
DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD
DASH.HARVARD.EDU


HARVARD LIBRARY
Office for Scholarly Communication

Passive Verification of the Strategyproofness of Mechanisms in Open Environments

Citation

Kang, Laura, and David C Parkes. 2006. Passive verification of the strategyproofness of mechanisms in open environments. In Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet: August 13-16, 2006, Fredericton, New Brunswick, Canada, ed. B. Spencer, 19-30. New York, NY: ACM Press.

Published Version

doi:10.1145/1151454.1151473

Permanent link

http://nrs.harvard.edu/urn-3:HUL.InstRepos:4039775

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. Submit a story .

Accessibility

Passive Verification of the Strategyproofness of Mechanisms in Open Environments

Laura Kang
Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
kang@eecs.harvard.edu

David C. Parkes
Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
parkes@eecs.harvard.edu

ABSTRACT

Consider an open infrastructure in which anyone can deploy mechanisms to support automated decision making and coordination amongst self-interested computational agents. Strategyproofness is a central property in the design of such mechanisms, allowing participants to maximize their individual benefit by reporting truthful private information about preferences and capabilities and without modeling or reasoning about the behavior of other agents. But, why should participants trust that a mechanism is strategyproof? We address this problem, proposing and describing a *passive verifier*, able to monitor the inputs and outputs of mechanisms and verify the strategyproofness, or not, of a mechanism. Useful guarantees are available to participants before the behavior of the mechanism is completely known, and metrics are introduced to provide a measure of partial verification. Experimental results demonstrate the effectiveness of our method.

Categories and Subject Descriptors

H.4 [**Multiagent Systems and Electronic Markets**]

General Terms

Keywords

Verification, strategyproofness, game theory, mechanism design, constraint networks.

1. INTRODUCTION

Recent work in computational mechanism design has uncovered many interesting methods for decision-making in domains with self-interested agents, each with private information about its utility for different outcomes [7; 1; 2; 15, e.g.]. Common to much of this work is the drive for *strategyproof* mechanisms; i.e., direct-revelation mechanisms in which truthful reporting of private information is a dominant strategy equilibrium. This is useful because it sidesteps the semantic and computational difficulties of more intricate equilibrium concepts, such as Bayes-Nash equilibrium. Many have argued that strategyproofness is important to encourage the adoption of agent-mediated decision making: with users able to trust autonomous agents to "do the right thing" because agents have a simple and provably optimal strategy.

Our work is motivated by a future in which multiple entities (e.g. firms, people, organizations, network services) are able to deploy decision mechanisms in an open computational infrastructure. These mechanisms can be used, for instance, to coordinate purchasing decisions, allocate resources, schedule bandwidth, or form coordinated plans of action.

A new issue arises for mechanism design in open settings. Why should participants *trust* that a mechanism has the strategyproof property that it claims? We address this problem, proposing and describing a *passive verifier*, able to monitor the inputs and outputs of mechanisms and verify the strategyproofness, or not, of a mechanism. Thus, our focus is not on the design of new mechanisms but on the verification of strategyproofness. Indeed, without verification careful design is worthless: it is only when a mechanism is both strategyproof *and known to be strategyproof* that participants gain the benefits of simple strategies and mechanisms behave as designed. Thus, verification is in the interest of both designers and participants.

The passive verifier that we design exploits a price-based characterization of truthful mechanisms, and is inspired by the work of Gui et al. [11] on graph-theoretic characterizations of truthful mechanisms. We formulate the verification problem as one of checking for feasible solutions to a constraint satisfaction problem defined on a graph representing the rules of a mechanism. Verifiers are situated in the computational infrastructure and intermediate between participants and a mechanism. A verifier has the power to *veto* the outcome of a mechanism when the mechanism is proved to violate strategyproofness.

