*, 29:19–47, 2007.

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

[20] P. Penna and C. Ventre. Optimal collusion-resistant mechanisms with verification. *Games and Economic Behavior*, 86:491–509, 2014.

[21] A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. *ACM Trans. Economics and Comput.*, 1(4):18, 2013.

[22] J. Rochet. A Condition for Rationalizability in a Quasi-Linear Context. *Journal of Mathematical Economics*, 16:191–200, 1987.

[23] P. Serafino and C. Ventre. Truthful mechanisms without money for non-utilitarian heterogeneous facility location. In *AAAI*, pages 1029–1035, 2015.

[24] C. Ventre. Truthful optimization using mechanisms with verification. *Theoretical Computer Science*, 518:64–79, 2014.

[25] R. Vohra. *Mechanism Design: A Linear Programming Approach.* Cambridge University Press, 2011.