

Improvements for Truthful Mechanisms with Verifiable One-Parameter Selfish Agents

A. Ferrante, G. Parlato, F. Sorrentino, and C. Ventre*

Dipartimento di Informatica ed Applicazioni “R.M. Capocelli”, Università di Salerno,
via S. Allende 2, I-84081 Baronissi (SA), Italy
{ferrante, parlato, sorrentino, ventre}@dia.unisa.it

Abstract. In this paper we study optimization problems with verifiable one-parameter selfish agents introduced by Auletta et al. [ICALP 2004]. Our goal is to allocate load among the agents, provided that the secret data of each agent is a single positive rational number: the cost they incur per unit load. In such a setting the payment is given after the load completion, therefore if a positive load is assigned to an agent, we are able to verify if the agent declared to be faster than she actually is. We design truthful mechanisms when the agents’ type sets are upper-bounded by a finite value. We provide a truthful mechanism that is $c \cdot (1 + \epsilon)$ -approximate if the underlying algorithm is c -approximate and weakly-monotone. Moreover, if type sets are also *discrete*, we provide a truthful mechanism preserving the approximation ratio of the used algorithm. Our results improve the existing ones which provide truthful mechanisms dealing only with finite type sets and do not preserve the approximation ratio of the underlying algorithm. Finally we give a full characterization of the $Q||C_{max}$ problem by using only our results. Even if our payment schemes need upper-bounded type sets, every instance of $Q||C_{max}$ can be “mapped” into an instance with upper-bounded type sets preserving the approximation ratio.

1 Introduction

Optimization problems dealing with resource allocation are classical algorithmic problems and they have been studied for decades in several models: centralized vs. distributed algorithms, on-line vs. off-line algorithms and so on. The underlying hypothesis has been that the input is available to the algorithm (either from the beginning in off-line algorithms or during its execution in on-line algorithms). This assumption turns out to be unrealistic in the context of modern networks like the Internet. Here, the various parts of the input are owned by *selfish* (but *rational*) agents as part of their private information (called the *type*) and thus the optimization algorithm will have to ask the agents for their type and then work on the *reported* types. In this context, it is realistic to assume

* Research supported by the European Project FP6-15964, Algorithmic Principles for Building Efficient Overlay Computers (AEOLUS). (Contract number 015964.)

that an agent will lie about her type if this leads to a solution S that she prefers, even in spite of the fact that S is not globally optimal.

The field of mechanism design is the branch of Game Theory and Microeconomics that studies ways of inducing, through payments, the agents to report their true type so that the optimization problem can be solved on the real input. In this paper we study the design of algorithms for solving (or approximately solving) combinatorial optimization problems in presence of selfish agents.

Following the standard notation used in the study of approximation of combinatorial optimization problems (see, e.g., [10]), we consider problems defined as four-tuples $(\mathcal{I}, m, \text{sol}, \text{goal})$, where \mathcal{I} is the set of *instances* of the problem; $\text{sol}(I)$ is the set of *feasible solutions* of instance I ; $m(S, I)$ is the *measure* of the feasible solution S of instance I and goal is either min or max. Thus, the optimization problem consists in finding a feasible solution S^* for instance I such that $m(S^*, I) = \text{opt}(I) := \text{goal}_{S \in \text{sol}(I)} m(S, I)$. A *c-approximation* algorithm A for $\Pi = (\mathcal{I}, m, \text{sol}, \text{goal})$ is such that for all $I \in \mathcal{I}$, $\max\{m(A(I), I)/\text{opt}(I), \text{opt}(I)/m(A(I), I)\} \leq c$.

In an optimization problem Π with *selfish agents*, there are m agents which *privately know* part of the input. Thus every instance $I \in \mathcal{I}$ consists of two parts $I = (T, \sigma)$, where the vector $T = (t_1, t_2, \dots, t_m)$ is the *private part* of the input and σ is the *public part* of the input. In particular, we assume that t_i is known only to agent i , for $i = 1, 2, \dots, m$ and we call t_i the *type* of agent i . The *type set* Θ_i of agent i is the set of the possible types of agent i . In this setting, each agent will report some value $b_i \in \Theta_i$ (which can be different from her true type t_i). An algorithm A for the optimization problem Π with selfish agents receives as input the vector of bids $B = (b_1, b_2, \dots, b_m)$, instead of the true instance T it is supposed to solve. Each selfish agent incurs some monetary *cost*, $\text{cost}_i(S, t_i)$, depending on the feasible solution S and her private data t_i . Since every agent i is selfish, she might declare $b_i \neq t_i$ so to induce A to return a cheaper solution for agent i . Unfortunately, even though A is c -approximating for the instance T , for $B \neq T$ the solution returned by A on input B might have measure, w.r.t. the true instance T , far-off the optimum $\text{opt}(T)$.

In order to obtain a correct solution, algorithm A is equipped with a *payment scheme* $P = (P_1, \dots, P_m)$ in order to induce every agent to report her true type. After a solution $S = A(B, \sigma)$ is computed, each agent i is awarded payment $P_i(S, B, \sigma)$. We assume that each agent i is *rational* in the sense that she picks her type declaration b_i so to maximize her *profit*.

Definition 1. *Let Π be an optimization problem with one-parameter selfish agents and A be an algorithm for Π , and P be a payment scheme. The profit function profit_i of agent i with respect to the pair (A, P) when B is the sequence of bids, σ is the public information, t_i is the true type of agent i , and $S = A(B, \sigma)$, is defined as $\text{profit}_i(S, B, \sigma, t_i) := P_i(S, B, \sigma) - \text{cost}_i(S, t_i)$.*

It is natural to consider mechanisms in which the profit of the i -th agent is maximized when she reports $b_i = t_i$. We have thus the following classical notion of a *truthful mechanism*. In the definition of a truthful mechanism (and

in the rest of the paper) the following notation turns out to be useful. Let $X = (x_1, \dots, x_k)$ be a vector. For any $1 \leq i \leq k$, the writing X_{-i} denotes the vector $X_{-i} := (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k)$ and the writing (y, X_{-i}) denotes the vector $(y, X_{-i}) := (x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_k)$.

