# Private capacities in mechanism design*

Vincenzo Auletta Paolo Penna Giuseppe Persiano

June 13, 2009

**Abstract**

Algorithmic mechanism design considers distributed settings where the participants, termed agents, cannot be assumed to follow the protocol but rather their own interests. The protocol can be regarded as an algorithm augmented with a suitable payment rule and the desired condition is termed *truthfulness*, meaning that it is never convenient for an agent to report false information.

Motivated by the applications, we extend the usual one-parameter and multi-parameter settings by considering agents with *private capacities*: each agent can misreport her cost for "executing" a single unit of work *and* the maximum amount of work that each agent can actually execute (i.e., the capacity of the agent). We show that truthfulness in this setting is equivalent to a simple condition on the underlying algorithm. By applying this result to various problems considered in the literature (e.g., makespan minimization on related machines) we show that only some of the existing approaches to the case "without capacities" can be adapted to the case with private capacities. This poses new interesting algorithmic challenges.

# 1 Introduction

Algorithmic mechanism design considers distributed settings where the participants, termed agents, cannot be assumed to follow the protocol but rather their own interests. The designer must ensure in advance that it is in the agents' interest to behave correctly. The protocol can be regarded as an algorithm augmented with a suitable payment rule and the desired condition is termed *truthfulness*, meaning that it is never convenient for an agent to report false information. We begin with an illustrative example:

**Example 1 (scheduling related machines [AT01])** We have two jobs of size, say, 1 and 2 to be scheduled on two machines. Each allocation specifies the amount of work that is allocated to each machine (the sum of the jobs sizes). Each machine has a *type* $t_i$ which

*Dipartimento di Informatica ed Applicazioni, Università di Salerno, Italy. Email:{auletta, penna, giuper}@dia.unisa.it. Research funded by the European Union through IST FET Integrated Project AEOLUS (IST-015964).

is the time (cost) for processing one unit of work; that is, the type is the inverse of the machine's speed. An allocation, $x$, assigns an amount of work $w_i(x)$ to machine $i$ and thus its completion time (cost) is equal to

$$w_i(x) \cdot t_i. \tag{1}$$

The goal is to compute an allocation that minimizes the "overall" cost

$$\max\{w_1(x) \cdot t_1, w_2(x) \cdot t_2\} \tag{2}$$

that is the so called makespan. The type of each machine is only known to its owner (agent) who incurs a cost equal to the completion time of her machine (the quantity in Equation 1). Each agent may find it convenient to misreport her type so to induce the underlying algorithm to assign less work to her machine.

This is a typical *one-parameter* mechanism design problem, meaning that each agent has a *private type* representing the cost for executing one unit of work. The goal is to compute a solution minimizing some "global" cost function which depends on the types of all agents (in the above example, the quantity in Equation 2). The underlying algorithm is then augmented with a suitable payment function so that each agent finds it convenient to report her type truthfully. This important requirement is commonly termed *truthfulness* of the resulting mechanism ("algorithm + payments"). Truthful mechanisms guarantee that the underlying algorithm receives the "correct" input because no agent has a reason to misreport her type.

In this work we introduce and study the natural extension of the one-parameter setting in which agents have *private capacities* and thus can "refuse" allocations that assign them amounts of work above a certain value (see Section 1.1 for the formal model). This setting is naturally motivated by various applications in which agents are not capable or willing to execute arbitrary amounts of work:

- A router can forward packets at a certain rate (per-packet cost), but an amount of traffic exceeding the capacity of the router will "block" the router itself.
- In wireless networks, each node acts as a router and the energy consumption determines the per-packet-cost of the node. The battery capacity of each node determines the maximum amount of packets (work) that can be forwarded.
- Agents can produce identical goods at some cost (per unit) and their capacities represent the maximum amount of goods that each of them can produce.

Truthful mechanisms for the case without capacities need not remain truthful in this new setting. We show that a simple "monotonicity" condition characterizes truthfulness for the case of one-parameter agents with private capacities (Theorem 12). This translates into an algorithmic condition on the underlying algorithm (Corollary 14). We apply this result to the various problems previously considered in the literature to see which of the existing techniques/results extend to our setting. Roughly, all mechanisms for one-parameter settings (without capacities) that are based on "lexicographically optimal" algorithms remain truthful (for the case with capacities) as they satisfy the above monotonicity condition (Section 3.1). This is not true for other mechanisms because the underlying algorithm is no longer monotone when capacities are introduced (Section 3.3).

We then move to *multidimensional* domains which provide a more powerful and general framework. By considering differend "kind" of work and a capacity for each of them one can easily model rather complex problems like, for instance, scheduling with restricted assignment on unrelated machines (i.e., each machine can execute only certain jobs and the execution times change arbitrarily from machine to machine). Here we observe that there is no "simple" monotonicity condition that characterizes truthfulness, even when the problem without capacities has a domain which does have such simple characterization (Section 4).

**Connections with existing work.** Algorithmic mechanism design questions have been raised in the seminal work by Nisan and Ronen [NR01]. Mechanism design is a central topic in game theory, with the celebrated Vickrey-Clarke-Groves [Vic61, Cla71, Gro73] mechanisms been probably the most general positive result. These mechanisms work for arbitrary domains, but require the problems' objective to be the so called (weighted) social welfare: essentially, to minimize the (weighted) sum of all agents' costs. Roberts' theorem [Rob79] says that these are the only possible truthful mechanisms for domains that allow for arbitrary valuations. Therefore, most of the research has been focused on specific (restricted) domains and to other global cost functions like, for instance, the makespan in scheduling problems [NR01, AT01, AAS07, MS07, LS08] or other min-max functions.

