Heterogeneous Facility Location without Money*

Paolo Serafino Carmine Ventre
School of Computing
Teesside University
U. K.
email: {p.serafino, c.ventre}@tees.ac.uk

Abstract

The study of the facility location problem in the presence of self-interested agents has recently emerged as the benchmark problem in the research on mechanism design without money. In the setting studied in the literature so far, agents are single-parameter in that their type is a single number encoding their position on a real line. We here initiate a more realistic model for several real-life scenarios. Specifically, we propose and analyze *heterogeneous facility location without money*, a novel model wherein: (*i*) we have multiple heterogeneous (i.e., serving different purposes) facilities, (*ii*) agents' locations are disclosed to the mechanism and (*iii*) agents bid for the set of facilities they are interested in (as opposed to bidding for their position on the network).

We study the heterogeneous facility location problem under two different objective functions, namely: *social cost* (i.e., sum of all agents' costs) and *maximum* cost. For either objective function, we study the approximation ratio of both deterministic and randomized truthful algorithms under the simplifying assumption that the underlying network topology is a line. For the social cost objective function, we devise an $(n-1)$-approximate deterministic truthful mechanism and prove a constant approximation lower bound. Furthermore, we devise an *optimal* and *truthful* (in expectation) randomized algorithm. As regards the maximum cost objective function, we propose a 3-approximate deterministic strategyproof algorithm, and prove a $3/2$ approximation lower bound for deterministic strategyproof mechanisms. Furthermore, we propose a $3/2$-approximate randomized strategyproof algorithm and prove a $4/3$ approximation lower bound for randomized strategyproof algorithms.

1 Introduction

Mechanism design without money is a relatively recent and challenging research agenda introduced by Procaccia and Tennenholtz in [16]. It is mainly concerned with the design of *truthful* (or *strategyproof*, *SP* for short) *mechanisms* in scenarios where monetary compensation cannot be used as a means to re-align the agents' interest to the mechanism designer's objective (as, e.g., done by VCG mechanisms). It has been noticed that such a circumstance occurs very frequently in real-life scenarios, as payments between agents and the mechanism are either illegal (e.g., organ transplant) or unethical (e.g., in the case of political decision making). To circumvent the impossibility of utilizing payments to enforce truthfulness, Procaccia and Tennenholtz propose instead to leverage the *approximation ratio* of the mechanism in those cases where the optimal outcome is not truthful. The facility location problem is arguably the archetypal problem in mechanism design without money [16]. It demands locating a set of facilities on a network, on input the bids of the agents for their locations, in such a way as to optimize a given *objective function* that depends on agents' costs. If we regard the problem of locating facilities as a political decision (e.g., a city council locating facilities of public interest on the basis of the population residing

*Preliminary versions of this paper have appeared as [19, 18, 20]. This work is partially supported by EPSRC grant EP/M018113/1.

in a certain area), the impossibility to utilize payments and the need to locate the facilities to minimize the social cost (e.g., the traffic in the city) in this context becomes immediately apparent.

Another application scenario that can be envisaged is *big data distribution in cloud networks*. Consider a multinational company having to decide how to distribute the data contained in its databases over its data network. Not all the various offices working for the company need access to the whole dataset, e.g., a payroll office arguably needs access to employees' data but not to customers', whilst sales offices need customers' data but not employees'. Thus, a demand-based allocation seems a sensible approach. However, due to well-known issues such as space consumption, consistency and query latency, it might be impractical to allow replication of the requested data at all the demanding offices' sites. Fast data access becomes then competitive and, guided by their willingness to have prompt access to the data they need, offices might strategize and amend their demands accordingly. The company, however, wants to minimize the maximum access time in order to guarantee a decent quality of service so that each office can work efficiently.

1.1 Our Contribution

Inspired by the work on facility location without money, and aiming at analyzing a richer and more realistic setting, we introduce and study the problem of *heterogeneous facility location without money*. With respect to the main stream of works on facility location, our model features *heterogeneous* (i.e., serving different purposes) as opposed to the *homogeneous* (i.e. serving the same purpose) facilities. Allowing heterogeneous facilities influences the agent cost model as in our setting the cost of an agent is the cost to access the set of facilities she is interested in, rather than accessing (as in the traditional setting) the nearest facility. Furthermore, we assume in our model that agents' locations are disclosed to the mechanism. This assumption fits many real-life applications (e.g., for the aforementioned examples, the city council can ask for proof of residence whilst the multinational company knows where its payroll offices are located).

	Social Cost		**Maximum Cost**	
	LB	UB	LB	UB
Deterministic	$9/8$	$n-1$	$3/2$	3
Randomized	1		$4/3$	$3/2$

Table 1: Summary of our results

In more detail, we focus on the heterogeneous facility location problem in the case in which the agents are on a discrete *line* and we have *two* facilities to locate. Despite its apparent simplicity, this class of instances already encodes many intricacies and showcases the tension between truthfulness without money and approximation. Moreover, these instances model the aforementioned content distribution scenario (the linear network being the backbone of the company's data network; facilities being employee and customer records). We study both utilitarian (i.e., social cost) and non-utilitarian (i.e., max-cost) objective functions. Under either objective function, we analyze both deterministic and randomized algorithms (see Table 1), prove that in both cases the optimal allocation does not preserve truthfulness, and provide lower and upper bounds for the approximation of truthful mechanisms.

As regards the social cost objective function, we prove a $9/8$ lower bound for the approximation of deterministic strategyproof algorithms. We then propose a *truthful* $(n-1)$*-approximate deterministic algorithm* named TWOEXTREMES, an adaptation to our model of a mechanism already proposed in [16], that assigns each facility to an extreme of the subnetwork of nodes requesting it. In order to provide better approximation guarantees, we then turn our attention to randomized algorithms and devise an *optimal randomized algorithm* that is *truthful in expectation*. At intuitive level, the reason for which deterministic optimal algorithms are not truthful resides in the richness of optimal solutions in very symmetric instances. For each way a deterministic optimum can break these ties, one side of the network will be disadvantaged and will then be able to manipulate the algorithm. The idea behind our randomized algorithm is to take care of these symmetries with randomization so that in expectation agents on either sides of the network are "happy". The technical challenge is that, in some cases, there are not enough optimal solutions to randomize upon and therefore a careful combination of deterministic and randomized solutions is designed and shown to preserve truthfulness.

As regards the maximum cost objective function, we prove a lower bound of $3/2$ on the approximation guarantee of deterministic *SP* mechanisms. The proof connects three different instances and uses truthfulness constraints on two agents to establish the lower bound. This is somehow more complex than typical lower bounds in literature wherein two instances and one lying agent are normally considered. We then analyze TWOEXTREMES for maximum cost and prove it is 3–approximate. We observe that TWOEXTREMES retains its strategyproofness as the latter is independent from the objective function of the mechanism but depends solely on the agents' cost function. Regarding randomized mechanisms we first prove a lower bound of $4/3$ and then design a $3/2$-approximate randomized *SP* mechanism. This algorithm is mainly based on the idea of allocating (in expectation) each facility on the average position of the subgraph comprised of agents requesting it. This way truthfulness is guaranteed since there is no advantage in hiding one's own requested facilities as the aforementioned subgraph can only move away from the lying agent. (Note that adding unneeded facilities does not help either.) A complication to this intuition is that facilities cannot always be located in the "middle" of the subgraph. Our algorithm works around this, while preserving truthfulness and guaranteeing a good approximation.

Roadmap. The remainder of this paper is organized as follows. Section 2 is devoted to survey some related literature. In Section 3 we formalize our model for the heterogeneous facility location problem on the line and give some definitions that will be used in the remainder of the paper. In Section 4 we discuss our results about the social cost objective function, whereas in Section 5 we analyze the maximum cost objective function Finally, in Section 6 we draw some conclusions and highlight some future research efforts.

2 Related Work

Over the years, the facility location problem has proved to be a fertile research problem and, as such, has been addressed by as diverse research communities as Operation Research, Algorithm Design, Social Choice and, relatively recently, Algorithmic Mechanism Design.

The Social Choice community has been mostly concerned with the problem of locating a single facility on the line. In his classical paper [13] Moulin characterizes the class of generalized median voter schemes as the only deterministic *SP* mechanisms for *single-peaked* agents on the line. Schummer and Vohra [17] extend the result of Moulin to the more general setting where *continuous graphs* are considered, characterizing *SP* mechanisms on *continuous lines and trees*. They show that on circular graphs every *SP* mechanism must be dictatorial.

From a Mechanism Design perspective, the aforementioned paper by Procaccia and Tennenholtz [16] initiated the field of approximate mechanism design without money. For the 2-facility location problem, they propose the Two-Extremes algorithm, that places the two facilities in the leftmost and rightmost location of the instance, and prove that it is group strategyproof and $(n-2)$-approximate, where n is the number of agents. Furthermore, they provide a lower bound of $3/2$ on the approximation ratio of any *SP* algorithm for the facility location problem on the line and conjecture a lower bound of $\Omega(n)$.

Lu et al. [12], improve several bounds studied in [16]. Particularly, as regards deterministic algorithms they prove a better (w.r.t. [16]) lower bound of $2 - \mathcal{O}(\frac{1}{n})$. Furthermore, they prove a 1.045

lower bound for randomized mechanisms for the 2-facility location problem on the line and present a randomized $n/2$-approximate mechanism.

Fotakis et al. [8] prove the conjectured lower bound of $\Omega(n)$ for deterministic *SP* algorithms for the facility location problem on the line. Their main result is the characterization of deterministic *SP* mechanisms with *bounded approximation ratio* for the 2-facility location problem on the line. They show that there exist only two such algorithms: (*i*) a mechanism that admits a unique dictator and (*ii*) the Two-Extremes mechanism proposed in [16].

In [11], Lu et al. focus on general metric spaces for the 2-facility game. They prove an $\Omega(n)$ lower bound for the approximation of deterministic strategyproof mechanisms and propose the so-called *Proportional* Mechanism, the first randomized algorithm to attain constant approximation ratio.

Alon et al. [1] derive a linear (in the number of agents) lower bound for *SP* mechanisms on continuous cycles. Furthermore, they derive a constant approximation bound for randomized mechanisms in the same settings. Dokow et al [2] shift the focus of research to *discrete* lines and cycles instead. They prove that *SP* mechanisms on *discrete large cycles* are nearly-dictatorial in that all agents can effect the outcome to a certain extent. Contrarily to the case of continuous cycles studied in [17], for small discrete graphs Dokow et al. prove that there are anonymous *SP* mechanisms. Furthermore, they prove a linear lower bound in the number of agents for the approximation ratio of *SP* mechanisms on discrete cycles.

Another interesting line of research in this area advocates the use of *imposing mechanisms*, i.e., mechanisms able to limit the way agents exploit the outcome of a game. For the facility location problem, imposing mechanisms typically prevent an agent from connecting to some of the facilities, thus increasing her connecting cost and penalizing liars. Following this wake, the authors of [7] consider *winner-imposing* mechanisms, namely mechanisms that (*i*) allocate a facility only at a location where there is an agent requesting it (as opposed to mechanisms that allocate facilities at arbitrary locations) and (*ii*) require that an agent that *wins* a facility (i.e. has a facility allocated to her location) must connect to it. They prove that the winner-imposing version of the Proportional Mechanism proposed in [11] is *SP* for the K-facility location problem and achieves an approximation ratio of at most $4K$, for $K \geq 1$. Furthermore they propose a deterministic non-imposing group strategyproof $\mathcal{O}(\log n)$-approximate mechanism for a variant of the facility location problem on the line with opening costs of facilities and no constraint on the number of facilities to be located.

Nissim et al. [14] combine imposing mechanisms and *differentially private mechanisms* (i.e., randomized algorithms whose outcomes are sufficiently "close" on instances that differ only on the declaration of one agent) to obtain the only known general technique for designing *SP* approximate mechanisms without money. In particular, they devise an SP approximate imposing mechanism for the K-facility location problem with a running time exponential in K. Unfortunately the algorithm proposed in [14] has an approximation ratio which is not bounded by any constant.

An approach similar to the work on imposing mechanisms is followed by a recent research avenue which attempts to import the advantages of *verification* into the field of mechanism design without money. Generally speaking, mechanisms with verification can prevent certain lies of the agents, by simply "observing" the output of the mechanism and punishing uncovered liars. This mechanism design paradigm has proved to be pretty powerful allowing for very general and strong, otherwise impossible, positive results in mechanism design *with* money, see [15] and references therein. Truthful mechanisms without money for scheduling selfish unrelated machines whose execution times can be (strongly) verified are considered in [9]. Mechanisms without money for combinatorial auctions are instead briefly treated in [10] and recently studied in [4, 5]. Verification has also recently been studied for homogenous facility location problem [3].

The literature on Min-Max objective function is quite rich in the case of mechanism design with money but sparse in the case of moneyless mechanisms. Procaccia and Tennenholtz prove in their model tight bounds for min-max approximation with 1 facility and nearly tight results with 2 facilities. Further results on min-max approximation for 2 facilities are contained in [6]. The paper [9] mentioned above studies min-max approximation for the scheduling selfish unrelated machines scenario considered.

3 Model and Preliminary Definitions

The *heterogeneous 2-facility location problem on the line* (hereinafter facility location, for short) consists of locating two facilities on a *linear unweighted graph*. More specifically, we are given a set of agents $N = \{1, \dots, n\}$; an undirected unweighted linear graph $G = (V, E)$, where $V \supseteq N$ is the set of vertices and E is the set of edges; and a set of facilities $\mathfrak{F} = \{\mathsf{F}_1, \mathsf{F}_2\}$. Graph G is *linear* in the sense that its vertices can be listed in the order $(v_1, v_2, \dots, v_{|V|})$ and E is the set of edges connecting v_i and v_{i+1}, for all $i = 1, \dots, |V| - 1$. Because of this property, for each node v we can identify a *successor*, denoted as $v + 1$, and a *predecessor*, denoted as $v - 1$. If v is the leftmost (rightmost, respectively) node of G, then its predecessor (successor, respectively) is the *fictitious node* NIL, and we will write $v - 1 = \text{NIL}$ ($v + 1 = \text{NIL}$, respectively). A *point* μ on G is either a vertex (which we will denote as $\mu \in V$) or a point on an edge connecting two nodes (which we will denote as $\mu \in E$).