Definition 2. *The pair $\mathcal{M} = (A, P)$ is a truthful mechanism for selfish agents if and only if for all σ , for all agents i , and all type declarations B , it holds $\text{profit}_i(A((t_i, B_{-i}), \sigma), (t_i, B_{-i}), \sigma, t_i) \geq \text{profit}_i(A(B, \sigma), B, \sigma, t_i)$.*

In such a way, every agent maximizes her profit when she is truthful. Thus, we assume that in a truthful mechanism every agent always reports her true type, and algorithm A always works on the true instance T . As a consequence, we say that a truthful mechanism $\mathcal{M} = (A, P)$ is *c-approximating* for an optimization problem Π with selfish agents, if A is a c -approximating algorithm for every instance I of Π . Since, in a truthful mechanism, agents are not sure to have a positive profit, they would not participate in such a mechanism unless they were coerced. This motivates the following definition.

Definition 3. *A truthful mechanism satisfies voluntary participation condition if agents who bid truthfully never incur a net loss, i.e. for all public information σ , for all agents i , and for all other agents' bids B_{-i} ,*

$$\text{profit}_i(A((t_i, B_{-i}), \sigma), (t_i, B_{-i}), \sigma, t_i) \geq 0.$$

We now review the concept of optimization problem Π with *one-parameter selfish agents* (as discussed in [2]). Here, each agent i has as private information a single parameter $t_i \in \mathbb{Q}$. Moreover, a feasible solution S of an instance I of Π defines, for each agent i , an amount $w_i(S)$ of assigned work. We call such a solution S *schedule*. Notice that in the definition of one-parameter problem ([2]) the total amount of work to schedule can depend on the private part of the input B . However we restrict ourselves, as in wide part of literature, to the case in which the amount of work assigned to all agents depends only on the public information (and not on the agent bids). We denote such an amount of load just as $\mathcal{W} > 0$. The cost function of agent i has the following special form.

Definition 4. *Let S be a feasible solution of Π . Then, the cost function $\text{cost}_i(S, t_i)$ is defined as $\text{cost}_i(S, t_i) := w_i(S) \cdot t_i$.*

Scheduling problems are typical examples of optimization problem for one-parameter selfish agents. In a *scheduling problem*, the input consists of m machine speeds $s = (s_1, s_2, \dots, s_m)$ and n job weights $W = (w_1, w_2, \dots, w_n)$. A schedule S is an assignment of jobs to machines. Let $w_i(S)$ be the sum of the weights of the jobs assigned to machine i by schedule S . In a scheduling problem the task consists in computing a schedule that minimizes a certain cost function associated with the schedule. For instance, in the $Q||C_{max}$ problem the cost of a schedule S is the *makespan* $MS(S)$ that is the maximum completion time of the machines. Formally, $MS(S) = \max_{1 \leq j \leq m} \left\{ \frac{w_j(S)}{s_j} \right\}$. We consider the setting in which each

machine i is owned by a different agent and the speed s_i of machine i is the private information of agent i . To be in a setting of one-parameter selfish agents, we consider $t_i = 1/s_i$ as the type of agent i . The public information σ is the sequence $W = (w_1, w_2, \dots, w_n)$ of job weights. We recall that $Q||C_{max}$ problem is NP-hard. Throughout the paper we use $Q||C_{max}$ as our main example.

A mechanism \mathcal{M} for verifiable one-parameter selfish agents is a pair $\mathcal{M} = (A, P)$ working as follows.

1. The allocation algorithm A takes as input the sequence of bids $B = (b_1, b_2, \dots, b_m)$ and the public part σ and outputs a schedule $S = A(B, \sigma)$ for the m agents. We recall that $w_i(S)$ denotes the amount of load assigned to agent i by the schedule S computed by algorithm A on input B and σ .
2. Each agent i is *observed* to complete her assigned load in time $\mathcal{T}_i \geq w_i(S) \cdot t_i$. Notice that agent i completes the load $w_i(S)$ assigned to her in time $w_i(S) \cdot t_i$. Agent i can however delay the release of the works and thus obtain a larger observed completion time and the mechanism has no way of detecting it. However, agent i cannot be observed to finish her load before the actual completion time $w_i(S) \cdot t_i$. Since $w_i(S) \cdot t_i$ is the request time for agent i to complete the load $w_i(S)$, we denote with $s_i = \frac{1}{t_i}$ the *speed* of agent i .
3. Finally, after agent i releases the assigned works, she is awarded payment computed by applying function P_i on arguments S, B, σ , and the *observed completion time* \mathcal{T}_i of machine i .

We stress that in this setting, payments are provided *after* the execution of the load and thus agents are (partially) *verifiable* in the following sense. If agent i receives an amount of load greater than 0, the mechanism can find out whether agent i has declared to be faster than she actually is (that is, $b_i < t_i$). Indeed, in this case the claimed completion time $w_i(S) \cdot b_i$ is smaller than the actual completion time $w_i(S) \cdot t_i$ and thus we have that $\mathcal{T}_i \geq w_i(S) \cdot t_i > w_i(S) \cdot b_i$. Since payments are provided *after* the completion of loads, the mechanism can make it inconvenient to claim faster speeds. On the other hand, the mechanism cannot find out if an agent has declared to be slower than she actually is, since the agent can decide to delay some of the jobs.

