Efficient Truthful Scheduling and Resource Allocation through Monitoring

Dimitris Fotakis, [1] Piotr Krysta, [2] Carmine Ventre [3]
[1] National Technical University of Athens, Greece
[2] University of Liverpool, UK
[3] King's College London, UK
fotakis@cs.ntua.gr, pkrysta@liverpool.ac.uk, carmine.ventre@kcl.ac.uk

Abstract

We study the power and limitations of the Vickrey-Clarke-Groves mechanism with *monitoring* (VCG^{mon}) for cost minimization problems with objective functions that are more general than the social cost. We identify a simple and natural sufficient condition for VCG^{mon} to be truthful for general objectives. As a consequence, we obtain that for any cost minimization problem with non-decreasing objective μ, VCG^{mon} is truthful, if the allocation is Maximal-in-Range and μ is 1-Lipschitz (e.g., μ can be the L_p-norm of the agents' costs, for any $p \geq 1$ or $p = \infty$). We apply VCG^{mon} to scheduling on *restricted-related machines* and obtain a polynomial-time truthful-in-expectation 2-approximate (resp. $O(1)$-approximate) mechanism for makespan (resp. L_p-norm) minimization. Moreover, applying VCG^{mon}, we obtain polynomial-time truthful $O(1)$-approximate mechanisms for some fundamental *bottleneck network optimization* problems with single-parameter agents. On the negative side, we provide strong evidence that VCG^{mon} could not lead to computationally efficient truthful mechanisms with reasonable approximation ratios for *binary covering* social cost minimization problems. However, we show that VCG^{mon} results in computationally efficient approximately truthful mechanisms for binary covering problems.

Introduction

The effective use of resources is an important goal of any digital system. For example, operating systems allocate the hardware (e.g., RAM memory, CPU time, etc.) to the processes under execution to guarantee "good" performance. However, there are cases in which the hardware is not directly available. Consider, for example, a cloud computing service provider P. Ideally, P would like to allocate customer tasks to the cloud resources so as to provide the "best" possible service. Relevant scenarios include the allocation of programs (a.k.a., jobs) to machines; of packets to routers or, more generally, the selection of a subset of resources that would complete the customer's task. The definition of "best" service might vary and range from interactivity (i.e., the maximum completion time of customers' tasks is minimized) to batch performance (i.e., the total completion time is minimized). The aforementioned optimization problems are known as *machine scheduling*, *bottleneck network optimization* and *total (a.k.a. social) cost optimization*, respectively; typically, the objective for the first two is interactivity, whilst the third is an example of batch optimization.

The question of how P should allocate resources optimally in these (and other similar) contexts is fundamental and has received significant attention by multiple research communities, see, e.g., (Liao 2014; Leyton-Brown 2003; Lombardi and Milano 2012) and the references therein. In modern digital infrastructure, dominated by outsourcing and distributed resource allocation, a notable additional obstacle for P is the fact that the resources are often controlled by self-interested entities, a.k.a. *selfish agents*, operating them according to their own goals (e.g., to avoid resource overloading), not necessarily aligned with P's objective. This is the standard setting of *Algorithmic Mechanism Design* (Nisan et al. 2007), where we seek *incentive compatibility*, in addition to the algorithmic objectives of computational efficiency and optimal resource allocation. Specifically, we aim at *truthful* mechanisms, that run in polynomial-time and approximate as well as possible the objective function at hand. A truthful mechanism guarantees that it is in each agent's interest to report to the mechanism her own private information (i.e., the features of the hardware they control), commonly termed *type*. In our examples above, a type could be the speed of a machine for a particular job to execute; the latency of a router, and, more generally, the hardware cost to execute a customer's task.

Unfortunately, for none of the problems of interest it is known how to design such mechanisms. The renown Vickrey-Clarke-Groves (VCG) mechanisms, see, e.g., (Nisan et al. 2007, Chapter 9), are about the only general technique known to obtain truthfulness. They require the computation of the optimum social cost solution, implying obvious limitations for their use. More specifically, for NP-hard social cost optimization problems, VCG does not run in polynomial-time, unless P = NP (when, as in our case, there is no suitable "Maximal-in-Range" approximation). Moreover, even when they can be implemented in a computationally efficient way, VCG mechanisms can return bad approximations to the min-max objective of machine scheduling and bottleneck network optimization (Nisan and Ronen 2001), because their truthfulness crucially depends on social cost optimization.

In this work, we investigate to which extent we can overcome these limitations of VCG mechanisms through *monitoring* (Kovács, Meyer, and Ventre 2015; Koutsoupias 2014). The idea is to let the mechanism designer (a.k.a. the principal) exert some control on the agents during the execution of the mechanism. Already in their seminal paper, (Nisan and Ronen 2001) considered a model wherein agents overbidding their cost could be *monitored* and forced to pay as much. This assumption is reasonable, when the principal has the power to appropriately decrease the agents' utility, if during the execution of the mechanism, she realizes that some agents have over-reported their cost. This is exactly the situation in which P is; if, for example, an agent exaggerates the time her machine takes to execute a certain job, then P can keep the machine busy that long by, e.g., charging the difference to the agent. Unlike (Nisan and Ronen 2001) (see also *compensation-and-penalty* mechanisms in (Shoham and Leyton-Brown 2009, Sec. 10.6.1)), we do not assume any punishment for underbidding and do not compute payments based on the actual costs incurred by the agents during the implementation of the chosen outcome; our only assumption is that the principal is able to monitor over-reported costs.

Monitoring is by now well-established in Algorithmic Mechanism Design. The difference between monitoring and the so-called *verification* is discussed by (Penna and Ventre 2014). Kovács, Meyer, and Ventre (2015) study mechanisms with monitoring where the principal is the operating system and the agents are computational processes. Different payment schemes for mechanisms with monitoring are studied in (Serafino, Ventre, and Vidali 2020). The monitoring paradigm is also studied in absence of transfers in (Koutsoupias 2014; Giannakopoulos, Koutsoupias, and Kyropoulou 2016), and in the context of truthfulness with bounded rationality in (Ferraioli and Ventre 2017; Kyropoulou and Ventre 2019).

Our Contributions. Motivated by applications of mechanism design to scheduling and resource allocation problems, where monitoring of over-reported costs is natural and easy to implement, we investigate the power and the limitations of the *Vickrey-Clarke-Groves mechanism with monitoring* (VCG^{mon}) for cost minimization problems with objective functions that are more general than the social cost.

We start with identifying two natural algorithmic properties, cf. (2) and (3) below, which together provide a simple necessary and sufficient condition for VCG^{mon} to be truthful for any objective function μ (i.e., μ is not necessarily the social cost). At the conceptual level, conditions (2) and (3) are subtle extensions of the classic monotonicity property, which has been extensively studied in the context of truthful mechanisms (Saks and Yu 2005; Nisan et al. 2007), and impose an additional continuity property on the algorithm. As a nice analogue of the VCG theorem, in our context, we prove that Maximal-in-Range (MIR) algorithms satisfy these conditions if the objective function μ is 1-Lipschitz (e.g., μ can be the L_p-norm of the agent costs, for any $p \geq 1$ or $p = \infty$).