Nodes where no agent resides, namely those in $V \setminus N$, are referred to as *empty nodes*. Agents' *types* are subsets of $\mathfrak{F}$, called their *facility set*. We denote the true type of agent i as $T_i \subseteq \mathfrak{F}$.[1] Furthermore, we will denote as $\mathcal{T} = (T_1, \dots, T_n)$ the vector of types of N, whereas $(T_i', \mathcal{T}_{-i})$ will denote the vector of types where agent i declares $T_i' \subseteq \mathfrak{F}$ instead of her true type. A mechanism M for the facility location problem takes as input a vector of types $\mathcal{T}$ and returns as output a *feasible* allocation $M(\mathcal{T}) = (F_1, F_2)$, such that $F_i \in V$ and $F_1 \neq F_2$.[2] Given a feasible allocation $\mathcal{F} = (F_1, F_2)$, agent i has a cost defined as $cost_i(\mathcal{F}) = \sum_{j \in T_i} d(i, F_j)$, where $d(i, F_j)$ denotes the length of the shortest path from i to F_j in G. Naturally, agents seek to minimize their cost. Therefore, they could misreport their facility sets to the mechanism if this reduces their cost. We are interested in the following class of mechanisms.

A mechanism M is *truthful* (or *strategyproof*, *SP*, for short) if for any vector of types $\mathcal{T}$, any agent i, and any declaration T_i', we have $cost_i(\mathcal{F}) \leq cost_i(\mathcal{F}')$, where $\mathcal{F} = M(\mathcal{T})$ and $\mathcal{F}' = M(T_i', \mathcal{T}_{-i})$. A *randomized* mechanism M is a *truthful in expectation* if the *expected cost* of every agent is minimized by truthtelling. In the study of randomized mechanisms we will require the concept of *mean set*, which we define here for convenience.

Definition 3.1. Let $S \subset V \times V$ be a set of feasible allocations on G and μ be a point on G. Let $\mathcal{M} \subseteq S$ and let X_k and X_{k+1} be the random variables defined as the positions of facilities F_k and F_{k+1}, respectively, if an allocation is drawn uniformly at random from $\mathcal{M}$. Then, $\mathcal{M}$ is a *mean set* of S centered around μ for facility $\mathsf{F_k}$ if $E[X_k] = \mu$. A solution extracted from a mean set is called a *mean set solution*.

To illustrate Definition 3.1, let us consider Figure 3, where we have a graph $G = (V, E)$ such that $V = \{v_1, v_2, v_3\}$ and $E = \{\{v_1, v_2\}, \{v_2, v_3\}\}$. In this case, a point on G can be either one of the vertices or any point on the *straight line edges* connecting the vertices. In Figure 3 the midpoints of edges $\{v_1, v_2\}$ and $\{v_2, v_3\}$ are denoted as μ_1 and μ_2, respectively. Let us consider the set of feasible allocations $S = \{(v_1, v_3), (v_2, v_1)\}$ where (v_1, v_3) signifies that F_1 is located at v_1 and F_2 at v_3, whereas (v_2, v_1) signifies that F_1 is located at v_2 and F_2 at v_1. If we extract uniformly at random an allocation from S, then we have that $E[X_1] = \mu_1$, whereas $E[X_2] = v_2$. Then $\mathcal{M} = S$ is a mean set of S centered around μ_1 (v_2, respectively) for F_1 (F_2, respectively).

In this paper, we are interested in truthful mechanisms M that return allocations $\mathcal{F} = M(\mathcal{T})$ minimizing a certain *objective function* $obj(\mathcal{F})$, dependent on the costs of individual agents. In particular, we

[1] Sometimes, slightly abusing notation, we will regard T_i as a set of *indices* j s.t. $\mathsf{F}_j \in T_i$.

[2] We note that when this constraint is lifted we are in the case of *capacious* nodes, meaning that we can accommodate all the facilities at one node. It is not difficult to see that in this case the optimal allocation is always truthful under both objective functions and computable in polynomial time. The same holds true even for the more general problem of K-facility location on general graphs if we allow capacious nodes, i.e., every single node can accommodate all K facilities.

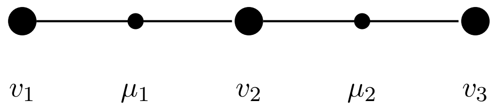

Figure 1: Illustrating the concept of mean set solutions.

will consider two objective functions: (*i*) the *social cost* function, namely $cost(\mathcal{F}) = \sum_{i \in N} cost_i(\mathcal{F})$ and (*ii*) the *maximum cost* function, namely $mc(\mathcal{F}) = \max_{i \in N} cost_i(\mathcal{F})$.

We call *optimal* a mechanism M such that $M(\mathcal{T}) \in \text{argmin}_{\mathcal{F} \text{ feasible}} obj(\mathcal{F})$ (where obj is either *cost* or mc) and denote an optimal allocation on declaration vector $\mathcal{T}$ as $OPT(\mathcal{T})$ if $obj(OPT(\mathcal{T})) = \min_{\mathcal{F} \text{ feasible}} obj(\mathcal{F})$.

Alas, sometimes we have to content ourselves with sub-optimal solutions. In particular, we say that a mechanism M is α-approximate if $obj(M(\mathcal{T})) \leq \alpha \cdot obj(OPT(\mathcal{T}))$. Furthermore, we denote as $N_j[\mathcal{T}] \subseteq N$ the set of agents wanting access to facility F_j according to a declaration vector $\mathcal{T}$, i.e., $N_j[\mathcal{T}] = \{i \in N | \mathsf{F}_j \in T_i\}$.

For the sake of notational conciseness, in the remainder of the paper we will often omit the declaration vector $\mathcal{T}$ (e.g., $N_k[\mathcal{T}]$ simply denoted as N_k) and denote an untruthful declaration $(T'_i, \mathcal{T}_{-i})$ of agent i by a prime symbol (e.g., $N_k[T'_i, \mathcal{T}_{-i}]$ simply denoted as N'_k). Finally, we define $\text{avg}(N_k) = \frac{\max(N_k)+\min(N_k)}{2}$ to be the function that computes the average point of a set of agents N_k. As an example, in the instance of Figure 3 if $N_1 = \{v_1, v_2\}$ and $N_2 = \{v_2, v_3\}$, then $\text{avg}(N_1) = \mu_1$ and $\text{avg}(N_2) = \mu_2$.

4 Social Cost Objective Function

In this section we discuss our results about the social cost objective function. In particular, in Section 4.1 we discuss deterministic algorithms, whereas in Section 4.2 we present our results concerning randomized algorithms.

4.1 Deterministic Mechanisms

In this section we study deterministic mechanisms for the 2-facility location problem under the social cost objective function. We first ask ourselves whether the optimal allocation for the facility location problem is truthful, to which we give a negative answer in Theorem 4.1, also providing a 9/8 lower bound for the approximation of deterministic SP algorithms. Afterwards, we discuss an $(n-1)$-approximate deterministic algorithm for the facility location problem.

Theorem 4.1. *No deterministic α-approximate SP mechanism can obtain an approximation ratio $\alpha < 9/8$ for the social cost objective function.*

Proof. Let us consider the instance depicted in Figure 2 according to the following declarations: $T_1 = \{\mathsf{F}_1\}$, $T_2 = \{\mathsf{F}_2\}$, $T_3 = \{\mathsf{F}_1, \mathsf{F}_2\}$, $T_4 = \{\mathsf{F}_2\}$, $T_5 = \{\mathsf{F}_1\}$. It can be easily checked that the optimal locations for this instance are the ones that locate a facility on node 3 and the other on either node 2 or 4, namely: $(F_1^* = 2, F_2^* = 3)$, $(F_1^* = 4, F_2^* = 3)$, $(F_1^* = 3, F_2^* = 2)$ and $(F_1^* = 3, F_2^* = 4)$. Let us note that any α-approximate algorithm with $\alpha < 9/8$ on input $\mathcal{T}$ would return an optimal solution. Indeed, it can be easily checked that the two second-best solutions $(F_1 = 2, F_2 = 4)$ and $(F_1 = 4, F_2 = 2)$ are $8/7$-approximate, their cost being 8 whereas $cost(OPT(\mathcal{T})) = 7$.

Let us consider the optimal solution $(F_1^* = 2, F_2^* = 3)$. If agent 5 reports $T'_5 = \{\mathsf{F}_1, \mathsf{F}_2\}$, then the only optimal solution is $OPT(T'_i, \mathcal{T}_{-i}) = (3, 4)$. We note that, since the cost (with respect to $(T'_i, \mathcal{T}_{-i})$) of this optimal solution is 8 whereas the cost of any second best solution (i.e, $(F_1 = 4, F_2 = 3)$, $(F_1 = 2, F_2 = 3)$ and $(F_1 = 2, F_2 = 4)$) is 9, any α-approximate algorithm with $\alpha < 9/8$ would return

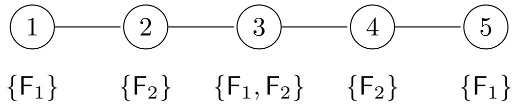

Figure 2: Instance showing that OPT is not truthful

the optimum. Furthermore, we note that the optimal solution is not *SP*, since $cost_5(OPT(T_i', \mathcal{T}_{-i})) = 2 < 3 = cost_5(OPT(\mathcal{T}))$. We note that, due to the intrinsic symmetry of the instance, a similar argument applies for solution $(F_1^* = 4, F_2^* = 3)$ when agent 1 reports $T_1' = \{\mathsf{F}_1, \mathsf{F}_2\}$.

Let us consider the optimal solution $(F_1^* = 3, F_2^* = 4)$. If agent 2 reports $T_2' = \{\mathsf{F}_1, \mathsf{F}_2\}$, then the only optimal solution is $OPT(T_i', \mathcal{T}_{-i}) = (F_1^* = 2, F_2^* = 3)$. We note that, since the cost (with respect to $(T_i', \mathcal{T}_{-i})$) of this optimal solution is 7 and the cost of any second best solution (i.e, $(F_1 = 2, F_2 = 4)$, $(F_1 = 3, F_2 = 4)$ and $(F_1 = 4, F_2 = 3)$) is 8, any α-approximate algorithm with $\alpha < 9/8$ would return the optimum. Furthermore, we note that the optimal solution is not *SP*, since $cost_2(OPT(T_i', \mathcal{T}_{-i})) = 1 < 2 = cost_2(OPT(\mathcal{T}))$. We note that, due to the intrinsic symmetry of the instance, a similar argument applies for solution $(F_1^* = 3, F_2^* = 2)$ when agent 4 reports $T_4' = \{\mathsf{F}_1, \mathsf{F}_2\}$. □

We now discuss TWOEXTREMES, a deterministic mechanism which is truthful and returns linear-approximate allocations. The algorithm, reported in Algorithm 1, is inspired by Two-Extremes of [16], the difference being that, due to the multi-dimensional nature of our problem, we need to check for the feasibility of solutions putting facilities at the extremes and handle cases of clash. In more detail, Algorithm 1 tries to allocate facility F_1 at the leftmost element of N_1 and F_2 at the rightmost element of N_2. If this allocation is unfeasible, i.e., $F_1 = F_2$, then it applies the following tie-breaking rule: if there is a node in the graph at the left of the clash node, i.e., $F_2 \neq$ NIL, then it moves F_2 to the left, i.e., $F_2 := F_2 - 1$; if the clash node is the leftmost node of the graph, it moves F_1 to the right[3], i.e. $F_1 := F_1 + 1$.

Algorithm 1: TWOEXTREMES

Require: Line G, facilities $\mathfrak{F} = \{\mathsf{F}_1, \mathsf{F}_2\}$, declarations $\mathcal{T} = \{T_1, \ldots, T_n\}$
Ensure: $F(\mathcal{T})$, a $(n-1)$-approximate allocation for 2-facility location on G

$F_1 := \min N_1[\mathcal{T}]$
$F_2 := \max N_2[\mathcal{T}]$
if $F_1 = F_2$ **then**
  if $F_2 - 1 \neq$ NIL **then**
    $F_2 := F_2 - 1$
  else
    $F_1 := F_1 + 1$
  end if
end if
return (F_1, F_2)

We begin by proving the truthfulness of the algorithm.

Theorem 4.2. *Algorithm* TWOEXTREMES *is SP.*

Proof. For the sake of contradiction, let us assume that there exist $i \in N$ with type T_i and an untruthful declaration T_i' such that $\sum_{j \in T_i} d(i, F_j(\mathcal{T})) > \sum_{j \in T_i} d(i, F_j(T_i', \mathcal{T}_{-i}))$, where $F_j(\mathcal{Z})$ denotes the location in which TWOEXTREMES, on input the declaration vector $\mathcal{Z}$, assigns facility F_j. We need to analyse three cases: (*a*) $i = \min N_1$, (*b*) $i = \max N_2$, and (*c*) $i \notin \{\min N_1, \max N_2\}$.

[3]We note that there must be at least two nodes otherwise the set of feasible solutions would be empty.

If case (a) occurs, it can be either $T_i = \{\mathsf{F}_1\}$ or $T_i = \{\mathsf{F}_1, \mathsf{F}_2\}$. If $T_i = \{\mathsf{F}_1\}$ then $F_1 = i$, $cost_i(F(\mathcal{T})) = 0$ and i cannot decrease her cost any further by misreporting her type. If $T_i = \{\mathsf{F}_1, \mathsf{F}_2\}$, then it can be either $i = \max N_2$ (in which case the algorithm returns $(F_1 = i-1, F_2 = i)$ or $(F_1 = i, F_2 = i+1)$, $cost_i(\mathcal{F}) = 1$ and i cannot decrease her cost any further by lying) or $i < \max N_2$ (in which case $F_1 = i$ and i cannot influence the location of facility F_2).

It is easy to check that case (b) is symmetric to case (a).

If case (c) occurs, then it can be either: $T_i = \{\mathsf{F}_1\}$, $T_i = \{\mathsf{F}_2\}$ or $T_i = \{\mathsf{F}_1, \mathsf{F}_2\}$. If $T_i = \{\mathsf{F}_1\}$, then $i > \min N_1$. It is easy to check that if $\min N_1 \neq \max N_2$ then i cannot influence the location of facility F_1. Let us assume then that $\ell = \min N_1 = \max N_2$. In this case the algorithm outputs either $(F_1 = \ell, F_2 = \ell - 1)$ or $(F_1 = \ell + 1, F_2 = \ell)$. In either case, if $T'_i = \emptyset$ the output of the algorithm does not change, whereas if $\mathsf{F}_2 \in T'_i$ then the algorithm outputs $(F'_1 = \ell, F'_2 = i)$ (as $i > \max N_2$) and $cost_i(F(\mathcal{T})) \leq cost_i(F(T'_i, \mathcal{T}_i))$. It is easy to check that the case when $T_i = \{\mathsf{F}_2\}$ is symmetric to the case when $T_i = \{\mathsf{F}_1\}$.