Henceforward we refer to Π as an optimization problem for verifiable one-parameter selfish agents. Let us now instantiate the definitions of profit and truthful mechanism in this new scenario.

Definition 5. Let A be an algorithm for Π , and P be a payment scheme. The profit function \mathbf{profit}_i of agent i with respect to the pair (A, P) , when B is the sequence of bids, σ is the public information, t_i is the true type of agent i , $S = A(B, \sigma)$, and \mathcal{T}_i is the observed completion time of agent i for the load $w_i(S)$, is defined as $\mathbf{profit}_i(S, B, \sigma, t_i, \mathcal{T}_i) := P_i(S, B, \sigma, \mathcal{T}_i) - \mathbf{cost}_i(S, t_i)$.

Definition 6. Let A be an algorithm for Π , and P be a payment scheme. A pair $\mathcal{M} = (A, P)$ is a truthful mechanism with respect to Π , if for all σ , for all i , for all bid vectors B , and for all observed completion times $\mathcal{T}_i \geq w_i(A(B, \sigma)) \cdot t_i$, it holds that $\mathbf{profit}_i(S, (t_i, B_{-i}), \sigma, t_i, w_i(S) \cdot t_i) \geq \mathbf{profit}_i(A(B, \sigma), B, \sigma, t_i, \mathcal{T}_i)$ where $S = A((t_i, B_{-i}), \sigma)$.

Note that these new definitions are not redundant, since in this case we have to take into account the observed completion time also.

Given a truthful mechanism $\mathcal{M} = (A, P)$ for Π , in [4] the authors give a necessary condition that algorithm A must satisfy.

Definition 7 (weakly-monotone algorithm). *Let Π be an optimization problem for verifiable one-parameter selfish agents and A be an algorithm for Π . Algorithm A is weakly-monotone if and only if, for all σ , for all i , for all declared bid vectors B such that $w_i(A(B, \sigma)) = 0$ and for all $b'_i \in \Theta_i$ with $b'_i > b_i$ it holds that $w_i(A((b'_i, B_{-i}), \sigma)) = 0$.*

In other words a weakly-monotone algorithm A has the following property. Fix some input (B, σ) for which algorithm A assigns no load to agent i . If agent i declares to be slower (that is, she declares $b'_i > b_i$) and the declared bids of the other agents remain the same, then A assigns no load to agent i .

Lemma 1 ([4]). *Let Π be an optimization problem for verifiable one-parameter selfish agents. If $\mathcal{M} = (A, P)$ is a truthful mechanism for Π , then A is a weakly-monotone algorithm.*

2 Previous Works and Our Contribution

The celebrated VCG mechanism [5,6,7,11] is the prominent technique to derive truthful mechanisms for optimization problems. However, this technique applies only to *utilitarian* problems, that are problems where the objective function is equal to the sum of the cost functions of the agent (e.g., shortest path, minimum spanning tree, etc.). In the seminal papers by Nisan and Ronen [8,9] it is pointed out that VCG mechanisms do not completely fit in a context where computational issues play a crucial role since they assume that it is possible to compute an optimal solution of the corresponding optimization problem (maybe a NP-hard problem). Scheduling is a classical optimization problem that is not utilitarian (since we aim at minimizing the maximum over all machines of their completion times) and it is NP-hard. Moreover, scheduling models important features of different allocation and routing problems in communication networks. Thus, it has been the first problem for which non VCG-based techniques have been introduced.

Nisan and Ronen [8,9] give an m -approximation truthful mechanism for the problem of scheduling tasks on m *unrelated* machines, when each machine is owned by a different agent that declares the processing times of the tasks assigned to her machine and the algorithm has to compute the scheduling based on the values declared by the agents. In [2], is considered the simpler variant of the task scheduling on *uniformly related* machines (in short $Q||C_{max}$), where each machine i has a speed s_i and the processing time of a task is given by the ratio between the weight of the task and the speed of the machine. They characterized the class of allocation algorithms A for one-parameter problems that admit payment scheme P for which $\mathcal{M} = (A, P)$ is a truthful mechanism.

Essentially, truthful mechanisms for one-parameter selfish agents must use *monotone* algorithms and, in this case, the payment scheme is uniquely determined (up to an additive factor). Intuitively, monotonicity means that increasing the speed of exactly one machine does not make the algorithm decrease the work assigned to that machine. The result of [2] reduces the problem of designing a truthful mechanism for $Q||C_{max}$ to the algorithmic problem of designing a good algorithm which also satisfies the additional monotonicity requirement. Efficient mechanisms for computing scheduling on related machines with small makespan (a special case of one-parameter agents) have been provided by Archer and Tardos [2] and, subsequently by Auletta et al. [3] and by Andelman, Azar and Sorani [1].

Afterwards, Auletta et al [4] consider optimization problem for verifiable one-parameter problems. In this model, payments are given to the agents only after the agents have completed the load assigned. This means that for each agent that receives a positive load, the mechanism can *verify* if the agent declared to be faster than she actually is. They showed that, in order to have a truthful mechanism for verifiable one-parameter selfish agents, a necessary condition is that the used algorithm must be weakly-monotone.

Our Contribution. In this work, we extend some results given in [4]. The authors were the first to study optimization problems for verifiable one-parameter selfish agents. Intuitively a verifiable agent is an agent that may lie in reporting its types but the mechanism can verify whether agent i underbids (i.e. declares a $b_i < t_i$), provided that the load assigned to this agent is positive. For instance, for scheduling problems the mechanism can *verify*, through the observed completion time of agent i , if she declares to be faster than she actually is, provided that at least one job has been assigned to her. In [4] was showed that if $\mathcal{M} = (A, P)$ is a truthful mechanism for an optimization problem for verifiable one-parameter selfish agents then A must be *weakly-monotone*. They also provide a payment scheme P which allows to have a truthful mechanism, when the cardinality of type sets is finite.