To establish the generality of our approach, we apply our truthful VCG mechanisms with monitoring to three broad classes of minimization problems: scheduling on unrelated machines, bottleneck network optimization and binary covering problems with social cost. For each of these classes, we prove essentially tight positive and corresponding negative results on the approximate optimality of such mechanisms. Our positive results are polynomial-time truthful approximate mechanisms with monitoring, and the matching negative results are either by known computational lower bounds or by our new unconditional impossibility results.

The makespan minimization for scheduling on unrelated machines is one of the flagship problems in Algorithmic Mechanism Design. Its approximability by (deterministic or randomized) truthful mechanisms has received significant attention. Finding a truthful mechanism with sublinear approximation ratio for this problem is a long-standing open problem, since the seminal work of (Nisan and Ronen 2001). The major complication is the multi-parameter agents-machines: their private information are the processing times for each job. Truthful mechanisms were only obtained with ratios of $O(n)$, where n is the number of machines (Nisan and Ronen 2001; Lu and Yu 2008b,a). Lower bounds of $\Omega(n)$ on the approximation ratio of certain classes of truthful mechanisms were shown in (Nisan and Ronen 2001; Saks and Yu 2005; Ashlagi, Dobzinski, and Lavi 2012). Christodoulou, Koutsoupias, and Kovács (2020) proved a lower bound of $\Omega(\sqrt{n})$ for the more general case where the machine costs are submodular. These lower bounds only assume truthfulness and hold even for exponential time mechanisms (i.e., even if exponential time was available, VCG could not achieve any nontrivial approximation ratio for makespan minimization on unrelated machines!). If we drop truthfulness, a classical 2-approximation algorithm is known (Shmoys and Tardos 1993). We apply our VCG^{mon} mechanisms to scheduling on unrelated machines, and investigate their power and limitations. We emphasize that monitoring for over-reported processing times is a natural and common assumption in mechanism design for scheduling problems (see e.g., the weak execution model in (Angel, Bampis, and Pascual 2006; Angel et al. 2009)). We can show that VCG^{mon} is not truthful for the fractional solution of the linear program (LP) used by (Shmoys and Tardos 1993), due to the parameter pruning step. Given that all known $O(1)$-approximation algorithms for this problem are based on either the LP of (Shmoys and Tardos 1993) or the so-called configuration LP (which also applies parameter pruning), an interesting open question is whether there exists a truthful (in expectation) $O(1)$-approximate mechanism with monitoring for makespan minimization on unrelated machines. On the positive side, we focus on the special case of makespan minimization on *restricted-related machines*, where each machine i has a private speed s_i and a private subset of jobs J_i that the machine can process (so, machine types are still multi-dimensional). We show that there is a truthful-in-expectation 2-approximate mechanism with monitoring. Moreover, our approach generalizes to L_p-norm minimization with $O(1)$ approximation. Azar et al. (2017) gave a truthful-in-expectation 2-approximation mechanism for makespan minimization on restricted-related machines, but under the assumption that sets J_i are public (which makes machine types single-dimensional). Our mechanism remains 2-approximate w.r.t. the makespan objective, if actual money transfers from the mechanism to the machines are not possible (e.g., imagine load balancing in volunteering) and the payments required for truthfulness are implemented by artificial delays in the schedule of each machine (i.e., in the *money burning* framework of (Hartline and Roughgarden 2008)).

In a bottleneck network optimization problem, we are given a network with edge costs and seek a certain feasible minimum-cost subnetwork where the cost of its costliest edge is minimized. For each of the bottleneck network optimization problems in Hochbaum and Shmoys (1986), we apply VCG^{mon} and show that there exists a deterministic approximation mechanism that is truthful with monitoring for single-dimensional agents, i.e., they own a single edge. Their approximation ratios are the same as those in (Hochbaum and Shmoys 1986) and are, for many of those problems, best polynomial-time approximations. We achieve this by proving that the generic bottleneck algorithm of Hochbaum and Shmoys (1986) possesses the required conditions (2) and (3), needed for the truthfulness of VCG^{mon}. These mechanisms cannot be extended to multi-dimensional agents for these problems without violating these conditions. We note that the bottleneck algorithm from (Hochbaum and Shmoys 1986) is monotone for single-dimensional agents, and therefore truthful in the standard sense, without monitoring. Our conditions (2) and (3), however, are more demanding and yield more flexible payment functions (cf. discussion in the next section about an equal-cost interpretation of VCG^{mon} and connections to money burning) – as opposed to the "threshold payment scheme" for standard truthfulness (see e.g., (Nisan et al. 2007, Sec. 13.1)). Moreover, VCG^{mon} payments have a more explicit/direct definition, which helps their computation and simplicity of the mechanism (i.e., how easy it is for humans to understand how to behave). For more detailed motivation, see also the second paragraph in the section on bottleneck network problems. Leucci, Mamageishvili, and Penna (2018) prove that no deterministic (standard) truthful mechanism with money can achieve an n-approximation for the bottleneck s-t-shortest path problem with multi-dimensional agents (each agent owns many edges), where n is the number of agents. This problem is solvable in polynomial time, if we do not insist on truthfulness. Interestingly, by applying monitoring along each dimension separately, we can show that VCG^{mon} with the optimal polynomial-time algorithm is truthful for this problem with multi-dimensional agents. We defer the details of this result to the full paper.

For binary covering problems with social cost objective, we interestingly connect approximation with truthfulness of our mechanisms. We first prove that no algorithm with a bounded approximation ratio is continuous, even for single-dimensional agents, thus implying that the only truthful mechanisms are either optimal (by using an optimal MIR algorithm) or have an approximation guarantee that is arbitrarily close to 1, i.e., (Fully) Polynomial-Time Approximation Schemes ((F)PTASs). This result, which might be of independent interest, is very general and applies to multiple covering problems and corresponding deterministic approximation algorithms[1]. The situation seems quite similar to VCG without monitoring, where if the algorithm is not optimal, or not MIR, we lose truthfulness, see, e.g., (Nisan et al. 2007). The parallel with VCG, and with classical truthfulness, is even more striking for binary covering problems; we, in fact, extend a result from (Dughmi and Roughgarden 2014) to prove that any MIR algorithm for a large class of objective functions is actually optimal. These are, to our best knowledge, the first results showing the limits of mechanisms with monitoring, where the principal is able to monitor the agent costs at runtime and to mildly penalize over-reported costs (see e.g., (Caragiannis et al. 2012; Fotakis and Zampetakis 2015; Ferraioli and Ventre 2018) for lower bounds for weaker notions of verification). Truthfulness is very fragile here and any kind of fixed approximation guarantee, no matter how good, leads to manipulability. On the positive side, we show that every deterministic approximation algorithm for any binary covering problem with social cost provides an approximately truthful mechanism with monitoring. Examples of such algorithms and problems include all of those mentioned in Footnote 1. This result complements our impossibility result and is especially interesting for problems with PTASs/FPTASs, e.g., minimum cost spanning tree with budget constraint and multi-unit reverse auctions (Grandoni et al. 2014). In such cases we can control the truthfulness factor to any desired accuracy by simply allowing for higher running time. Interestingly, for many of these problems and algorithms, there exist instances, where the "truthfulness gap" actually reaches the best possible approximation ratio, and thus, our approximate truthfulness results are tight for those algorithms and problems. For instance, the tight example of the greedy algorithm for minimum cost set cover problem, see (Vazirani 2001, Example 2.5), shows that VCG^{mon} can only be $\Omega(\log(n))$-truthful (see Def. 3 of approximate truthfulness).

