Collusion-Resistant Mechanisms with Verification Yielding Optimal Solutions*

Paolo Penna[1] and Carmine Ventre[2,**]

[1] Dipartimento di Informatica ed Applicazioni, Università di Salerno, Italy
penna@dia.unisa.it
[2] Computer Science Department, University of Liverpool, UK
Carmine.Ventre@liverpool.ac.uk

Abstract. A *truthful mechanism* consists of an algorithm augmented with a suitable payment function which guarantees that "players" cannot improve their utilities by "cheating". Mechanism design approaches are particularly appealing for designing "protocols" that cannot be manipulated by rational players.

We present new constructions of so called mechanisms *with verification* introduced by Nisan and Ronen [STOC 1999]. We first show how to obtain mechanisms that, for single-parameter domains, are resistant to *coalitions* of colluding agents even in the case in which compensation among members of the coalition is allowed (i.e., n-*truthful* mechanisms). Based on this technique we derive a class of *exact* truthful mechanisms with verification for *arbitrary* bounded domains. This class of problems includes most of the problems studied in the algorithmic mechanism design literature and for which exact solutions cannot be obtained with truthful mechanisms *without verification*. This result improves over all known previous constructions of exact mechanisms with verification.

1 Introduction

A large body of the literature studies ways to incorporate economic and game-theoretic considerations in the design of algorithms and protocols. One of the most studied and acknowledged paradigms is *mechanism design* (see, e.g., [1, 4, 7, 12, 13, 16]). Distributed computations over the Internet often involve self-interested parties (referred to as *selfish agents*) which may manipulate the protocol by misreporting a fundamental piece of information they hold (their own *type*). The protocol runs some algorithm which, because of the misreported information, is no longer guaranteed to return a "globally optimal" solution (optimality is naturally expressed as a function of agents' types) [13, 16]. Since agents can manipulate the algorithm by misreporting their types, one augments the algorithms with carefully designed payment functions which make it disadvantageous for an agent to do so. A *mechanism* consists of an algorithm (also termed social choice function) and a payment rule which associates a payment to every agent. Each agent derives a *utility* which depends on the solution computed by the algorithm, on

* Research funded by the European Union through IST FET Integrated Project AEOLUS (IST-015964).
** The author is also supported by DFG grant Kr 2332/1-2 within Emmy Noether Program.

D. Halperin and K. Mehlhorn (Eds.): ESA 2008, LNCS 5193, pp. 708–719, 2008.

the type of the agent, and on the payment that the agent receives from the mechanism (the solution and the payment depend on the *reported* types). A mechanism is *truthful* if truthtelling is a *dominant strategy* for all agents. That is, the utility of any agent is maximized when this agent reports her type truthfully, no matter which strategy the other agents follow. An even stronger solution concept is that of *c-truthful* mechanism [9] which requires that no *coalition* of up to c agents can increase the utility of its members even when compensations (or side payments) among them occur.

The construction of a truthful mechanism is a challenging problem since the mechanism must fix the "rules" in advance without knowing the types of the agents. The only available information is that each agent's type belongs to some *domain* which depends on the problem and agents can only report types in that domain. Intuitively speaking, constructing truthful mechanisms for richer domains is more difficult because there are more ways in which an agent can cheat the mechanism.

In their seminal work on algorithmic game theory, Nisan and Ronen [13] suggested a rather innovative paradigm called *mechanisms with verification*. They showed that these mechanisms can overcome the main limitations of the "classical" approach which cannot guarantee exact (or even approximate solutions) for some interesting problems. Intuitively speaking, these mechanisms can optimize only certain global cost functions. Suppose that each agent i has a type t^i which, for every feasible solution x, specifies a cost $t^i(x)$ associated to this solution. The only known general technique for designing truthful mechanisms are the classical Vickrey-Clarke-Groves (VCG) mechanisms [5, 10, 23]; these mechanisms optimize only *utilitarian* problems, that is, global cost functions of the form

$$\sum_i \alpha_i \cdot t^i(x) \tag{1}$$

where each α_i is some nonnegative constant. Moreover, the celebrated Robert's Theorem [20] states that these are the only global cost functions that can be optimized, i.e., for which there exists an exact truthful mechanism, when agents' domains are unrestricted. Furthermore, no positive result on the construction of "classical" c-truthful mechanisms is known, even for simple domains.

