

Truthful Facility Assignment with Resource Augmentation: An Exact Analysis of Serial Dictatorship ^{*}

Ioannis Caragiannis¹, Aris Filos-Ratsikas², Søren Kristoffer Stiil Frederiksen³,
Kristoffer Arnsfelt Hansen³, and Zihan Tan⁴

¹ Department of Computer Engineering and Informatics, University of Patras, Greece
`caragian@ceid.upatras.gr`

² Department of Computer Science, University of Oxford, UK
`Aris.Filos-Ratsikas@cs.ox.ac.uk`

³ Department of Computer Science, Aarhus University, Denmark
`sorensf@gmail.com, arnsfelt@cs.au.dk`

⁴ Department of Computer Science, University of Chicago, USA
`zihantan@uchicago.edu`

Abstract. We study the *truthful facility assignment* problem, where a set of agents with private most-preferred points on a metric space are assigned to facilities that lie on the metric space, under capacity constraints on the facilities. The goal is to produce such an assignment that minimizes the social cost, i.e., the total distance between the most-preferred points of the agents and their corresponding facilities in the assignment, under the constraint of truthfulness, which ensures that agents do not misreport their most-preferred points.

We propose a *resource augmentation framework*, where a truthful mechanism is evaluated by its worst-case performance on an instance with enhanced facility capacities against the optimal mechanism on the same instance with the original capacities. We study a well-known mechanism, Serial Dictatorship, and provide an exact analysis of its performance.

Among other results, we prove that Serial Dictatorship has approximation ratio $g/(g-2)$ when the capacities are multiplied by any integer $g \geq 3$. Our results suggest that even a limited augmentation of the resources can have wondrous effects on the performance of the mechanism and in particular, the approximation ratio goes to 1 as the augmentation factor becomes large. We complement our results with bounds on the approximation ratio of Random Serial Dictatorship, the randomized version of Serial Dictatorship, when there is no resource augmentation.

^{*} Ioannis Caragiannis was partially supported by a Caratheodory research grant E.114 from the University of Patras. Aris Filos-Ratsikas was partially supported by the COST Action IC1205 on “Computational Social Choice” and by the ERC Advanced Grant 321171 (ALGAME). Aris Filos-Ratsikas, Søren Kristoffer Stiil Frederiksen, and Kristoffer Arnsfelt Hansen acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61361136003) for the Sino-Danish Center for the Theory of Interactive Computation and from the Center for Research in Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council.

1 Introduction

We study the *facility assignment problem*, in which there is a set of agents and a set of *facilities* with finite capacities; facilities are located on a metric space at points F_i and each agent has a most-preferred point A_i , which is her private information. The goal is to produce an *assignment* of agents to facilities, such that no capacity is exceeded and the sum of distances between agents and their assigned facilities, *the social cost*, is minimized. A *mechanism* is a function that elicits the points A_i from the agents and outputs an assignment. We will be interested in *truthful* mechanisms, i.e., mechanisms that do not incentivize agents to misreport their most-preferred locations and we will be aiming to find mechanisms that achieve a social cost as close as possible to that of the optimal assignment when applied to the true points A_i of the agents. Our setting has various applications such as assigning patients to personal GPs, vehicles to parking spots, children to schools and pretty much any matching environment where there is some notion of distance involved.

Our work falls under the umbrella of *approximate mechanism design without money*, a term coined by Procaccia and Tennenholtz [16] to describe problems where some objective function is optimized under the hard constraints imposed by the requirement of truthfulness. The standard measure of performance for truthful mechanisms is the *approximation ratio*, which for our objective, is the worst-case ratio between the social cost of the truthful mechanism in question over the minimum social cost, calculated over all input instances of the problem.

However, it is arguably unfair to compare the performance of a mechanism that is severely limited by the requirement of truthfulness to that of an omnipotent mechanism that operates under no restrictions and has access to the real inputs of the agents, without giving the truthful mechanism any additional capabilities. This is even more evident in general settings, where strong impossibility results restrict the performance of all truthful mechanisms to be rather poor. The need for a departure from the worst-case approach has been often advocated in the literature, but the suggestions mainly involve some average case analysis or experimental evaluations.

Instead, we will adopt a different approach, that has been made popular in the field of online algorithms and competitive analysis [13, 17]; the approach suggests enhancing the capabilities of the mechanism operating under some very limiting requirement (such as truthfulness or lack of information) before comparing to the optimal solution. Our main conceptual contribution is the adoption of a *resource augmentation approach to approximate mechanism design*. In the resource augmentation framework, we evaluate the performance of a truthful mechanism on an input with additional resources, when compared to the optimal solutions for the set of original resources. For our problem, we consider the social cost achievable by a truthful mechanism on some input with augmented facility capacities against the optimal assignment under the original capacities given as input.