Rochet's [Roc87] is able to characterize truthfulness in terms of the so-called "cycle monotonicity" property. In this paper, we refer to the interpretation of cycle monotonicity given by Gui *et al* [GMV04] in terms of graph cycles which gives us a simple way for computing the payments. However, cycle monotonicity is difficult to interpret and to use. To our knowledge, the work by Lavi and Swamy [LS08] is the first (and only) one to obtain truthful mechanisms for certain two-values scheduling domains directly from Rochet's cycle monotonicity [Roc87].

Bikhchandani *et al* [BCL+06] propose the simpler two-cycle monotonicity property (also known as weak-monotonicity) and showed that it characterizes truthfulness for rather general domains. We refer to domains for which two-cycle monotonicity characterizes truthfulness as monotonicity domains. Monotonicity domains turn out to be extremely important because there the construction of the mechanism (essentially) reduces to ensuring that the algorithm obeys relatively simple (two-cycle monotonicity) conditions. Saks and Yu [SY05] showed that every convex domain is a monotonicity domain. Our main result is that also one-parameter domains with private capacities are monotonicity domains. The resulting characterization generalizes prior results by Myerson [Mye81] and by Archer and Tardos [AT01] when the set of possible solutions is finite. We remark that domains obtained by adding private capacities are not convex and our result for the two-parameter case implies that the characterization by Saks and Yu [SY05] cannot be used here. Dobzinski *et al* [DLN08] studied auctions with budget-constrained bidders. Also these domains are different from ours because they put a bound on the payment capability of the agents, while in our problems bounds are put on the assignment of the algorithm. Our positive results on min-max objective functions use ideas by Archer and Tardos [AT01], Andelman *et al* [AAS07], Mu'alem and Shapira [MS07], and they extend some results therein to the case with private capacities.

## 1.1 Agents with private capacities

We are given a finite set of feasible solutions. In the *one-parameter* setting, every solution $x$ assigns an amount of *work* $w_i(x)$ to agent $i$. Agent $i$ has a monetary *cost* equal to $\mathrm{cost}_i(t_i, x) = w_i(x) \cdot t_i$ where $t_i \in \Re^+$ is a private number called the *type* of the agent. This is the cost per unit of work and the value is known only to agent $i$. We extend the one-parameter setting by introducing *capacities* for the agents. An agent will incur an *infinite* cost whenever she gets an amount of work exceeding her capacity $c_i \in \Re^+$, that is

$$\mathrm{cost}_i(t_i, c_i, x) = \begin{cases} w_i(x) \cdot t_i & \text{if } w_i(x) \leq c_i \\ \infty & \text{otherwise} \end{cases}$$

Each agent makes a *bid* consisting of a type-capacity pair $b_i = (t'_i, c'_i)$, possibly different from the true ones. An *algorithm* $A$ must pick a solution on input the bids $b = (b_1, \ldots, b_n)$ of all agents, and a suitable payment function $P$ assigns a payment $P_i(b)$ to every agent $i$. Thus, the *utility* of this agent is

$$\mathrm{utility}_i(t_i, c_i, b) := P_i(b) - \mathrm{cost}_i(t_i, c_i, A(b)).$$

Truthtelling is a *dominant strategy* with respect to both types and capacities if the utility of each agent is maximized when she reports truthfully her type and capacity, no matter how we fix the types and the capacities reported by the other agents. Formally, for every $i$, for every $t_i$ and $c_i$, and for every $b$ as above

$$\mathrm{utility}_i(t_i, c_i, (t_i, c_i, b_{-i})) \geq \mathrm{utility}_i(t_i, c_i, b)$$

where $(t_i, c_i, b_{-i}) := (b_1, \ldots, b_{i-1}, (t_i, c_i), b_{i+1}, \ldots, b_n)$ is the $n$-vector obtained by replacing $b_i = (t'_i, c'_i)$ with $(t_i, c_i)$.

**Definition 2** *An algorithm $A$ is truthful for one-parameter agents with private capacities if there exists a payment function $P$ such that truthtelling is a dominant strategy with respect to both types and capacities.*

We consider only algorithms that produce an allocation which respects the capacities (no agent gets more work than her reported capacity). A simple (standard) argument reduces truthfulness of $A$ to the truthfulness of the work functions of the single agents (see Section 1.1.2).

**Multidimensional settings.** In the *multidimensional* or *k-parameter* setting each type, capacity, and work is a vector of length $k$. Agent $i$ has a type $t_i = (t_i^1, \ldots, t_i^k)$, a capacity $c_i = (c_i^1, \ldots, c_i^k)$, and she is assigned some amount of work $w_i = (w_i^1, \ldots, w_i^k)$. The resulting cost is

$$w_i \cdot t_i = \sum_j w_i^j \cdot t_i^j$$

provided the $j$-th amount of work $w_i^j$ does not exceed the corresponding capacity $c_i^j$ for all $j$ ($w_i \leq c_i$ component-wise). The cost is instead $\infty$ if some capacity is violated ($w_i \not\leq c_i$).