This work presents new constructions of mechanisms with verification which guarantee c-truthful mechanisms for certain domains or exact solutions for a much more general class of global cost functions. Before discussing these and prior results in detail, we describe informally the main idea of mechanisms with verification:

Mechanisms with verification. Nisan and Ronen [13] introduced mechanisms with verification for a task scheduling problem in which each agent corresponds to a machine of type t^i. Tasks needs to be allocated to the machines, and each task allocation x results in a completion time $t^i(x)$ for a machine of type t^i. The key observation, made by Nisan and Ronen [13], is that machine i cannot release its tasks before $t^i(x)$ time steps. Therefore, if agent i reports a type b^i and a solution x is implemented, the mechanism is able to detect that b^i is *not* the true type of machine i if $b^i(x) < t^i(x)$.

Mechanisms *with verification* are based on this idea and apply to the following general framework (see Section 1.2 for a formal definition). For every feasible solution x, an agent of type t^i has a cost $t^i(x)$ associated to this solution. This cost is the time that this agent must spend for implementing solution x (artificial delays can be introduced at no cost since an agent can use the idle time for other purposes).[1] Agent i is caught lying if her reported type b^i and the computed solution x are such that $b^i(x) < t^i(x)$. Agents who are caught lying receive no payment.

In contrast, the classical approach in mechanism design is to provide always each agent with a payment that depends only on the reported types. In order to distinguish these mechanisms from mechanisms with verification, in the sequel we use the term mechanisms *without verification*.

1.1 Our Contribution and Related Work

We study the existence of truthful (or even c-truthful) mechanisms with verification that guarantee *exact* solutions for problems in which the objective is to minimize some global cost function of interest. Intuitively speaking, our basic question is whether one can augment an optimal algorithm with a suitable payment function in order to guarantee that no agent (or even coalitions of colluding agents) can benefit from misreporting their types (i.e., part of the input of the algorithm). We consider a rather general class of objective functions in which the global cost of a solution depends on the various costs that the agents associate to that solution; Naturally, the overall cost cannot decrease if the cost of one agent increases (see Section 2 for a formal definition). The contribution of this work is twofold:

- We provide a sufficient condition for which an algorithm can be turned into a c-*truthful* mechanism with verification, for any $c \geq 1$. This result applies to the class of *single-parameter* bounded domains (see Section 3 for a formal definition).
- We then show how to obtain optimal truthful mechanisms with verification for the much more general case of *arbitrary* bounded domains, i.e., the mechanism needs only an upper bound on the agents' costs (see Section 1.2). Despite the fact that these domains are extremely rich, we provide *exact* truthful mechanisms with verification for *every* problem in which the global cost function is of the form

$$\mathrm{Cost}(t^1(x), \ldots, t^n(x)) \tag{2}$$

  where $t^i(x)$ is the cost that agent i associates to solution x and the above function is naturally nondecreasing in its arguments.

The conditions for obtaining these mechanisms are stated in terms of *algorithmic properties* so that the design of the entire mechanism reduces to the design of an algorithm that fulfills these conditions. All our mechanisms satisfy also the *voluntary participation* condition saying that truthful agents have always a nonnegative utility.

The result on single-parameter bounded domains is the first technique for obtaining c-truthful mechanisms, for $c > 1$, without restricting to a particular class of global cost functions and it might be of some independent interest. For instance, certain non-utilitarian graph problems studied in [19] have single-parameter domains and thus are

[1] Nisan and Ronen [13] considered the case in which an agent introducing an artificial delay to her computations will pay *this* augmented cost. This is one of the differences between mechanisms with verification we consider and those in [13]. See [18] for a discussion.

the right candidate for studying exact n-truthful mechanisms based on our constructions (namely Theorems 4 and 5). Interestingly enough, the only way to guarantee c-truthfulness without verification, for $c \geq 2$, is to run a (useless) mechanism which returns always the same *fixed* solution [9, 21].

The result on arbitrary bounded domains improves significantly over the best known constructions of mechanisms with verification. In particular, [22] shows exact mechanisms for cost functions like (2) in the case of *finite* domains, i.e., there is a finite set of possible types that each agent can report to the mechanism. Exact n-truthful mechanisms with verification for a *subclass* of the cost functions in (2) are presented in [18]: For instance, it cannot give exact mechanisms for global cost functions of the form of the form $\max_i t^i(x)$. These so called min-max problems received a lot of attention in the algorithmic mechanism design literature [4, 8, 11, 12, 13]. These works prove that there is no exact or even r-approximate mechanism without verification, for some $r > 1$; results apply also to finite domains and to mechanisms without verification that run in exponential time and/or use randomization [12].