If $T_i = \{\mathsf{F}_1, \mathsf{F}_2\}$ then $\min N_1 < i < \max N_2$, and it is easy to check that i cannot influence the outcome of the algorithm. □

In order to prove the approximation guarantee of TWOEXTREMES, we initially prove a lower bound on the value of the optimal social cost.

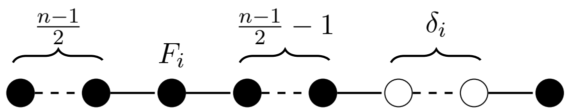

Figure 3: Bounding OPT_i from below

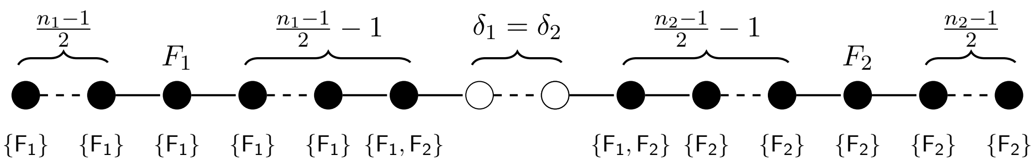

Figure 4: Bounding OPT from below

Lemma 4.1. *Let $\mathcal{T}$ be an instance of the 2-facility location problem, such that $n_1 = |N_1|$, $n_2 = |N_2|$, and $\delta_i \geq 0$ is the number of empty nodes in in the interval $[\min N_i, \max N_i]$. Then the following holds: $cost(OPT(\mathcal{T})) \geq \frac{n_1^2}{4} + \frac{n_2^2}{4} - \frac{1}{2} + \delta_1 + \delta_2$.*

Proof. We first consider the problem of allocating only one facility. Let us consider the instance depicted in Figure 3, where we have a sequence of $n-1$ adjacent agents requesting the facility, δ empty nodes and then the n-th agent requesting the facility. For the sake of exposition, le us assume n is odd. In this configuration, the minimum social cost is obtained when the facility is located at the median node (node $\frac{n-1}{2} + 1$). We argue that this configuration, along with its specular counterpart where (some of) the empty nodes are located between agent 1 and agent 2, yields the least social cost. In this configuration only the n–th agent incurs the cost of travelling through the empty nodes. Let us suppose we move one empty node to another location, say between agent $\ell - 1$ and agent ℓ, still to the right of the median node, i.e. $\frac{n-1}{2} + 1 \leq \ell - 1 < n$. Then the n–th node will still have to travel through the same number of empty nodes as before, but now $n - \ell - 1$ nodes will have to travel through an empty node, which increases the social cost of $n - \ell - 1$. If we move the empty node to a location between agent $\ell - 1$ and agent ℓ to the left of the median node, i.e. $1 < \ell \leq \frac{n-1}{2} + 1$, then agent n will have to travel through $\delta - 1$ empty nodes, whereas $\ell - 1$ agents will have to travel through 1 empty node, which increases the social cost of $\ell - 1$. The same reasoning applies if we move more than one empty node.

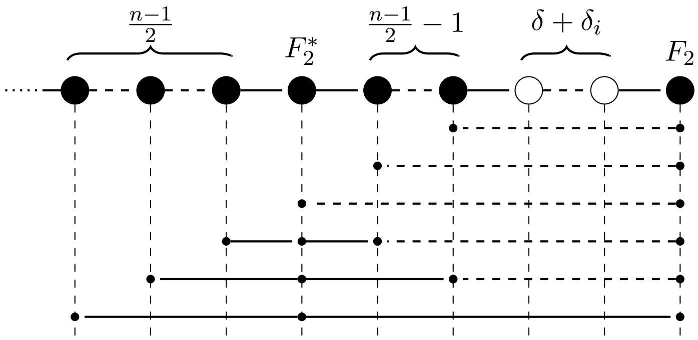

Figure 5: Computing $cost(\mathcal{LR}_2(\mathcal{T}))$. Full edges denote links used by OPT_2 while dashed edges denote links used in $cost(\mathcal{LR}_2(\mathcal{T})) - OPT_2$.

Let us take into consideration the family of instances depicted in Figure 4. It is easy to see that this instance is a composition of two min-cost instances of the type depicted in Figure 3, and that this family of instances yields the least cost for the social cost objective function when locating two facilities. It can be easily checked that the following holds:

$$OPT \geq 2 \sum_{i=1}^{\frac{n_1-1}{2}} i + \delta_1 + 2 \sum_{i=1}^{\frac{n_2-1}{2}} i + \delta_2 = \frac{n_1^2}{4} + \frac{n_2^2}{4} - \frac{1}{2} + \delta_1 + \delta_2,$$

where $2 \sum_{i=1}^{\frac{n_j-1}{2}} i + \delta_j$ represents the social cost of the instance in Figure 3. This concludes the proof. □

Theorem 4.3. *Algorithm* TWOEXTREMES *is* $(n-1)$*-approximate for the social cost objective function.*

Proof. Let us consider a generic instance $\mathcal{T}$. Moreover, let (F_1^*, F_2^*) be an optimal solution for such an instance, and let $cost(OPT(\mathcal{T})) = OPT_1 + OPT_2$, where $OPT_1 = \sum_{i \in N_1} d(i, F_1^*)$ and $OPT_2 = \sum_{i \in N_2} d(i, F_2^*)$ denote the cost incurred by the agents to connect to facility F_1 and F_2, respectively. Let $\mathcal{LR}(\mathcal{T})$ be the solution output by TWOEXTREMES on input $\mathcal{T}$ and let $(F_1 = \mathcal{LR}_1(\mathcal{T}), F_2 = \mathcal{LR}_2(\mathcal{T}))$ denote the locations that $\mathcal{LR}(\mathcal{T})$ computes for the two facilities. We can express the cost of location (F_1, F_2) as a function of the optimal allocation (F_1^*, F_2^*) as follows:

$$\begin{aligned} cost(\mathcal{LR}(\mathcal{T})) = OPT_1 + 2 \sum_{i \in N_1^L \setminus F_1} d(i, F_1) + d(F_1, F_1^*) \\ + OPT_2 + 2 \sum_{i \in N_2^R \setminus F_2} d(i, F_2) + d(F_2, F_2^*), \end{aligned}$$

where N_j^R (N_j^L, respectively) denotes the set of nodes in $N_j[\mathcal{T}]$ to the right (left, respectively) of F_j^* (excluding the node where F_j^* is located). Figure 5 gives the geometric intuition behind this equality. We note that F_1 and F_2 are excluded from the summation as $d(F_1, F_1) = d(F_2, F_2) = 0$.

We can then observe that:

$$\begin{aligned} cost(\mathcal{LR}(\mathcal{T})) \leq & OPT + (n_1 - 3) \cdot d(F_1, F_1^*) + d(F_1, F_1^*) \\ & + (n_2 - 3) \cdot d(F_2, F_2^*) + d(F_2, F_2^*) \\ \leq & OPT + (n - 2) \cdot (d(F_1, F_1^*) + d(F_2, F_2^*)) \end{aligned}$$

where: (*i*) the first inequality follows from: upper-bounding $d(i, F_1)$ and $d(i, F_2)$, respectively, by $d(F_1, F_1^*)$ and $d(F_2, F_2^*)$ and observing that $2 \cdot |N_j^R \setminus F_j| = 2(\lceil \frac{n_j}{2} \rceil - 2) \leq 2(\frac{n_j+1}{2} - 2) = n_j - 3$;

whereas the second inequality follows from upper-bounding n_1 and n_2 by $n-1$ (i.e., $\max\{n_1, n_2\} \leq n$ since $|N_1| > 0$ and $|N_2| > 0$). In order to upper bound $d(F_1, F_1^*)$ and $d(F_2, F_2^*)$, let us consider the instance depicted in Figure 6, where α is the number of "empty" nodes between F_2^* and F_1^*, α_1 is the number of empty nodes between F_1 and F_2^* and α_2 is the number of empty nodes between F_1^* and F_2. We argue that this instance is the one exhibiting the maximum possible distance $d(F_1, F_1^*)$ (respectively, $d(F_2, F_2^*)$) as between F_1 and F_1^* (F_2 and F_2^*, respectively) there is the maximum possible number of agents requesting facility F_2 (respectively, F_1) and the maximum number of empty nodes (the number of agents requesting F_1 is dictated by the fact that F_1^* is the median node of N_1). It is easy to check that $d(F_1^*, F_1) \leq (\frac{n_1}{2} + n_2 - 1 + \alpha + \alpha_1)$ and $d(F_2^*, F_2) \leq (\frac{n_2}{2} + n_1 - 1 + \alpha + \alpha_2)$, which applied to the last inequality yields:

$$cost(\mathcal{LR}(\mathcal{T})) \leq OPT + (n-2)\left(\frac{3}{2}(n_1 + n_2) + 2\alpha + \alpha_1 + \alpha_2\right).$$

In virtue of Lemma 4.1, by observing that $\alpha + \alpha_i \leq \delta_i$, we conclude that $\frac{3}{2}(n_1+n_2)+2\alpha+\alpha_1+\alpha_2$ is bounded from above by OPT. Applying the above lower bound to the last inequality yields the following: $cost(\mathcal{LR}(\mathcal{T})) \leq (n-1) \cdot OPT$ which proves the claim. □

We finish this section by proving that the analysis of TWOEXTREMES presented above is tight.

Theorem 4.4. *The upper bound of Theorem 4.3 for the* TWOEXTREMES *algorithm is tight.*

Proof. We are going to exhibit an instance for which the TWOEXTREMES algorithm obtains an approximation ratio of $(n-1)$. The instance we consider is the one depicted in Figure 7 and is such that $|N_1| = n$, $|N_2| = 1$ and n is odd. The number of nodes of the graph is $n + \delta$, where δ is the number of empty nodes. The declarations, depicted in brackets below each nodes are as follows: $T_i = \{\mathsf{F}_1\}$ for each $1 \leq i < n$, $T_n = \{\mathsf{F}_1, \mathsf{F}_2\}$. As before, (F_1^*, F_2^*) and (F_1, F_2) denote the optimal allocation and the outcome of the TWOEXTREMES algorithm, respectively. It is easy to check that (1) gives the cost of the optimal location, whereas (2) gives the cost of (F_1, F_2):

$$cost(OPT(\mathcal{T})) = 2 \cdot \sum_{i=1}^{\frac{(n-1)}{2}} (i) + \delta = \frac{n^2 - 1 + 4\delta}{4} \tag{1}$$

$$\begin{aligned} cost(\mathcal{LR}(\mathcal{T})) &= \sum_{i=1}^{n-1} (\delta + i + 1) \\ &= \frac{n^2 - 3n + 2(n-1)\delta - 2}{2}. \end{aligned} \tag{2}$$

Equation (3) below expresses the approximation ratio of the TWOEXTREMES algorithm with respect to the instance of Figure 7 as a function of both the number of players n and the number of empty nodes δ.

$$\alpha(n, \delta) = 2 \cdot \frac{n^2 - 3n + 2(n-1)\delta - 2}{n^2 - 1 + 4\delta} \tag{3}$$

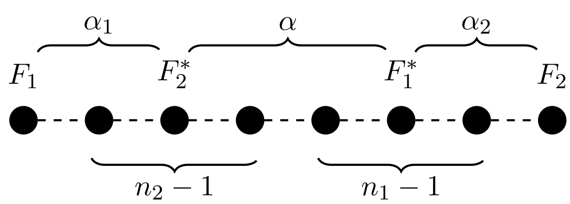

Figure 6: Upper bound to $d(F_1, F_1^*)$ and $d(F_2, F_2^*)$

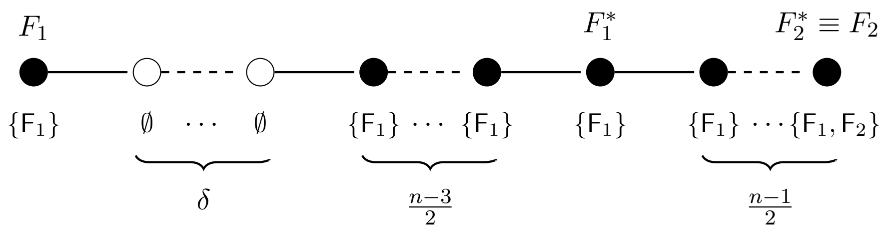

Figure 7: TWOEXTREMES is $\Theta(n-1)$-approximate

We can see from (3) that if $\delta \in \omega(n^2)$ then $\alpha(n, \delta)$ tends to $n-1$. □

4.2 Randomized Mechanisms for the Social Cost Objective Function

In this section we present and analyze RANDOPT, a truthful randomized optimal algorithm for the 2-facility location problem. To adequately describe the algorithm, we first need to define some concepts and ideas of interest. For notational convenience, in this section we let the index of the two facilities be binary and all the operations involving indexes be modulo 2. Hence, we will refer indistinctly to one facility as F_k and to the other one as F_{k+1}.

Definition 4.1. We denote as S_k the set of *optimal locations* on G for facility $\mathsf{F_k}$ taking into consideration the requests for facility F_k alone.

By the results in [13], we know that the optimal location for a single facility is the *median point*, and therefore the set of optimal locations S_k is either a singleton, i.e., when the number of requests is odd, or has size greater than 1, i.e., when the number of requests is even.[4] We observe that when only one agent changes her declaration for facility $\mathsf{F_k}$, i.e. $N_k \neq N'_k$, due to the change of the parity of N_k we have that if $|S_k| = 1$ then $|S'_k| \neq 1$ and, vice versa, if $|S_k| \neq 1$ then $|S'_k| = 1$. We will extensively use this simple observation in the rest of this section, and refer to it as the *parity argument*.

Definition 4.2. A solution F_k is *extreme* for S_k w.r.t. S_{k+1}[5] if: (*i*) $|S_k| = 2$; (*ii*) $|S_k \cap S_{k+1}| \leq 1$; and (*iii*) $F_k = \text{argmax}_{\ell \in S_k}\{d(\ell, S_{k+1})\}$, where $d(\ell, S_{k+1}) = \min_{s \in S_{k+1}} d(\ell, s)$.

The main idea of algorithm RANDOPT is to return either an extreme solution or a mean set solution centered on $\text{avg}(S_k)$ in order to preserve truthfulness: we will show why this is necessary before giving the details of algorithm RANDOPT.