We identify techniques to both accelerate the process of verification as well as reduce the space complexity of verification. The idea is to place (weak) restrictions on the space of allowable mechanisms in order to simplify the task of verification. We provide three illustrations of this idea. First, we introduce the notion of *summarization*, which requires that a mechanism place restrictions on the complexity of its pricing rule, in identifying some subset $w < n$ of n participants, that define the price faced by each agent. Summarization provides an exponential reduction in the space complexity of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICEC '06, August 14-16, 2006, Fredericton, Canada
Copyright 2006 ACM 1-59593-392-1 ...$5.00.

verification, from $O(md^{n-1})$ to $O(md^w)$, in domains with m alternatives and with d possible agent valuations. Second, when a mechanism is required to satisfy a standard property on payments, that we term *natural payments*, we can leverage constraint propagation to further accelerate verification. Third, when a mechanism is required to satisfy a property of *envy-freeness*, reasonable for simple domains, additional acceleration is achieved.

The biggest challenge is to provide useful feedback to participants before the complete input space for a mechanism has been observed. Here, we identify a property that allows the verifier to guarantee that an agent's best strategy is truthful reporting (conditioned on the mechanism passing the verifier) even though only a subset of possible inputs has been observed. We identify a weaker condition that ensures that truthful reporting maximizes the *worst-case* utility of a participant against an adversarial (but ultimately non-strategyproof) mechanism in early stages of verification. We also provide two metrics, namely *probability-of-strategyproofness* and *price-flexibility*, to measure the degree of strategyproofness that is ensured for a mechanism that is still not completely verified. These metrics facilitate an empirical study in which we demonstrate the speed of verification (or failure of verification when using a weaker, baseline method) on various mechanisms, both strategyproof and non-strategyproof.

1.1 Related Work

Plenty of prior work has leveraged the methods of cryptography and secure function evaluation [18; 5, e.g.] to achieve absolute trust (i.e. the rules of a mechanism are published and correctness is verifiable to any party). When designing an auction that will be used for a multi-million dollar allocation of government bonds the overhead of cryptographically-secured proofs is well justified. Our motivation comes from a vision of dynamic, open environments that can support a multitude of frequent, perhaps even mundane, decisions. For instance, we wish to enable infrastructure that can support the minute-by-minute allocation of computational services, the buying and selling of limited-play songs, and services to make dinner reservations, solicit information about flight prices, and find answers to trivia questions.

Similarly, there is a large literature on the use of logic formalisms to verify the correctness of mechanisms and other institutions [20, 21, 8]. Here, a logic specification makes explicit the rules of a mechanism and properties such as strategyproofness can (in principle) be established through methods such as model checking. Of particular relevance is the work of Guerin and Pitt [10], who share the same motivation of allowing for the deployment of trusted mechanisms in open environments. Moreover, these authors discuss the use of "sentinel agents" which can be used to verify the properties of mechanisms at run time. Here, we are *only* interested in the verification of strategyproofness and not in the verification of other properties. Second, we deliberately eschew logic-based formalisms because model checking is intractable for appropriate logics [22], and because logic formalisms are a bad fit for quantitative properties such as strategyproofness and for the methods of combinatorial optimization that are important for internal decision making by mechanisms.

2. PRELIMINARIES

Formally, a mechanism consists of a social choice function $f : V_1 \times \ldots \times V_n \rightarrow A$, that chooses an alternative $f(v) = a \in A$ from a space of alternatives A, and a payment function $\tilde{p} : V_1 \times \ldots \times V_n \rightarrow \mathbb{R}^n$ that defines a payment $\tilde{p}_i(v)$ by each agent. Here, $v_i \in V_i$ is the valuation of agent i, where $v_i(a) \in \mathbb{R}$ is the value of agent i for alternative a. Also, denote $v = (v_1, \ldots, v_n) \in V$ and $v_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$. The valuation (or *type*) of an agent is private information, although we assume the type space V_i of each agent is common knowledge. Define $E_i(a) = \{d \in A | v_i(a) = v_i(d), \forall v_i \in V_i\}$[1], and let $R_f(v_{-i})$ denote the range of possible alternatives given social choice function f and reports v_{-i} from all but one agent.