We instead show exact mechanisms with verification for any global cost function of the form (2) without assuming finite domains like in [22] (see Definition 2 and Theorem 6). Indeed, we only need to consider an (arbitrarily large) upper bound on the agents' costs, which turns out to be reasonable in practice. These arbitrary bounded domains are, in general, infinite because there are infinitely many types that an agent can report. Since the "cycle-monotonicity" approach adopted in all recent constructions [2, 3, 22] cannot deal with infinite domains, we use a totally different idea which is to turn c-truthful mechanisms for single-parameter domains into truthful mechanisms for arbitrary domains (see Section 4). The result of Theorem 6 is "tight" in the sense that one cannot relax any of the assumptions without introducing additional conditions (see Theorems 7 and 8). Finally, an explicit formula for the payments guarantees that the entire mechanism runs in polynomial-time if the algorithm is polynomial-time and the domain is finite (Corollary 1).

In this work we do not consider frugality issues, that is, how much the mechanism pays the agents. The optimality of the payments is an important issue *in general* since even truthful mechanisms must have large payments for rather simple problems [6]. Our positive results pose another interesting question that is to design *computationally-efficient* algorithms satisfying the conditions required by our methods.

Roadmap. Preliminary definitions are given in Section 1.2. In Section 2 we introduce the class of optimal algorithms leading to (c-)truthful mechanisms. Mechanisms for single-parameter domains are given in Section 3, while those for arbitrary domains are presented in Section 4. Due to lack of space some of the proofs are sketched or missing. The interested reader may refer to the full version of the paper [17].

1.2 Preliminaries

We have a finite set $\mathcal{O}$ of feasible alternative *solutions* (or *outcomes*). Without loss of generality, we assume that $\mathcal{O} = \{1, \ldots, a\}$, where $a = |\mathcal{O}|$, and sometimes write $x \leq y$

to denote the fact that outcome x precedes outcome y in this fixed order. There are n selfish agents, each of them having a so called *type*

$$t^i : \mathcal{O} \to \mathbb{R}^+$$

which associates a monetary *cost* to every feasible outcome. If an agent i receives a payment equal to r^i and an outcome x is selected, then her *utility* is equal to

$$r^i - t^i(x). \tag{3}$$

Each type t^i belongs to a so called *domain* D^i which consists of all admissible types, that is, a subset of all functions $u : \mathcal{O} \to \mathbb{R}$. The type t^i is *private knowledge*, that is, it is known to agent i only. Everything else, including each domain D^i, is *public knowledge*. Hence, each agent i can *misreport* her type to *any* other element b^i in the domain D^i. We sometimes call such b^i the *bid* or *reported type* of agent i. We let D being the cross product of all agents domains, that is, D contains all bid vectors $\mathbf{b} = (b^1, \ldots, b^n)$ with b^i in D^i. An *algorithm* A is a function

$$A : D \to \mathcal{O}$$

which maps all agents (reported) types $\mathbf{b}$ into a feasible outcome $x = A(\mathbf{b})$.[2] A *mechanism* is a pair (A, p), where A is an algorithm and $p = (p^1, \ldots, p^n)$ is a vector of suitable *payment functions*, one for each agent, where each payment function

$$p^i : D \to \mathbb{R}$$

associates some amount of money to agent i. We say that D is a *bounded domain* if there exists ℓ such that $b^i(x)$ belongs to the interval $[0, \ell]$, for all outcomes x, for all b^i in D^i, and for all agents i. Unless we make further assumptions on the domain D, we have (algorithms over) *arbitrary bounded domains*. Throughout the paper we consider only type vectors $\mathbf{t}$ in the domain D and we denote by t^i the type corresponding to agent i.

We say that an agent i is *truthtelling* if she reports her type, that is, the bid b^i coincides with her type t^i. Given an algorithm A and bids $\mathbf{b} = (b^1, \ldots, b^i, \ldots, b^n)$, we say that agent i is *caught lying by the verification* if the following inequality holds:

$$t^i(A(\mathbf{b})) > b^i(A(\mathbf{b})).$$

A mechanism (A, p) is a *mechanism with verification* if, on input bids $\mathbf{b}$, every agent that is caught lying does not receive any payment, while every other agent i receives her associated payment $p^i(\mathbf{b})$. Hence, the utility of an agent i whose type is t^i is equal to

$$\text{Utility}^i(\mathbf{b}) := \begin{cases} p^i(\mathbf{b}) - t^i(A(\mathbf{b})) & \text{if } i \text{ is not caught lying,} \\ 0 - t^i(A(\mathbf{b})) & \text{otherwise.} \end{cases}$$

On the contrary, we say that (A, p) is a *mechanism without verification* if every agent receives *always* her associated payment $p^i(\mathbf{b})$.

[2] In the game theory literature A is often referred to as *social choice function*.