In Section 3, we give very simple and efficient payment schemes, leading to polynomial-time truthful mechanisms, for a wide class of optimization problems with verifiable one-parameter selfish agents. In particular, we provide a payment $P^{(1)}$ that works for *discrete* and *upper-bounded* type sets (see Section 3). In this setting, we need that agents bid from sets in which there is always a gap between the inverse of two types. Considering scheduling problems (where types are the inverse of machines' speed), our assumption is satisfied when it is not possible to have machines executing j instructions per second, for every possible $j \in \mathbb{Q}$. Indeed, in the market there are only machines of certain (sufficiently far apart) speeds. Moreover, we need that the agents cannot declare more than a finite value. In scheduling problems, this means that an infinitely slow machine does not exist. Thus our hypothesis applies to many real life applications.

From a theoretical point of view, our results improve the ones given in [4], as follows: (i) the class of the discrete and upper-bounded type sets properly includes the class of finite type sets; (ii) our mechanism preserves the approxi-

Table 1. Comparing Results (c is the approximation of a given weakly monotone algorithm)

Problem Version	Payments Time Complexity	Apx Ratio
Θ_i finite and discrete [4]	$\text{poly}(\Theta_i , m, n)$	c
Smooth problems [4]	$\text{poly}(\log_{1+\epsilon} \Theta_i , m, n)$	$c \cdot (1 + \epsilon)$
Θ_i upper bounded and discrete	$\text{poly}(m, n)$	c
Smooth problems with Θ_i upper bounded (continuous)	$\text{poly}(m, n)$	$c \cdot (1 + \epsilon)$

mation ratio c of the algorithm it uses, while the mechanism given in the paper [4] needs that the problem is *smooth* (see Def. 11) in order to obtain a $c \cdot (1 + \epsilon)$ -approximation. (This assumption is required to round the input bids in order to have payments computable in polynomial time.)

In Section 4, we give a payment scheme $P^{(2)}$, leading to polynomial-time truthful mechanisms (Theorem 3), for agents having rational type sets upper-bounded (but not discrete). In order to obtain truthful mechanism we round the agents' bid. Using this rounding technique, if the algorithm used by the mechanism is c -approximate, then nothing can be said about the approximation of the same algorithm when it runs on rounded bids. However, if the problem is smooth then the mechanism is $c \cdot (1 + \epsilon)$ -approximate (see Theorem 4). To best of our knowledge this is the *first* result showing that weakly-monotonicity of algorithms is a sufficient condition for the existence of truthful mechanisms for optimization problems with verifiable one-parameter selfish agents with continuous type sets. It left open the case when type sets are not upper-bounded. In Table 1 we summarize our results comparing them to the previous ones.

Finally, in Section 5, as application of our results, we fully characterize $Q||C_{max}$ problem with verifiable one-parameter selfish agents reducing any unbounded instance to a bounded one, so obtaining a polynomial-time $c \cdot (1 + \epsilon)$ -approximate truthful mechanism, given a c -approximate weakly-monotone polynomial-time algorithm.

3 A Payment Scheme for Discrete Types

In this section, we consider only type sets Θ_i having the following property.

Definition 8. A set Θ_i is said *discrete and upper-bounded* if: (i) there exists a value $\Delta_i \in \mathbb{R}^+$ such that, for all $b, \bar{b} \in \Theta_i$, $b \neq \bar{b}$, $|b^{-1} - \bar{b}^{-1}| \geq \Delta_i$ (discrete), and (ii) there exists a finite value $\sup_i \in \mathbb{R}^+$ such that $\sup_i \geq b$, $\forall b \in \Theta_i$ (upper-bounded).

Next we define a payment scheme which allows us to construct truthful mechanism for Π , when agents have type sets discrete and upper-bounded.

Definition 9. Let S be a schedule, B be a bid vector, σ be the public part of the input, \mathcal{T}_i be the observed completion time and $c_i^{(1)} \in \mathbb{R}^+$ be a constant (to be given). For each $i = 1, \dots, m$, we define

$$P_i^{(1)}(S, B, \sigma, \mathcal{T}_i) := \begin{cases} \frac{\mathcal{W}}{b_i} \cdot c_i^{(1)} & \text{if } w_i(S) \neq 0 \text{ and } \mathcal{T}_i = w_i(S) \cdot b_i; \\ 0 & \text{otherwise.} \end{cases}$$

The idea behind the payment $P_i^{(1)}$ is to give the agent i a disincentive to declare to be slower than she actually is. On the other hand, agent i is also discouraged to declare to be faster, if we use verification and weakly-monotone algorithms, as shown in the next theorem.