More precisely, let I be an input instance to the facility assignment problem and let I_g be the same instance where each capacity has been multiplied by some

integer constant g , that we call the *augmentation factor*. Then, the *approximation ratio with augmentation g* of a truthful mechanism M is the worst-case ratio of the social cost achievable by M on I_g over the social cost of the optimal assignment on I , over all possible inputs of the problem. The idea is that if the ratio achievable by a mechanism with small augmentation is much better when compared to the standard approximation ratio, it might make sense to invest in additional resources. At the same time, such a result would imply that the set of “bad” instances in the worst-case analysis is rather pathological and not very likely to appear in practice. To the best of our knowledge, this is the first time that such a resource augmentation framework has been explicitly proposed in algorithmic mechanism design.

1.1 Our Results

As our main contribution, we study the well-known truthful mechanisms for assignment problems, Serial Dictatorship (SD) and Random Serial Dictatorship (RSD). For SD, we provide an *exact analysis*, obtaining tight bounds on the approximation ratio of the mechanism for all possible augmentation factors g . Specifically, we prove that when n is the number of agents, while without any augmentation, the approximation ratio of SD is $2^n - 1$, the approximation ratio with augmentation factor $g = 2$ is exactly $\log(n + 1)$ whereas for $g \geq 3$, the approximation ratio is $g/(g - 2)$, i.e., a small constant. In particular, our results imply that as the augmentation factor becomes large, the approximation ratio of SD with augmentation goes to 1 and the convergence is rather fast. Our results for SD improve and extend some results in the field of online algorithms [12].

To prove the approximation ratios for all augmentation factors, we use an interesting technique based on linear programming. Specifically, we first provide a directed graph interpretation of the assignment produced by SD and the optimal assignment, and then prove that the worst-case instances appear on g -trees, i.e., trees where (practically) every vertex has exactly g successors. Then, we formulate the problem of calculating the worst ratio on such trees as a linear program and bound the ratio by obtaining feasible solutions to its dual. Such a solution can be seen as a “path covering” of the assignment graph and we obtain the bounds by constructing appropriate path coverings of low cost.

We also consider randomized mechanisms and the very well-known Random Serial Dictatorship mechanism. We prove that for augmentation factor 1 (i.e., no resource augmentation), the approximation ratio of the mechanism is between $n^{0.26}$ and n ; the result suggests that even a small augmentation ($g = 2$) is a more powerful tool than randomization.

1.2 Related Work

Assignment problems are central in the literature of economics and computer science. The literature on one-sided matchings dates back to the seminal paper by Hylland and Zeckhauser [10] and includes many very influential papers [5, 18] in economics as well as a rich recent literature in computer science [9, 2, 8,

15]. Serial Dictatorships (or their randomized counterparts) have been in the focus of much of this literature, mainly due to their simplicity and the fragile nature of truthfulness, which makes it quite hard to construct more involved truthful mechanisms. In a celebrated result, Svensson [18] characterized a large class of truthful mechanisms by serial dictatorships. Random Serial Dictatorship has also been extensively studied [15, 1] and recently it was proven [8] that is asymptotically the best truthful mechanism for one-sided matchings under the general cardinal preference domain.

The facility assignment problem can be interpreted as a matching problem; somewhat surprisingly, matching problems in metric spaces have only recently been considered in the mechanism design literature. Emek et al. [7] study a setting very closely related to ours, where the goal is to find matchings on metric spaces, but they are interested in how well a mechanism that produces a stable matching can approximate the cost of the optimal matching. In a conceptually similar work, Anshelevich and Shreyas [3] study the performance of *ordinal* matching mechanisms on metric spaces, when the limitation is the lack of information. The fundamental difference between those works and ours is that we consider truthful mechanisms and bound their performance due to the truthfulness requirement; to the best of our knowledge, this is the first time where truthful mechanisms have been considered in a matching setting with metric preferences. Another difference between our work and the aforementioned papers is that they do not consider resource augmentation and only bound the performance of mechanisms on the same set of resources.⁵ However, given the generality of the augmentation framework, the same idea could be applied to their settings. In that sense, our paper proposes a *resource augmentation approach to algorithmic mechanism design* that could be adopted in most resource allocation and assignment settings.

As we mentioned earlier, the idea of resource augmentation was popularized by the field of online algorithms and competitive analysis and is tightly related to the literature on *weak adversaries* where an online competitive algorithm is compared to the adversary that uses a smaller number of resources. The idea for this approach originated in the seminal paper by Sleator and Tarjan [17] and has been adopted by others ever since [14, 19]; the term “resource augmentation” was explicitly introduced by Kalyanasundaram and Pruhs [13].

Most closely related to our problem is the *online transportation problem* [12] (also known as the minimum online metric bipartite matching). In particular, results about the greedy algorithm in the online transportation problem imply bounds for the facility assignment problem. However, contrary to [12], our analysis is *exact*, i.e. our results involve no asymptotics. Furthermore, compared to the related result in [12], we remark that our analysis is substantially different due to the use of linear programming; our primal-dual technique could be applicable for greedy assignment mechanisms on other resource augmentation settings, beyond the problem studied here. For a detailed discussion of the connection between the two settings, the reader might refer to the full version of this paper.

⁵ With the exception of the bi-criteria result in [3].