For any two type vectors $\mathbf{t}$ and $\mathbf{b}$, we say that a coalition C *can misreport* $\mathbf{t}$ *to* $\mathbf{b}$ if the vector $\mathbf{b}$ is obtained by changing the type of some of the agents in C, i.e., $t^i = b^i$ for every agent i *not* in the coalition C. For any two type vectors $\mathbf{t}$ and $\mathbf{b}$, we say that *verification does not catch* $\mathbf{t}$ *misreported to* $\mathbf{b}$ if $t^i(A(\mathbf{b})) \leq b^i(A(\mathbf{b}))$ for every agent i. Conversely, we say that *verification catches* $\mathbf{t}$ *misreported to* $\mathbf{b}$ if $t^i(A(\mathbf{b})) > b^i(A(\mathbf{b}))$ for some agent i.

Mechanisms (with verification) which are resistant to coalitions of $c \geq 1$ colluding agents that can exchange side payments satisfy the following definition.

Definition 1 (c-truthfulness [9]). *A mechanism (with verification) is* c-truthful *if, for any coalition of size at most c and any bid of agents not in the coalition, the sum of the utilities of the agents in the coalition is maximized when all agents in the coalition are truthtelling.*

Mechanisms (with verification) satisfying the definition above only for $c = 1$ are called *truthful* mechanisms (with verification).

Since the above condition must hold for *all* possible bids of agents outside the coalition under consideration, one can restrict the analysis to the case in which these agents are actually truthtelling. Thus the following known fact holds:

Fact 1. *A mechanism (with verification) is c-truthful if and only if, for any coalition C of size at most c and for any two type vectors* $\mathbf{t}$ *and* $\mathbf{b}$ *such that C can misreport* $\mathbf{t}$ *to* $\mathbf{b}$*, the corresponding agents' utilities satisfy*

$$\sum_{i \in C} \mathrm{Utility}^i(\mathbf{t}) \geq \sum_{i \in C} \mathrm{Utility}^i(\mathbf{b}). \tag{4}$$

Throughout the paper we make use of the following standard notation. Given a type vector $\mathbf{v} = (v^1, \ldots, v^n)$, we let $\mathbf{v}^{-i}$ being the vector of length $n - 1$ obtained by removing v^i from $\mathbf{v}$, i.e., the vector $(v^1, \ldots, v^{i-1}, v^{i+1}, \ldots, v^n)$. We also let $(w, \mathbf{v}^{-i})$ be the vector $(v^1, \ldots, v^{i-1}, w, v^{i+1}, \ldots, v^n)$, which is obtained by replacing the i-th entry of $\mathbf{v}$ with w.

2 A Class of Optimal Algorithms

We focus on algorithms which minimize some global cost function of interest. Our ultimate goal is to derive a general technique to augment these algorithms with a suitable payment function so that the resulting mechanism with verification is truthful or even n-truthful (i.e., resistant to any coalition of colluding agents).

Towards this end, we consider algorithms that satisfy the following:

Definition 2 (exact algorithm with fixed tie breaking rule). *Let* $\mathrm{Cost} : \mathcal{O} \times D \to \mathbb{R}$ *be a function of the form*

$$\mathrm{Cost}(x, \mathbf{t}) = \mathrm{Cost}(t^1(x), \ldots, t^n(x)),$$

which is monotone non-decreasing *in each* $t^i(x)$*. We say that an algorithm A is an* exact algorithm *if there exists* $\mathcal{O}' \subseteq \mathcal{O}$ *such that, for all type vectors* $\mathbf{t}$*, it holds that*

$$A(\mathbf{t}) \in \arg\min_{x \in \mathcal{O}'} \{\mathrm{Cost}(x, \mathbf{t})\}.$$

Further, we say that A uses a fixed tie breaking rule *if, for any two type vectors* $\mathbf{t}$ *and* $\mathbf{b}$*,* $\mathrm{Cost}(A(\mathbf{t}), \mathbf{t}) = \mathrm{Cost}(A(\mathbf{b}), \mathbf{t})$ *implies that the outcomes $A(\mathbf{t})$ and $A(\mathbf{b})$ in the outcome set $\mathcal{O}$ satisfy: $A(\mathbf{t}) \leq A(\mathbf{b})$.*[3] *We say that A is an* exact algorithm with fixed tie breaking rule *if it is an exact algorithm and it uses a fixed tie breaking rule.*

Note that the definition of exact algorithm requires only the algorithm being optimal with respect to an arbitrarily fixed subset of solutions. Of course all positive results apply to algorithms that are optimal with respect to all solutions, i.e., the case $\mathcal{O}' = \mathcal{O}$. Observe also that the class of exact algorithm with fixed tie breaking rules strictly generalizes the class of algorithms that admit VCG-based truthful mechanisms (without verification) [15] and that optimizes *utilitarian* cost functions, that is, functions of the form (1).