**Example 3 (agents with several related machines)** *Each agent $i$ owns* two *related machines whose processing times are $t_i^1$ and $t_i^2$. Every job allocation $x$ assigns an amount of work $w_i^1(x)$ and $w_i^2(x)$ to these machines, respectively. The cost for the agent is the sum of the costs of her machines, that is, $w_i^1(x) \cdot t_i^1 + w_i^2(x) \cdot t_i^2$.*

**Example 4 (unrelated machines [NR01])** *Each machine corresponds to an agent. The corresponding type is a vector $t_i = (t_i^1, \ldots, t_i^k)$, where $t_i^j$ is the processing time of job $j$ on machine $i$ and $k$ is the number of jobs that need to be scheduled. Each job allocation $x$ specifies a binary vector $w_i(x)$ with $w_i^j(x) = 1$ iff job $j$ is allocated to machine $i$. The cost for agent $i$ is the completion time of her machine, that is $w_i(x) \cdot t_i = \sum_j w_i^j(x) \cdot t_i^j$. The variant in which machines can execute only certain jobs (restricted assignment) can be modelled with binary capacity vectors $c_i$: machine $i$ can execute jobs $j$ iff $c_i^j = 1$.*

### 1.1.1 A simple reduction to the single-agent case

In this section we present a simple (standard) reduction that allows us to study truthfulness for the case of a single agent. The truthfulness of an algorithm $A$ can be reduced to a condition on what we call below its single agent work functions: That is, the amount of work that $A$ assigns to a fixed agent $i$, depending on her reported type and capacity, and having fixed the types and the capacities of all other agents. Each agent receives an amount of work and a payment according to some *work function* $f$ and a suitable *payment function* $p$. Recall that the algorithm (and thus $f$) will always assign an amount of work that does not exceed the reported capacity. Hence, infinite costs occur only when the agent misreports her capacity. In the following definition, we consider $t'$ and $c'$ as the type and capacity reported by the agent, and $t$ and $c$ are the true ones.

**Definition 5** *A work function $f$ is truthful for a one-parameter agent with private capacity if there exist a payment function $p$ such that, for every types $t$ and $t'$ and capacities $c$ and $c'$,*

$$p(t,c) - f(t,c) \cdot t \geq p(t',c') - \begin{cases} f(t',c') \cdot t & \text{if } f(t',c') \leq c \\ \infty & \text{otherwise} \end{cases}$$

For every $i$ and for every fixed sub-vector $b_{-i} = (b_i, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$, agent $i$ receives an amount of work

$$w_i^A(t_i', c_i', b_{-i}) := w_i(A(b_1, \ldots, b_{i-1}, (t_i', c_i'), b_{i+1}, \ldots, b_n)$$

where $t'_i$ and $c'_i$ are the type and the capacity reported by $i$. Dominant strategies are equivalent to the fact that $i$'s utility is maximized when she is truthtelling, no matter how we have fixed $i$ and $b_{-i}$. Since the utility of $i$ is given by the work function $f(\cdot) = w_i^A(\cdot, b_{-i})$ and by the payment function $p(\cdot) = P_i(\cdot, b_{-i})$, we have

**Fact 6** *An algorithm $A$ is truthful if and only if every single agent work function $f(\cdot) = w_i^A(\cdot, b_{-i})$ is truthful.*

### 1.1.2 Cycle monotonicity and truthfulness

The material in this section is based on the cycle monotonicity approach by Rochet [Roc87] and its recent interpretation by Gui *et al* [GMV04] in terms of graph cycles. Because of the reduction in the previous subsection, we consider the case of a single agent and its work function $f$. We shall see that $f$ is truthful if and only if a suitable weighted graph associated to $f$ contains no negative cycles [Roc87, GMV04]. The graph in question is defined as follows. Since the set of feasible solutions is finite, the amount of work that can be allocated to this agent must belong to some finite set $W = \{\ldots, w, \ldots, w', \ldots\}$. We associate to $f$ the following complete directed graph over $|W|$ nodes, one for each possible workload. The length of an edge $w \to w'$ is

$$\delta_{ww'} := \inf\{t \cdot (w' - w)|\ \ t \in \Re^+ \text{ and there exists } c \geq w' \text{ such that } f(t, c) = w\}$$

where $\inf \emptyset = \infty$. The length of a cycle in this graph is the sum of the lengths of its edges.

**Remark 7** *Intuitively speaking, we think of $w$ as the work when reporting the "true" type and capacity ($t$ and $c$) and $w'$ being the work when reporting some "false" type and capacity ($t'$ and $c'$). Since every "lie" leading to a work exceeding the true capacity cannot be beneficial, we need to consider only the case $w' \leq c$. Then the condition for being truthful can be rewritten as $p_w + \delta_{ww'} \geq p_{w'}$, where $p_w$ and $p_{w'}$ are the payments received in the two cases, respectively. When there is no "lie" that is potentially beneficial for the agent, we set $\delta_{ww'} = \infty$ which is mathematically equivalent to the fact that we do not add any constraint between these two payments.*

The length of a cycle in this graph is the sum of the lengths of its edges.

**Definition 8 (monotone [Roc87, GMV04])** *A function $f$ is monotone if its associated graph contains no cycle of negative length.*

Rochet [Roc87] showed that the above condition characterizes truthfulness. In particular, the weaker condition that every two-cycle has nonnegative length is always necessary. We restate the latter (necessary) condition for our setting and Rochet's theorem below.

**Definition 9 (two-cycle monotone)** *A function $f$ is two-cycle monotone if for every $(t, c)$ and $(t', c')$ it holds that*

$$(t - t') \cdot (w' - w) \geq 0 \quad \text{or} \quad c \geq w' \quad \text{or} \quad c' \geq w$$

*where $w = f(t, c)$ and $w' = f(t', c')$.*

**Theorem 10 ([Roc87])** *Every truthful function must be two-cycle monotone. Every monotone function is truthful.*

While the above result has been originally stated for finite valuations/costs, it can be easily extended to our setting (where "unfeasible" solutions are modelled by means of infinite costs) using the arguments in [GMV04] (see Appendix A.1 for the proof).