Finally, there is some resemblance between our problem and the facility location problem [16] that has been studied extensively in the literature of approximate mechanism design, in the sense that in both settings, agents specify their most preferred positions on a metric space. Note that the settings are fundamentally different however, since in the facility location problem, the task is to identify the appropriate point to locate a facility whereas in our setting, facilities are already in place and we are looking for an assignment of agents to them.

2 Preliminaries

In the *facility assignment* problem, there is a set $N = \{1, \dots, n\}$ of agents and a set $M = \{1, \dots, m\}$ of *facilities*, where agents and facilities are located on a metric space, equipped with a distance function d . Each facility has a *capacity* $c_i \in \mathbb{N}_+$, which is the number of agents that the facility can accommodate. We assume that $\sum_{i=1}^m c_i \geq n$, i.e., all agents can be accommodated by some facility. Each agent has a most preferred position A_i on the space and his cost $d_i(j)$ from facility j is the distance $d(A_i, F_j)$ between A_i and the position F_j of the facility. Let $A = (A_1, \dots, A_n)$ be a vector of preferred positions and call it a *location profile*. Let $F = (F_1, \dots, F_m)$ be the corresponding set of points of the facilities. A pair of agents' most preferred points and facility points (A, F) is called an *instance* of the facility assignment problem and is denoted by I .

The locations of the facilities are known but the location profiles are not known; agents are asked to report them to a central planner, who then decides on an *assignment* S , i.e., a pairing of agents and facilities such that no agent is assigned to more than one facility and no facility capacity is exceeded. Let S_i be the restriction of the assignment to the i 'th coordinate, i.e., the facility to which agent i is assigned in S and let \mathcal{S} be the set of all assignments. The *social cost* of an assignment S on input I is the sum of the agents' costs from their facilities assigned by S i.e., $\sum_{i=1}^n d_i(S_i)$. A deterministic mechanism maps instances to assignments whereas a randomized mechanism maps instances to probability distributions over assignments.

A mechanism is *truthful* if no agent has an incentive to misreport his most preferred location. Formally, this is guaranteed when for every location profile A , any report A'_i , and any reports A_{-i} of all agents besides agent i , it holds that $d_i(S_i) \geq d_i(S'_i)$, where $S = M(I)$ and $S' = M(I')$, with $I = (A, F)$ and $I' = ((A'_i, A_{-i}), F)$. For randomized mechanisms, the corresponding notion is *truthfulness-in-expectation*, where an agent can not decrease her expected distance from the assigned facilities by deviating, i.e., it holds that $\mathbb{E}_{S \sim D}[d_i(S_i)] \geq \mathbb{E}_{S \sim D'}[d_i(S_i)]$, where D and D' are the probability distributions output by the mechanism on inputs I and I' respectively. A stronger notion of truthfulness for randomized mechanisms is that of *universal truthfulness*, which guarantees that for every realization of randomness, there will not be any agent with an incentive to deviate. Alternatively, one can view a universally truthful mechanism as a mechanism that runs a deterministic truthful mechanism at random, according to some distribution.

As our main conceptual contribution, we will consider a *resource augmentation* framework where the minimum social cost of any assignment will be compared with the social cost achievable by a mechanism on a location profile with augmented facility capacities. Given an instance I , we will use the term *g -augmented instance* to refer to an instance of the problem where the input is I and the facility of each capacity has been multiplied by g . We will denote that instance by I_g and we will call g the *augmentation factor* of I . For example, when $g = 2$, we will compare the minimum social cost with the social cost of a mechanism on the same inputs but with double capacities.

For the facility assignment problem, the optimal mechanism computes a minimum cost matching (which can be computed using an algorithm for maximum weight bipartite matching) and it can be easily shown that it is not truthful; in order to achieve truthfulness, we have to output suboptimal solutions. As performance measure, we define the *approximation ratio with augmentation* of a mechanism M as

$$ratio_g(M) = \sup_I \frac{SC_M(I_g)}{SC_{OPT}(I)}$$

where $SC_M(I_g) = \sum_{i=1}^n d_i(M(I_g)_i)$ is the social cost of the assignment produced by mechanism M on input instance I with augmentation factor g and $SC_{OPT}(I)$ is the minimum social cost of any assignment on I i.e., $SC_{OPT}(I) = \min_{S \in \mathcal{S}} \sum_{i=1}^n d_i(S_i)$. For randomized mechanisms, the definitions involve the expected social cost and are very similar. Obviously, if we set $g = 1$, we obtain the standard notion of the approximation ratio for truthful mechanisms [16]. For consistency with the literature, we will denote $ratio_1(M)$ by $ratio(M)$.

We will be interested in two natural truthful mechanisms that assign agents to facilities in a greedy nature. A *serial dictatorship* (SD) is a mechanism that first fixes an ordering of the agents and then assigns each agent to his most preferred facility, from the set of facilities with non-zero residual capacities. Its randomized counterpart, *Random Serial Dictatorship* (RSD), is the mechanism that first fixes the ordering of agents uniformly at random and then assigns them to their favorite facilities that still have capacities left. In other words, RSD runs one of the $n!$ possible serial dictatorships uniformly at random and hence it is universally truthful.