Theorem 1. *Let Π be an optimization problem for verifiable one-parameter selfish agents and A be a polynomial-time weakly-monotone algorithm for Π . If every Θ_i is upper-bounded by a finite value \sup_i and discrete w.r.t. a known value Δ_i , then for every $1 \leq i \leq m$ there exists a value for the constant $c_i^{(1)}$ such that $\mathcal{M} = (A, P^{(1)})$ is a polynomial-time truthful mechanism for Π . Moreover, \mathcal{M} satisfies voluntary participation condition.*

Proof. Let S_{t_i} be the schedule computed by A when takes as input (t_i, B_{-i}) , and S_{b_i} be the one on the input (b_i, B_{-i}) . To demonstrate that \mathcal{M} is a truthful mechanism, we show that for all $b_i \in \Theta_i$ and for all $\mathcal{T}_i \geq w_i(S_{b_i}) \cdot t_i$ the following relation holds

$$A_i = \text{profit}_i(S_{t_i}, (t_i, B_{-i}), \sigma, t_i, w_i(S_{t_i}) \cdot t_i) - \text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) \geq 0.$$

For sake of readability we denote (t_i, B_{-i}) as \mathbf{T} and $w_i(S_{t_i}) \cdot t_i$ as \mathcal{T}_i^* . We first consider the case $w_i(S_{b_i}) = 0$. Since $\text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) = 0$ we have

$$A_i = \frac{\mathcal{W}}{t_i} \cdot c_i^{(1)} - w_i(S_{t_i}) \cdot t_i \geq \mathcal{W} \cdot \left(\frac{c_i^{(1)}}{\sup_i} - \sup_i \right) \geq 0 \tag{1}$$

for all the values $c_i^{(1)} \geq \sup_i^2$. By the above calculations, we also have that $\text{profit}_i(S_{t_i}, \mathbf{T}, \sigma, t_i, \mathcal{T}_i^*) \geq 0$, and thus \mathcal{M} satisfies voluntary participation condition. Let $w_i(S_{b_i}) > 0$. We distinguish two cases.

Case 1 ($b_i > t_i$). Since A is weakly-monotone it holds that $w_i(S_{t_i}) > 0$. If $\text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) < 0$, from Eq. 1 we have $A_i > 0$ for $c_i^{(1)} \geq \sup_i^2$. Let $\text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) \geq 0$. Then we have:

$$\begin{aligned} A_i &= \mathcal{W} \cdot \left(\frac{1}{t_i} - \frac{1}{b_i} \right) \cdot c_i^{(1)} - (w_i(S_{t_i}) - w_i(S_{b_i})) \cdot t_i \\ &\geq \mathcal{W} \cdot \Delta_i \cdot c_i^{(1)} - \mathcal{W} \cdot \sup_i \geq 0 \end{aligned}$$

for all the values $c_i^{(1)} \geq \sup_i / \Delta_i$.

Case 2 ($b_i \leq t_i$). Since $T > w_i(S_{b_i}) \cdot b_i$, we have that $P_i^{(1)}(S_{b_i}, B, \sigma, \mathcal{T}_i) = 0$ and $\text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) < 0$. Therefore, from Eq. 1 we have $A_i > \text{profit}_i(S_{t_i}, \mathbf{T}, \sigma, t_i, \mathcal{T}_i^*) \geq 0$ for $c_i^{(1)} \geq \sup_i^2$.

Hence, for $c_i^{(1)} \geq \max\{\sup_i^2, \frac{\sup_i}{\Delta_i}\}$, \mathcal{M} is truthful. It is straightforward that payment scheme $P^{(1)}$ is computable in polynomial time. \square

As argued in Section 1, if A is the algorithm used in a truthful mechanism, then it always works on true types, since every agent always reports her true type. As a consequence, if A is c -approximate and $\mathcal{M} = (A, P)$ is a truthful mechanism then \mathcal{M} is c -approximate as well. Thus, from Theorem 1 we have the following.

Theorem 2. *Let Π be an optimization problem for verifiable one-parameter selfish agents and A be a polynomial-time c -approximating weakly-monotone algorithm for Π . If every Θ_i is upper-bounded by a finite value \sup_i and is discrete w.r.t. a known value Δ_i , then $\mathcal{M} = (A, P^{(1)})$ is a polynomial-time c -approximate truthful mechanism for Π , satisfying voluntary participation condition.*

Notice that if a type set is finite then it is discrete and finitely upper-bounded. Conversely if a type set is discrete and finitely upper-bounded it could contain infinite values. For instance consider the case in which for every $i = 1, \dots, m$, $\Theta_i \subseteq \{i^{-1} | i \in \mathbb{N}\}$. This is a special case of the discrete and upper-bounded type set: $\Delta_i = 1$ and $\sup_i = 1$, for every type set Θ_i .

4 A Payment Scheme for Rational Types

In this section, we show how to extend our payments in order to deal with rational type set which are only upper-bounded by a finite value \sup_i . To do that, we apply a rounding technique on types. Given a bid vector B , we denote by B^R the vector obtained by B by replacing each element b_i with a rounded value b_i^R of b_i . If $\alpha^\gamma < b_i^{-1} \leq \alpha^{\gamma+1}$, then $b_i^R = 1/\alpha^{\gamma+1}$ for some $\gamma \in \mathbb{Z}$. Thus, if $B = (b_1, b_2, \dots, b_m)$ then $B^R = (b_1^R, b_2^R, \dots, b_m^R)$. Given an algorithm A for Π , we define algorithm A_α as the algorithm that, on input B and σ , simply run algorithm A on input B^R and σ .

Definition 10. *Let S be a schedule, B be a bid vector, σ be the public part of the input, \mathcal{T}_i be the observed completion time and $c_i^{(2)} \in \mathbb{R}^+$ be a constant (to be given). For each $i = 1, \dots, m$, we define*

$$P_i^{(2)}(S, B, \sigma, \mathcal{T}_i) := \begin{cases} \frac{w_i}{b_i^R} \cdot c_i^{(2)} & \text{if } w_i(S) \neq 0 \text{ and } \mathcal{T}_i = w_i(S) \cdot b_i; \\ 0 & \text{otherwise.} \end{cases}$$

The idea behind payment scheme $P^{(2)}$ is similar to the one for $P^{(1)}$. The difference is that we consider the rounded bid b_i^R instead of the declared bid b_i and the used constant $c_i^{(2)}$ is essentially different from $c_i^{(1)}$. In the next theorem, we will better clarify the meaning of constant $c_i^{(2)}$.