VCG with Monitoring for General Objectives

Let Π be an optimization problem with n agents and $\mathcal{O}$ the set of outcomes, i.e., feasible solutions, to problem Π. Each agent i has a *cost function*, called *type*, $t_i : \mathcal{O} \to \mathbb{R}_{>0}$. For $x \in \mathcal{O}$, $t_i(x)$ is the cost paid by agent i to implement x. The type t_i is *private knowledge* of agent i. The set of all legal cost functions t_i, denoted by D_i, is called the *domain* of agent i. After each agent has reported or *bid* a (true or false) cost function $b_i \in D_i$, a *mechanism* determines an outcome $x \in \mathcal{O}$ and a payment p_i to each agent i. In summary, by letting $D = D_1 \times \ldots \times D_n$, a *mechanism* $\mathcal{M}$ is a pair (f, p), where $f : D \to \mathcal{O}$ is an algorithm (a.k.a. *social choice function*) that maps agents' costs to a feasible solution in $\mathcal{O}$; and $p : D \to \mathbb{R}^n$ maps cost vectors to payments to each agent i. For mechanism $\mathcal{M} = (f, p)$, let $u_i(b_i, \mathbf{b}_{-i})$ denote the *utility* of agent i for the output computed by $\mathcal{M}$ on input

[1] To name a few, the minimum cost set cover (SC) problem and the primal-dual, deterministic LP rounding and Chvatal's greedy algorithms for SC; MST-based algorithm for metric Steiner Tree; primal-dual algorithm for Steiner Forest and Jain's iterative rounding algorithm for Steiner Network; see (Vazirani 2001) for an overview of these approximation algorithms.

$(b_i, \mathbf{b}_{-i})$ and evaluated by t_i. Since the type t_i is private knowledge of agent i, she might find it profitable to bid $b_i \neq t_i$. We are interested in mechanisms for which truthtelling is a dominant strategy for each agent.

Definition 1 (Truthful mechanisms)**.** *A mechanism $\mathcal{M} = (f, p)$ is* truthful *if for any i, and for all bids $\mathbf{b}_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$ of the agents other than i, and any $b_i \in D_i$, $u_i(t_i, \mathbf{b}_{-i}) \geq u_i(b_i, \mathbf{b}_{-i})$.*

Often, u_i is equal to the payments $p_i(\mathbf{b})$ received from the mechanism minus the *true* cost $t_i(f(\mathbf{b}))$ paid by agent i for the mechanism's outcome $f(\mathbf{b})$. We focus on the mechanism design paradigm of *mechanisms with monitoring*, where this quasi-linear definition is retained, but costs paid by the agents for the allocated solution are tied to their bids. Intuitively, monitoring means that agents with over-reported cost for the chosen outcome, i.e., if $b_i(f(\mathbf{b})) > t_i(f(\mathbf{b}))$, have to "work" up to cost $b_i(f(\mathbf{b}))$ instead of the true cost $t_i(f(\mathbf{b}))$.

Definition 2 (Mechanism with monitoring)**.** *In a* mechanism with monitoring *$\mathcal{M}^{mon} = (f, p)$, the bid b_i is a* lower bound *on agent i's cost of using $f_i(b_i, \mathbf{b}_{-i})$. So, agent i is allowed to have a real cost higher than $b_i(f(\mathbf{b}))$, but not lower. Formally, $u_i(b_i, \mathbf{b}_{-i}) := p_i(\mathbf{b}) - \max\{t_i(f(\mathbf{b})), b_i(f(\mathbf{b}))\}$.*

The $\mathrm{VCG}_\mu^{\mathrm{mon}}$ mechanism. Let f be an algorithm for problem Π and $\mu : \mathcal{O} \times D \to \mathbb{R}_{\geq 0}$ be the objective function of Π mapping outcomes and bid vectors to non-negative reals. The second argument of μ specifies the bids used to calculate the value of a solution (first argument of μ). A *VCG_μ^{mon} mechanism* (f, p) pays agent i an amount of $p_i(b_i, \mathbf{b}_{-i}) = h_i(\mathbf{b}_{-i}) - \mu(f(\mathbf{b}), \mathbf{b}) + b_i(f(\mathbf{b}))$. Hence, if i bids truthfully, i's utility becomes $u_i(t_i, \mathbf{b}_{-i}) = h_i(\mathbf{b}_{-i}) - \mu(f(t_i, \mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i})) + t_i(f(t_i, \mathbf{b}_{-i})) - t_i(f(t_i, \mathbf{b}_{-i})) = h_i(\mathbf{b}_{-i}) - \mu(f(t_i, \mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i}))$, for some function $h_i(\mathbf{b}_{-i})$ not depending on i's bid. Since truthfulness is independent of the choice of $h_i(\mathbf{b}_{-i})$, as in the case without monitoring, we omit $h_i(\mathbf{b}_{-i})$ for sake of brevity and simplicity. This mechanism is a simple extension of VCG, where the generic cost function μ plays the role of the social cost. Then, truthfulness of $\mathrm{VCG}_\mu^{\mathrm{mon}}$ is equivalent to: for all $i, \mathbf{b}_{-i}, t_i, b_i$,

$$\begin{aligned} \mu(f(t_i,\mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i})) \leq{}& \mu(f(\mathbf{b}), \mathbf{b}) \\ &- b_i(f(\mathbf{b})) + \max\{t_i(f(\mathbf{b})), b_i(f(\mathbf{b}))\}. \end{aligned} \tag{1}$$

In fact, (1) is equivalent to the following: for all $i, \mathbf{b}_{-i}, t_i, b_i$,

$$\begin{gathered} t_i(f(\mathbf{b})) \leq b_i(f(\mathbf{b})) \Longrightarrow \\ \mu(f(t_i, \mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i})) \leq \mu(f(\mathbf{b}), \mathbf{b}); \end{gathered} \tag{2}$$

$$\begin{aligned} t_i(f(\mathbf{b})) > b_i(f(\mathbf{b})) \Longrightarrow{}& \mu(f(t_i, \mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i})) \\ &\leq \mu(f(\mathbf{b}), \mathbf{b}) - b_i(f(\mathbf{b})) + t_i(f(\mathbf{b})). \end{aligned} \tag{3}$$

(2) requires that μ (and f) should be monotone in the agent bids. Furthermore, (3) requires that μ (and f) should be continuous, in the sense that a change δ in an agent's bid should change the μ-value of f's outcome by at most δ.