3 Approximation Guarantees for Serial Dictatorships

In this section we provide our main results, the upper bounds on the approximation ratio with augmentation of Serial Dictatorship, for all possible augmentation factors. In Section 4, we state the theorem that ensures that the bounds proven here are tight. At the end of the section, we also consider Random Serial Dictatorship, when there is no resource augmentation.

Theorem 1. *The approximation ratio of SD with augmentation factor g in facility assignment instances with n agents is*

1. $\text{ratio}(SD) \leq 2^n - 1$,
2. $\text{ratio}_2(SD) \leq \log(n + 1)$,
3. $\text{ratio}_g(SD) \leq \frac{g}{g-2}$ when $g \geq 3$.

In order to prove the theorem,⁶ we first need to introduce a different interpretation of the assignment produced by SD and the optimal assignment, in terms of a directed graph. We begin with a roadmap of the proof of Theorem 1.

1. We show how to represent an instance of facility assignment together with an optimal solution and a solution computed by the SD mechanism as a directed graph and argue that the instances in which the SD mechanism has the worst approximation ratio are specifically structured as directed trees.
2. We observe that the cost of the SD mechanism in these instances is upper-bounded by the objective value of a maximization linear program defined over the corresponding directed trees.
3. We use duality to upper-bound the objective value of this LP by the value of a feasible solution for the dual LP. This reveals a direct relation of the approximation ratio of the SD mechanism to a graph-theoretic quantity defined on a directed tree, which we call the cost of a path covering.
4. Our last step is to prove bounds on this quantity; these might be of independent interest and could find applications in other contexts.

Consider an instance I of facility assignment. Recall the interpretation of the problem as a metric bipartite matching and note that without loss of generality, each facility can be assumed to have capacity 1 and $m \geq n$. Unless otherwise specified, agents and facilities are identified by the integers in $[n]$ and $[m]$, respectively.

Now, let O be any assignment on input I , and let S be an assignment returned by the SD mechanism when applied on the instance I_g (where each facility has capacity g). We use a directed graph to represent the triplet I , O , and S as follows. The graph has a node for each facility. Each directed edge corresponds to an agent. A directed edge from a node corresponding to facility j_1 to a node corresponding to facility j_2 indicates that the agent corresponding to the edge is assigned to facility j_1 in O and facility j_2 in S . Observe that there is at most one edge outgoing from each node; this edge corresponds to the agent that is assigned to the facility corresponding to the node in solution O . Furthermore, a node may have up to g incoming edges, corresponding to agents assigned to the facility by the SD mechanism.

Representations as directed g -trees are of particular importance. A *directed g -tree* T is an acyclic directed graph that has a root node r of in-degree 1 and out-degree 0, leaves with in-degree 0 and out-degree 1, and intermediate nodes with in-degree g and out-degree 1. We now show that it suffices to restrict our

⁶ We point out here that statement 1 and a weaker version of statement 2 in Theorem 1 can be obtained as corollaries of results in the literature for the online transportation problem (see [11, 12]). However, we will prove the three statements of Theorem 1 as part of our more general framework.

attention to directed g -trees as graph representations of instances in which the SD mechanism achieves its worst performance.

Lemma 1. *Given a instance I with n agents, an optimal solution O for I and a solution S consistent with the SD mechanism when applied to instance I_g , there is another instance I' with at most n agents, with an optimal solution O' and a solution S' consistent with the application of the SD mechanism on the instance I'_g such that the representation graph of the triplet (I', O', S') is a directed g -tree and such that*

$$\frac{\text{cost}(S, I_g)}{\text{cost}(O, I)} \leq \frac{\text{cost}(S', I'_g)}{\text{cost}(O', I')}.$$

Proof: Let o_i and s_i denote the facility to which agent i is connected in assignments O and S , respectively. We say that agent i is *optimal* if $o_i = s_i$. We say that agent i is *greedy* if $s_i \neq o_i$ and less than g agents are assigned to facility o_i when SD decides the assignment of agent i . This means that $d(A_i, F_{s_i}) \leq d(A_i, F_{o_i})$. We say that agent i is *blocked* if g agents are already assigned to facility o_i when SD decides the assignment of agent i .

Starting from (I, O, S) , we construct a new triplet (I', O', S') as follows:

- First, we remove all optimal agents. This corresponds to removing loops from the representation graph.
- Then, we repeat the following process as long as there exists a blocked agent i that is connected under S to a facility j that is the optimal facility of a greedy agent. In this case, we introduce a new facility j' at point $F_{j'}$ such that $d(A_i, F_{j'}) = d(A_i, F_j)$ and $d(F_{j'}, X) = d(A_i, F_{j'}) + d(A_i, X)$ for every other point X of the space. The second equality guarantees that the set of all points corresponding to locations of agents and facilities that have survived and the newly introduced point $F_{j'}$ is a metric. This can easily be achieved by placing the new facility j' such that it coincides with j on the metric space. We assign agent i to facility j' instead of j ; by the first equality above, this is consistent to the definition of the SD mechanism. In the representation graph, this step adds a new node corresponding to the new facility j' and modifies the directed edge corresponding to blocked agent i so that it is directed to the new node.
- Then, we remove all greedy agents that are not connected under S to optimal facilities of blocked agents together with their optimal facilities.
- Then, for each facility j that is used by $t \geq 2$ agents i_1, i_2, \dots, i_t in S but is not used by any agent in O , we remove facility j and introduce t new facilities j_1, j_2, \dots, j_t such that $d(A_{i_k}, j_k) = d(A_{i_k}, j)$ for $k = 1, \dots, t$ and $d(X, j_k) = d(X, A_{i_k}) + d(A_{i_k}, j_k)$ for every other point X of the space. Again, the second equality guarantees that the set of all points corresponding to locations of agents and facilities that have survived and the newly introduced points F_{j_1}, \dots, F_{j_t} is a metric. For $k = 1, \dots, t$, we assign agent i_k to facility j_k ; by the first equality above, this is consistent to the definition of the SD mechanism. In the representation graph, this step adds t nodes corresponding to the new facilities j_1, \dots, j_t and, for $k = 1, \dots, t$, it modifies the directed

- edge corresponding to blocked agent i_k so that it is directed to the new node j_k , and removes node corresponding to facility j .
- Finally, we remove any facility that is not used by any of the non-removed agents in any of the two solutions.

We denote by I' the resulting instance and by O' the restriction of O to the survived agents. Also, S' is the assignment obtained by the modification of S and considering the survived agents only. We remark that the representation graph of (I', O', S') is a forest of directed g -trees. Indeed, the optimal facility of a greedy agent is not used by any agent in S' ; the corresponding node is a leaf in the representation graph. Now, assume that the representation graph contains a directed cycle; this should consist of directed edges corresponding to blocked agents. By the definition above, this would mean that, for every agent j in this cycle, the assignment of all agents that were assigned by the SD mechanism to the optimal facility o_j took place before the assignment of agent j to a facility; this yields a contradiction and no such cycle exists. The optimal facility of a blocked agent has out-degree 1 and in-degree g . Nodes with zero out-degree have degree exactly 1; these are nodes corresponding to the newly added facilities and serve as roots of the directed g -trees.

Let R be the set of (greedy and optimal) agents removed and observe that $d(A_i, F_{s_i}) \leq d(A_i, F_{o_i})$ for each such agent $i \in R$. Hence, it is

$$\begin{aligned} \frac{\text{cost}(S, I_g)}{\text{cost}(O, I)} &= \frac{\sum_{i \in [n]} d(A_i, F_{s_i})}{\sum_{i \in [n]} d(A_i, F_{o_i})} \leq \frac{\sum_{i \in [n]} d(A_i, F_{s_i}) - \sum_{i \in R} d(A_i, F_{s_i})}{\sum_{i \in [n] \setminus R} d(A_i, F_{o_i}) - \sum_{i \in R} d(A_i, F_{o_i})} \\ &= \frac{\sum_{i \in [n] \setminus R} d(A_i, F_{s'_i})}{\sum_{i \in [n] \setminus R} d(A_i, F_{o'_i})}. \end{aligned}$$

Clearly, if the representation of triplet I', O', S' consists of more than one g -trees, there is an instance I'' and assignments O'' and S'' corresponding to the restriction of (I', O', S') in one of the g -trees which satisfies $\frac{\text{cost}(S, I_g)}{\text{cost}(O, I)} \leq \frac{\text{cost}(S'', I''_g)}{\text{cost}(O'', I'')}$. If O'' is indeed an optimal solution for instance I'' , the proof is complete. Otherwise, we repeat the whole process using instance I'' as I , solution O to be the optimal solution for instance I'' , and the SD solution S'' until the solution O'' obtained is optimal for the g -tree instance obtained at the final step (this condition will eventually be satisfied as the optimal cost decreases in each application of the process). By setting $\tilde{I} = I''$, $\tilde{O} = O''$, and $\tilde{S} = S''$ will then yield the triplet with the desired characteristics. \square

So, in the following, we will focus on triplets (I, O, S) of a facility assignment instance I with at most n agents, with an optimal solution O , and with an SD solution S for instance I_g that have a graph representations as a directed g -tree T . Below, we use \mathcal{P} to denote the set of all paths that originate from leaves. Given an edge e of a g -tree, we use \mathcal{P}_e (respectively, $\tilde{\mathcal{P}}_e$) to denote the set of all paths that originate from a leaf and cross (respectively, terminate with) edge e . We always use e_r to denote the edge incident to the root of a g -tree.