**Remark 11** *The above result applies also to the* multidimensional *case.*

# 2 Characterizations for one-parameter agents

We show that two-cycle monotonicity *characterizes* truthfulness for one-parameter agents with private capacities. In particular, this necessary condition is also sufficient:

**Theorem 12** *A function is truthful for one-parameter agents with private capacities if and only if it is two-cycle monotone.*

Proof. Since two-cycle monotonicity is a necessary condition (see Theorem 10), we only need to show that it is also sufficient. We prove that every two-cycle monotone function, for one-parameter agents with private capacities, is monotone (and thus truthful because of Theorem 10).

We consider only cycles whose edges have finite length (because otherwise the total length is obviously non-negative). We show that for any cycle with at least three edges, there exists another cycle with fewer edges and whose length is not larger. This fact, combined with the two-cycle monotonicity, implies that there is no cycle of negative length.

Given an arbitrary cycle of three or more edges, we consider the node with maximal work $\hat{w}$ in the cycle. We thus have three consecutive edges in the path, say $w \to \hat{w} \to w'$ with

$$\hat{w} > w \quad \text{and} \quad \hat{w} > w'.$$

If nodes $w$ and $w'$ coincide, then the path $w \to \hat{w} \to w'$ is actually a two-cycle. The two-cycle monotonicity says that $\delta_{w\hat{w}} + \delta_{\hat{w}w} \geq 0$. If we remove these two edges we obtain a cycle with fewer edges and whose length is not larger compared to the original cycle.

Otherwise, we show that a shorter cycle can be obtained by replacing the path $w \to \hat{w} \to w'$ with edge $w \to w'$. Towards this end, we show that

$$\delta_{ww'} \leq \delta_{w\hat{w}} + \delta_{\hat{w}w'}. \tag{3}$$

For every $\epsilon > 0$ and for every $w^{(1)}$ and $w^{(2)}$ such that $\delta_{w^{(1)}w^{(2)}} < \infty$, there exist $t^{(1)}$ and $c^{(1)} \geq w^{(2)}$ such that $w^{(1)} = f(t^{(1)}, c^{(1)})$ and

$$t^{(1)} \cdot (w^{(2)} - w^{(1)}) = \delta_{w^{(1)}w^{(2)}} + \epsilon^*$$

for some $\epsilon^*$ satisfying $0 \leq \epsilon^* \leq \epsilon$. In particular, since $\delta_{w\hat{w}}$ and $\delta_{\hat{w}w'}$ are both different from $\infty$, we can find $t$, $c \geq \hat{w}$ and $\hat{t}$, $\hat{c} \geq w'$ such that

$$t \cdot (\hat{w} - w) + \hat{t} \cdot (w' - \hat{w}) = \delta_{w\hat{w}} + \delta_{\hat{w}w'} + \epsilon^*$$

where $\epsilon^*$ satisfies $0 \leq \epsilon^* \leq \epsilon$. Observe that $\hat{c} \geq \hat{w} > w$ and thus the two-cycle monotonicity

$$(t - \hat{t})(\hat{w} - w) \geq 0$$

implies $\hat{t} \leq t$. This and $\hat{w} > w'$ imply that

$$t \cdot (\hat{w} - w) + \hat{t} \cdot (w' - \hat{w}) \geq t \cdot (w' - w).$$

Since $c \geq \hat{w} > w'$, we have

$$\delta_{ww'} \leq t \cdot (w' - w).$$

By putting things together we obtain

$$\delta_{ww'} \leq \delta_{w\hat{w}} + \delta_{\hat{w}w'} + \epsilon$$

for *every* $\epsilon > 0$. This implies Equation 3. Hence, by replacing the two edges $w \to \hat{w} \to w'$ with edge $w \to w'$ we obtain a cycle with fewer edges and whose length is not larger than the length of the original cycle. □

The two-cycle monotonicity condition can be expressed in a more convenient form:

**Fact 13** *A function $f$ is two-cycle monotone if and only if for every $(t, c)$ and $(t', c')$ with $t' > t$ it holds that $w' \leq w$ or $w' > c$, where $w = f(t, c)$ and $w' = f(t', c')$.*

We thus obtain a simple algorithmic condition:

**Corollary 14** *An algorithm $A$ is truthful for one-parameter agents with private capacities if and only if every work function is two-cycle monotone. That is, for every $i$ and for every $b_{-i}$ the following holds. For any two capacities $c_i$, and $c'_i$, and for any two types $t_i$ and $t'_i$ with $t'_i > t_i$, it holds that*

$$w_i^A((t'_i, c'_i), b_{-i}) \leq w_i^A((t_i, c_i), b_{-i}) \quad \text{or} \quad w_i^A((t'_i, c'_i), b_{-i}) > c_i.$$

For fixed capacities, this condition boils down to the usual monotonicity of one-parameter agents [AT01].

# 3 Applications to min-max problems

In this section we apply the characterization result on one-parameter agents with capacities to several optimization problems. We show that exact solutions are possible for min-max objectives (e.g., makespan) and that some (though not all) known techniques for obtaining approximation mechanisms for scheduling can be adapted to the case with private capacities.

## 3.1 Exact mechanisms are possible

**Theorem 15** *Every min-max problem for one-parameter agents (with private capacities) admits an exact truthful mechanism.*

PROOF. We show that the optimal lexicographically minimal algorithm is monotone. We prove the theorem for the case of two agents since the proof can be generalized to any number of agents in a straightforward manner. Fix and agent $i$, and a type $\bar{t}$ and capacity $\bar{c}$ for the other agent. Also let $w_{other}$ and $w'_{other}$ denote the work assigned to the other agent when agent $i$ gets assigned work $w$ and $w'$, respectively (these two values are defined below).