Now we define a strategyproof mechanism.

DEFINITION 1. *A mechanism M is strategyproof if for all agents i with type v_i and for every $v_{-i} \in V_{-i}$: $v_i(a) - \tilde{p}_i(v_i, v_{-i}) \geq v_i(b) - \tilde{p}_i(v_i', v_{-i})$ where $a = f(v_i, v_{-i})$ and $b = f(v_i', v_{-i})$, for all types $v_i' \in V_i$, i.e., no agent can do better by misreporting her type, no matter what the other agents report.*

Well-known examples of strategyproof mechanisms include the family of Vickrey-Clarke-Groves(VCG) mechanisms [13], of which the second-price Vickrey auction is a special case. Other examples include: the greedy mechanism for single minded agents due to Lehmann et al. [15], mechanisms for one-parameter agents [1, e.g.], and truthful and competitive auctions for digital goods [7].

The key observation that we use for verifying the truthfulness (or strategyproofness) of mechanisms is that they must be *price-based* (see for instance [3, 23]).

THEOREM 1. *A mechanism $M =< f, \tilde{p} >$ is strategyproof if and only if for every agent i and every v_{-i}:*

(A1) the mechanism charges an agent-independent price $p_i(a, v_{-i})$ whenever alternative a is selected

(A2) for any report v_i, any v_{-i}, the mechanism chooses an alternative $a \in R_f(v_{-i})$ that maximizes $v_i(a) - p_i(a, v_{-i})$.

Here is some intuition for the sufficiency of (A1) and (A2) for strategyproofness: an agent cannot change the price that it faces (A1), and the mechanism maximizes its utility with respect to its reported valuation given these prices (A2). A simple argument can also be constructed for the necessary direction.

2.1 Problem Definition

Our verifiers act as lightweight trusted intermediaries (or "wrappers") that are situated as a default interface between published mechanisms and participants. A verifier receives bids from agents, passes them on the a mechanism and then receives the outcome and payments from the mechanism. At this point the verifier checks that the mechanism has not violated strategyproofness, and if this is OK it will pass the outcome on to the agents. Otherwise the verifier can exercise *veto* power on the outcome.

[1]For instance, in a resource allocation setting with no allocative-externalities the set of equivalent alternatives $E_i(a)$ would be all allocations that give agent i the same bundle of goods.

A mechanism is treated as a black box, i.e., the verifier can only observe a sequence of inputs, $v^1, v^2, \ldots v^k, \ldots$, where v^k denotes the reported valuations of the agents in the kth run of the mechanism, and a sequence of outputs, $< a^1, p^1 >$, $< a^2, p^2 >, \ldots, < a^k, p^k >, \ldots$ where $< a^k, p^k >$ denotes the alternative and vector of payments produced by the mechanism in the kth run.

DEFINITION 2 (PASSIVE VERIFICATION PROBLEM). *An online decision problem in which a verifier observes a sequence of inputs and outputs to a mechanism (assumed fixed and deterministic) and determines whether the mechanism is strategyproof or not strategyproof.*

A useful verifier will also reject a non-strategyproof mechanism quickly, and provide guidance on whether or not a mechanism is likely to be strategyproof even before a final "reject" or "accept" is generated. We also demonstrate that a verifier can prove that a mechanism is strategyproof given the private valuation of a bidder, even before the mechanism has been fully verified.

Here are our main assumptions:

1. *The space of alternatives and the type space are* finite.
2. *The verifier is trusted, is able to observe all inputs and outputs to a mechanism, and has* veto *power on any decision made by the mechanism (e.g. as soon as some decision, perhaps this decision, provides proof that the mechanism is not strategyproof).*
3. *Once the mechanism has decided on an alternative and payments for an instance, it can no longer change that decision in the future.*[2]
4. *The mechanism is* anonymous*, meaning that $f(v_i, v_{-i}) \in E_i(a)$ for all permutations of v_{-i}.*

The first assumption is needed to ensure that a sound and complete verifier can operate with bounded memory requirements. The second assumption is essential to our approach to passive verification. The third assumption ensures that the entire history of instances is always relevant for validating or invalidating strategyproofness. The final assumption allows the valuation profile, v_{-i}, of all agents except i to be treated as an unordered set of $n-1$ elements (allowing for repeated elements). This is helpful computationally, but should be relaxed in settings where anonymity is not provided (e.g. *optimal auctions* [17] where prior information about bidder valuations is used to bias the outcome of a mechanism.) These anonymous semantics for v_{-i} will be assumed for the rest of the paper.