3 Collusion-Resistant Mechanisms for Single-Parameter Agents

In this section we consider the case of *single-parameter* agents (see e.g. [9]). Here, each outcome partitions the agents into two sets: those that are *selected* and those that are *not selected*. The value $t^i(x)$ depends *uniquely* on the fact that i is selected in x or not and it is completely specified by a *parameter* t_i, which is a real number such that

$$t^i(x) = \begin{cases} t_i & \text{if } i \text{ is selected in } x, \\ 0 & \text{if } i \text{ is not selected in } x. \end{cases} \tag{5}$$

Whether i is selected in x is publicly known, for every outcome x, and thus each agent can only specify (and misreport) the parameter t_i. We assume *single-parameter bounded domains*, that is, each parameter t_i belongs to the interval $[0, \ell]$.

In the sequel we will provide sufficient conditions for the existence of c-truthful mechanisms, for any given $c \leq n$.

3.1 Sufficient Conditions for c-Truthfulness

We begin with a *necessary* condition. Observe that in order to have truthful mechanisms for single-parameter agents (even when using verification [2]) the algorithm must select agents "monotonically":

Definition 3 (monotone). *We say that algorithm A is* monotone *if the following holds. Having fixed the bids of all agents but i, agent i is selected if bidding a cost less than a threshold value $b_i^{\oplus}$, and is not selected if bidding a cost more than a threshold value $b_i^{\oplus}$. In particular, for every $\mathbf{b} \in D$ and for every i, there exists a value $b_i^{\oplus}$ which depends only on $\mathbf{b}^{-i}$ and such that (i) i is selected in $A(b^i, \mathbf{b}^{-i})$ for $b^i < b_i^{\oplus}$ and (ii) i is not selected in $A(b^i, \mathbf{b}^{-i})$ for $b^i > b_i^{\oplus}$.*

From Definition 3 we can easily obtain the following:

Fact 2. *If algorithm A is monotone and i is selected in $A(\mathbf{b})$, then $b_i \leq b_i^{\oplus}$. Moreover, if i is not selected in $A(\mathbf{b})$ then $b_i \geq b_i^{\oplus}$. Hence, for bounded domains the threshold values of Definition 3 are in the interval $[0, \ell]$.*

[3] Recall that we identify solutions with integers and thus fix an arbitrary order of them.

From (5) we immediately get the following:

Fact 3. *For single-parameter agents, it holds that verification does not catch* $\mathbf{t}$ *misreported to* $\mathbf{b}$ *if and only if* $t_i \leq b_i$ *for all* i *that is selected in* $A(\mathbf{b})$.

We next give a rather technical *sufficient* condition for c-truthfulness on single-parameter bounded domains. Below we show that, in the case of exact algorithms, this leads to a simpler condition for n-truthfulness on these domains.

Definition 4 (c-resistant). *We say that* $\mathbf{b}$ *is* c-different *from* $\mathbf{t}$ *if these two type vectors differ for at most* c *agents' types. A monotone algorithm* A *is* c-resistant *if, for every* $\mathbf{b}$ *which is* c*-different from* $\mathbf{t}$ *and such that verification does not catch* $\mathbf{t}$ *misreported to* $\mathbf{b}$*, it holds that* $t_i^{\oplus} \leq b_i^{\oplus}$ *for all* i *that is not selected in* $A(\mathbf{b})$.

Theorem 4. *Every* c*-resistant algorithm* A *admits a* c*-truthful mechanism with verification for single-parameter bounded domains.*

Proof. We define the payment functions as follows:

$$p^i(\mathbf{b}) := \begin{cases} \hbar - b_i^{\oplus} & \text{if } i \text{ is not selected in } A(\mathbf{b}) \\ \hbar & \text{otherwise} \end{cases} \tag{6}$$

where $\hbar := c \cdot \ell$.

Let us consider an arbitrary coalition C of size at most c and any two type vectors $\mathbf{t}$ and $\mathbf{b}$ such that C can misreport $\mathbf{t}$ to $\mathbf{b}$. Because of Fact 1, it suffices to prove (4). Either verification does not catch $\mathbf{t}$ misreported to $\mathbf{b}$ or verification catches $\mathbf{t}$ misreported to $\mathbf{b}$. We consider the two cases separately.