Theorem 3. *Let Π be an optimization problem for verifiable one-parameter selfish agents whose types are positive rational, and let A be a polynomial-time weakly-monotone algorithm for Π . If every Θ_i is upper-bounded by a finite value \sup_i , then for every $1 \leq i \leq m$ there exists a value for the constant $c_i^{(2)}$, such that $\mathcal{M} = (A_\alpha, P^{(2)})$ is a polynomial-time truthful mechanism for Π . Moreover, \mathcal{M} satisfies voluntary participation condition.*

Proof. First note that, if A is weakly-monotone then A_α is weakly-monotone as well. Let B be a vector of bids, and S_{t_i} be the schedule computed by algorithm A_α when it takes as input (t_i, B_{-i}) , and S_{b_i} be the one on input (b_i, B_{-i}) . To show that \mathcal{M} is truthful, we prove that for all $b_i \in \Theta_i$ and $\mathcal{T}_i \geq w_i(S_{b_i}) \cdot t_i$,

$$A_i = \text{profit}_i(S_{t_i}, (t_i, B_{-i}), \sigma, t_i, w_i(S_{t_i}) \cdot t_i) - \text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) \geq 0.$$

For the sake of readability we denote $\mathbf{T} = (t_i, B_{-i})$ and $\mathcal{T}_i^* = w_i(S_{t_i}) \cdot t_i$. We first consider the case $w_i(S_{b_i}) = 0$. In this case we have:

$$A_i = \text{profit}_i(S_{t_i}, \mathbf{T}, \sigma, t_i, \mathcal{T}_i^*) \geq \mathcal{W} \cdot \left(\alpha^{\gamma+1} \cdot c_i^{(2)} - \frac{1}{\alpha^\gamma} \right) \geq 0 \quad (2)$$

when

$$\gamma \geq -\frac{\log_\alpha c_i^{(2)}}{2} - \frac{1}{2}. \quad (3)$$

At the end of the theorem, we discuss how to choose $c_i^{(2)}$ in order \mathcal{M} to be truthful. From Eq. 2, we also have that $\text{profit}_i(S_{t_i}, \mathbf{T}, \sigma, t_i, \mathcal{T}_i^*) \geq 0$, thus \mathcal{M} satisfies voluntary participation condition.

It remains to show the case $w_i(S_{b_i}) > 0$. We distinguish two cases:

Case 1 ($b_i > t_i$). Since A is weakly-monotone and $b_i^R \geq t_i^R$ it holds that $w_i(S_{t_i}) > 0$. W.l.o.g. we only consider the case in which $\text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) \geq 0$. We first analyze the case $b_i^R = t_i^R$, i.e. b_i and t_i are rounded to the same power of α .

$$A_i = \mathcal{W} \cdot \left(\frac{1}{t_i^R} - \frac{1}{b_i^R} \right) \cdot c_i^{(2)} - (w_i(S_{t_i}) - w_i(S_{b_i})) \cdot t_i = 0.$$

Here, we analyze the remaining case in which $b_i^R > t_i^R$. Then, for some $\gamma \in \mathbb{Z}$, it holds:

$$\begin{aligned} A_i &= \mathcal{W} \cdot \left(\frac{1}{t_i^R} - \frac{1}{b_i^R} \right) \cdot c_i^{(2)} - (w_i(S_{t_i}) - w_i(S_{b_i})) \cdot t_i \geq \\ &\geq \mathcal{W} \cdot \left(\frac{1}{t_i^R} - \frac{1}{b_i^R} \right) \cdot c_i^{(2)} - \mathcal{W} \cdot t_i \geq \mathcal{W} \cdot \left((\alpha^{\gamma+1} - \alpha^\gamma) \cdot c_i^{(2)} - \frac{1}{\alpha^\gamma} \right) \end{aligned} \quad (4)$$

By simple calculations we have that Eq. 4 is greater or equal to 0 when:

$$\gamma \geq -\frac{\log_\alpha(c_i^{(2)})}{2} - \frac{\log_\alpha(\alpha - 1)}{2}. \quad (5)$$

As in the previous case, we postpone the discussion of choosing $c_i^{(2)}$ for the end of the theorem.

Case 2 ($b_i \leq t_i$). Since $T > w_i(S_{b_i}) \cdot b_i$, we have that $P_i^{(2)}(S_{b_i}, B, \sigma, \mathcal{T}_i) = 0$ and $\text{profit}_i(S_{b_i}, B, \sigma, t_i, \mathcal{T}_i) < 0$, implying that $A_i \geq \text{profit}_i(S_{t_i}, \mathbf{T}, \sigma, t_i, \mathcal{T}_i^*) \geq 0$.

Here we discuss how to choose the constant $c_i^{(2)}$ in order to satisfy both Eq. 3 and Eq. 5, for any value of γ . In particular, we just show for the case in which $\gamma = \gamma_{\min}$, where γ_{\min} is the minimal value that γ can have. Since $\frac{1}{\sup_i} \leq (\frac{1}{\sup_i})^R = \alpha^{\gamma_{\min}}$, by simple calculations we have $\gamma_{\min} = \lceil \log_{\alpha} \frac{1}{\sup_i} \rceil$. Since $\log_{\alpha}(c_i^{(2)})$ can have as value any real number by varying $c_i^{(2)}$, we can compute a value of $c_i^{(2)}$ such that both Eq. 3 and Eq. 5 are satisfied when $\gamma = \gamma_{\min}$. Hence, it is straightforward that payment scheme $P^{(2)}$ is computable in polynomial time. \square

Note that, if in Def. 10 the constant $c_i^{(2)}$ is not used, then from Eq. 3 and Eq. 5 we may observe that in order \mathcal{M} to be truthful, type sets Θ_i must be upper-bounded by a constant which depends on the value of $\max\{-\frac{1}{2}, -\frac{\log_{\alpha}(\alpha-1)}{2}\}$. Thus, $c_i^{(2)}$ allows us to deal with any type set Θ_i that is upper-bounded by any constant \sup_i .