To give an example, let us consider the instance depicted in Figure 8, where $V = N = \{1, 2, 3, 4, 5\}$ and $T_1 = \{\mathsf{F}_0, \mathsf{F}_1\}$, $T_2 = \{\mathsf{F}_0\}$, $T_3 = \{\mathsf{F}_0, \mathsf{F}_1\}$, $T_4 = \{\mathsf{F}_0, \mathsf{F}_1\}$ and $T_5 = \{\mathsf{F}_0\}$. In this case, $S_0 = \{2, 3\}$, $S_1 = \{3, 4\}$ and a mean set solution for F_0 and F_1 centered around $\text{avg}(S_0) = 2.5$ and $\text{avg}(S_1) = 3.5$ is $\mathcal{M} = \{(F_0 = 2, F_1 = 3), (F_0 = 3, F_1 = 4)\}$, which is also a set of *optimal solutions* for the instance. For the same instance, $F_0 = 2$ is an extreme solution for S_0 w.r.t. S_1, as: (*i*) $|S_0| = 2$, (*ii*) $|S_0 \cap S_1| \leq 1$ and (*iii*) $F_0 = \text{argmax}_{\ell \in S_0}\{d(\ell, S_1)\}$; and $F_1 = 4$ is an extreme solution for S_1 w.r.t. S_0 as: (*i*) $|S_1| = 2$, (*ii*) $|S_0 \cap S_1| \leq 1$ and (*iii*) $F_1 = \text{argmax}_{\ell \in S_1}\{d(\ell, S_0)\}$. Let us now consider the instance of Figure 9, where we have the same graph and same agents as before, but the declarations are $T_1 = \{\mathsf{F}_0, \mathsf{F}_1\}$, $T_2 = \{\mathsf{F}_0\}$, $T_3 = \{\mathsf{F}_0, \mathsf{F}_1\}$, $T_4 = \{\mathsf{F}_0\}$ and $T_5 = \{\mathsf{F}_1\}$. In this case, $S_0 = \{2, 3\}$ and $S_1 = \{3\}$. We note that there is no feasible optimal mean set solution centered around $avg(S_0) = 2.5$, as to do so we need to allocate F_0 half of the times on node 2 and half of the times on node 3, whereas node 3 is the only optimal allocation point for facility F_1. The only feasible optimal

[4] All empty nodes between $\min S_k$ and $\max S_k$ are optimal solutions.

[5] For the sake of notational conciseness, when referring to extreme and mean solutions we omit S_k and S_{k+1} as they can be easily deduced from the context.

allocation in this case is $(F_0 = 2, F_1 = 3)$, which is an extreme solution for S_0 w.r.t S_1, as $|S_0| = 2$, $|S_0 \cap S_1| = 1$ and $F_0 = \text{argmax}_{\ell \in S_0}\{d(\ell, S_1)\}$. Furthermore, if we consider the instance in Figure 9 as the truthful instance and, as a consequence of that, we regard agent 4 in the instance of Figure 8 as reporting $T'_4 = \{\mathsf{F}_0, \mathsf{F}_1\}$ instead of her true type $T_4 = \{\mathsf{F}_0\}$, we can easily see that returning a mean set solution in the instance of Figure 8 is not truthful. Indeed, in the instance of Figure 9 the expected cost of agent 4 for the (only) optimal allocation $(F_0 = 2, F_1 = 3)$ is 2, whereas in Figure 8 the true expected cost of agent 4 for the mean set solution for F_0 centered around 2.5 is 1.5. To preserve truthfulness, in the instance of Figure 8 an extreme solution must be returned.

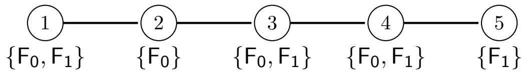

Figure 8: Extreme solution example

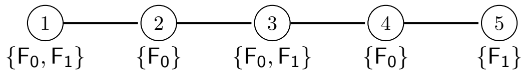

Figure 9: Extreme solution example - truthful instance

We can now discuss algorithm RANDOPT, whose pseudocode is reported in Algorithm 2. The algorithm meaningfully discerns the cases when an optimal mean set solution exists and can be outputted from the cases when an extreme solution must be returned in order to preserve strategyproofness. More specifically, when an extreme solution exists for both facilities (i.e., Line 6), algorithm RANDOPT returns it with probability 1, whereas when an extreme solution exists for one facility only (i.e., Line 9), say facility F_k, it returns a solution extracted uniformly at random from a mean set solution for facility $\mathsf{F}_{\mathsf{k}+1}$ where facility F_k is always allocated as an extreme solution. When no extreme solutions exists for either facility, the algorithm then returns either (*i*) a solution extracted uniformly at random from a mean set solution centered around $\text{avg}(S_k)$ and $\text{avg}(S_{k+1})$ (i.e., Line 20), computed by means of procedure COMPUTEOPTMEANSET, or (*ii*) in the very restricted case when $S_k = S_{k+1} = \{s\}$ and s is the first or last node of the graph, a solution where, due to feasibility reasons, only facility F_k is allocated as a mean set solution whereas the other facility is allocated to the nearest node (i.e., lines 14 and 16).

Procedure COMPUTEOPTMEANSET, whose pseudocode is reported in Algorithm 3, takes as input the sets of optimal locations S_k and S_{k+1} and returns a mean set solution for F_k and $\mathsf{F}_{\mathsf{k}+1}$ centered around $\text{avg}(S_k)$ and $\text{avg}(S_{k+1})$, respectively. It constructs mean set solutions by carefully considering the structure of S_k and S_{k+1} in order to avoid unfeasible solutions. The pseudocode of procedure COMPUTEOPTMEANSET is an algorithmic transposition of case $c.3$ of the proof of Theorem 4.5, which offers a constructive proof of the existence of mean set solutions.

The remainder of this section is devoted to prove two important properties of algorithm RANDOPT, namely: its optimality (Theorem 4.5) and its strategyproofness (Theorem 4.6).

Theorem 4.5. *Algorithm* RANDOPT *always returns an optimal solution for the social cost objective function.*

Proof. RANDOPT returns solutions that are extracted from the set of optimal solutions S_k and S_{k+1}, which proves by definition the optimality of the algorithm[6]. What is left to be proven is that the solutions returned by the algorithm are actually feasible. The solutions returned by RANDOPT are always feasible

[6] Strictly speaking, this is always true but in one case: when $|S_k| = 1$, $|S_{k+1}| = 1$ and $|S_k \cap S_{k+1}| = 1$, as the solutions computed by RANDOPT at lines 14, 16 and 19 (when the mean set returned by COMPUTEOPTMEANSET is computed at Line 7) allocate facility F_1 at the nodes that are immediately to the left (i.e., $s-1$) and right (i.e., $s+1$) of node $s = S_k \cap S_{k+1}$. It is easy to check that, in this special case, $F_1 = s-1$ and $F_1 = s+1$, although suboptimal for the problem of locating facility F_1 alone, are optimal for the 2-facility location problem.

Algorithm 2: RANDOPT

Require: Line G, facilities $\mathfrak{F} = \{\mathsf{F}_1, \mathsf{F}_2\}$, declarations $\mathcal{T} = \{T_1, \ldots, T_n\}$
Ensure: $F(\mathcal{T})$ optimal allocation for 2-facility location on G

1: $\forall k \; S_k :=$ OPT$(N_k[\mathcal{T}])$
2: **if** $\exists k \in \{0, 1\}$ s.t. $|S_k| = 2$ **and** $|S_k \cap S_{k+1}| \leq 1$ **then**
3: $\quad F_k := \underset{v \in S_k}{\text{argmax}}\{\, d(v, S_{k+1})\}$
4: $\quad$ **if** $|S_{k+1}| = 2$ **then**
5: $\quad\quad F_{k+1} := \underset{v \in S_{k+1}}{\text{argmax}}\{\, d(v, S_k)\}$
6: $\quad\quad$ **return** (F_k, F_{k+1}) w.p. 1
7: $\quad$ **else**
8: $\quad\quad \mathcal{M} = \{(F_k, \min(S_{k+1})), (F_k, \max(S_{k+1}))\}$
9: $\quad\quad$ **return** $(F_k, F_{k+1}) \in \mathcal{M}$ w.p. $1/|\mathcal{M}|$
10: $\quad$ **end if**
11: **else**
12: $\quad$ **if** $|S_k| = |S_{k+1}| = 1$ **and** $S_k = S_{k+1} = \{s\}$ **and (** $s - 1 =$ NIL **or** $s + 1 =$ NIL **) then**
13: $\quad\quad$ **if** $s - 1 =$ NIL **then**
14: $\quad\quad\quad$ **return** $(F_0 = s, F_1 = s + 1)$ w.p. 1
15: $\quad\quad$ **else**
16: $\quad\quad\quad$ **return** $(F_0 = s, F_1 = s - 1)$ w.p. 1
17: $\quad\quad$ **end if**
18: $\quad$ **else**
19: $\quad\quad \mathcal{M} :=$COMPUTEOPTMEANSET$(S_k, S_{k+1})$
20: $\quad\quad$ **return** $(F_k, F_{k+1}) \in \mathcal{M}$ w.p. $1/|\mathcal{M}|$
21: $\quad$ **end if**
22: **end if**

whenever $S_k \cap S_{k+1} = \emptyset$ (Line 2 of Algorithm 3, invoked at Line 20 of algorithm RANDOPT), as a clash in the allocation of the two facilities cannot occur, so in the remainder we are going to assume that $S_k \cap S_{k+1} \neq \emptyset$. The solutions returned by RANDOPT are also feasible by construction when $S_k = S_{k+1} = \{s\}$ and either $s - 1 =$ NIL or $s + 1 =$ NIL (lines 14 and 16), so we can also assume in the remainder that this case does not occur. We need to consider three cases: (c.1) both facilities are allocated as extreme solutions (Line 6 of algorithm RANDOPT), denoted in the sequel as (E, E); (c.2) one facility is allocated as an extreme solution while the other facility is allocated as a mean solution (Line 9 of algorithm RANDOPT), referred to as either (E, M) or (M, E); and (c.3) both facilities are allocated as mean solutions (Line 20 of Algorithm RANDOPT), denoted as (M, M).

In Line 6 (case c.1) Algorithm RANDOPT allocates both facilities as extreme solutions, so $|S_k| = 2$, $|S_{k+1}| = 2$ and $|S_k \cap S_{k+1}| \leq 1$. Let us suppose w.l.o.g. that $S_k = \{l, l+1\}$ $S_{k+1} = \{l+1, l+2\}$. It is easy to check that $(l, l+2)$, where the first (second, respectively) element of the ordered couple denotes the location of facility F_k (F_{k+1}, respectively), is a feasible extreme solution for F_k and F_{k+1}.

In Line 9 (case c.2) algorithm RANDOPT allocates a facility as an extreme solution and the other one as a mean solution. W.l.o.g. let us suppose that F_k is allocated as an extreme solution and F_{k+1} is allocated as a mean solution. Therefore, we have $|S_k| = 2$, $|S_k \cap S_{k+1}| \leq 1$ and $|S_{k+1}| \neq 2$. Let us denote $S_k = \{l, l+1\}$ and let us suppose w.l.o.g. that $S_k \cap S_{k+1} = \{l+1\}$ (i.e., the case when $S_k \cap S_{k+1} = \{l\}$ is symmetric). There are two cases to consider: (i) $|S_{k+1}| = 1$, (ii) $|S_{k+1}| > 2$. We notice that in both cases $F_k = l$ is a feasible extreme solution for S_k. When case (i) occurs, then $S_{k+1} = \{l+1\}$ and $\mathcal{M} = \{(l, l+1)\}$ is a feasible mean set for S_{k+1}. When case (ii) occurs, $\mathcal{M} = \{(l, \min(S_{k+1})), (l, \max(S_{k+1}))\}$ is a feasible mean set for S_{k+1}.

In Line 20 (case $c.3$) algorithm RANDOPT invokes COMPUTEOPTMEANSET and returns an (M, M) solution, so either (*i*) $|S_k| \neq 2$ and $|S_{k+1}| \neq 2$ or (*ii*) $|S_k \cap S_{k+1}| > 1$. Let us consider case (*i*). Let us suppose that $|S_k| > 2$. If allocations $(\min(S_k), \min(S_{k+1}))$ and $(\max(S_k), \max(S_{k+1}))$ are feasible (Line 14 of COMPUTEOPTMEANSET) then $\mathcal{M} = \{(\min(S_k), \min(S_{k+1})), (\max(S_k), \max(S_{k+1}))\}$ is trivially a mean set, and the claim is true. The same holds if $(\min(S_k), \max(S_{k+1}))$ and $(\max(S_k), \min(S_{k+1}))$ are feasible (Line 16 of COMPUTEOPTMEANSET). If neither of the previous holds, then $\min(S_{k+1}) = \max(S_{k+1})$, hence $S_{k+1} = \{s\}$ and $s \in \{\min(S_k), \max(S_k)\}$ (Line 11 of COMPUTEOPTMEANSET). Then both $(\min(S_k) + 1, s)$ and $(\max(S_k) - 1, s)$ are optimal and $\mathcal{M} = \{(\min(S_k) + 1, s), (\max(S_k) - 1, s)\}$ is a mean set. Let us consider the case when $|S_k| = 1$, and let $S_k = \{s\}$. If $|S_{k+1}| = 1$ then $S_k = S_{k+1}$ (Line 7 of COMPUTEOPTMEANSET). We note that in this case $\{(s, s-1), (s, s+1)\}$ is a feasible mean set for both S_k and S_{k+1}. If $|S_k| = 1$ and $|S_{k+1}| > 2$, this case is analogous to the case when $|S_k| > 2$ and $|S_{k+1}| = 1$ that we analysed above. Let us now consider case (*ii*) (Line 20 of COMPUTEOPTMEANSET). Since $|S_k \cap S_{k+1}| > 1$, then $|S_k| \geq 2$ and $|S_{k+1}| \geq 2$. Then, either $\{(\min(S_k), \min(S_{k+1})), (\max(S_k), \max(S_{k+1}))\}$ (Line 22 of COMPUTEOPTMEANSET) or $\{(\min(S_k), \max(S_{k+1})), (\max(S_k), \min(S_{k+1}))\}$ (Line 24 of COMPUTEOPTMEANSET) is a feasible mean set for both S_k and S_{k+1}. □

We are now going to prove that algorithm RANDOPT is strategyproof. Before we can do that, we need to prove four auxiliary lemmata.