If verification catches $\mathbf{t}$ misreported to $\mathbf{b}$, then we have at least one agent $j \in C$ which does not receive any payment for $\mathbf{b}$. Moreover, the payment received by every other agent i in the coalition is at most $\hbar$. Hence, we have

$$\sum_{i \in C} \text{Utility}^i(\mathbf{b}) \leq (c-1)\hbar = c\hbar - \hbar.$$

We next show that the utility of every truthtelling agent is at least $\hbar - \ell$. Indeed, the definition of $p^i()$ implies that $\text{Utility}^i(\mathbf{t})$ is either $\hbar - t_i^{\oplus}$ if i is not selected in $A(\mathbf{t})$, or $\hbar - t_i$ if i is selected in $A(\mathbf{t})$. Fact 2 says that $t_i^{\oplus} \leq \ell$ and, if i is selected in $A(\mathbf{t})$, then $t_i \leq t_i^{\oplus}$. Hence, $\text{Utility}^i(\mathbf{t}) \geq \hbar - \ell$. From this and from our choice of $\hbar$, we obtain

$$\sum_{i \in C} \text{Utility}^i(\mathbf{t}) \geq c(\hbar - \ell) = c\hbar - c\ell = c\hbar - \hbar.$$

The two inequalities above clearly imply (4).

If verification does not catch $\mathbf{t}$ misreported to $\mathbf{b}$ then we can show that for any $i \in C$ it holds

$$\text{Utility}^i(\mathbf{t}) \geq \text{Utility}^i(\mathbf{b}),$$

which clearly implies (4). There are four possible cases:

Case 1 (i is selected in $A(\mathbf{t})$ and i is selected in $A(\mathbf{b})$). In this case nothing changes for i. Indeed, by the definition of $p^i()$, we have $\text{Utility}^i(\mathbf{t}) = \hbar - t_i = \text{Utility}^i(\mathbf{b})$.

Case 2 (i is not selected in $A(\mathbf{t})$ and i is selected in $A(\mathbf{b})$). Fact 2 implies that $t_i^{\oplus} \leq t_i$. This and the definition of $p^i()$ imply $\text{Utility}^i(\mathbf{t}) = \hbar - t_i^{\oplus} \geq \hbar - t_i = \text{Utility}^i(\mathbf{b})$.

Case 3 (i is not selected in $A(\mathbf{t})$ and i is not selected in $A(\mathbf{b})$). Since A is c-resistant, we have that $t_i^{\oplus} \leq b_i^{\oplus}$. This and the definition of $p^i()$ imply $\text{Utility}^i(\mathbf{t}) = \hbar - t_i^{\oplus} \geq \hbar - b_i^{\oplus} = \text{Utility}^i(\mathbf{b})$.

Case 4 (i is selected in $A(\mathbf{t})$ and i is not selected in $A(\mathbf{b})$). Since i is selected in $A(\mathbf{t})$, Fact 2 implies $t_i \leq t_i^{\oplus}$. Since i is not selected in $A(\mathbf{b})$ and as A is c-resistant, we have that $t_i^{\oplus} \leq b_i^{\oplus}$, thus implying $t_i \leq b_i^{\oplus}$. This and the definition of $p^i()$ imply $\text{Utility}^i(\mathbf{t}) = \hbar - t_i \geq \hbar - b_i^{\oplus} = \text{Utility}^i(\mathbf{b})$.

This concludes the proof. □

We next "specialize" the above result for the class of exact algorithm with fixed tie breaking rule and obtain a more easy-to-handle *sufficient* condition for obtaining n-truthful mechanisms with verification.

Definition 5 (threshold-monotone). *A monotone algorithm A is* threshold-monotone *if, for every $\mathbf{t}$ and every $\mathbf{b}$ obtained by increasing one agent entry of $\mathbf{t}$, the inequality $t_i^{\oplus} \leq b_i^{\oplus}$ holds for all i, where $t_i^{\oplus}$ and $b_i^{\oplus}$ are the threshold values of Definition 3.*

By showing that every threshold-monotone exact algorithm with fixed tie breaking rule is n-resistant from Theorem 4 we obtain another sufficient condition for n-truthful mechanisms. We believe this result might be useful in that the threshold-monotone condition might be simpler to exhibit.

Theorem 5. *Every threshold-monotone exact algorithm with fixed tie breaking rule admits an n-truthful mechanism with verification for single-parameter bounded domains.*

4 Truthful Mechanisms for Arbitrary Bounded Domains

In this section we derive truthful mechanisms for any exact algorithm with fixed tie breaking rule over arbitrary bounded domains. The main idea is to regard each agent as a *coalition* of (virtual) single-parameter agents.