By contradiction, assume that the function associated to this agent is not monotone. By virtue of Theorem 10 and from Fact 13 this means that $t' > t$, $w' > w$, and $c \geq w'$. The latter inequality says that $w'$ is feasible for capacity $c$ and thus the optimality of the algorithm implies

$$\max\{w \cdot t, w_{other} \cdot \bar{t}\} \leq \max\{w' \cdot t, w'_{other} \cdot \bar{t}\}. \quad (4)$$

Similarly, we have $c' \geq w'$ because $w'$ must be feasible for $c'$. Thus $w' > w$ implies that $w$ is feasible for $c'$ and the optimality of the algorithm yields

$$\max\{w' \cdot t', w'_{other} \cdot \bar{t}\} \leq \max\{w \cdot t', w_{other} \cdot \bar{t}\}. \quad (5)$$

We consider two cases:

1. $(w \cdot t' > w_{other} \cdot \bar{t}.)$ Since $w' > w$, we have $\max\{w' \cdot t', w'_{other} \cdot \bar{t}\} \geq w' \cdot t' > w \cdot t' = \max\{w \cdot t', w_{other} \cdot \bar{t}\}$, thus contradicting Inequality (5).

2. $(w \cdot t' \leq w_{other} \cdot \bar{t}.)$ Since $t < t'$, we have $w' \cdot t \leq w' \cdot t'$ thus implying that we can chain the inequality in (4) with the one in (5). This and $w \cdot t \leq w \cdot t' \leq w_{other} \cdot \bar{t}$ imply that

$$\max\{w \cdot t, w_{other} \cdot \bar{t}\} = \max\{w \cdot t', w_{other} \cdot \bar{t}\}.$$

   Hence, both the inequalities in (4) and (5) hold with '='. This will contradict the fact that the algorithm picks the lexicographically minimal solution. On input $t$ and $c$ assigning work $w'$ is feasible and gives the same cost as assigning work $w$. Since the algorithm picks $w$, instead of $w'$, we have that $w$ precedes lexicographically $w'$. Similarly, on input $t'$ and $c'$, the work $w$ is also feasible and has the same cost as $w'$. This implies that $w'$ precedes lexicographically $w$, which is a contradiction.

We conclude that each function associated to some agent must be monotone. □

## 3.2 Makespan on related machines in polynomial time

Andelman *et al* [AAS07] have obtained a truthful polynomial-time approximation scheme for a constant number of machines. Their idea is that one precomputes, in polynomial-time, a set of allocations and then obtains $(1+\epsilon)$-approximation by picking the best solution out of a precomputed set. We can use the very same idea and pick the solution in a lexicographically minimal fashion as we did to prove Theorem 15 and obtain the following:

**Corollary 16** *There exists a polynomial-time* $(1+\epsilon)$*-approximation truthful mechanism for scheduling selfish machines with private capacities, for any constant number of machines and any* $\epsilon > 0$*.*

PROOF. All we need to show is that we can also compute the payments in polynomial time. Using the characterization by Gui *et al* [GMV04] the payments can be computed as the shortest path in the graph defined in Section 1.1.2 (for each agent we fix the bids of the others and consider the resulting graph). Notice that the graph has size polynomial because we have precomputed a polynomial number of feasible solutions [AAS07]. The length of each edge corresponds to some breakpoint in which the work assigned to the machine (agent) under consideration reduces from $w$ to some $w' < w$. The breakpoint is the value $\alpha$ for which

$$\max(\alpha \cdot w, M(w)) = \max(\alpha \cdot w', M(w'))$$

where $M(z)$ is the minimum makespan, over all solutions assigning work $z$ to the machine under consideration and ignoring the completion time of this machine (i.e., the makespan with respect to the other machines). □

## 3.3 Limitations of the greedy algorithm

We show that the monotone 3-approximation algorithm by Kovacs [Kov05] *cannot* be extended in the "natural" way to the case with private capacities. This algorithm is the greedy LPT algorithm which processes jobs in decreasing order of their sizes; the current job is assigned to the machine resulting in the smallest completion time (ties are broken in a fixed order).

The modified version of the greedy algorithm simply assigns a job under consideration to the "best" machine among those for which adding this job does not exceed the corresponding capacity. It turns out that this modified greedy algorithm is not monotone, for the case with private capacities, even if we restrict to speeds (processing times) that are power of *any* constant $\gamma > 1$. (Kovacs [Kov05] proved the monotonicity for $\gamma = 2$ and obtained a 3-approximation by simply rounding the speeds.)

**Theorem 17** *The modified greedy algorithm is not truthful, even for fixed capacities and when restricting to speeds that are power of any* $\gamma > 1$*.*

PROOF. There are three jobs of size 10, 6, and 5, and two machines both having capacity 11. The processing time of the second machine is $\gamma > 1$. We show that the work function corresponding to the first machine is not two-cycle monotone (the theorem then follows from Corollary 14).

When the first machine has processing time $t_1 = 1$, the algorithm produces the allocation in Figure 1(left) because after the first job is allocated to the fastest machine, the other two jobs must go to the other machine because of the capacity. Now observe that when the first machine has processing time $t_1' = \gamma^2 > t_1$, the algorithm simply "swaps" the previous

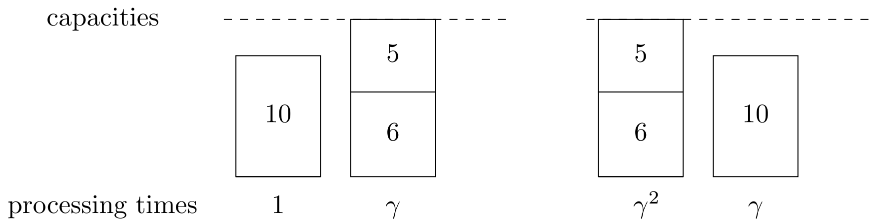


Figure 1: The proof of Theorem 17.