In order to have truthful mechanism for the problem at hand, involving agents having type set upper-bounded by a finite value, we round the bids. But what about the approximation? If A is a c -approximation algorithm, then nothing can be said about the approximation of A_{α} . Next, we define the class of problems for which the rounding increases the approximation of A_{α} by a guarantee factor with respect to the approximation guarantee of A . Henceforth, we restrict our attention only to minimization problems. We stress that similar arguments can be applied for maximization problems as well.

Definition 11. Fix $\epsilon > 0$ and $\delta > 1$. A one-parameter minimization problem $\Pi = (\mathcal{I}, \mathbf{m}, \text{sol}, \min)$ is (δ, ϵ) -smooth if, for any pair of instances $I = (T, \sigma)$ and $\tilde{I} = (\tilde{T}, \sigma)$ such that $t_i \leq \tilde{t}_i \leq \delta \cdot t_i$ for $i = 1, 2, \dots, m$, and for all $S \in \text{sol}(\sigma)$, it holds that $\mathbf{m}(S, I) \leq \mathbf{m}(S, \tilde{I}) \leq (1 + \epsilon) \cdot \mathbf{m}(S, I)$.

For instance, observe that $Q||C_{max}$ is $(\alpha, \alpha - 1)$ -smooth for all $\alpha > 1$. From the above definition, the following remark is straightforward.

Remark 1. Let Π be a (δ, ϵ) -smooth one-parameter minimization problem and let $I = (T, \sigma)$ and $\tilde{I} = (\tilde{T}, \sigma)$ be two instances of Π such that $t_i \leq \tilde{t}_i \leq \delta \cdot t_i$, for $i = 1, 2, \dots, m$. Then, any c -approximate solution S for I is $c \cdot (1 + \epsilon)$ -approximate for \tilde{I} and any c -approximate solution \tilde{S} for \tilde{I} is $c \cdot (1 + \epsilon)$ -approximate for I .

From Theorem 3 and the above remark we have the next theorem.

Theorem 4. Let Π be a $(\alpha, \alpha - 1)$ -smooth optimization problem for verifiable one-parameter selfish agents whose types are positive rational, and let A be polynomial-time c -approximate weakly-monotone algorithm for Π . If every Θ_i is upper-bounded by a finite value \sup_i , then $\mathcal{M} = (A_{\alpha}, P^{(2)})$ is $(\alpha \cdot c)$ -approximate polynomial-time truthful mechanism for Π , satisfying voluntary participation condition.

5 Applications to $Q||C_{max}$ Problem

In this section we give a non-trivial application of our results to the well known $Q||C_{max}$ problem. In the case in which type sets are discrete, then given a c -approximate polynomial-time weakly-monotone algorithm for $Q||C_{max}$ problem, we can construct c -approximate polynomial-time truthful mechanism for $Q||C_{max}$. On the other hand, when we have no constraints on the type sets, then given a c -approximate polynomial-time weakly-monotone algorithm for $Q||C_{max}$ problem, we can construct $c \cdot (1 + \epsilon)$ -approximate polynomial-time truthful mechanism for $Q||C_{max}$, for every $\epsilon > 0$.

We refer to the Section 1 for the definition of the problem. In the $Q||C_{max}$ problem with verifiable one-parameter selfish agents¹, the machines are owned by verifiable selfish agents wishing to maximize their own profit (as discussed in the section 1) disregarding the global makespan minimization. In particular, the job weights $W = (w_1, w_2, \dots, w_n)$, are the public part of the input, and the speeds of the machines are the private part of the input, that is, each agent i privately knows the speed of her machine. As usual, we assume that the types of the agents are the inverse of the speed.

As shown in Theorem 1 and Theorem 3, to apply our payment schemes, type sets must be upper-bounded by a finite value. For $Q||C_{max}$ problem, since types are the inverse of the speeds, it means that the speed of every agent (the inverse of the declared bid) has to be lower-bounded by a constant greater than 0, but this could not be the case. Therefore, here we show a method to deal with these cases. We show that it is always possible to reduce any instance of $Q||C_{max}$ to the one where every type set is upper-bounded by a finite value preserving the optimum of the instance.

The idea is to give a lower bound on the speed (an upper bound on the declared bid) to each agent depending on the declaration of the other agents. Thus, if an agent declares a speed value too small with respect to the other declared speeds, then she can be discarded. Let us proceed more formally.

To compute the lower bound of each agent we execute the following algorithm taking as input the bid vector B and the weight of the jobs W . (We call this algorithm *BoundTypes*(B, W).

1. For all $i \in \{1, \dots, m\}$, if s_i is lower-bounded by a constant $\hat{s}_i > 0$, then use this value as lower bound for the machine i , otherwise execute the steps 2 – 3.
2. Let k be a fastest machine in $\{1, \dots, i - 1\} \cup \{i + 1, \dots, m\}$ w.r.t. the bid vector B (that is a machine with a smallest bid without considering machine i); let time_i be the time needed (considering the bid b_k) for machine k to execute all the jobs: $\text{time}_i = \mathcal{W} \cdot b_k$.
3. Let w_j be a minimum weight job; use the value $\hat{s}_i = \frac{w_j}{\text{time}_i}$ as a lower bound for the speed of machine i .