4.1 Arbitrary Domains as Coalitions of Single-Parameter Agents

We call every agent whose domain is an arbitrary bounded domain a *multidimensional* agent. Since there are a alternative outcomes, and n multidimensional agents, we simply consider n coalitions $C_1, \ldots, C_n$, where each coalition C_i consists of a (virtual) single-parameter agents that correspond to the multidimensional agent i. (These are actually "known" coalitions which we use only for the purpose of defining the payments and analyzing the resulting mechanism.) This "new game" has $N = n \cdot |\mathcal{O}|$ single-parameter agents and a alternative outcomes. For each outcome x, we have a unique single-parameter agent per coalition being selected: denoted by $1^{(i)}, \ldots, |\mathcal{O}|^{(i)}$

the agents in coalition C_i, we have agent $x^{(i)}$ being selected in x, and every other agent in C_i being not selected in x; this holds for all coalitions above. We choose the parameter of the (virtual) single-parameter agents in the coalition C_i so that the cost for an agent $x^{(i)}$ selected is equal to the cost of the multidimensional agent i when outcome x is selected. That is, for all i and all outcomes x, the parameter $t_{x^{(i)}}$ of agent $x^{(i)}$ is equal to $t^i(x)$.

Observe that any type b^i in the domain of the multidimensional agent i can be seen as a vector

$$\mathbf{b}^i := (b^i_1, \dots, b^i_a),$$

with $b^i_x = b^i(x)$ for every alternative outcome x. In particular, $\mathbf{b}^i$ is the vector of the parameters of the a agents in C_i, that is, b^i_x is the parameter of agent $x^{(i)}$. Consider an exact algorithm with fixed tie breaking rule B over the multidimensional agents, and fix the bids $\mathbf{b}^{-i}$ of all agents but i. Then the resulting *single player function* $B(b^i, \mathbf{b}^{-i})$ can be seen as another exact algorithm with fixed tie breaking rule $A(\mathbf{b}^i)$ whose domain (input) is restricted to the domains of the $a = |\mathcal{O}|$ single-parameter agents in C_i.

4.2 The Mechanism and Its Analysis

It turns out that every single player function $B(b^i, \mathbf{b}^{-i})$ as above is a-resistant. Based on this fact, we can apply the techniques developed for single-parameter agents and define the following class of mechanisms:

Definition 6 (threshold-based mechanism). *For any exact algorithm with fixed tie breaking rule B we consider its* single player function*, depending on $\mathbf{b}^{-i}$, as $A(\mathbf{b}^i) := B(b^i, \mathbf{b}^{-i})$. In this case, the single player function A has C_i as the set of* virtual *single-parameter agents. We define payment functions $q^i(b^i, \mathbf{b}^{-i}) := \sum_{j \in C_i} p^j(\mathbf{b}^i)$ where each $p^i()$ is the payment function of Theorem 4 when applied to A above and to the single-parameter agents in C_i. The resulting mechanism with verification (B, q) is called* threshold-based mechanism*.*

In the sequel we prove that every threshold-based mechanism is truthful *for multidimensional agents*. In order to prove this result, we first observe that the threshold-based mechanism needs only be resistant to the "known" coalitions defined above (recall that we have one virtual single-parameter agent per solution and thus coalitions are of size at most $|\mathcal{O}|$):

Lemma 1. *If every single player function A of B is $|\mathcal{O}|$-resistant with respect to its virtual single-parameter agents, then the threshold-based mechanism is truthful for the multidimensional agents.*

PROOF SKETCH. We observe that the utility of a multidimensional agent i is the sum of the utilities of all single-parameter agents in the coalition C_i. Therefore, if (B, q) was not truthful, then the mechanism (A, p) would not be a-truthful, thus contradicting Theorem 5. □

Theorem 6. *Every exact algorithm with fixed tie breaking rule admits a truthful mechanism with verification over any arbitrary bounded domain.*

PROOF SKETCH. It is possible to show that every single player function A of B is threshold-monotone. But then every single player function is a-resistant (see discussion above Theorem 5). The theorem thus follows from Lemma 1. □

We next observe that one cannot extend the result of Theorem 6 by relaxing the definition of exact algorithm with fixed tie breaking rule. Indeed, the "non-decreasing cost function", the "fixed tie breaking rule" and the "optimality" assumptions are necessary in order to guarantee the existence of exact truthful mechanisms with verification without introducing other conditions. The optimality condition is necessary as we obtain exact mechanisms. As for the other two assumptions we can prove the next two theorems.

Theorem 7. *For any cost function that is* not *monotone nondecreasing there exists a bounded domain such that no algorithm that minimizes such a cost function admits a truthful mechanism with verification.*

We next remove the fixed tie breaking rule from our assumptions and show that there exists an exact algorithm not admitting truthful mechanisms with verification.