We now discuss the role of $h_i(\mathbf{b}_{-i})$. Standard VCG uses the "Clarke tax" to ensure individual rationality, i.e., the utility of truthtelling agents being non-negative. However, the payment's flexibility can be leveraged to explore different properties of $\mathrm{VCG}_\mu^{\mathrm{mon}}$ mechanisms. By setting $h_i(\mathbf{b}_{-i}) = 0$, for all i and $\mathbf{b}_{-i}$, we obtain an *equal-cost* mechanism, where all agents have the same utility (defined as in Def. 2) equal to $\mu(f(t_i, \mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i}))$. Such mechanisms were introduced in context of facility location in (Fotakis and Tzamos 2014) and are particularly useful when there are no monetary tranfers and payments should be implemented as, e.g., waiting times (an agent is delayed by amount "equal" to the payments imposed by the mechanism). This is the interpretation used in (Kovács, Meyer, and Ventre 2015). Since the transfers required to equalize agents' utilities take a form of wasted resources in this case, it is reasonable to consider money burning objectives and include payments in the objective value. One can see that such a variant of $\mathrm{VCG}_\mu^{\mathrm{mon}}$ provides approximation guarantees in this (more demanding) money burning setting that are not far from the ratio to the optimum cost alone. We defer these details to the full paper.

$\mathrm{VCG}_\mu^{\mathrm{mon}}$ and Maximal-in-Range Mechanisms

Let Π be an optimization problem with an objective function μ to be minimized. A deterministic *Maximal-in-Range* (MIR) mechanism for Π with a range $\mathcal{R} \subseteq \mathcal{O}$ of feasible solutions uses the following MIR algorithm f: given the bids $\mathbf{b}$, it computes a minimizer of the objective value $\mu(x, \mathbf{b})$ over all $x \in \mathcal{R}$. Note that range $\mathcal{R}$ is independent of the bids $\mathbf{b}$. Based on (2) and (3), we next show that if f is Maximal-in-Range, (sub)linearity of μ suffices for truthfulness of $\mathrm{VCG}_\mu^{\mathrm{mon}}$.

If f is an MIR algorithm for μ, its range is $\mathcal{R} = \{f(\mathbf{b}) : \mathbf{b} \in D\}$. An MIR algorithm f for μ satisfies

$$\mu(f(t_i, \mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i})) \leq \mu(f(\mathbf{b}), (t_i, \mathbf{b}_{-i})) \tag{4}$$

for all $t_i, b_i, \mathbf{b}_{-i}$. A non-decreasing function μ (i.e., $t_i(x) > b_i(x)$ implies $\mu(x, (t_i, \mathbf{b}_{-i})) \geq \mu(x, \mathbf{b})$) is 1-*Lipschitz* in dimension i if for any $x \in \mathcal{O}$, $\mathbf{b} \in D$ and $t_i \in D_i$,

$$t_i(x) \geq b_i(x) \Longrightarrow \mu(x, (t_i, \mathbf{b}_{-i})) - \mu(x, \mathbf{b}) \leq t_i(x) - b_i(x).$$

This means that μ grows at most linearly in dimension i, and if it is differentiable, $\frac{\partial \mu(x,\mathbf{b})}{\partial b_i(x)} \leq 1$. If μ is 1-Lipschitz in every dimension $i = 1, \ldots, n$, then we omit the dimensions.

Theorem 1. *Let μ be a non-decreasing 1-Lipschitz function and f be MIR for μ. Then VCG_μ^{mon} using f is truthful.*

Proof. f is MIR for μ, so by (4) truthfulness is implied by

$$\begin{aligned} \mu(f(\mathbf{b}), \mathbf{t}) \leq{}& \mu(f(\mathbf{b}), \mathbf{b}) \\ &- b_i(f(\mathbf{b})) + \max\{t_i(f(\mathbf{b})), b_i(f(\mathbf{b}))\}, \end{aligned} \tag{5}$$

where $\mathbf{t} = (t_i, \mathbf{b}_{-i})$. If $\max\{t_i(f(\mathbf{b})), b_i(f(\mathbf{b}))\} = b_i(f(\mathbf{b}))$ then (5) simply requires μ to be non-decreasing. Otherwise, if we have that $\max\{t_i(f(\mathbf{b})), b_i(f(\mathbf{b}))\} = t_i(f(\mathbf{b}))$, then (5) follows by 1-Lipschitzness. □

Notably, Theorem 1 applies to objectives μ that minimize the maximum agent cost, the L_p-norm of agent costs, and the social cost (i.e., the L_1-norm of agent costs).

Scheduling on Unrelated Machines

Next, we present a randomized truthful-in-expectation mechanism with monitoring for makespan minimization. In scheduling on unrelated machines, we seek a balanced schedule of set J of n jobs to a set M of m selfish machines. Each machine i has a vector $\mathbf{p}_i = (p_{ij})_{j\in J}$, where $p_{ij} \in \mathbb{N} \cup \{\infty\}$ denotes the processing time of job j on machine i. $p_{ij} = \infty$ means that job j cannot be processed by machine i. Wlog., we assume that for each job j, there are at least two machines with finite processing time for j (otherwise, scheduling j is trivial; in our mechanism with monitoring, falsely declaring a finite processing time as infinite, or vice versa, is dominated by truthful reporting). An assignment $\mathcal{J} = (J_1, \dots, J_m)$ is a partition $J_1, \dots, J_m$ of J, where jobs J_i are processed by machine i. The load ℓ_i of machine i in $\mathcal{J}$ is $\ell_i(\mathcal{J}) = \sum_{j\in J_i} p_{ij}$.

The standard objective μ is the *makespan*, i.e., compute an assignment $\mathcal{J}$ that minimizes $\max_{i\in M}\{\ell_i(\mathcal{J})\}$. The vector $\mathbf{p}_i = (p_{ij})_{j\in J}$ of job processing times on machine i is i's private type. Each machine i aims to maximize her utility (expected utility, for randomized mechanisms), defined as i's payment for assignment $\mathcal{J}$ minus i's load in $\mathcal{J}$.