Lemma 4.2 states that if an agent i "hides" facility $\mathsf{F_k}$ from her declaration, then the average point $\text{avg}(S'_k)$ is located further away from i than $\text{avg}(S_k)$. Put in other words, if facility $\mathsf{F_k}$ is assigned as mean set solution, agent i cannot gain by misreporting on $\mathsf{F_k}$.

Lemma 4.2. *If $i \in N_k$ and $N'_k = N_k \setminus \{i\}$, then $d(i, \text{avg}(S_k)) < d(i, \text{avg}(S'_k))$.*

Proof. Let us consider the case when $i \leq \min\{S_k\}$, the case when $i \geq \max\{S_k\}$ is symmetric. If $|S_k| = 1$, let s_k denote the sole element of S_k. If $i \notin N'_k$, then $|S'_k| > 1$ by the parity argument and S'_k is such that $\min(S'_k) = s_k$ and $\max(S'_k) = \ell$, where $\ell \in N'_k$ is the location of the leftmost agent such that $\ell > s_k$ and $k \in T_\ell$. Clearly, $i \leq \text{avg}(S_k) < \text{avg}(S'_k)$, which implies the claim. If $|S_k| > 2$, then $|S'_k| = 1$ by the parity argument. If $i \notin N'_k$ then $S'_k = \{\max(S_k)\}$ from which it follows that $i \leq \text{avg}(S_k) < \text{avg}(S'_k)$, and the claim. □

Lemma 4.3 states that if an agent does not report facility $\mathsf{F_k}$ then the set of optimal allocations S'_k does not get any closer.

Lemma 4.3. *Let $i \in N$ be an agent such that $i \in N_k$, $N'_k = N_k \setminus \{i\}$. Then $\min_{\ell \in S_k}\{d(i, \ell)\} \leq \min_{\ell' \in S'_k}\{d(i, \ell')\}$.*

Proof. Let us assume that $i \leq \min(S_k)$. S_k can be either a singleton (if $|N_k|$ is odd) or have cardinality greater than 1 (if $|N_k|$ is even). Let $S_k = \{s_k\}$, then $|S'_k| > 1$ by the parity argument. Let $r = \max(S'_k)$. The thesis holds since $\min_{\ell \in S_k} |i - \ell| = |i - s_k| = \min_{\ell' \in S'_k} |i - \ell'|$. If $|S_k| > 1$, let $l = \min(S_k)$ and $r = \max(S_k)$. Then $S'_k = \{r\}$. The thesis holds in this case since $\min_{\ell \in S_k} |i - \ell| = |i - l| < |i - r| = \min_{\ell' \in S'_k} |i - \ell'|$. The same argument holds for the case when $i \geq \max(S_k)$. Finally, we observe that when $\min(S_k) < i < \max(S_k)$ then $i \notin N_k$. □

In essence, the previous lemma states that in a monodimensional setting if an agent does not declare a facility she is interested in, the space of optimal allocation points gets further away from her.

Lemma 4.4. *Let F_k and F'_k be two extreme solutions. Then it must be $F_k = F'_k$.*

Proof. We note that since F_k and F'_k are by hypothesis two extreme solutions, by definition $|S_k| = |S'_k| = 2$. We argue that it must be $S_k = S'_k = \{l, r\}$. Indeed, if only one agent changes her declaration between S_k and S'_k either $|S_k| \neq 2$ or $|S'_k| \neq 2$ Let us suppose w.l.o.g. that $F_k = r$. Since both F_k and F'_k are extreme solutions, it must be the case that $|S_k \cap S_{k+1}| \leq 1$ and $|S'_k \cap S'_{k+1}| \leq 1$. This implies that $s \leq l < r$, where s is the element of S_{k+1} nearest to S_k. Let us suppose, for the sake of contradiction, that $F'_k = l$. In this case it must be $l < r \leq s'$, where s' is the element of S'_{k+1} nearest to S_k. We observe that whenever this happens $s \in S'_{k+1}$ (as we assume that only one agent changes her declaration), which implies that $|S'_k \cap S'_{k+1}| \geq 2$ and contradicts the hypothesis that F'_k is an extreme solution. □

The previous lemma essentially states that an agent cannot gain on a facility assigned as an extreme solution, unless she changes the declaration for that facility.

Lemma 4.5. *Let $|S_k| = 1$, F'_k be an extreme solution for S'_k, and let $N'_k = N_k \setminus \{i\}$. Then $d(F_k, i) \leq d(F'_k, i)$.*

Proof. Since $F_k = \mathrm{argmin}_{\ell \in S_k}\{d(i, \ell)\}$, and since, in the best case for agent i, $F'_k = \mathrm{argmin}_{\ell' \in S'_k}\{d(i, \ell')\}$, by Lemma 4.3 $d(i, F_k) \leq d(i, F'_k)$. □

We now ready to prove that the algorithm RANDOPT is truthful.

Theorem 4.6. *Algorithm* RANDOPT *is SP.*

Proof. Consider the outcomes $\mathcal{F} = \text{RANDOPT}(\mathcal{T})$ and $\mathcal{F}' = \text{RANDOPT}(T'_i, \mathcal{T}_{-i})$. We next show that $cost_i(\mathcal{F}) \leq cost_i(\mathcal{F}')$. Assume by contradiction that $cost_i(\mathcal{F}) > cost_i(\mathcal{F}')$; this implies that there exists at least a facility $k \in \{0, 1\}$ such that $d(i, E[F_k]) > d(i, E[F'_k])$, where F_k (F'_k, respectively) denotes the position of facility k in $\mathcal{F}$ ($\mathcal{F}'$, respectively). Recall that we will denote as S_k and S'_k the optimal locations of facility k in the instances $\mathcal{T}$ and $(T'_i, \mathcal{T}_{-i})$, respectively. Let us assume for now that algorithm RANDOPT always returns either a mean set solution or an extreme solution for both facilities. We will analyze the special case when this is infeasible (lines 14 and 16 of algorithm RANDOPT) at the end of the proof.

We denote a possible *output transition* of RANDOPT as $(F_0, F_1) \to (F'_0, F'_1)$, where the left-hand side pair denotes the outcome of the algorithm when each agent reports truthfully, whereas the right-hand side pair denotes the outcome of the algorithm when agent i misreports her type. It can be easily showed that all possible output transitions of algorithm RANDOPT (with the exception of a particular case treated at the end of the proof) can be represented by the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that $\mathcal{V} = \{(E, E), (E, M), (M, E), (M, M)\}$ and $\mathcal{E} = \mathcal{V} \times \mathcal{V} \setminus \{((E, E), (E, E))\}$, where E and M stand, respectively, for an extreme and a mean set solution being returned. Notice that the set of arcs of $\mathcal{G}$ comprises all possible transitions but $(E, E) \to (E, E)$: we are now going to prove that this transition cannot occur. Firstly, we notice that if a solution of type (E, E) is returned then either $S_k \neq S'_k$ or $S_{k+1} \neq S'_{k+1}$, since if $S_k = S'_k$ and $S_{k+1} = S'_{k+1}$ then the output of the algorithm does not change and strategyproofness trivially holds. Let us suppose w.l.o.g. that $S_k \neq S'_k$ (i.e. the case when $S_{k+1} \neq S'_{k+1}$ is symmetric). If F_k is an extreme solution for S_k and $S_k \neq S'_k$ then F'_k is a mean solution for S'_k, which would result in a transition $(E, E) \to (M, E)$. To prove the claim, we are now going to prove that every arc of $\mathcal{G}$ represents an *SP* transition.

It can be easily verified that transition $(M, M) \to (M, M)$ is *SP* by Lemma 4.2. Indeed, we can regard transition $(M, M) \to (M, M)$ as two $M \to M$ transitions. According to Lemma 4.2 if an agent misreports hiding a facility she is interested in from her bid, then her distance from that facility increases. Transitions $(M, E) \to (M, E)$ and $(E, M) \to (E, M)$ are *SP* by Lemmata 4.2 and 4.4. Indeed, they can be regarded as two transitions $M \to M$ and $E \to E$. As before, Lemma 4.2 implies transition $M \to M$ is *SP*, as no agent can improve her utility function by misreporting on a facility allocated as a mean set (transition $M \to M$). Transition $E \to E$ is *SP* by Lemma 4.4, which implies that no agent can improve her utility function on a facility allocated as an extreme solution if, when the agent lies, the facility is still allocated as an extreme solution (transition $E \to E$).

Algorithm 3: ComputeOptMeanSet

Require: S_k and S_{k+1} sets of optimal locations for $\mathsf{F_k}$ and $\mathsf{F_{k+1}}$.
Ensure: a set of allocations $\mathcal{M}$ such that when an allocation (F_1, F_2) is extracted uniformly at random from $\mathcal{M}$ the expected value of F_1 and F_2 is, respectively, $\mathrm{avg}(S_1)$ and $\mathrm{avg}(S_2)$.

if $S_k \cap S_{k+1} = \emptyset$ **then**
  return $\mathcal{M} = \{(\min S_k, \min S_{k+1}), (\max S_k, \max S_{k+1})\}$
end if
if $|S_k| \neq 2$ **and** $|S_{k+1}| \neq 2$ **and** $|S_k \cap S_{k+1}| = 1$ **then**
  if $|S_k| = 1$ **and** $|S_{k+1}| = 1$ **then**
    $s = S_k \cap S_{k+1}$
    return $\mathcal{M} = \{(F_0 = s, F_1 = s + 1), (F_0 = s, F_1 = s - 1)$
  end if
  if $|S_k| > 2$ **and** $|S_{k+1}| = 1$ **then**
    $s = S_k \cap S_{k+1}$
    return $\mathcal{M} = \{(F_k = \min(S_k) + 1, F_{k+1} = s), (F_k = \max(S_k) - 1, F_{k+1} = s)\}$
  end if
  if $\min(S_k) \neq \min(S_{k+1})$ **and** $\max(S_k) \neq \max(S_{k+1})$ **then**
    return
    $\mathcal{M} = \{(F_k = \min(S_k), F_{k+1} = \min(S_{k+1})), (F_k = \max(S_k), F_{k+1} = \max(S_{k+1}))\}$
  end if
  if $\min(S_k) \neq \max(S_{k+1})$ **and** $\max(S_k) \neq \min(S_{k+1})$ **then**
    return
    $\mathcal{M} = \{(F_k = \min(S_k), F_{k+1} = \max(S_{k+1})), (F_k = \max(S_k), F_{k+1} = \min(S_{k+1}))\}$
  end if
else
  if $|S_k \cap S_{k+1}| > 1$ **then**
    if $\min(S_k) \neq \min(S_{k+1})$ **and** $\max(S_k) \neq \max(S_{k+1})$ **then**
      return
      $\mathcal{M} = \{(F_k = \min(S_k), F_{k+1} = \min(S_{k+1})), (F_k = \max(S_k), F_{k+1} = \max(S_{k+1}))\}$
    else
      return
      $\mathcal{M} = \{(F_k = \max(S_k), F_{k+1} = \min(S_{k+1})), (F_k = \min(S_k), F_{k+1} = \max(S_{k+1}))\}$
    end if
  end if
end if

We note that we can regard $(M, M) \to (E, M)$ and $(M, M) \to (M, E)$ as one case, as in both cases one facility makes a transition $M \to M$ and the other one makes a transition $M \to E$. Lemma 4.2 assures that transition $M \to M$ is *SP*. Let us focus then on transition $M \to E$. Two cases can occur: (*i*) $S_k = S'_k$, in which case it must be $|S_k| = |S'_k| = 2$ and (*ii*) $S_k \neq S'_k$, in which case $|S_k| = 1$ and $|S'_k| = 2$ by the parity argument. In case (*i*), let $S_k = \{l, l+1\}$. We notice that $|S_k \cap S_{k+1}| > 1$ and $E[F_k] = \mathrm{avg}(S_k) = l + \frac{1}{2}$ must hold, the latter since F_k is assigned by assumption as a mean set solution. Let us suppose w.l.o.g. $i \leq l$ (i.e. the case when $i \geq l + 1$ is symmetric). We note that i can gain on $\mathsf{F_k}$ only if $F'_k = l$ (i.e., the nearest node of S'_k to i is selected as extreme solution when i lies). In order for this to happen, S'_{k+1} needs to be to the right of S'_k, which implies that $l < l + 1 \leq s$ where s is the nearest point of S'_{k+1} to S_k. As we assume no other agent changes her bid, this can only happen if $\mathsf{F}_{k+1} \in T_i$ and $\mathsf{F}_{k+1} \notin T'_i$ and, as a consequence of that, S'_{k+1} moves to the right with respect to S_{k+1}. It follows that $E[|F'_k - i|] = E[|F_k - i|] - \frac{1}{2}$ but $E[|F'_{k+1} - i|] \geq E[|F_{k+1} - i||] + \frac{1}{2}$ (since

S'_{k+1} moves away from i, $\mathrm{avg}(S'_{k+1})$ moves away of at least 1/2), which implies that $cost_i(F'_k, F'_{k+1}) = d(i, E[F'_k]) + d(i, E[F'_{k+1}]) \geq d(i, E[F_k]) + d(i, E[F_{k+1}]) = cost_i(F_k, F_{k+1})$. In case (*ii*), we note that $\mathsf{F}_k \in T_i$ (i.e., if $\mathsf{F}_k \notin T_i$ the location of facility F_k is irrelevant to agent i) and $\mathsf{F}_k \notin T'_i$. Since, $|S_k| = 1$, F'_k is an extreme solution for S'_k and $N'_k = N_k \setminus \{i\}$, by Lemma 4.5 this transition is *SP*.

We note that we can regard cases $(E, M) \to (E, E)$ and $(M, E) \to (E, E)$ as the same case, since in both cases we have a transition $E \to E$ and a transition $M \to E$. We notice that transition $E \to E$ is *SP* by Lemma 4.4. Let us now focus on transition $M \to E$. We notice that in this case $|S_k| = 1$ and F'_k is an extreme solution. By Lemma 4.5 this transition is *SP*.

We note we can regard $(E, E) \to (E, M)$ and $(E, E) \to (M, E)$ as one case, in both cases we have a transition $E \to E$ and a transition $E \to M$. For the transition $E \to E$, we note that by Lemma 4.4 the facility does not move, so truthfulness is preserved. Let us now analyse transition $E \to M$. Agent i can only gain if $\mathsf{F}_k \in T_i$, so the only possible lie is $T'_i = T_i \setminus \{k\}$. Since F_k is an extreme solution, $|S_k| = 2$. Let us denote $S_k = \{l, l+1\}$. Let us suppose w.l.o.g. that $i \leq l < l+1$ (i.e., the case when $l < l+1 \leq i$ is symmetric). When i misreports her type, the number of requests for F_k becomes odd, so $|S'_k| = 1$. It is easy to check that $S'_k = \{l+1\}$ and $i < F_k \leq F'_k$, which implies that $d(i, F_k) \leq d(i, F'_k)$.