Our next observation is that $\text{cost}(S, I_g)$ is upper-bounded by the objective value of the following linear program.

$$\begin{aligned} & \text{maximize } \sum_{e \in T} z_e \\ & \text{subject to: } z_e - \sum_{a \in p \setminus \{e\}} z_a \leq \sum_{a \in p} d(A_a, F_{o_a}), e \in T, p \in \tilde{\mathcal{P}}_e \\ & \quad z_e \geq 0, e \in T \end{aligned}$$

To see why, interpret variable z_e as the distance of agent corresponding to edge e of T to the facility it is connected to under assignment S . Then, clearly, the objective $\sum_{e \in T} z_e$ represents $\text{cost}(S, I_g)$. Now, how high can $\text{cost}(S, I)$ be? The LP essentially answers this question (partially, because it does not use all constraints of the SD mechanism but sufficiently for our purposes). In particular, the LP takes into account the fact that the distance of agent e to the facility to which it is connected in S is not higher than the distance from the agent to any leaf facility in its subtree; this follows by the definition of the SD mechanism since leaf facilities are by definition available throughout the execution of the SD mechanism. Indeed, consider agent e and a path $p \in \tilde{\mathcal{P}}_e$. Since agent e is connected to facility s_e under SD and not to the facility corresponding to the leaf from which path p originates from, this means that the distance $d(A_e, F_{s_e})$ is not higher than the distance of A_e from the location of the facility corresponding to that leaf. Since d is a metric, this distance is at most $d(A_e, F_{o_e}) + \sum_{a \in p \setminus \{e\}} d(F_{s_a}, F_{o_a}) \leq d(A_e, F_{o_e}) + \sum_{a \in p \setminus \{e\}} (d(A_a, F_{s_a}) + d(A_a, F_{o_a}))$. So, the constraint associated with path $p \in \tilde{\mathcal{P}}_e$ in the LP captures the inequality $d(A_e, F_{s_e}) \leq d(A_e, F_{o_e}) + \sum_{a \in p \setminus \{e\}} d(F_{s_a}, F_{o_a}) \leq d(A_e, F_{o_e}) + \sum_{a \in p \setminus \{e\}} (d(A_a, F_{s_a}) + d(A_a, F_{o_a}))$, by replacing $d(A_e, F_{s_e})$ with z_e and $d(A_a, F_{s_a})$ with z_a and rearranging the terms.

By duality, the cost $\text{cost}(I_g, S)$ of solution S is upper-bounded by the objective value of the dual linear program, defined as follows:

$$\begin{aligned} & \text{minimize } \sum_{p \in \mathcal{P}} x_p \sum_{e \in p} d(A_e, F_{o_e}) \\ & \text{subject to: } \sum_{p \in \mathcal{P}_{e_r}} x_p \geq 1 \\ & \quad \sum_{p \in \tilde{\mathcal{P}}_e} x_p - \sum_{p \in \mathcal{P}_e \setminus \tilde{\mathcal{P}}_e} x_p \geq 1, e \in T, e \neq e_r \\ & \quad x_p \geq 0, p \in \mathcal{P} \end{aligned}$$

Actually, for any feasible solution x of the dual LP, $\text{cost}(S, I_g)$ is upper bounded by the quantity $\sum_{p \in \mathcal{P}} x_p \sum_{e \in p} d(A_e, F_{o_e})$. We will refer to any assignment x over the paths of \mathcal{P} that satisfies the constraints of the dual LP as a *path covering* of the directed g -tree T and will denote its cost by $c(x) = \max_{e \in T} \sum_{p \in \mathcal{P}_e} x_p$. We repeat these definitions for clarity:

Definition 1. Let T be a directed tree. A function $x : \mathcal{P} \rightarrow \mathbb{R}^+$ is called a *path covering* of T if the following conditions hold:

- $\sum_{p \in \mathcal{P}_{e_r}} x_p \geq 1$ for the edge e_r incident to the root of T ;
- $\sum_{p \in \tilde{\mathcal{P}}_e} x_p - \sum_{p \in \mathcal{P}_e \cap \mathcal{P}_f} x_p \geq 1$ if $e \neq e_r$ and f denotes the parent edge of e .

The cost $c(x)$ of x is equal to $\max_{e \in T} \sum_{p \in \mathcal{P}_e} x_p$.

Lemma 2. *Let $g \geq 2$ be an integer, I be a facility assignment instance with an optimal solution O , S be a solution of the SD mechanism when applied on instance I_g , so that the triplet (I, O, S) is represented as a directed g -tree T which has a path covering x . Then, $\text{cost}(S, I_g) \leq c(x) \cdot \text{cost}(O, I)$.*

Proof: Using the interpretation of the variables of the primal LP, duality, and the definition of the cost of path covering x , we have that

$$\begin{aligned} \text{cost}(S, I_g) &= \sum_{e \in T} z_e \leq \sum_{p \in \mathcal{P}} x_p \sum_{e \in p} d(A_e, F_{o_e}) = \sum_{e \in T} d(A_e, F_{o_e}) \cdot \sum_{p \in \mathcal{P}_e} x_p \\ &\leq c(x) \cdot \sum_{e \in T} d(A_e, F_{o_e}) = c(x) \cdot \text{cost}(O, I) \end{aligned}$$

as desired. \square

In order to establish the upper bounds in Theorem 1, it remains to show that path coverings with low cost do exist; this is what we do in the next three lemmas. We start with the Lemma for no augmentation. The proof of the lemma is omitted due to lack of space.

Lemma 3. *Let T be a 1-tree. Then, there is a path covering of T of cost $2^n - 1$.*

In the following, we identify path coverings of low cost for the case of $g \geq 3$ and $g = 2$. The next two lemmas complete the part of Theorem 1 that regards the upper bounds.