¹ In the rest of the paper, with an abuse of notation, we will simply call $Q||C_{max}$ problem the one with verifiable one-parameter selfish agents since here we deal only with the latter.

To understand the motivation of this method, we consider the following: if the machine i declares $b_i > \frac{1}{\hat{s}_i}$, then for any optimum solution OPT we have $w_i(OPT) = 0$, since there exists a machine requiring less time to execute all jobs with respect to the time needed to machine i to complete a job having the smallest weight. Let A be a weakly-monotone algorithm for $Q||C_{max}$ problem. Now, we describe a weakly-monotone algorithm A' for $Q||C_{max}$ which uses A as a subroutine. A' has the same approximation ratio of A and can be used to deal with machines having unbounded speeds. It takes as input the bid vector B and the weight vector W and outputs a schedule S .

1. Let $(\hat{s}_1, \dots, \hat{s}_m) = BoundTypes(B, W)$; let \hat{B} the bids vector B without the machines bidding $b_i > \frac{1}{\hat{s}_i}$; let \hat{S} be the schedule returned by A executed on \hat{B} and W ;
2. Let S be a schedule equal to \hat{S} for all machines declaring $b_i < \frac{1}{\hat{s}_i}$ assigning 0 to all the other machines; return S as the schedule.

Now, we show that this algorithm leads to a truthful mechanism (together with our payment schemes) and that it has the same approximation ratio as the algorithm A .

Lemma 2. *If A is a weakly-monotone c -approximate algorithm for $Q||C_{max}$ problem, then A' is a weakly-monotone c -approximate algorithm for $Q||C_{max}$ problem.*

Proof. We first show that A' is a scheduling algorithm. Since A is a scheduling algorithm, we only have to show that at least one machine is given as input to algorithm A . Now, we show that we never discard the fastest machines. Let i be a fastest machine in $\{1, \dots, m\}$ and k a fastest machine in $\{1, \dots, i-1\} \cup \{i+1, \dots, m\}$. Then, obviously $s_k = \frac{1}{b_k} \leq \frac{1}{b_i} = s_i$. Let T be the time needed by machine k to execute all jobs and let w be a smallest job. From the definition of \hat{s}_i , we have $\hat{s}_i = \frac{w}{T} = \frac{w}{W} \cdot s_k \leq s_k \leq s_i$. This implies that the fastest machines are surely not discarded. We now prove that A' is weakly-monotone. Fix a bid vector B , and suppose that $w_i(A'(B, W)) = 0$. We prove that $w_i(A'((b', B_{-i}), W)) = 0$, for every $b' \geq b_i$. In the case $b' > \frac{1}{\hat{s}_i}$, trivially $w_i(A'((b', B_{-i}), W)) = 0$, since machine i will be discarded. If $b'_i \leq \frac{1}{\hat{s}_i}$, then machine is not discarded and $w_i(A'((b', B_{-i}), W)) = 0$, given that algorithm A is weakly-monotone. Finally, to show that the algorithm A' is a c -approximate algorithm, we only prove that the deletion of the "slowest" machines that does not modify the optimum. More specifically, let I be the initial instance of the problem and OPT be an optimum solution for I . If $b_i > \frac{1}{\hat{s}_i}$ (i.e. machine i is a discarded machine), then $w_i(OPT) = 0$. In fact, the time needed by the machine i to complete the smallest job w is greater than the time needed to the fastest machine to complete the overall jobs. \square

Algorithm A' , using the algorithm $BoundTypes$, reduces any (potentially unbounded) instance I of $Q||C_{max}$ to a bounded instance \hat{I} of $Q||C_{max}$. Thus we can apply our payment scheme. By Lemma 2 and Theorem 2 we have:

Theorem 5. *Let A be a c -approximate polynomial-time weakly-monotone algorithm for $Q||C_{max}$ problem. If every Θ_i is discrete w.r.t. a known value Λ_i , then there exists a c -approximate polynomial-time truthful mechanism $\mathcal{M} = (A', P^{(1)})$ for $Q||C_{max}$, satisfying voluntary participation condition.*

By Lemma 2, Theorem 4 and since $Q||C_{max}$ problem is $(1 + \epsilon, \epsilon)$ -smooth we have:

Theorem 6. *Let A be a c -approximate polynomial-time weakly-monotone algorithm for $Q||C_{max}$ problem. Then, for any $\epsilon > 0$, there exists a $c \cdot (1 + \epsilon)$ -approximate polynomial-time truthful mechanism $\mathcal{M} = (A', P^{(2)})$ for $Q||C_{max}$, satisfying voluntary participation condition.*

Acknowledgments. We wish to thank the authors of [4] for providing us with a full version of their paper.

References

1. N. Andelman, Y. Azar, and M. Sorani. Truthful approximation mechanisms for scheduling selfish related machines. In *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*. LNCS, 2005.
2. A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.
3. V. Auletta, R. De Prisco, P. Penna, and P. Persiano. Deterministic Truthful Mechanisms for Scheduling on Selfish Machines. In *Proceedings of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2996, pages 608–619. LNCS, 2004.
4. V. Auletta, R. De Prisco, P. Penna, and P. Persiano. The Power of Verification for One-Parameter Agents. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3142, pages 171–182. LNCS, 2004.
5. E. H. Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.
6. R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, (45):1563–1581, 1966.
7. R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal of Applied Mathematics*, 17(2), 1969.
8. N. Nisan and A. Ronen. Algorithmic Mechanism Design. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 129–140, 1999.
9. N. Nisan and A. Ronen. Computationally Feasible VCG Mechanisms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC)*, pages 242–252, 2000.
10. V. Vazirani. *Approximation Algorithms*. Springer, 2001.
11. W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.