We consider the special case of *restricted-related machines* (Azar et al. 2017), where each job has a publicly known processing time p_j. The private type of each machine i consists of the subset $J^i \subseteq J$ of jobs that i can process and of i's speed $s_i \in \mathbb{N}^*$. Hence, $p_{ij} = p_j/s_i$, if $j \in J^i$ and $p_{ij} = \infty$, if $j \notin J^i$. The load ℓ_i of machine i in assignment $\mathcal{J}$ is $\ell_i(\mathcal{J}) = \sum_{j\in J_i} p_j/s_i$, if $J_i \subseteq J^i$, and $\ell_i(\mathcal{J}) = \infty$, otherwise. (In (Azar et al. 2017), only the speeds s_i are private).

We study properties of the following randomized version of $\text{VCG}_\mu^{\text{mon}}$ for scheduling on unrelated machines.

Fractional Solution. We compute the minimum value of T for which the following linear program is feasible:

$$\sum_{i\in M} x_{ij} = 1 \qquad \forall j \in J$$

$$\sum_{j\in J} x_{ij}p_{ij} \leq T \qquad \forall i \in M$$

$$x_{ij} \geq 0\,,\ \ p_{ij} > T \Rightarrow x_{ij} = 0 \qquad \forall i \in M,\ j \in J$$

We refer to this system as $\text{LP}(T)$. Let T^* be the minimum value of T for which $\text{LP}(T)$ is feasible and let $(x^*_{ij})_{i\in M, i\in J}$ be a basic feasible solution (bfs) of $\text{LP}(T^*)$.

Randomized Rounding. We obtain an integral assignment by applying randomized rounding of Kumar et al. (2009); Lavi and Swamy (2009) to a bfs $(x^*_{ij})_{i\in M, j\in J}$ of $\text{LP}(T^*)$. Let X_{ij} be indicator random variables denoting that job j is assigned to machine i. The randomized rounding procedure ensures that for all $i \in M$ and $j \in J$, $\mathbb{E}[X_{ij}] = x^*_{ij}$, and for any machine i, the following holds with certainty (see also (Lavi and Swamy 2009, Lemma 4.2)): $\sum_{j\in J} X_{ij}p_{ij} \leq \sum_{j\in J} x^*_{ij}p_{ij} + \max_{j:x^*_{ij}>0}\{p_{ij}\}$.

Payments. Machine i receives payment $\sum_{j\in J} x^*_{ij}p_{ij} - T^*$ from the mechanism (recall that we omit the h_i term from the payments, because truthfulness does not depend on it).

Approximation Guarantee. Lavi and Swamy (2009, Lemma 4.2) proved that the above algorithm is 2-approximate for makespan minimization on unrelated machines (and also on restricted-related machines). In fact, (Lavi and Swamy 2009, Lemma 4.2) implies that the load of each machine i is at most $T^* + \max_{j:x^*_{ij}>0}\{p_{ij}\} \leq 2T^*$, with certainty[2].

Truthfulness of $\text{VCG}_\mu^{\text{mon}}$. The expected utility of machine i with monitoring is

$$\sum_{j\in J} x^*_{ij}p'_{ij} - T^* - \mathbb{E}\left[\max\left\{\sum_{j\in J} X_{ij}p_{ij}, \sum_{j\in J} X_{ij}p'_{ij}\right\}\right] \qquad (6)$$

where p_{ij} (resp. p'_{ij}) is the true (resp. reported) processing time of each job j on machine i (recall that the solution x^*_{ij} is computed wrt. the reported processing rimes). The last term in (6) corresponds to $-\max\{t_i(f(\mathbf{b})), b_i(f(\mathbf{b}))\}$ in Definition 2, and the first term corresponds to $b_i(f(\mathbf{b}))$ in the definition of $\text{VCG}_\mu^{\text{mon}}$ payments (we use expected values here, because the mechanism is randomized). For truthfulness-in-expectation, we compute the payments using the optimal fractional value T^* of $\text{LP}(T)$, instead of the expected makespan $\mathbb{E}[\mu(f(\mathbf{b}), \mathbf{b})]$ of the integral assignment obtained by randomized rounding (which may not satisfy the equivalent of (2) and (3)). Since T^* is an upper bound on the expected machine load, we can use $-T^*$, instead of $-\mathbb{E}[\mu(f(\mathbf{b}), \mathbf{b})]$ for the payments of $\text{VCG}_\mu^{\text{mon}}$ and i's expected utility in (6).

The randomized version of $\text{VCG}_\mu^{\text{mon}}$ above is not truthful for the general setting of scheduling on unrelated machines, due to the parameter pruning step in 3rd line of $\text{LP}(T^*)$. So, we focus on restricted-related machines. We consider non-trivial instances, where each job j belongs to at least two sets J^i. Then, (6) becomes $\left(\frac{1}{s'_i} - \max\left\{\frac{1}{s_i}, \frac{1}{s'_i}\right\}\right)\sum_{j\in J} x^*_{ij}p_j - T^*$, provided all jobs with $x^*_{ij} > 0$ belong to the true set of admissible jobs of machine i. Otherwise, the expected utility of machine i is $-\infty$.

Theorem 2. *The randomized version of $\text{VCG}_\mu^{\text{mon}}$ is truthful-in-expectation for scheduling on restricted-related machines (note that types are multidimensional).*

If monetary transfers are infeasible, we can set $h_i = 0$ and implement the payments as a delay of $T^* - \sum_{j\in J} x^*_{ij}p_{ij}$, introduced before machine i starts processing jobs. Thm. 2 applies and the mechanism is truthful-in-expectation for restricted-related machines. An approximation ratio of 2 for makespan (resp. of $2n^{1/p}$ for L_p-norm minimization) follows from (Lavi and Swamy 2009, Lemma 4.2) (resp. from (Kumar et al. 2009, Sec. 4)), even in the money burning setting.

[2]We can extend the proof of Theorem 2 to show that the randomized rounding algorithm in (Kumar et al. 2009, Sec. 4) is truthful with payments (6). The second case in the proof of Theorem 2 can deal with (Kumar et al. 2009, constraints (18)) by convexity of (18), giving a randomized truthful-in-expectation $O(1)$-approximation for minimizing the L_p-norm of machine loads, for any $p \geq 1$.

Bottleneck Network Optimization Problems

The *bottleneck traveling salesperson* (BTSP) is the following problem. Given a weighted complete graph, with edge weights obeying triangle inequality, find a Hamiltonian cycle in this graph where the most costly edge is as cheap as possible. We prove here that the 2-approximation algorithm of Hochbaum and Shmoys (1986) for BTSP implies a truthful $\text{VCG}_\mu^{\text{mon}}$ mechanism for single-dimensional agents, where μ corresponds to the above objective. This implies truthful mechanisms with monitoring for all bottleneck network optimization problems in (Hochbaum and Shmoys 1986), with the same approximation guarantees as the algorithms therein.