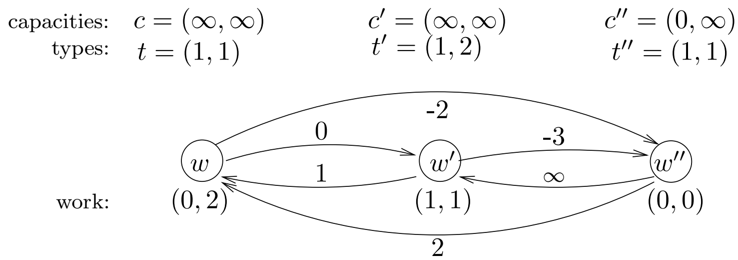


Figure 2: Proof of Theorem 18.

allocation and assigns jobs as shown in Figure 1(right). It is easy to see that this violates the (two-cycle monotonicity) condition of Corollary 14 because

$$10 = w_1^A((1, 11), (\gamma, 11)) < w_1^A((\gamma^2, 11), (\gamma, 11)) = 11 = c_1 = c_1'.$$

This concludes the proof. □

# 4 Multidimensional domains

In this section we show that two-cycle monotonicity does not characterize truthful mechanisms for the *multidimensional* case. We prove the result even for the case of *two-parameter* domains where each agent gets two amounts of different kind of work.

**Theorem 18** *Two-cycle monotonicity does not characterize truthfulness for two-parameter agents with private capacities.*

Proof. We show that there exists a function over a domain with three elements such that the associated graph is like in Figure 2 (for the moment ignore the numbers associated to the nodes). Thus the function is two-cycle monotone but not monotone (there exists a cycle with three edges and negative length).

Each node corresponds to some work which is given in output for the type and the capacity shown above this node: for example, $w = (0, 2) = f(t, c)$ where $t = (1, 1)$ and $c = (\infty, \infty)$.

Observe that edge $w' \leftarrow w''$ has length $\infty$ because work $w'_1 = 1$ exceeds the capacity $c''_1 = 0$. The length of every other edge $w^a \rightarrow w^b$ is given by the formula

$$\delta_{w^a w^b} = t^a \cdot (w^b - w^a) = t^a_1 \cdot (w^b_1 - w^a_1) + t^a_2 \cdot (w^b_2 - w^a_2).$$

It is easy to check that the length of each edge is the one shown in Figure 2. This example can be easily extended to a convex domain (details in Appendix A.3). □

**Acknowledgements.** We wish to thank Riccardo Silvestri and Carmine Ventre for useful comments on an earlier version of this work. We are also grateful to an anonymous referee for a careful reading of the paper and for several suggestions.

# References

[AAS07] Nir Andelman, Yossi Azar, and Motti Sorani. Truthful Approximation Mechanisms for Scheduling Selfish Related Machines. *Theory of Computing Systems*, 40(4):423–436, 2007.

[AT01] Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

[BCL+06] Sushil Bikhchandani, Shurojit Chatterji, Ron Lavi, Ahuva Mu'alem, Noam Nisan, and Arunava Sen. Weak monotonicity characterizes deterministic dominant strategy implementation. *Econometrica*, 74(4):1109–1132, 2006.

[Cla71] Edward H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, pages 17–33, 1971.

[DLN08] Shahar Dobzinski, Ron Lavi, and Noam Nisan. Multi-unit auctions with budget limits. In *FOCS*, pages 260–269. IEEE Computer Society, 2008.

[GMV04] Hongwei Gui, Rudolf Muller, and Rakesh V. Vohra. Dominant strategy mechanisms with multidimensional types. Discussion Papers 1392, Northwestern University, Center for Mathematical Studies in Economics and Management Science, July 2004.

[Gro73] Theodore Groves. Incentive in Teams. *Econometrica*, 41:617–631, 1973.

[Kov05] Annamária Kovács. Fast monotone 3-approximation algorithm for scheduling related machines. In Gerth Stølting Brodal and Stefano Leonardi, editors, *ESA*, volume 3669 of *Lecture Notes in Computer Science*, pages 616–627. Springer, 2005.

[LS08] Ron Lavi and Chaitanya Swamy. Truthful mechanism design for multidimensional scheduling via cycle monotonicity. *Games and Economic Behavior*, 2008. In press, doi:10.1016/j.geb.2008.08.001.

[MS07] Ahuva Mu'alem and Michael Schapira. Setting lower bounds on truthfulness. In *Proc. of annual ACM symposium on discrete algorithms (SODA)*, pages 1143–1152, 2007.

[Mye81] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

[NR01] Noam Nisan and Amir Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35:166–196, 2001. Extended abstract in the Proc. of the 31st Annual ACM Symposium on Theory of Computing (STOC), pages 129–140, 1999.

[Rob79] Kevin Roberts. The characterization of implementable choice rules. *Aggregation and Revelation of Preferences*, pages 321–348, 1979.

[Roc87] Jean-Charles Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16(2):191–200, 1987.

[SY05] Michael Saks and Lan Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proceedings of the 6th ACM conference on Electronic commerce (EC)*, pages 286–293, New York, NY, USA, 2005. ACM.

[Vic61] William Vickrey. Counterspeculation, Auctions and Competitive Sealed Tenders. *Journal of Finance*, pages 8–37, 1961.