Let us now consider the case $(M, M) \to (E, E)$. We notice that in this case $|S_k| \neq 2$ and $|S_{k+1}| \neq 2$. To prove this, let us suppose for the sake of contradiction that $|S_k| = 2$. In order to have an (M, M) pair it must be the case that $|S_k \cap S_{k+1}| > 1$ which implies that $|S_{k+1}| \geq 2$. We notice that in this case $|S'_{k+1}| = 1$ by the parity argument, which would not result in a (E, E) pair. The same argument holds if we assume by contradiction that $|S_{k+1}| = 2$. Let us then consider the case when $|S_k| \neq 2$ and $|S_{k+1}| \neq 2$. We highlight that, since $|S_k| \neq 2$ and $|S_{k+1}| \neq 2$ but $|S'_k| = |S'_{k+1}| = 2$, it must be that $N_k \neq N'_k$ and $N_{k+1} \neq N'_{k+1}$, from which by the parity argument it follows that $|S_k| = |S_{k+1}| = 1$. Furthermore it must be the case that $\mathsf{F}_k \in T_i$ (i.e. otherwise the location of facility F_k would be irrelevant for the cost of agent i) and $\mathsf{F}_k \notin T'_i$. We can apply to both $M \to E$ transitions Lemma 4.5 to show that strategyproofness is preserved.

Let us now consider the case $(E, E) \to (M, M)$. We have $|S_k| = 2$, $|S_{k+1}| = 2$. We are going to prove that $S_k \neq S'_k$ and $S_{k+1} \neq S'_{k+1}$. For the sake of contradiction, if $S_k = S'_k$, then $S_{k+1} \neq S'_{k+1}$ and $|S'_{k+1}| = 1$. Since F'_k must be a mean solution for S'_k, it must be that $|S'_k \cap S'_{k+1}| > 1$ (as otherwise F'_k would be allocated as an extreme solution), which is a contradiction, since $|S'_{k+1}| = 1$. Furthermore, we can assume that $\mathsf{F}_k \in T_i$ (i.e., otherwise the location of facility F_k would be irrelevant for agent i) and $\mathsf{F}_k \notin T'_i$. We observe that $|S'_k| = |S'_{k+1}| = 1$, because $\mathsf{F}_k \notin T'_i$ and by the parity argument. Since (in the best case for agent i) $F_k = \min_{\ell \in S_k}\{d(i, \ell)\}$ and since $F'_k = \min_{\ell' \in S'_k}\{d(i, \ell')\}$ (as S'_k is a singleton) by Lemma 4.3 strategyproofness is preserved.

We note we can regard $(E, M) \to (M, E)$ and $(M, E) \to (E, M)$ as the same case, since both cases have a transition $E \to M$ for one facility and a transition $M \to E$ for the other one. To fix ideas, let us assume facility F_k makes transition $E \to M$ and facility F_{k+1} makes transition $M \to E$. We are going to prove that $S_k \neq S'_k$ and $S_{k+1} \neq S'_{k+1}$. We note that $|S_k| = 2$ and $|S'_{k+1}| = 2$. Let us suppose for the sake of contradiction that $S_k = S'_k$. We reach a contradiction since $|S'_k| = |S'_{k+1}| = 2$ can yield either a (M, M) solution (if $|S'_k \cap S'_{k+1}| \geq 2$) or a (E, E) solution (if $|S'_k \cap S'_{k+1}| \leq 2$). Let us suppose now that $S_{k+1} = S'_{k+1}$. As before, we reach a contradiction since $|S_k| = |S_{k+1}| = 2$ can yield either a (M, M) solution or a (E, E) solution. Furthermore, it can be easily checked that $|S'_k| = 1$ and $|S_{k+1}| = 1$. We can now analyse each transition singularly. Let us focus on transition $E \to M$. We can restrict ourselves to the case when $\mathsf{F}_k \in T_i$ (i.e., otherwise the location of facility F_k does not affect the cost of agent i) and $\mathsf{F}_k \notin T'_i$. Since (in the best case for agent i) $F_k = \mathrm{argmin}_{\ell \in S_k}\{d(i, \ell)\}$ and $F'_k = \mathrm{argmin}_{\ell' \in S'_k}\{d(i, \ell')\}$ (as S'_k is a singleton), and strategyproofness is guaranteed by Lemma 4.3. Let us consider transition $M \to E$. Once again, we can restrict to the case when $\mathsf{F}_{k+1} \in T_i$ and $\mathsf{F}_{k+1} \notin T'_i$. Since $|S_{k+1}| = 1$ and F_{k+1} is an extreme solution, by Lemma 4.5 strategyproofness is preserved.

We now analyze the special case when $S_k = S_{k+1} = \{s\}$ and s is either the first or the last node of the graph. We next prove that this can only happen when there is only one agent s located either at the first or the last node of the graph and who is bidding for both facilities. Let us consider the case when s is the first node of the graph, i.e. $s = 1$, the case when $s = |V|$ being symmetric. It is easy to see that $S_k = S_{k+1} = \{s\}$ only if there is no other agent to the right of s bidding for either F_k or F_{k+1}, as this would imply the existence of at least one agent to the left of s making the same bid, contradicting the assumption that s is the first node of the graph. Furthermore, we notice that if we assume that no agent is misreporting her bid when this configuration occurs, than the only agent located at $s = 1$ has no incentive to lie, since $F_0 = s$, $F_1 = s + 1$ and $cost_i(\mathcal{F}) = 1$ and s cannot increase her utility function any further. We will hence analize the case when case when $\mathcal{T}' = \{T_s = \{\mathsf{F}_0, \mathsf{F}_1\}\}$ and there is an agent $i \neq s$ misreporting her bid.

For any $i > s$ we need to consider $T_i = \{\mathsf{F}_0\}$, $T_i = \{\mathsf{F}_1\}$ and $T_i = \{\mathsf{F}_0, \mathsf{F}_1\}$. If $T_i = \{\mathsf{F}_0\}$, then if $i = 2$ F_0 is assigned as an extreme solution, i.e. $F_0 = 1$, and $cost_i(\mathcal{F}) = cost_i(\mathcal{F}')$, whereas if $i > 2$ then F_0 is assigned as a mean set solution, i.e. the expected location of F_0 is $\text{avg}(S_0)$, and $cost_i(\mathcal{F}) = d(i, \text{avg}(S_0)) \leq d(i, F'_0 = 1) = cost_i(\mathcal{F}')$. If $T_i = \{F_1\}$, then if $i = 2$ the allocation returned is $\mathcal{F} = (F_0 = 1, F_1 = 2)$ and the output does not change, whereas if $i > 2$ then F_1 is allocated in expectation on $\text{avg}(S_1)$ and $cost_i(\mathcal{F}) = d(i, \text{avg}(S_1)) \leq d(i, F'_1 = 2) = cost_i(\mathcal{F}')$. Finally, if $T_i = \{\mathsf{F}_0, \mathsf{F}_1\}$ then both facilities are assigned as mean set solutions and the expected locations of F_0 and F_1 are at $\text{avg}(S_0) = \text{avg}(S_1)$. It is easy to check that $cost_i(\mathcal{F}) = d(i, \text{avg}(S_0)) + d(i, \text{avg}(S_1)) \leq d(i, F'_0 = 1) + d(i, F'_1 = 2) = cost_i(\mathcal{F}')$. □

5 Maximum cost Objective Function

In this section we analyze the heterogeneous 2-facility location problem under the maximum cost objective function. Section 5.1 is devoted to study deterministic mechanisms, whereas in Section 5.2 we explore randomized mechanisms.

5.1 Deterministic Mechanisms for the Maximum Cost Objective Function

In this section we analyze deterministic mechanisms for the maximum cost heterogeneous facility location problem. We start by presenting a negative result stating the impossibility of approximating the optimal allocation within 3/2 of the optimal value while maintaining strategyproofness.

Theorem 5.1. *There exists no α-approximate deterministic SP algorithm for the facility location problem with $\alpha < 3/2$.*

Proof. Let us first consider the two instances depicted in Figure 10(a) and Figure 10(b). The agents in the instance of Figure 10(a) have declarations $T_1 = \{\mathsf{F}_1, \mathsf{F}_2\}$, $T_2 = \{\mathsf{F}_1\}$ and $T_3 = \{\mathsf{F}_1\}$, whereas the agents in Figure 10(b) have declarations $T_1 = \{\mathsf{F}_1\}$, $T_2 = \{\mathsf{F}_1\}$ and $T_3 = \{\mathsf{F}_1, \mathsf{F}_2\}$. It is easy to check that the optimal allocation for the instance of Figure 10(a) is $\mathcal{F}_1 = (F_1 = 2, F_2 = 1)$, whereas the optimal allocation for the instance of Figure 10(b) is $\mathcal{F}_2 = (F_1 = 2, F_2 = 3)$, having both cost 1. We note that, in both cases, any second-best solution has cost 2, so any $3/2$-approximate algorithm would return the optimal solution for these instances. Let us now consider the case when agent 3 in instance 10(a) lies declaring $T'_3 = \{\mathsf{F}_1, \mathsf{F}_2\}$ (see Figure 10(c)). In this case we have two optimal solutions: $\mathcal{F}'_1 = (F_1 = 3, F_2 = 1)$ and $\mathcal{F}'_2 = (F_1 = 1, F_2 = 3)$, with cost 2. We note that we obtain the same instance (and hence the same optimal solutions) if we consider the case when agent 1 in instance 10(b) lies declaring $T'_1 = \{\mathsf{F}_1, \mathsf{F}_2\}$ instead of her true type. We now note that neither $\mathcal{F}'_1$ nor $\mathcal{F}'_2$ are *SP*. In fact, if $\mathcal{F}'_1$ is returned, we can then regard the instance of Figure 10(c) as resulting from the instance of Figure 10(a) when agent 3 lies, in which case agent 3 would gain by lying, as $cost_3(\mathcal{F}_1) = 1 > 0 = cost_3(\mathcal{F}'_1)$. On the other hand, if $\mathcal{F}'_2$ is returned, we can then regard the instance

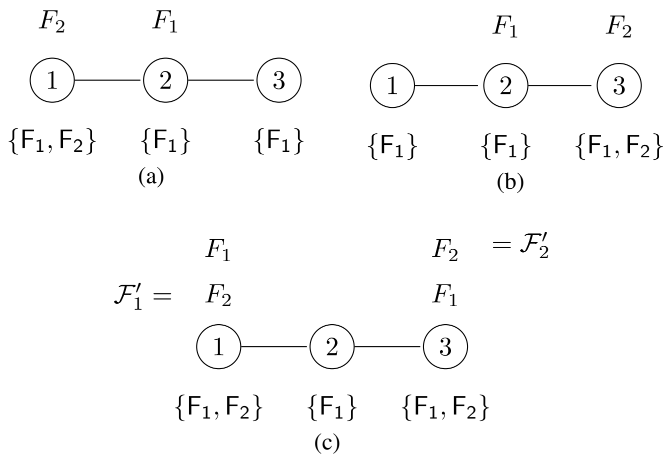

Figure 10: Instances used to prove the lower bound of $3/2$

of Figure 10(c) as resulting from the instance of Figure 10(b) when agent 1 lies, in which case agent 1 would gain by lying, as $cost_1(\mathcal{F}_2) = 1 > 0 = cost_1(\mathcal{F}'_2)$. It is clear from the above argument that an *SP* solution for the instance of Figure 10(c) locates facility F_1 at node 2, hence the only *SP* solutions are $(F_1 = 2, F_2 = 1)$ and $(F_1 = 2, F_2 = 3)$. Since the cost of these solutions is 3, the claim is proven. □

Since we have proved a $3/2$ approximation lower bound for deterministic SP algorithms, the following corollary easily follows.

Corollary 5.1. *There is no optimal deterministic SP algorithm for the facility location problem.*

We now show that algorithm TWOEXTREMES, presented in Section 4.1 is tightly 3-approximate for the maximum cost objective function. We note that the strategyproofness of algorithm TWOEXTREMES follows from Theorem 4.2, since truthfulness is independent of the objective function but is rather dependent on the agents' cost model.

Theorem 5.2. *The* TWOEXTREMES *algorithm is* 3*-approximate.*

Proof. Let $\mathcal{F}^*$ denote the optimal allocation and $\mathcal{F}$ denote the allocation returned by the TWOEXTREMES algorithm. Let us consider an agent i such that $i \in \operatorname{argmax}_{i \in N} cost_i(\mathcal{F})$ and let us denote $EXT = cost_i(\mathcal{F})$. It is easy to check that the following holds:

$$cost_i(\mathcal{F}^*) = EXT - \Delta F \leq OPT \tag{4}$$

where $\Delta F = \sum_{j \in T_i} \Delta F_j$ and $\Delta F_j = d(i, F_j) - d(i, F_j^*)$. Intuitively, i can be regarded as the *worse off* agent under the TWOEXTREMES, and (4) formalizes the fact that the optimal allocation locates the facilities closer to i with respect to TWOEXTREMES in order to lower the cost of agent i and, as a consequence, to lower the value of the objective function. On the other hand, because the position of the facilities changes, there are some agents that are better off under the TWOEXTREMES allocation and that are made worse off by the optimal allocation. For instance, unless the TWOEXTREMES allocation is optimal, the agents that reside at the extremes of the graph are strictly better off under the TWOEXTREMES allocation (since they have a facility located at their node) than they are under the optimal allocation. To formalize this intuition, let S denote the set $\{\min N_1, \max N_2\}$ if $T_i = \{\mathsf{F}_1, \mathsf{F}_2\}$, $\{\min N_1\}$ if $T_i = \{\mathsf{F}_1\}$ and $\{\max N_2\}$ if $T_i = \{\mathsf{F}_2\}$. Then there exists $x \in S$ and a facility $k \in T_i \cap T_x$ such that $d(x, F_k) \leq d(x, F_k^*)$. Intuitively, agent s is the agent made worse off by the optimal allocation in order to make agent i better off and lower the overall cost of the allocation. Furthermore,

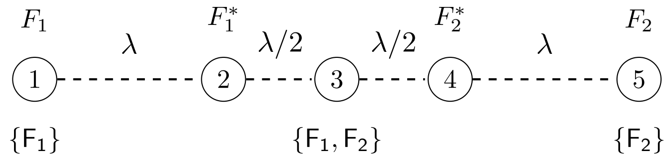

Figure 11: Instance showing that the bound of Theorem 5.2 is tight.