To further motivate the results in this section, we look at the equal-cost implementation of $\text{VCG}_\mu^{\text{mon}}$ and money burning. Agent i's payment in mechanism $\text{VCG}_\mu^{\text{mon}}(f,p)$ is $p_i(b_i, \mathbf{b}_{-i}) = h_i(\mathbf{b}_{-i}) - \mu(f(\mathbf{b}), \mathbf{b}) + b_i(f(\mathbf{b}))$. To compute all the p_i's, one only needs to run algorithm $f(\mathbf{b})$ once, and, possibly, again when computing $h_i(\mathbf{b}_{-i})$'s. But if we used the "threshold" payment, following from monotonicity of f in the standard sense, we would need to use binary search for each agent, that requires more time. Importantly, if, for instance, $h_i(\mathbf{b}_{-i}) = 0$, for all i and $\mathbf{b}_{-i}$, $\text{VCG}_\mu^{\text{mon}}$ would be an equal cost mechanism, where payments can be interpreted as, e.g., waiting times imposed by the mechanism. In this case, our mechanisms below maintain the *same* approximation factor also for the money burning objective accounting for the payments as well as the maximum cost. Finally, our mechanism is a simple example of continuous mechanism, i.e., obeying (2)-(3), with an easy-to-follow proof.

The Mechanism. To model bottleneck network optimization problems, suppose that $G_C = (V, E_C)$ is the complete input graph with edge costs $c_{e_1} \leq c_{e_2} \leq \cdots \leq c_{e_m}$, $|V| = n$ and $|E_C| = m = \binom{n}{2}$. We assume that costs obey the triangle inequality (this assumption is only needed for the approximation guarantees but not for truthfulness). We will define these problems using BTSP as a running example.

Let $\mathcal{G}$ be the set of all feasible solutions to the problem on graph G_C. For instance, $\mathcal{G}$ contains all subgraphs $(V, E') \subseteq G_C$ that are Hamiltonian cycles. Given any positive $c \geq 0$, we define the bottleneck subgraph as $bottleneckG(c) = (V, E')$, where $E' = \{e \in E_C : c_e \leq c\}$. Let $G = (V, E')$ be an arbitrary subgraph of G_C, then let $max(G) = \max_{e \in E'} c_e$.

Algorithm 1 is a generic approximation algorithm for solving bottleneck problems, with $t > 0$ a fixed, usually

Algorithm 1: The generic bottleneck algorithm.

procedure $bottleneck(\mathcal{G}, G_C, t)$
  $i := 0$
  repeat
    $i := i + 1;\ \ G_i := bottleneckG(c_{e_i})$
    $test' := test(\mathcal{G}, G_i, t)$
  until $test'$ is not a certificate of failure
  return $test'$

Algorithm 2: The test procedure.

procedure $test(\mathcal{G}, G, 2)$
  if graph G is not biconnected **then**
    return certificate of failure
  else
    return a Hamilton cycle in G^2

small, integer. It either returns as $test'$ a certificate of failure or returns a feasible solution to the bottleneck problem in the t-th power $(G_i)^t$ of the graph $G_i = (V, E')$, where $(G_i)^t = (V, (E')^t)$ and $(E')^t = \{(v_0, v_\ell) : \exists v_1, v_2, \ldots, v_{\ell-1}$ s.t. $(v_{s-1}, v_s) \in E', s = 1, 2, \ldots, \ell, \ell \leq t\}$. Note, for all the problems in (Hochbaum and Shmoys 1986), finding a feasible solution to the problem in mind in G_i is NP-complete, e.g., finding a Hamiltonian cycle. That is why they relax the problem and find a feasible solution in the graph $(G_i)^t$ with $t \geq 2$ instead.

Hochbaum and Shmoys (1986) prove that if procedure $test$ is a poly-time algorithm outputting a certificate of failure if there is no feasible solution to the problem in G_i and outputs a feasible solution in $(G_i)^t$ otherwise, then Algorithm 1 is a poly-time t-approximation to the given bottleneck problem. Given the output solution $test'$ of Algorithm 1, the cost of this solution is c_{e_i}, where i is the last value of the variable i. For BTSP problem there is a poly-time $test$ procedure for $t = 2$, Algorithm 2 (Hochbaum and Shmoys 1986).

A generic network bottleneck mechanism design problem has m single-dimensional agents, each owns an edge $e \in E_C$ and has cost c_e as private data. Let $\mathbf{b}, \mathbf{t} \in \mathbb{R}_{\geq 0}^{|E_C|}$ be the vector of the declared costs and agents' true costs, respectively. We assume that the declared costs obey the triangle inequality. Let (f, p) be the $\text{VCG}_\mu^{\text{mon}}$ mechanism for the bottleneck problem. For instance, for BTSP, f is Algorithm 1 with $t = 2$, and p is the $\text{VCG}_\mu^{\text{mon}}$ payment with respect to cost function μ defined as follows. We run Algorithm 1 with the declared costs $\mathbf{b}$ and the cost of $f(\mathbf{b})$ is the cost b_{e_i} of the returned solution $test'$, i.e., $\mu(f(\mathbf{b})) = b_{e_i}$. Algorithm 1 is 2-approximate for the BTSP. We can prove the following:

Theorem 3. *Let the procedure test($\mathcal{G}$, G, t) run in deterministic polynomial time and correctly output a certificate of failure in G_i or a feasible solution in $(G_i)^t$ for a given bottleneck network optimization problem. Then, the $\text{VCG}_\mu^{\text{mon}}$ mechanism for the bottleneck problem is truthful for single-dimensional agents, and provides the following approximations, given in $\{\}$, to these bottleneck problems: k-clustering* $\{2\}$*, k-switching network $\{3\}$, $(k, \mathcal{G})$-partition with diameter d $\{2d\}$, k-center* $\{2\}$*, weighted k-center $\{3\}$, weighted k-center with at most ℓ centers $\{3\}$, m-weighted k-center $\{9m-6\}$, wandering salesperson and BTSP* $\{2\}$*, repeated city TSP* $\{2\}$*, k-supplier* $\{3\}$*, k-path vehicle routing* $\{2\}$*, single depot k-vehicle routing* $\{2\}$.

Mechanism $\text{VCG}_\mu^{\text{mon}}$ has best approximation guarantees possible in polynomial time for many problems in Theorem 3, indicated by "*", see (Hochbaum and Shmoys 1986). We can also show that Algorithm 1 is not truthful for 2-

dimensional agents, i.e., owning 2 edges.