3. PASSIVE VERIFICATION

In this section we define a set of rules that can be implemented by a passive verifier, and prove that verification with these rules is sound and complete. A constraint network formulation is given for the rules, which leads to a concrete algorithmic instantiation for a verifier.

3.1 SimpleChecker: Rules for Verification

Fix v_{-i}. By condition (A2) from Theorem 1, a strategyproof mechanism must choose an alternative $a \in A$ such that $v_i(a) - p_i(a, v_{-i})$ is maximized. Hence, if $f(v_i, v_{-i}) = a$, then $v_i(a) - p_i(a, v_{-i}) \geq v_i(b) - p_i(b, v_{-i})$ for all $b \neq a \in A$.

[2]For ties, this requires that ties are broken the same way each time. Notice that this does not rule out some forms of adaptiveness: a mechanism can update its prices via learning on any (a, v_{-i}) that has not been observed.

SIMPLECHECKER. Initialize history H to an empty history.

1. For a new instance $< v^k, a^k, p^k >$, and for each agent i, consider history $f^H(v^k_{-i})$ and if non-empty:
   (a) if $v^k_i \in v^H_i(v^k_{-i})$ check $a^k \in E_i(f^H(v^k))$.
   (b) if $a^k \in f^H(v^k_{-i})$ check $p^k_i = p^H_i(a^k, v^k_{-i})$.
   (c) if $a^k \notin f^H(v^k_{-i})$ check the price p^k_i is no less than:
   $$\sup_{b \in f^H(v^k_{-i})} \left\{ p^H_i(b, v^k_{-i}) + \sup_{v_i \in v^H_i(b, v^k_{-i})} (v_i(a^k) - v_i(b)) \right\} \quad (5)$$
   (d) if $v^k_i \notin v^H_i(v^k_{-i})$ check the price p^k_i is no greater than:
   $$v^k_i(a^k) + \inf_{b \in f^H(v^k_{-i})} \left\{ p^H_i(b, v^k_{-i}) - v^k_i(b) \right\} \quad (6)$$
2. If any check fails, then **reject** the mechanism, else update history for v^k_{-i} as necessary (i.e. whenever a new v_i and/or new alternative was observed).
3. Once all $v_i \in V_i$ for all $v_{-i} \in V_{-i}$ have been observed, then **pass** the mechanism.

Let $G_a \subseteq \{v_i \in V_i | a \in E_i(f(v_i, v_{-i}))\}$. Rearranging, we have, $p_i(a, v_{-i}) - p_i(b, v_{-i}) \leq v_i(a) - v_i(b)$ $\forall b \neq a$ for all $v_i \in G_a$. Hence, we get the following inequality:

$$p_i(a, v_{-i}) - p_i(b, v_{-i}) \leq \inf_{v_i \in G_a} \{v_i(a) - v_i(b)\} \quad (1)$$

Similarly, considering cases where $f(v_i, v_{-i}) = b$, we get:

$$p_i(b, v_{-i}) - p_i(a, v_{-i}) \leq \inf_{v_i \in G_b} \{v_i(b) - v_i(a)\} \quad (2)$$

Combining the two, for each pair $< a, b >$, we have:

$$\sup_{v_i \in G_b} \{v_i(a) - v_i(b)\} \leq p_i(a, v_{-i}) - p_i(b, v_{-i}) \quad (3)$$

$$\leq \inf_{v_i \in G_a} \{v_i(a) - v_i(b)\} \quad (4)$$

This is well known, see for instance Gui et al. [11] and Lavi et al. [14], and suggests the following simple verification procedure.