# A Postponed proofs

## A.1 Proof of Theorem 10

Proof. Suppose by contradiction that a truthful $f$ is not two-cycle monotone. That is, there exist $w$ and $w'$ such that

$$(t - t')(w' - w) < 0$$

where $w = f(t, c)$ and $w' = f(t', c')$ for some $c \geq w'$ and $c' \geq w$. Consider the case in which the true type and capacity are $t$ and $c$, respectively. Since $c \geq w' = f(t', c')$, the agent report a type $t'$ and a capacity $c'$. Then, truthfulness implies that

$$p_w - w \cdot t \geq p_{w'} - w' \cdot t. \tag{6}$$

Similarly, by considering the case in which the true type and capacity are $t'$ and $c'$, because of $c' \geq w$ and because of truthfulness we obtain

$$p_{w'} - w' \cdot t' \geq p_w - w \cdot t'. \tag{7}$$

By summing up these two inequalities we get

$$(t - t')(w' - w) \geq 0$$

thus contradicting the fact that $f$ is not two-cycle monotone.

Since the function is monotone, the associated graph contains no negative cycles. We can thus define the payments $p = \{p_w\}_{w \in W}$ by means of shortest-path distances: Fix arbitrarily a node $w_0$ in the graph, and let each $p_w$ be the length of the shortest path from node $w$ to node $w_0$. Because of the triangle inequality, for every $w, w' \in W$ we have

$$p_{w'} - p_w \leq \delta_{ww'}.$$

Now observe that, for every $t$, $c$, $t'$, and $c'$ such that $c \geq w' = f(t', c')$

$$\delta_{ww'} \leq t \cdot (w' - w)$$

where $w = f(t, c)$. Hence

$$p_w - w \cdot t \geq p_{w'} - w' \cdot t$$

meaning that $(f, p)$ is a truthful mechanism. □

## A.2 Proof of Fact 13

For any $(t, c)$ and $(t', c')$, let $w = f(t, c)$ and $w' = f(t', c')$. Since the algorithm respects the capacities, if $w' > w$ then $c' \geq w' > w$. Therefore, the following two conditions are equivalent:

$$w' \leq w \text{ or } w' > c \tag{8}$$
$$w' \leq w \text{ or } w' > c \text{ or } w > c' \tag{9}$$

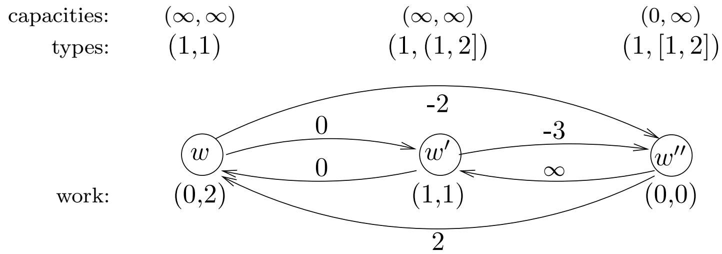


Figure 3: Proof of Theorem 18 for convex domains with private capacities.

Observe that the two-cycle monotonicity condition in Definition 9 can be rewritten as follows: For all $(t, c)$ and $(t', c')$ with $t' > t$ it holds that

$$w' - w \leq 0 \quad \text{or} \quad c \geq w' \quad \text{or} \quad c' \geq w$$

that is (9). Fact 13 is an immediate consequence of the equivalence between (8) and (9).

## A.3 Proof of Theorem 18 for convex domains

We extend the types domain of the graph in Figure 2 to a convex domain so that we obtain (roughly) the same graph. So, the underlying function is two-cycle monotone but not monotone. The types domain is now defined as

$$D := \{1\} \times [1, 2]$$

meaning that we consider all $t = (t_1, t_2)$ with $t_1 = 1$ and $t_2 \in [1, 2]$. The types domain is clearly convex since, for any $t$ and $t'$ in the domain, their convex combination

$$\lambda t + (1 - \lambda)t' = (1, \lambda t_2 + (1 - \lambda)t'_2)$$

is also an element of the domain.

The function is specified by the graph in Figure 3, where the labels attached to the nodes specify a *range* of values for the types and the corresponding work: for example, work $w' = (1, 1)$ is given in output for capacities $c' = (\infty, \infty)$ and for all $t' = (t'_1, t'_2)$ such that $t'_1 = 1$ and $t'_2 \in (1, 2]$.

Now we show that the length of the edges is indeed as shown in Figure 3:

$$\begin{aligned}
\delta_{ww'} &= \inf\{t \cdot (w' - w)|\ t \in D \text{ and there exists } c \geq w' \text{ such that } f(t, c) = w\} \\
&= (1, 1) \cdot (w' - w) = (1, 1) \cdot (1, -1) = 0 \\
\delta_{w'w''} &= \inf\{t' \cdot (w'' - w')|\ t' \in D \text{ and there exists } c' \geq w'' \text{ such that } f(t', c') = w'\} \\
&= \inf\{t' \cdot (-1, -1)|\ t'_1 = 1 \text{ and } t'_2 \in (1, 2]\} = -3 \\
\delta_{w''w} &= \inf\{t'' \cdot (w - w'')|\ t'' \in D \text{ and there exists } c'' \geq w \text{ such that } f(t'', c'') = w''\} \\
&= \inf\{t'' \cdot (0, 2)|\ t''_1 = 1 \text{ and } t''_2 \in [1, 2]\} = 2 \\
\delta_{ww''} &= \inf\{t \cdot (w'' - w)|\ t \in D \text{ and there exists } c \geq w'' \text{ such that } f(t, c) = w\}
\end{aligned}$$