Social Cost Optimization Problems

Let us define a binary covering minimization problem, see, e.g., (Dughmi and Roughgarden 2014). Let U be a finite set, *universe*, $|U| = m$, and $\mathbf{b} = (b_e)_{e\in U} \in \mathbb{R}^m_{\geq 0}$ be a vector of costs. The agents $e \in U$ here are again single-dimensional. An instance $I \in \Pi$ of a *binary covering minimization problem* Π is defined by a family of feasible solutions $\mathcal{F}(I) \subseteq 2^U$, such that, if $S \in \mathcal{F}(I)$ is a feasible solution then for any superset $S' \in U$ of S, $S \subseteq S'$, S' is also feasible, $S' \in \mathcal{F}(I)$. That is, given any feasible solution to problem Π if we add any other element of U to this solution, we again obtain a feasible solution. We usually assume that vector $\mathbf{b}$ is part of the instance of Π, but formally, the set $\mathcal{F}(I)$ contains only all combinatorial feasible solutions on instance $I \in \Pi$. Thus, formally, the full instance of problem Π is $(I, \mathbf{b})$.

We define now an objective function of Π abstractly, as a function $\mu : \mathcal{F}(I) \longrightarrow \mathbb{R}_{\geq 0}$ depending on the costs, i.e., given a feasible solution $S \in \mathcal{F}(I)$ and vector $(b_e)_{e\in U}$, $\mu(S, \mathbf{b}) = \mu((b_e)_{e\in S}) = \mu(b_{e_1}, \ldots, b_{e_l})$, where $S = \{e_1, \ldots, e_l\}$. The value of μ depends only on the costs of elements from set S, but sometimes we will write $\mu((b_e)_{e\in S})$ as $\mu(\mathbf{b})$, i.e., specifying all elements of $\mathbf{b}$.

In this section we will be interested in functions μ which are *strictly all-monotone*, that is, for any vector $\mathbf{b} \in \mathbb{R}^m_{\geq 0}$ and any single element $e \in S$ with $b'_e > b_e$, we have that $\mu((b_e, \mathbf{b}_{-e})) < \mu((b'_e, \mathbf{b}_{-e}))$. In words, if we strictly increase any of the arguments of function μ, its value increases by a strictly positive amount, which might be tiny. Note, that the social cost function $\mu((b_e)_{e\in S}) = \sum_{e\in S} b_e$ is strictly all-monotone. Also, the L_p norm with any fixed $p \geq 1$ is strictly all-monotone but not the L_p norm with $p = +\infty$.

Impossibility Result. Let Π be an NP-hard binary covering minimization problem with a strictly all-monotone objective μ. Let f be a deterministic polynomial time α-approximation algorithm for Π. Our first goal in this section is to prove that no such algorithm f can fulfill condition (2) if $\alpha > 1$ and $\mathcal{F}(I)$ is finite for any $I \in \Pi$, which is obviously true for our class of problems. This result will imply that none of these algorithms can be used by a truthful $\text{VCG}^{\text{mon}}_{\mu}$ mechanism.

To prove that f violates condition (2), we will show that it is *discontinuous*, that is, there always exists an instance of Π, vector $\mathbf{b}$ and value t_i such that $t_i = t_i(f(\mathbf{b})) < b_i(f(\mathbf{b})) = b_i$ and $\mu(f(t_i, \mathbf{b}_{-i}), (t_i, \mathbf{b}_{-i})) > \mu(f(\mathbf{b}), \mathbf{b})$.

Theorem 4. *Let Π be a binary covering minimization problem with a strictly all-monotone objective μ, and f be a deterministic α-approximation algorithm for Π, with $\alpha > 1+\varepsilon$ for some $\varepsilon > 0$. Then there exists an instance of Π on which algorithm f is discontinuous, i.e., does not fulfill condition (2). This means that no truthful VCG^{mon}_{μ} mechanism for Π can use f as algorithm, even for single-dimensional agents.*

Proof. (*Sketch*) Let $\alpha > 1$ be the worst-case approximation ratio of f. Then there exists an instance $I \in \Pi$, vector $b \in \mathbb{R}^m_{\geq 0}$, and two feasible solutions $S_0, S_1 \in \mathcal{F}(I)$ s.t.

$$S_0 = opt(I, b),\ S_1 = f(I, b),$$
$$c_0 = \mu(S_0, b) < \mu(S_1, b) = \alpha \cdot c_0 = c_1,$$

with $\mu(S, b) = \mu((b_e)_{e\in S})$ for any $S \in \mathcal{F}(I)$. S_0 is the optimal solution on instance (I, b) minimizing μ with cost vector b; S_1 shows tightness of the approximation ratio α.

Given the solutions $(S_0, b), (S_1, b)$, we will construct a sequence of pairwise distinct feasible solutions $S_0, S_1, \ldots, S_k \in \mathcal{F}(I)$ output by f and corresponding cost vectors $b = b^1, b^2, \ldots, b^k$, where b^k is obtained from b^{k-1} by increasing a single coordinate (the first vector b^1 is the same for S_0 and S_1), and $\alpha \cdot c_0 = \mu(S_1, b^1) = \mu(S_2, b^2) = \cdots = \mu(S_k, b^k)$. We will prove that algorithm f is either discontinuous when "switching" from solution (S_{k-1}, b^{k-1}) to solution (S_k, b^k), or if f "switches" continuously between these two solutions (i.e., (2) holds with equality, $\mu(S_{k-1}, b^{k-1}) = \mu(S_k, b^k)$) then we will show how to extend this sequence with a new solution (S_{k+1}, b^{k+1}), distinct from all previous $S_1, \ldots, S_k$. Because $\mathcal{F}(I)$ is finite, this process must eventually end with $S_\ell = S_0$ for some ℓ, showing that f is discontinuous when "switching" from solution $(S_{\ell-1}, b^{\ell-1})$ to (S_ℓ, b^ℓ), because $\alpha \cdot c_0 = \mu(S_{\ell-1}, b^{\ell-1}) > \mu(S_\ell, b^\ell) = c_0$. □

Corollary 1. *Let Π be a binary covering minimization problem with a strictly all-monotone objective μ, and f be any deterministic α-approximation algorithm for Π with $\alpha \geq 1$. If f fulfils (2) on any instance of Π, then $\forall\varepsilon > 0 : \alpha \leq 1+\varepsilon$. That is, if f is the algorithm used by a truthful VCG^{mon}_{μ} mechanism for Π then f must be optimal or have arbitrarily good approximation ratio, even for single-dimensional agents.*

Remark 1. *We believe that Theorem 4 and Corollary 1 can be extended to certain classes of randomized approximation algorithms. Moreover, we believe that the assumption of Π being a binary covering problem can also be relaxed.*

Approximate Truthfulness. Let A be any α-approximation algorithm for a binary covering minimization problem with a social cost objective function. We will show that any such algorithm A fulfills (2) and (3) up to a multiplicative factor of α. This will imply an α-approximate truthfulness with monitoring. These results together with the above impossibility results provide a characterization of approximate mechanisms which are truthful with monitoring for the class of binary covering minimization problems with social cost objectives.