Let H denote the *history* of instances currently available to the verifier. Define the following: $v \in v^H$ denotes bids $v \in V^N$ received so far; $v_i \in v^H_i(v_{-i})$ denotes the bids from agent i that have been observed for v_{-i} from the other agents; $v_i \in v^H_i(a, v_{-i})$ denote the bids received from agent i with the additional restriction that alternative a (or an equivalent) was selected; $a = f^H(v)$ denotes the alternative selected given input v (it is sufficient to choose any one of a set of equivalent alternatives); $f^H(v_{-i}) \subseteq A$ denotes the set of alternatives selected for v_{-i}, and $p^H_i(a, v_{-i})$ denotes the payment made by agent i given alternative a and v_{-i}. For every new instance we first check whether the exact same value profile has been seen before. If this is the case then the same alternative must be selected. By case (b), if a^k (or equivalent) has been seen before for v^k_{-i} then by (A1) the payment by i must be the same. Case (c) ensures that *previous agents could not have done better* given this new

information: if a^k is a new alternative for v^k_{-i} then the price on this alternative must be high enough for (A2), and thus for inequality (3) to hold for bids that have already been observed. Case (d) ensures that the *current agent could not have done better given the current history*: if v^k_i is a new bid for v^k_{-i} then the price on this alternative must be low enough for (A2), and thus for inequality (4) to hold for other alternatives that have been observed.

EXAMPLE 1. *Consider an auction mechanism for an allocation problem with multiple identical items. Suppose that the sequence of instances in Table 1 are observed (all with two bidders and two items). Instance 1 indicates that bidder 1 has value 4 for 1 unit and 8 for 2 units. Bidder 2 has value 4 for 1 unit and 9 for 2 units. The allocation gives both units to bidder 2, and bidder 2 makes payment 8. The auction implements the VCG mechanism for instances 1–3 but deviates in instance 4 (The VCG mechanism would choose the same allocation but with payments (0, 9)).*

Consider the build-up of history for $v_{-i} = (4, 9)$, and consider allocations in which bidder i receives 0, 1 or 2 items. Let $p_i(n)$ denote the price that agent i faces when it wins n items. After instance 1, we get $p_i(0) = 0$. After the second instance, the verifier learns that $p_i(1) = 5$, and checks that the constraints (c) $p_i(1) \geq p_i(0) - (v^1_i(0) - v^1_i(1)) = 4$ and (d) $p_i(1) \leq p_i(0) + v^2_i(1) - v^2_i(0) = 5$ are satisfied. After instance 3, agent i wins one item and pays the price of 5, so the verifier checks that (b) $p^3_i(1) = p_i(1) = 5$ holds. After instance 4, the verifier learns $p_i(2) = 10$, and checks that (c) $p_i(2) \geq \max\{p_i(0) - (v^1_i(0) - v^1_i(2)), p_i(1) - (v^2_i(1) - v^2_i(2))\} = 9$. However, the verifier also checks for the constraint (d) $p_i(2) \leq p_i(1) + v^4_i(2) - v^4_i(1) = 9$, which is violated (the price is too high). Hence, this mechanism is rejected by SIMPLECHECKER *after the fourth instance.*

Alternatively, suppose that the fourth instance is changed to $v^4 = ((7, 9), (4, 9)), a^4 = (2, 0), p^4 = (8, 0)$, where the mechanism still deviates from the VCG outcome ($a^4_{VCG} = (1, 1)$ and $p^4_{VCG} = (5, 2)$). This time, after instance 4, the verifier learns $p_i(2) = 8$, and checks that (c) $p_i(2) \geq \max\{p_i(0) - (v^1_i(0) - v^1_i(2)), p_i(1) - (v^2_i(1) - v^2_i(2))\} = 9$. Now this constraint is violated (the price too low), and the mechanism is rejected after the fourth instance.