$d(x, F_k^*) - d(x, F_k) \geq d(i, F_k) - d(i, F_k^*)$, since x is at least as much better of in the optimal allocation as i is worse off, according to facility $\mathsf{F_k}$. It is not hard to check that the following holds for $x \in S$:

$$\begin{aligned} OPT \geq \quad & cost_x(\mathcal{F}^*) \geq d(x, F_k^*) \\ \geq \quad & d(x, F_k^*) - d(x, F_k) \\ \geq \quad & d(i, F_k) - d(i, F_k^*) \geq \quad \Delta F_k. \end{aligned} \tag{5}$$

Two cases can occur: (*i*) $|T_i| = 1$ and (*ii*) $|T_i| = 2$. If case (*i*) occurs, we notice that $\Delta F = \Delta F_k$, $k \in T_i$, and from (4) and (5) we derive $EXT \leq 2 \cdot OPT$. If case (*ii*) occurs, we notice that applying (5) with k and $k+1$, we obtain $2 \cdot OPT \geq \Delta F_k + \Delta F_{k+1}$, and, finally, from (4), $EXT \leq 3 \cdot OPT$. □

Theorem 5.3. *The upper bound of Theorem 5.2 is tight.*

Proof. We are now going to prove that the bound is tight. Let us consider the 3-agent family of instances depicted in Figure 11, where between nodes 1 and 2 and between nodes 4 and 5 there are λ "empty" nodes whereas between nodes 2 and 3 and between nodes 3 and 4 there are $\frac{\lambda}{2}$ "empty" nodes. It is easy to check that the optimal allocation in this case is $\mathcal{F}^* = (F_1^* = 2, F_2^* = 4)$ and the optimal cost is $mc(\mathcal{F}^*) = \lambda$. Furthermore, it is easy to check that the cost of the allocation $\mathcal{F} = (F_1 = 1, F_2 = 5)$ computed by algorithm TwoExtremes is $mc(\mathcal{F}) = cost_3(\mathcal{F}) = 3\lambda$, which is indeed 3-approximate. □

5.2 Randomized Mechanisms for the Min-Max Objective Function

In this section we shift our focus to randomized algorithms. Similarly to the case of deterministic mechanisms, our results are twofold: (*i*) we prove a negative result regarding the impossibility of obtaining arbitrarily good approximations by means of *SP* algorithms and (*ii*) we present a $3/2$-approximate randomized *SP* algorithm.

We begin by proving the impossibility of approximating the optimum within a factor of $4/3$ while preserving strategyproofness.

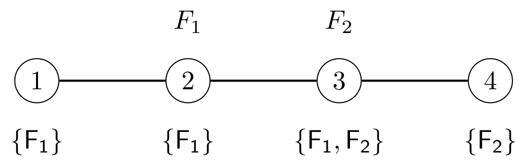

Figure 12: Truthful instance used to prove the bound of Theorem 5.4

Theorem 5.4. *There is no SP randomized α-approximate algorithm for the maximum cost objective function with $\alpha < 4/3$.*

Proof. Let us consider the instance depicted in Figure 12. It is easy to check that the (unique) optimal solution $\mathcal{F}^* = (F_1 = 2, F_2 = 3)$ has cost $mc(\mathcal{F}^*) = 1$, whereas any suboptimal allocation has cost at least 2. Let us consider a generic randomized algorithm A that returns the optimal allocation with probability ρ and some suboptimal allocations with probability $1 - \rho$. If A is $4/3$-approximate, then $4/3 \geq mc(A(\mathcal{T})) \geq 1 \cdot \rho + 2(1 - \rho)$ which implies that A must return the exact solution with probability $\rho \geq 2/3$. In particular, this means that for agent 4 the allocation computed by A has cost $cost_4(A(\mathcal{T})) \geq 2/3$. If agent 4 lies declaring $T_4' = \{\mathsf{F}_1, \mathsf{F}_2\}$ (see Figure 13), then the optimal allocations are $\mathcal{F}_1' = (F_1 = 2, F_2 = 4)$ and $\mathcal{F}_2' = (F_1 = 3, F_2 = 4)$, having cost 2, whereas any suboptimal solution has cost at least 3. Once again, A would return an optimal solution with a certain probability π and a suboptimal solution with probability $1 - \pi$. In particular, to preserve strategyproofness $cost_4(A(\mathcal{T}')) \geq 2/3$ must hold, which can only happen if a suboptimal solution is returned with probability greater than $2/3$, as both optimal solutions $\mathcal{F}_1'$ and $\mathcal{F}_2'$ locate F_2 and $cost_4(\mathcal{F}_1') = cost_4(\mathcal{F}_2') = 0$. This implies that $1 - \pi \geq 2/3$ and $\pi \leq 1/3$. Hence, $mc(A(\mathcal{T}')) \geq 2 \cdot 1/3 + 3 \cdot 2/3 = 8/3$, which yields an approximation ratio of $\alpha \geq 4/3$. □

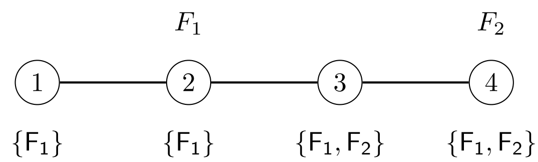

Figure 13: Instance used in the proof of Theorem 5.4 wherein agent 4 lies

Since we have proved a $4/3$ approximation lower bound for randomized SP algorithms, the following corollary easily follows.

Corollary 5.2. *There is no optimal randomized SP algorithm for the facility location problem with respect to the maximum cost objective function.*

We now present a randomized *SP* algorithm, which returns $3/2$-approximate solutions. The main idea behind the algorithm, called RANDAVG and presented in Algorithm 5, is to locate in expectation facility F_k on the mean location of N_k, thus guaranteeing that hiding $\mathsf{F_k}$ from one's own type is not profitable (i.e., N_k can only shrink away from the lying agent). Much like algorithm RANDOPT, algorithm RANDAVG uses mean set solutions, although we do not require mean set solutions to be optimal any more. Mean set solutions are computed by means of COMPUTEMEANSET(x, y) that returns a mean set (see Definition 3.1) centered around two points x and y on the graph.[7] There are, however, certain extreme situations in which the existence of mean sets is not guaranteed (cf. Lemma 5.1). RANDAVG needs to consider these cases separately (cf. lines 3 and 6 of the algorithm) and return deterministic solutions instead.

In this section, we let μ_k denote the average location of N_k, i.e., $\mu_k = \text{avg}(N_k) = \frac{\min N_k + \max N_k}{2}$, $k = 0, 1$. Depending on the parity of $|N_k|$, μ_k might either lie on a vertex of G (if $|N_k|$ is odd) or in between two vertices (if $|N_k|$ is even); we denote the former case as $\mu_k \in V$ and the latter as $\mu_k \in E$ (meaning that μ_k lies on an edge of G, formally: $\exists (u, v) \in E$ such that $\mu_k = (u + v)/2$). We let RIGHT$(\mu_k) = \lceil \mu_k \rceil$ if $\mu_k \in E$ and $\mu_k + 1$ otherwise. Similarly, LEFT$(\mu_k) = \lfloor \mu_k \rfloor$ if $\mu_k \in E$ and $\mu_k - 1$ otherwise.

Algorithm 4 reports the pseudocode for procedure COMPUTEMEANSET, which is used by algorithm RANDAVG to compute mean set solutions. It takes as input two points μ_1 and μ_2 on the graph and returns a set of allocations $\mathcal{M}$ such that when a solution is extracted uniformly at random from $\mathcal{M}$, the expected value of F_0 and F_1 are, respectively, μ_0 and μ_1. Lemma 5.1 gives sufficient conditions for

[7]To ease the notation, in this section we use binary indexes for the facilities.

Algorithm 4: COMPUTEMEANSET

Require: Two points μ_0 and μ_1 on G.
Ensure: a set of allocations $\mathcal{M}$ such when an allocation (F_0, F_1) is extracted uniformly at random from $\mathcal{M}$ the expected value of F_0 and F_1 is, respectively, μ_0 and μ_1.
if $|\mu_0 - \mu_1| \geq 1$ **then**
  if $\mu_0 \in V$ **and** $\mu_1 \in V$ **then**
    return $\mathcal{M} = \{(\mu_0, \mu_1)\}$
    if $\mu_k \in V$ **and** $\mu_{k+1} \in E$ **then**
      return $\mathcal{M} = \{(\mu_k, \text{LEFT}(\mu_{k+1})), (\mu_k, \text{RIGHT}(\mu_{k+1}))\}$
    else
      return $\mathcal{M} = \{(\text{LEFT}(\mu_0), \text{LEFT}(\mu_1)), (\text{RIGHT}(\mu_0), \text{RIGHT}(\mu_1))\}$
    end if
  end if
else
  if $\mu_0 \in V$ **and** $\mu_1 \in V$ **then**
    return $\{(\mu_0, \text{LEFT}(\mu_1)), (\mu_0, \text{RIGHT}(\mu_1)), (\text{LEFT}(\mu_0), \mu_1), (\text{RIGHT}(\mu_0), \mu_1)\}$
  end if
  if ($\mu_k \in V$ **and** $\mu_{k+1} \in E$**) or (**$\mu_k \in E$ **and** $\mu_{k+1} \in E$**) then**
    return $\mathcal{M} = \{(\text{RIGHT}(\mu_k), \text{LEFT}(\mu_{k+1})), (\text{LEFT}(\mu_k), \text{RIGHT}(\mu_{k+1}))\}$
  end if
end if

a mean set solution to be feasible. Algorithm 4 is an algorithmic polynomial-time transposition of the constructive proof of Lemma 5.1.

Lemma 5.1. *There always exists a feasible mean set for graph $G = (V, E)$ if either of the following holds:* (i) $|\mu_k - \mu_{k+1}| \geq 1$, (ii) $\forall k \in \{0, 1\}$, $\text{RIGHT}(\mu_k) \neq \text{NIL}$ *and* $\text{LEFT}(\mu_k) \neq \text{NIL}$.

Proof. Let us focus on case (*i*) initially. We distinguish the cases in which μ_k is in V from those in which it is in E. If $\mu_k \in V$ and $\mu_{k+1} \in V$ then $\mathcal{M} = \{(\mu_k, \mu_{k+1})\}$ is a mean set for G (note that this solution is feasible as $|\mu_k - \mu_{k+1}| \geq 1$ by hypothesis). If $\mu_k \in E$ and $\mu_{k+1} \in V$ then: $\text{LEFT}(\mu_k) \neq \text{NIL}$, $\text{RIGHT}(\mu_k) \neq \text{NIL}$ and both $\text{LEFT}(\mu_k) \neq \mu_{k+1}$ and $\text{RIGHT}(\mu_k) \neq \mu_{k+1}$ (as, by hypothesis, $|\mu_k - \mu_{k+1}| \geq 1$). Hence, $\mathcal{M} = \{(\text{LEFT}(\mu_k), \mu_{k+1}), (\text{RIGHT}(\mu_k), \mu_{k+1})\}$ is a feasible mean set for G. If both $\mu_k \in E$ and $\mu_{k+1} \in E$, then since $|\mu_k - \mu_{k+1}| \geq 1$, we have that $\mathcal{M} = \{(\text{LEFT}(\mu_k), \text{LEFT}(\mu_{k+1})), (\text{RIGHT}(\mu_k), \text{RIGHT}(\mu_{k+1}))\}$ is a mean set for G.

Let us now focus on case (*ii*). We assume that $|\mu_k - \mu_{k+1}| < 1$ (for otherwise the arguments above apply). It is easy to check that in this case we have either: (*i*) $\mu_k \in V$ and $\mu_{k+1} \in E$ for some $k \in \{0, 1\}$ or (*ii*) $\mu_k = \mu_{k+1}$ and $\mu_k, \mu_{k+1} \in V$ or (*iii*) $\mu_k = \mu_{k+1}$ and $\mu_k, \mu_{k+1} \in E$. Then $\mathcal{M} = \{(\text{LEFT}(\mu_k), \text{RIGHT}(\mu_{k+1})), (\text{RIGHT}(\mu_k), \text{LEFT}(\mu_{k+1}))\}$ is a feasible mean set for G. □

Theorem 5.5. *Algorithm* RANDAVG *is SP in expectation.*

Proof. It is easy to check that algorithm RANDAVG returns either a feasible mean set solution (i.e., line 10) or a feasible deterministic solution (lines 3 and 6) when no mean set solutions exist. For the sake of notation, in the remainder we will denote a mean set solution as $\mathcal{M}$ and a deterministic solution as $\mathcal{D}$. Let us denote by i the lying agent; we shall prove that $cost_i(\mathcal{F}') \geq cost_i(\mathcal{F})$, where $\mathcal{F}$ and $\mathcal{F}'$ denote the outcomes of RANDAVG on input the true type of i and a misreport, respectively. (The value of $cost_i(\cdot)$ must be intended here with respect to the expected locations of the facilities in T_i.) The analysis distinguishes what type of allocation (i.e., $\mathcal{M}$ or $\mathcal{D}$) $\mathcal{F}$ and $\mathcal{F}'$ are. By letting $\mathcal{X} \to \mathcal{Y}$ symbolize that $\mathcal{F}$

Algorithm 5: RANDAVG

Require: Line G, facilities $\mathfrak{F} = \{\mathsf{F}_0, \mathsf{F}_1\}$, declarations $\mathcal{T} = \{T_1, \dots, T_n\}$
Ensure: $F_{AVG}(\mathcal{T})$, a $3/2$-approximate allocation for 2-facility location on G
1: $\mu_k := \text{avg}(N_k), \ \ \forall k \in \{0, 1\}$
2: **if** $\exists k \in \{0, 1\}$ s.t. $\textsc{Right}(\mu_k) = \textsc{Nil}$ AND $|\mu_0 - \mu_1| < 1$ **then**
3: **return** $\left(\bar{F}_k := \mu_k, \bar{F}_{k+1} := \textsc{Left}(\mu_{k+1})\right)$
4: **end if**
5: **if** $\exists k \in \{0, 1\}$ s.t. $\textsc{Left}(\mu_k) = \textsc{Nil}$ AND $|\mu_0 - \mu_1| < 1$ **then**
6: **return** $\left(\bar{F}_k := \mu_k, \bar{F}_{k+1} := \textsc{Right}(\mu_{k+1})\right)$
7: **end if**
8: $S := \{(u, v) : u \in V, v \in V, u \neq v\}$
9: $\mathcal{M} := \textsc{ComputeMeanSet}(\mu_k, \mu_{k+1}, S)$
10: **return** $(F_k, F_{k+1}) \in \mathcal{M}$ with probability $1/|\mathcal{M}|$

is of type $\mathcal{X}$ and $\mathcal{F}'$ is of type $\mathcal{Y}$, we consider three cases: (*a*) $\mathcal{M} \to \mathcal{M}$; (*b*) $\mathcal{M} \to \mathcal{D}$; (*c*) $\mathcal{F}$ is of type $\mathcal{D}$.