Lemma 4. *Let $g \geq 3$ be an integer and T be a g -tree. Then, there is a path covering of T of cost $\frac{g}{g-2}$.*

Proof: We prove the lemma using the following assignment x : for every path p of length ℓ , we set $x_p = \frac{1}{g-2}g^{2-\ell}$ if it contains an edge that is adjacent to the root and $x_p = \frac{g-1}{g-2}g^{1-\ell}$ otherwise.

We will first show that $\sum_{p \in \mathcal{P}_e} x_p = \frac{g}{g-2}$ for every edge e using induction. We will do so by visiting the edges in a bottom-up manner (i.e., an edge will be visited only after its child-edges have been visited) and prove that the equality for edge e using the information that the equality holds for its child-edges. As the basis of our induction, consider an edge e that is adjacent to a leaf at depth $\ell \geq 1$ from the root. If $\ell = 1$, this means that the tree consists of a single edge and there is a single path p with $x_p = \frac{g}{g-2}$. If $\ell \geq 2$, then the paths that contain edge e are those who end at each ancestor of the leaf adjacent to e . Hence,

$$\sum_{p \in \mathcal{P}_e} x_p = \sum_{i=1}^{\ell-1} \frac{g-1}{g-2} g^{1-i} + \frac{1}{g-2} g^{2-\ell} = \frac{g}{g-2}.$$

Now, let us focus on a non-leaf edge e and assume that $\sum_{p \in \mathcal{P}_{e_i}} x_p = \frac{g}{g-2}$ for each child-edge e_i (for $i \in [g]$) of e (this is the induction hypothesis). Let u be the node to which edges e and e_i with $i \in [g]$ are incident. The set of paths in \mathcal{P}_e consists of the following disjoint sets of paths: for each edge e_i and for each path $p \in \tilde{\mathcal{P}}_{e_i}$, set \mathcal{P}_e contains all super-paths of p , i.e., paths originating from the leaf-node reached by p and ending at each ancestor of node u ; we use the notation $\text{sup}(p)$ to denote the set of super-paths of p . Observe that, the definition of x implies that a super-path q of p that is longer than p by j has $x_q = \frac{1}{g-1} g^{1-j} x_p$ if q is adjacent to the root and $x_q = g^{-j} x_p$ otherwise. Hence, assuming that node u is at depth $\ell \geq 1$ from the root, we have that

$$\begin{aligned} \sum_{p \in \mathcal{P}_e} x_p &= \sum_{i=1}^g \sum_{p \in \tilde{\mathcal{P}}_{e_i}} \sum_{q \in \text{sup}(p)} x_q = \left(\sum_{j=1}^{\ell-1} g^{-j} + \frac{1}{g-1} g^{1-\ell} \right) \sum_{i=1}^g \sum_{p \in \tilde{\mathcal{P}}_{e_i}} x_p \\ &= \frac{1}{g-1} \left(\sum_{i=1}^g \sum_{p \in \mathcal{P}_{e_i}} x_p - \sum_{p \in \mathcal{P}_e} x_p \right), \end{aligned}$$

which yields $\sum_{p \in \mathcal{P}_e} x_p = \frac{g}{g-2}$ as desired, since $\sum_{p \in \mathcal{P}_{e_i}} x_p = \frac{g}{g-2}$ by the induction hypothesis.

It remains to show feasibility. Clearly, $\sum_{p \in \mathcal{P}_e} x_p = \frac{g}{g-2} \geq 1$ if e is adjacent to the root. Otherwise, consider an edge e , its parent edge f , and their common endpoint u . Assuming that u is at depth ℓ from the root (and using definitions and observations we used above), we have

$$\sum_{p \in \mathcal{P}_e \cap \mathcal{P}_f} x_p = \sum_{p \in \tilde{\mathcal{P}}_e} \sum_{q \in \text{sup}(p)} x_q = \left(\sum_{j=1}^{\ell-1} g^{-j} + \frac{1}{g-1} g^{1-\ell} \right) \sum_{p \in \tilde{\mathcal{P}}_e} x_p = \frac{1}{g-1} \sum_{p \in \tilde{\mathcal{P}}_e} x_p,$$

which, together with the fact that $\frac{g}{g-2} = \sum_{p \in \mathcal{P}_e} x_p = \sum_{p \in \mathcal{P}_e \cap \mathcal{P}_f} x_p + \sum_{p \in \tilde{\mathcal{P}}_e} x_p$ yields $\sum_{p \in \mathcal{P}_e \cap \mathcal{P}_f} x_p = \frac{1}{g-2}$ and $\sum_{p \in \tilde{\mathcal{P}}_e} x_p = \frac{g-1}{g-2}$ and, consequently, $\sum_{p \in \tilde{\mathcal{P}}_e} x_p - \sum_{p \in \mathcal{P}_e \cap \mathcal{P}_f} x_p = 1$ as desired. \square

Finally, we state the lemma for augmentation factor $g = 2$. The proof is omitted due to lack of space.