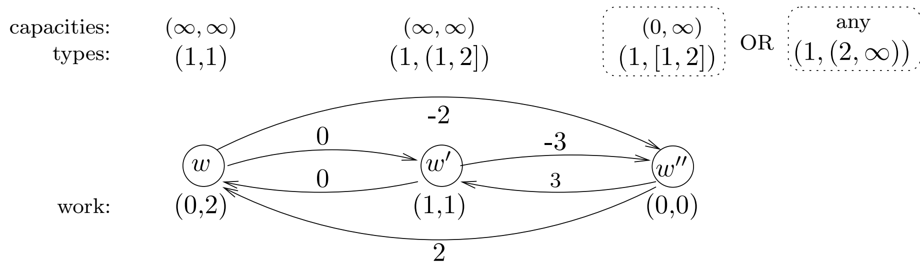


Figure 4: An extension of Theorem 18 to unbounded convex domains with private capacities.

$$
\begin{aligned}
&= (1,1)\cdot(w''-w) = (1,1)\cdot(0,-2) = -2\\
\delta_{w''w'} &= \inf\{t''\cdot(w'-w'')|\ t''\in D \text{ and there exists } c''\geq w' \text{ such that } f(t'',c'')=w''\}\\
&= \inf\emptyset = \infty\\
\delta_{w'w} &= \inf\{t'\cdot(w-w')|\ t'\in D \text{ and there exists } c'\geq w \text{ such that } f(t',c')=w'\}\\
&= \inf\{t'\cdot(-1,1)|\ t'_1=1 \text{ and } \ t'_2\in(1,2]\} = 0
\end{aligned}
$$

where for $\delta_{w''w'}$ we use the fact that, if $f(t'',c'')=w''$, then $c''=(0,\infty)$ and thus $c''\not\geq w'=(1,1)$.

Now we further extend the result by allowing the second parameter to be arbitrarily large. We thus consider the following unbounded types domain

$$D_\infty := \{1\}\cup[1,\infty)$$

and extend the example as shown in Figure 4: The only difference is that work $w''=(0,0)$ is also given in output whenever $t_2\geq 2$ and for all possible capacities. Hence, only the edges whose first endpoint is $w''$ can have a different length, compared to Figure 3. For them, we have

$$
\begin{aligned}
\delta_{w''w} &= \inf\{t''\cdot(w-w'')|\ t''\in D_\infty \text{ and there exists } c''\geq w \text{ such that } f(t'',c'')=w''\}\\
&= \inf\{t''\cdot(0,2)|\ t''_1=1 \text{ and } t''_2\geq 1\} = 2\\
\delta_{w''w'} &= \inf\{t''\cdot(w'-w'')|\ t''\in D_\infty \text{ and there exists } c''\geq w' \text{ such that } f(t'',c'')=w''\}\\
&= \inf\{t''\cdot(1,1)|\ t'\in D_\infty \text{ such that } f(t'',c'')=w'' \text{ for } c''=(\infty,\infty)\}\\
&= \inf\{t''\cdot(1,1)|\ t''_1=1 \text{ for some } t''_2\geq 2\} = 3
\end{aligned}
$$

where for $\delta_{w''w'}$ we use the fact that, if $c''\geq w'$ and $f(t'',c'')=w''$, then it must be $c''=(\infty,\infty)$.

# B What happens when greedy is monotone for fixed capacities

We show that the only obstacle to an extension of the approach by Kovacs [Kov05] is the fact that the greedy LPT approximation algorithm is *not* monotone for fixed capacities.

Essentially, the monotonicity for fixed capacities of the modified greedy implies truthfulness with capacities of the approach by Kovacs [Kov05]. This is important because, for certain job instances, the algorithm is in fact monotone.

**Theorem 19** *If the modified greedy algorithm is monotone for fixed capacities and speeds that are power of two, then it is also monotone for the case with (arbitrary) capacities and speeds that are power of two.*

Proof. By contradiction, suppose that the function associated to some agent in not monotone. That is, there exist $t' > t$, $w' > w$, and $c \geq w'$. We consider

$$\tilde{w} := f(t', c)$$

that is the work assigned to the agent on input type $t'$ and capacity $c$. We show that $\tilde{w} \geq w'$ and conclude, by the monotonicity for fixed capacities, that $w = f(t, c) \geq f(t', c) = w'$.

We show a stronger fact, that is, the algorithm must produce the same allocation on input type $t$ and capacity $c'$ as on input type $t'$ and capacity $c$ (types are different but the capacity is the same). We distinguish to cases:

1. $c' > c$. If a job under consideration is not allocated to this agent because it exceeds capacity $c'$, then the same happens for capacity $c$. The job is then allocated to another machine which is the same in both cases. If the job is allocated to this agent on input capacity $c'$, then the current work after allocating the job is at most $w'$ and thus it is feasible also for capacity $c \geq w'$. So, the algorithm makes the same allocation on input capacity $c$.

2. $c' < c$. If $\tilde{w} \geq w'$ then we are done. Otherwise, for $\tilde{w} < w'$, we show that the allocation on input $c$ is the same as that on input $c'$. If a job is not allocated because of capacity $c$, then the same happens because $c > c'$. If instead the job is allocated to the agent, then the resulting work is at most $\tilde{w}$ and thus at most $c' \geq w' > \tilde{w}$.

□

Kovacs [Kov05] proved that the algorithm is 3/2 approximated for speeds that are power of two, and thus 3 approximations can be obtained by rounding the speeds.

**Corollary 20** *There exists a polynomial-time* 3*-approximation mechanism for scheduling selfish machines with private capacities which is truthful whenever the set of jobs is such that the modified greedy algorithm is monotone for fixed capacities.*

Proof. The very same argument used in the proof of the theorem above works also for the modified version of greedy LPT in which speeds are rounded to the closest power of two. Indeed, the rounding preserves the monotonicity (for fixed capacities). □