Theorem 8. *There exists a bounded domain and a monotone cost function such that the following holds. There exists an exact algorithm (not using a fixed tie breaking rule) which does not admit any truthful mechanism with verification.*

We conclude this section by observing that the mechanisms presented here have a further advantage of giving an explicit formula for the payments (see Equation 6 and Definition 6). In particular, this improves over the construction in [22] since it gives efficient mechanisms for the case of arbitrary finite domains. The idea is to perform a binary search to determine the threshold values of Definition 3. For threshold-based mechanisms the running time is polynomial in the size of the input $\mathbf{t}$, where each t^i is a vector of $|\mathcal{O}|$ values, one for each outcome. Such an "explicit" representation of the input is in general necessary, as implied by communication complexity lower bounds for certain instances of combinatorial auction [14] which fall into the class of finite domains.

Corollary 1. *Every polynomial-time exact algorithm with fixed tie breaking rule over an arbitrary finite domain admits a polynomial-time truthful mechanism with verification. For finite single-parameter domains, every polynomial-time c-resistant exact algorithm with fixed tie breaking rule admits a polynomial-time c-truthful mechanism with verification.*

Acknowledgements. We wish to thank Riccardo Silvestri for several useful comments on an earlier version of this work.

References

1. Archer, A., Tardos, E.: Truthful mechanisms for one-parameter agents. In: Proc. of FOCS, pp. 482–491 (2001)
2. Auletta, V., De Prisco, R., Penna, P., Persiano, G.: The power of verification for one-parameter agents. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 171–182. Springer, Heidelberg (2004)

3. Auletta, V., De Prisco, R., Penna, P., Persiano, G., Ventre, C.: New constructions of mechanisms with verification. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 596–607. Springer, Heidelberg (2006)
4. Christodoulou, G., Koutsoupias, E., Vidali, A.: A lower bound for scheduling mechanisms. In: Proc. of SODA, pp. 1163–1170 (2007)
5. Clarke, E.H.: Multipart Pricing of Public Goods. Public Choice, 17–33 (1971)
6. Elkind, E., Sahai, A., Steiglitz, K.: Frugality in path auctions. In: Proc. of SODA, pp. 701–709 (2004)
7. Feigenbaum, J., Papadimitriou, C.H., Sami, R., Shenker, S.: A bgp-based mechanism for lowest-cost routing. Distributed Computing 18(1), 61–72 (2005)
8. Gamzu, I.: Improved lower bounds for non-utilitarian truthfulness. In: Kaklamanis, C., Skutella, M. (eds.) WAOA 2007. LNCS, vol. 4927, pp. 15–26. Springer, Heidelberg (2008)
9. Goldberg, A.V., Hartline, J.D.: Collusion-resistant mechanisms for single-parameter agents. In: Proc. of SODA, pp. 620–629 (2005)
10. Groves, T.: Incentive in Teams. Econometrica 41, 617–631 (1973)
11. Koutsoupias, E., Vidali, A.: A lower bound of $1 + \phi$ for truthful scheduling mechanisms. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708, pp. 454–464. Springer, Heidelberg (2007)
12. Mu'alem, A., Schapira, M.: Setting lower bounds on truthfulness. In: Proc. of SODA, pp. 1143–1152 (2007)
13. Nisan, N., Ronen, A.: Algorithmic Mechanism Design. Games and Economic Behavior 35, 166–196 (2001)
14. Nisan, N., Segal, I.: The communication requirements of efficient allocations and supporting prices. Journal of Economic Theory (2006)
15. Nisan, N., Ronen, A.: Computationally Feasible VCG Mechanisms. In: Proc. of EC, pp. 242–252 (2000)
16. Papadimitriou, C.H.: Algorithms, games, and the internet. In: Proc. of STOC (2001)
17. Penna, P., Ventre, C.: Collusion-resistant mechanisms with verification yielding optimal solutions. Technical report (2008), http://www.dia.unisa.it/~penna/papers/esa08full.pdf
18. Penna, P., Ventre, C.: Optimal collusion-resistant mechanisms with verification. Technical Report (2008)
19. Proietti, G., Widmayer, P.: A truthful mechanism for the non-utilitarian minimum radius spanning tree problem. In: Proc. of SPAA, pp. 195–202 (2005)
20. Roberts, K.: The characterization of implementable choice rules. Aggregation and Revelation of Preferences, 321–348 (1979)
21. Schummer, J.: Manipulation through bribes. Journal of Economic Theory 91(3), 180–198 (2000)
22. Ventre, C.: Mechanisms with verification for any finite domain. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 37–49. Springer, Heidelberg (2006)
23. Vickrey, W.: Counterspeculation, Auctions and Competitive Sealed Tenders. Journal of Finance, 8–37 (1961)