Lemma 5. *Let T be an N -node 2-tree. Then, there is a path covering of T of cost at most $\log N$.*

We have shown that the performance of SD significantly improves even with a small augmentation factor. A natural next question is to study its randomized counterpart, RSD. Could randomization help in achieving much better ratios? In the following, we state an approximation guarantee for RSD, when there is no resource augmentation. The proof is omitted due to lack of space.

Theorem 2. *The approximation ratio of RSD without resource augmentation is $\text{ratio}(\text{RSD}) \leq n$.*

4 Lower Bounds

In this section, we provide lower bounds on the approximation ratio with augmentation of the two mechanisms that we study. Interestingly, the constructed instances are all on a simple metric space, the *real line metric*.

Theorem 3. *The approximation ratio of Serial Dictatorship with augmentation factor g in facility assignment instances with n agents is*

1. $\text{ratio}(\text{SD}) \geq 2^n - 1$
2. $\text{ratio}_2(\text{SD}) \geq \log(n + 1)$
3. $\text{ratio}_g(\text{SD}) \geq \frac{g}{g-2} - \delta$ for any $\delta > 0$ when $g \geq 3$.

The approximation ratio of Random Serial Dictatorship is at least $\text{ratio}(\text{RSD}) \geq n^{0.26}$ (without resource augmentation).

We omit the proof of the theorem due to lack of space. The instances that provide the lower bounds as well as the proofs are included in the full version of the paper.

5 Discussion

We proposed a resource augmentation framework for algorithmic mechanism design, where a mechanism, severely limited by the need for truthfulness is given some additional allocative power before being compared to the optimal mechanism, which operates under no restrictions. The framework is applicable to other related problems as well; for example, the bi-criteria algorithms of [3] can be seen as instances of resource augmentation. The framework can also be applied to broader settings where the loss in performance is due to restrictions other than truthfulness, such as fairness [6], stability [7] or ordinality [8, 4]; all the problems in those papers can be studied through the resource augmentation lens. It is not hard to imagine that similar notions like the *price of fairness* [6], could be redefined in terms of resource augmentation.

For the facility assignment problem, we took a positive step in the study of Random Serial Dictatorship, proving approximation ratio bounds when there is no augmentation. It seems like an interesting technical question to obtain (tight) bounds for RSD and for different augmentation factors. It would also be meaningful to consider augmentation factors smaller than 2; note that a similar construction to the one in our main lower bound can be used to show that additive factors can not achieve significantly improved approximations. Finally, it makes sense to consider other truthful mechanisms, beyond the greedy ones. In the full version, we actually prove that for two facilities and no resource augmentation, the approximation ratio of SD is 3, which is optimal among all truthful mechanisms, even randomized ones.

References

1. A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66:689–701, 1998.
2. E. Anshelevich and S. Das. Matching, cardinal utility, and social welfare. *ACM SIGECom Exchanges*, 9(1):4, 2010.
3. E. Anshelevich and S. Sekar. Blind, greedy, and random: Algorithms for matching and clustering using only ordinal information. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 390–396, 2016.
4. H. Aziz, J. Chen, A. Filos-Ratsikas, S. Mackenzie, and N. Mattei. Egalitarianism of random assignment mechanisms. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2016.
5. A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100:295–328, 2001.
6. I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, and M. Kyropoulou. The efficiency of fair division. *Theory of Computing Systems*, 50(4):589–610, 2012.
7. Y. Emek, T. Langner, and R. Wattenhofer. The price of matching with metric preferences. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA)*, pages 459–470. 2015.
8. A. Filos-Ratsikas, S. K. S. Frederiksen, and J. Zhang. Social welfare in one-sided matchings: Random priority and beyond. In *Proceedings of the 7th International Symposium on Algorithmic Game Theory (SAGT)*, pages 1–12, 2014.
9. M. Guo and V. Conitzer. Strategy-proof allocation of multiple items between two agents without payments or priors. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 881–888, 2010.
10. A. Hylland and R. Zeckhauser. The efficient allocation of individuals to positions. *The Journal of Political Economy*, 87(2):293–314, 1979.
11. B. Kalyanasundaram and K. Pruhs. Online weighted matching. *Journal of Algorithms*, 14(3):478–488, 1993.
12. B. Kalyanasundaram and K. Pruhs. The online transportation problem. *SIAM Journal on Discrete Mathematics*, 13(3):370–383, 2000.
13. B. Kalyanasundaram and K. Pruhs. Speed is as powerful as clairvoyance. *Journal of the ACM*, 47(4):617–643, 2000.
14. E. Koutsoupias. Weak adversaries for the k-server problem. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 444–449, 1999.
15. P. Krysta, D. Manlove, B. Rastegari, and J. Zhang. Size versus truthfulness in the House Allocation problem. In *Proceedings of the 15th ACM Conference on Economics and Computation (EC)*, pages 453–470, 2014.
16. A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. *ACM Transactions on Economics and Computation*, 1(4):art. 18, 2013.
17. D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.
18. L.-G. Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4):557–567, 1999.
19. N. Young. The k-server dual and loose competitiveness for paging. *Algorithmica*, 11(6):525–541, 1994.