Definition 3 (α-truthful mechanisms)**.** *A mechanism $\mathcal{M}$ is* α-truthful*, for some $\alpha > 1$, if for any i, any bids $\mathbf{b}_{-i}$, and any $b_i \in D_i$: $u_i(t_i, \mathbf{b}_{-i}) \geq \alpha \cdot u_i(b_i, \mathbf{b}_{-i})$.*

Thus to prove that $\text{VCG}^{\text{mon}}_{\mu}$ is α-truthful it suffices to prove the relaxed variants of conditions (2) and (3).

Theorem 5. *Let Π be a binary covering minimization problem with the social cost objective function μ, and f be any deterministic α-approximation algorithm for Π. Then f fulfils the relaxed conditions (2) and (3), implying an α-truthful VCG^{mon}_{μ} mechanism for Π with single-dimensional agents.*

Acknowledgments

Dimitris Fotakis is supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers' and the procurement of high-cost research equipment grant", project BALSAM, HFRI-FM17-1424. Piotr Krysta is supported by the Network Sciences and Technologies (NeST) initiative of the School of EEECS at the University of Liverpool. Carmine Ventre is supported by EPSRC grant EP/V00784X/1 "UKRI Trustworthy Autonomous Systems Hub".

References

Angel, E.; Bampis, E.; and Pascual, F. 2006. Truthful algorithms for scheduling selfish tasks on parallel machines. *Theoretical Computer Science* 369(1-3): 157–168.

Angel, E.; Bampis, E.; Pascual, F.; and Tchetgnia, A. 2009. On truthfulness and approximation for scheduling selfish tasks. *Journal of Scheduling* 12(5): 437–445.

Ashlagi, I.; Dobzinski, S.; and Lavi, R. 2012. Optimal Lower Bounds for Anonymous Scheduling Mechanisms. *Math. Oper. Res.* 37(2): 244–258.

Azar, Y.; Hoefer, M.; Maor, I.; Reiffenhäuser, R.; and Vöcking, B. 2017. Truthful mechanism design via correlated tree rounding. *Math. Program.* 163(1-2): 445–469.

Caragiannis, I.; Elkind, E.; Szegedy, M.; and Yu, L. 2012. Mechanism design: from partial to probabilistic verification. In *Proc. of the 13th ACM Conference on Electronic Commerce (EC 2012)*, 266–283.

Christodoulou, G.; Koutsoupias, E.; and Kovács, A. 2020. On the Nisan-Ronen conjecture for submodular valuations. In *Proc. of the 52nd ACM SIGACT Symposium on Theory of Computing (STOC 2020)*, 1086–1096. ACM.

Dughmi, S.; and Roughgarden, T. 2014. Black-Box Randomized Reductions in Algorithmic Mechanism Design. *SIAM J. Comput.* 43(1): 312–336.

Ferraioli, D.; and Ventre, C. 2017. Obvious Strategyproofness Needs Monitoring for Good Approximations. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 516–522.

Ferraioli, D.; and Ventre, C. 2018. Probabilistic Verification for Obviously Strategyproof Mechanisms. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, 240–246. ijcai.org.

Fotakis, D.; and Tzamos, C. 2014. On the Power of Deterministic Mechanisms for Facility Location Games. *ACM Trans. Economics and Comput.* 2(4): 15:1–15:37.

Fotakis, D.; and Zampetakis, E. 2015. Truthfulness Flooded Domains and the Power of Verification for Mechanism Design. *ACM Trans. Economics and Comput.* 3(4): 20:1–20:29.

Giannakopoulos, Y.; Koutsoupias, E.; and Kyropoulou, M. 2016. The Anarchy of Scheduling Without Money. In *Algorithmic Game Theory - 9th International Symposium, SAGT 2016*, 302–314.

Grandoni, F.; Krysta, P.; Leonardi, S.; and Ventre, C. 2014. Utilitarian Mechanism Design for Multiobjective Optimization. *SIAM J. Comput.* 43(4): 1263–1290.

Hartline, J.; and Roughgarden, T. 2008. Optimal Mechanism Design and Money Burning. In *Proc. of the 40th ACM Symposium on Theory of Computing*, 75–84.

Hochbaum, D. S.; and Shmoys, D. B. 1986. A unified approach to approximation algorithms for bottleneck problems. *J. ACM* 33(3): 533–550.

Koutsoupias, E. 2014. Scheduling Without Payments. *Theory Comput. Syst.* 54(3): 375–387.

Kovács, A.; Meyer, U.; and Ventre, C. 2015. Mechanisms with Monitoring for Truthful RAM Allocation. In *Web and Internet Economics - 11th International Conference, WINE 2015*, 398–412.

Kumar, V. S. A.; Marathe, M. V.; Parthasarathy, S.; and Srinivasan, A. 2009. A unified approach to scheduling on unrelated parallel machines. *J. ACM* 56(5): 28:1–28:31.

Kyropoulou, M.; and Ventre, C. 2019. Obviously Strategyproof Mechanisms without Money for Scheduling. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Lavi, R.; and Swamy, C. 2009. Truthful mechanism design for multidimensional scheduling via cycle monotonicity. *Games and Economic Behavior* 67(1): 99–124.

Leucci, S.; Mamageishvili, A.; and Penna, P. 2018. No truthful mechanism can be better than n approximate for two natural problems. *Games and Economic Behavior* 111: 64–74.

Leyton-Brown, K. 2003. *Resource Allocation in Competitive Multiagent Systems (PhD thesis).* Stanford University.

Liao, K. 2014. *Approximation Algorithms for Resource Allocation Optimization (PhD thesis).* The University of Adelaide.

Lombardi, M.; and Milano, M. 2012. Optimal methods for resource allocation and scheduling: a cross-disciplinary survey. *Constraints* 17: 51–85.

Lu, P.; and Yu, C. 2008a. An Improved Randomized Truthful Mechanism for Scheduling Unrelated Machines. In *STACS 2008, 25th Annual Symposium on Theoretical Aspects of Computer Science*, 527–538.

Lu, P.; and Yu, C. 2008b. Randomized Truthful Mechanisms for Scheduling Unrelated Machines. In *Internet and Network Economics, 4th International Workshop, WINE 2008*, 402–413.

Nisan, N.; and Ronen, A. 2001. Algorithmic Mechanism Design. *Games and Economic Behavior* 35(1-2): 166–196.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. 2007. *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press. ISBN 0521872820.

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

Saks, M. E.; and Yu, L. 2005. Weak monotonicity suffices for truthfulness on convex domains. In *Proceedings 6th ACM Conference on Electronic Commerce (EC-2005)*, 286–293. ACM.

Serafino, P.; Ventre, C.; and Vidali, A. 2020. Truthfulness on a Budget: Trading Money for Approximation through Monitoring. *Auton. Agents Multi Agent Syst.* 34(1): 5.

Shmoys, D. B.; and Tardos, É. 1993. An approximation algorithm for the generalized assignment problem. *Math. Program.* 62: 461–474.

Shoham, Y.; and Leyton-Brown, K. 2009. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

Vazirani, V. 2001. *Approximation Algorithms*. Springer.