Case (*a*). Let $i \in \{\min N_0, \max N_0, \min N_1, \max N_1\}$. In this case each facility is located independently (in expectation) and truthfulness follows from the simple observation that $d(i, \text{avg}(N_k)) \leq d(i, \text{avg}(N'_k))$, for $k \in T_i$. If $i \notin \{\min N_0, \max N_0, \min N_1, \max N_1\}$, it is easy to check that i cannot alter the outcome of RANDAVG, as avg is computed based on the extreme elements of N_0 and N_1.

Case (*b*). Let us consider the case $\mathcal{M} \to \mathcal{D}$. Since $\mathcal{F}'$ is of type $\mathcal{D}$, $|\mu'_0 - \mu'_1| < 1$ and there exists $k \in \{0, 1\}$ such that either $\textsc{Left}(\mu'_k) = \textsc{Nil}$ or $\textsc{Right}(\mu'_k) = \textsc{Nil}$. We focus on the former case; the other case follows by symmetry. Since $\textsc{Left}(\mu'_k) = \textsc{Nil}$ then, by definition of LEFT, $\mu'_k \in V$ and, in turns, by definition of μ'_k, $N'_k = \{\ell\}$ and $\mu'_k = \ell$. Note also that since $|\mu'_k - \mu'_{k+1}| < 1$ then $\mu'_{k+1} \in E$, with $\textsc{Left}(\mu'_{k+1}) = \ell$. In this case we then have $T'_\ell = \{\mathsf{F_k}, \mathsf{F_{k+1}}\}$, $T'_{\ell+1} = \{\mathsf{F_{k+1}}\}$ and $T'_r = \emptyset$ for all $r \geq \ell + 2$, but $T_l = T'_l$ for all $l \neq i$ (i.e. we assume that only agent i is lying). On this instance RandAvg returns $\mathcal{F}' = (F'_k = \ell, F'_{k+1} = \ell + 1)$ (see Line 6 of algorithm RANDAVG). Let us assume that $i = \ell$. In this case, if $T_i = \{\mathsf{F_k}\}$, then $\mathcal{F} = (F_k = \ell, F_{k+1} = \ell + 1)$ (line 3 of RANDAVG) and the cost of agent i is unchanged, whereas if $T_i = \{\mathsf{F_{k+1}}\}$ then $N_{k+1} = \{\ell, \ell + 1\}$, $F_{k+1} = \ell + 0.5$ and $cost_i(\mathcal{F}') = 1 > 0.5 = cost_i(\mathcal{F})$. Let us assume that $i = \ell + 1$. We need to consider two cases: $T_i = \{\mathsf{F_k}\}$ and $T_i = \{\mathsf{F_k}, \mathsf{F_{k+1}}\}$. If $T_i = \{\mathsf{F_k}\}$, then $N_k = \{1, 2\}$, $N_{k+1} = \{1\}$ and RANDAVG outputs $\mathcal{F} = (F_k = \ell + 1, F_{k+1} = \ell)$ (line 6 or RANDAVG) and $cost_i(\mathcal{F}) = 0 < 1 = cost_i(\mathcal{F}')$. If $T_i = \{\mathsf{F_k}, \mathsf{F_{k+1}}\}$, then $N_k = N_{k+1} = \{1, 2\}$, the expected position of the solution returned by RANDAVG is $\ell{+}0.5$ for both F_k and F_{k+1}, and $cost_i(\mathcal{F}) = 1 = cost_i(\mathcal{F}')$. Let us assume that $i \geq \ell{+}2$. In this case it can be $T_i = \{\mathsf{F_k}\}$, $T_i = \{\mathsf{F_{k+1}}\}$ or $T_i = \{\mathsf{F_k}, \mathsf{F_{k+1}}\}$. If $T_i = \{\mathsf{F_k}\}$, then $N_k = \{\ell, i\}$ and $N_{k+1} = \{\ell, \ell + 1\}$. In this case the expected position of facility F_k is $\text{avg}(N_k) = \frac{\ell + i}{2} > \ell$ and $cost_i(\mathcal{F}) = \frac{i - \ell}{2} < i - \ell = cost_i(\mathcal{F}')$. If $T_i = \{\mathsf{F_{k+1}}\}$, then $N_k = \{\ell\}$ and $N_k + 1 = \{\ell, \ell + 1, i\}$. In this case the expected position of facility F_{k+1} is $\frac{\ell + i}{2}$ and $cost_i(\mathcal{F}) = \frac{i - \ell}{2} < \ell + 1 - 1 = cost_i(\mathcal{F}')$. If $T_i = \{\mathsf{F_k}, \mathsf{F_{k+1}}\}$, then $N_k = \{\ell, i\}$ and $N_{k+1} = \{\ell, \ell + 1, i\}$. In this case the expected position for both facilities F_k and F_{k+1} is $\frac{\ell + 1}{2}$ and $cost_i(\mathcal{F}) = i - \ell < 2(i - \ell) - 1 = cost_i(\mathcal{F}')$.

Case (*c*). Since $\mathcal{F}$ is of type $\mathcal{D}$, by the same reasoning of case (*b*) we conclude that there exists $k \in \{0, 1\}$ such that $N_k = \{\ell\}$, $\mu_k = \ell$ and either $\textsc{Left}(\mu_k) = \textsc{Nil}$ or $\textsc{Right}(\mu_k) = \textsc{Nil}$. We focus on the former case, the other being symmetric. We note that in this case we have a 2-agents instance such that $T_\ell = \{\mathsf{F_k}, \mathsf{F_{k+1}}\}$, and $T_{\ell+1} = \{\mathsf{F_{k+1}}\}$. Since the allocation outputted by RANDAVG is $(F_k = \ell, F_{k+1} = \ell + 1)$ it is easy to check that no agent can lower her cost any further. □

Theorem 5.6. *Algorithm* RANDAVG *is* $3/2$*-approximate for the maximum cost objective function.*

Proof. We note that whenever algorithm RANDAVG returns a deterministic solution, then it returns an optimal allocation. Hence, in the remainder we restrict ourselves to considering only the case when RANDAVG returns a mean set solution.

Let us denote a *bottleneck agent* (i.e., an agent incurring the maximum cost) of RANDAVG by i, namely: $cost_i(\mathcal{F}) = mc(\mathcal{F})$, where $\mathcal{F} = (F_0, F_1)$ denotes the output of RANDAVG. (Again, $cost_i(\cdot)$ must be considered w.r.t. the expected locations of the facilities in T_i.) Hereinafter, $\mathcal{F}^* = (F_0^*, F_1^*)$ will denote the optimal solution. We can assume that $T_i = \{\mathsf{F}_0, \mathsf{F}_1\}$, as otherwise RANDAVG would return an optimal allocation. Indeed, if we let $T_i = \{k\}$, we have $AVG = mc(\mathcal{F}) = \frac{R_k - L_k}{2} \leq mc(\mathcal{F}^*) = OPT$, where $L_k = \min N_k$ and $R_k = \max N_k$, hence $\mathcal{F}$ must be optimal.

Let us denote $\Delta F_j = d(i, F_j) - d(i, F_j^*)$, for $j \in \{0, 1\}$. It is easy to check that:

$$cost_i(\mathcal{F}^*) = AVG - \Delta F \leq OPT, \tag{6}$$

where $\Delta F = \Delta F_0 + \Delta F_1$. Intuitively, the optimal allocation locates the facilities closer to i with respect to RANDAVG in order to lower the cost of agent i. Because of this, there is an agent $x \in \{L_0, R_0\}$ such that $d(x, F_0) \leq d(x, F_0^*)$. For instance, any agent at or near $\text{avg}(N_0)$ is made worse off if the F_0 is moved away from $\text{avg}(N_0)$. It is not too hard to check that the following holds:

$$OPT \geq cost_x(\mathcal{F}^*) \geq \frac{R_0 - L_0}{2} + \Delta F_0. \tag{7}$$

Likewise, there is an agent $y \in \{L_1, R_1\}$ such that $d(y, F_1) \leq d(y, F_1^*)$ and we have:

$$OPT \geq cost_y(\mathcal{F}^*) \geq \frac{R_1 - L_1}{2} + \Delta F_1. \tag{8}$$

We now need to consider two cases: $\Delta F \leq \frac{OPT}{2}$ and $\Delta F > \frac{OPT}{2}$. If $\Delta F \leq \frac{OPT}{2}$, the claim follows immediately from (6), as $AVG - \frac{OPT}{2} \leq AVG - \Delta F$ holds. If $\Delta F > \frac{OPT}{2}$, then the following holds:

$$\begin{aligned} \frac{OPT}{2} &< \Delta F_0 + \Delta F_1 \\ &\leq 2 \cdot OPT - \frac{R_0 - L_0}{2} - \frac{R_1 - L_1}{2} \end{aligned} \tag{9}$$

where the second inequality follows from (7) and (8). From (9), we obtain

$$\frac{R_0 - L_0}{2} + \frac{R_1 - L_1}{2} < \frac{3}{2} \cdot OPT.$$

By observing that

$$AVG \leq \frac{R_0 - L_0}{2} + \frac{R_1 - L_1}{2},$$

the claim follows. □

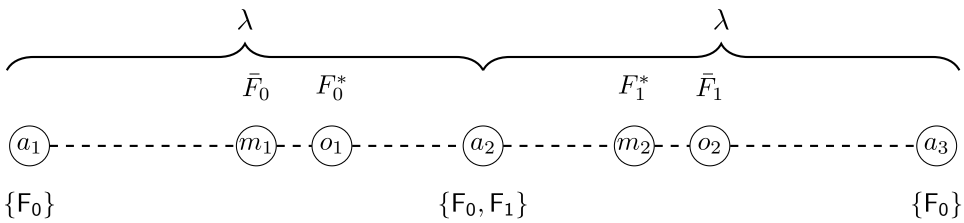

Figure 14: Tight instance for RANDAVG

Theorem 5.7. *The upper bound of Theorem 5.6 is tight.*

Proof. Figure 14 depicts a family of instances for which the RANDAVG algorithm always returns a $3/2$-approximate solution, thus showing that the analysis above is tight. This family of instances consists of (at least) 3 agents a_1, a_2 and a_3 such that: (*i*) $a_1 < a_2 < a_3$; (*ii*) $d(a_1, a_2) = d(a_2, a_3) = \lambda$, where $\lambda \in \mathbb{Z}$; (*iii*) $a_1 = \min N_0$, $a_2 = \max N_0 = \min N_1$ and $a_3 = \max N_1$; (*iv*) $T_a = \{\mathsf{F}_0\}$, $T_b = \{\mathsf{F}_0, \mathsf{F}_1\}$ and $T_c = \{\mathsf{F}_1\}$. It is easy to check that, for this family of instances, the optimal allocation is $(F_0^* = o_1, F_1^* = o_2)$, such that $o_1 = \frac{2}{3}\lambda$ and $o_2 = \frac{4}{3}\lambda$, and $mc((F_0^*, F_1^*)) = \frac{2}{3}\lambda$. Algorithm RANDAVG returns allocation $(F_0 = m_1, F_1 = m_2)$ such that: $m_1 = \frac{a_1+a_2}{2}$, $m_2 = \frac{a_2+a_3}{2}$ and $mc((F_0, F_1)) = \lambda$. Hence, the approximation ratio of algorithm RANDAVG on this family of instances is $3/2$. □

6 Conclusions

In this paper, we have introduced and analyzed a multi-dimensional variant of the facility location problem, arguably the paradigmatic case study in the literature on approximate mechanism design without money. Moreover, works falling in this research agenda often only deal with single-parameter agents (exceptions being the studies on mechanisms without money and verification [9, 4, 5]).

In greater detail, we have studied the heterogeneous facility location problem, which features heterogeneous facilities (i.e., serving different purposes). Our study encompasses both utilitarian and non utilitarian objective functions, namely: (*i*) social cost, which consist of the sum of the agents' individual costs, and (*ii*) maximum cost, which accounts for the highest cost incurred by any of the agents.

In both cases, we have shown that even for very simple agents' domains comprised of only 2 bits (as in the case of heterogeneous 2-facility location), truthfulness might impose a penalty on the quality of the solutions computed by deterministic mechanisms. Indeed, we have proved a $9/8$ approximation lower bound for deterministic mechanisms when the social cost objective is concerned, and a $3/2$ approximation lower bound for maximum cost. We have coupled these negative results with an $(n-1)$-approximate truthful deterministic mechanism for the social cost objective function and a 3-approximate truthful deterministic mechanism for maximum cost.

Randomization provably helps to improve the approximation quality, as it enables us to obtain the optimal allocation via truthful mechanisms with respect to the social cost objective. As regards the maximum cost objective function, we have proved that in order to impose truthfulness, we still have to content ourselves with suboptimal allocations even when resorting to randomized algorithms, although we can provide better approximations.

Naturally, our results leave a gap between upper and lower bounds for both deterministic and randomized truthful mechanisms. To close these gaps, a better understand of truthfulness without money of multi-dimensional agents is needed, which is left as a future direction for further investigation.

References

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

[2] E. Dokow, M. Feldman, R. Meir, and I. Nehama. Mechanism design on discrete lines and cycles. In *ACM EC*, pages 423–440, 2012.

[3] D. Ferraioli, P. Serafino, and C. Ventre. What to verify for optimal truthful mechanisms without money. In *AAMAS*, 2016. In press.

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

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

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

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

[8] 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.

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

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

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

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

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

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

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

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

[17] J. Schummer and R. V. Vohra. Strategy-proof location on a network. *Journal of Economic Theory*, 104:405–428, 2002.

[18] P. Serafino and C. Ventre. Heterogeneous facility location without money on the line. In *ECAI*, pages 807–812, 2014.

[19] P. Serafino and C. Ventre. Truthful mechanisms for the location of different facilities. In *AAMAS*, pages 1613–1614, 2014.

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