instance	values	allocation	payments
1	((4, 8), (4, 9))	(0, 2)	(0, 8)
2	((4, 9), (5, 9))	(1, 1)	(4, 5)
3	((5, 8), (4, 9)	(1, 1)	(5, 4)
4	((4, 9), (5.5, 10))	(0, 2)	(0, 10)

Table 1: Sequence of Instances: 2 Agents and 2 Identical Items

3.2 Establishing Soundness and Correctness

The first task is to establish soundness and correctness of the verifier. We hold these properties to be necessary for an (exact) verifier, although insufficient to show the utility of passive verification. Soundness simply proves that SIMPLECHECKER will detect the failure of strategyproofness eventually, once all inputs are observed. The main challenge is to provide intermediate feedback, discussion of which is delayed until Section 5.

THEOREM 2 (SOUNDNESS). *The rules as defined by* SIMPLECHECKER *will detect a non-strategyproof mechanism even if all agents are not truthful as long as all inputs are eventually observed.*

PROOF. A mechanism is not strategyproof if either:

¬ (A1) there exists some i, v_{-i} and $v_i, v'_i \in V_i$ such that $f(v_i, v_{-i}) = f(v'_i, v_{-i})$, but $\tilde{p}_i(v_i, v_{-i}) \neq \tilde{p}_i(v'_i, v_{-i})$, or

¬ (A2) There exists some i, v_{-i} and $v_i, v'_i \in V_i$, $v_i \neq v'_i$ such that $v_i(f(v_i, v_{-i})) - p_i(f(v_i, v_{-i}), v_{-i}) < v_i(f(v'_i, v_{-i})) - p_i(f(v'_i, v_{-i}), v_{-i})$.

Suppose toward contradiction that a non-strategyproof mechanism passes the SIMPLECHECKER after all possible reports $v \in V$ are observed. First, suppose ¬ (A1). This is not possible because check (b) of SIMPLECHECKER would catch the price deviation. Second, suppose ¬ (A2), such that there exists some i, v_{-i} and $v_i, v'_i \in V_i$ such that

$$v_i(f(v_i, v_{-i})) - p_i(f(v_i, v_{-i}), v_{-i}) < v_i(f(v'_i, v_{-i})) - p_i(f(v'_i, v_{-i}), v_{-i}) \quad (7)$$

Let $v_i = v^k_i$ and $v'_i = v^l_i$. If $k > l$, then when instance k is observed, $a^l \in f^H(v_{-i})$. By (7),

$$p^k_i(a^k, v_{-i}) > p^l_i(a^l, v_{-i}) + v^k_i(a^k) - v^k_i(a^l) \geq \inf_{b \in f^H(v^k_{-i})} [p^H_i(b, v^k_{-i}) + (v^k_i(a^k) - v^k_i(b))],$$

and step (d) of SIMPLECHECKER would fail. A contradiction. If $l > k$, then when instance l is observed, $a^k \in f^H(v_{-i})$. By (7), we have $p^l_i(a^l, v_{-i}) <$

$$p^k_i(a^k, v_{-i}) + v^k_i(a^k) - v^k_i(a^l) \leq p^k_i(a^k, v_{-i}) + \sup_{v_i \in v^H_i(a^k, v_{-i})} (v_i(a^k) - v_i(a^l)) \leq \sup_{b \in f^H(v_{-i})} [p^H_i(b, v_{-i}) + \sup_{v_i \in v^H_i(b, v_{-i})} (v_i(a^l) - v_i(b))],$$

and step (c) would fail. A contradiction. □

THEOREM 3 (CORRECTNESS). *The rules as defined by* SIMPLECHECKER *will never halt a strategyproof mechanism even for agents that are untruthful.*

PROOF. Suppose toward contradiction that a strategyproof mechanism M is rejected by SIMPLECHECKER after k instances. Since M satisfies (A1), it will not fail step (b). So it either fails (c) or fails (d). Suppose it fails (c). Then there exists $a^l \in f^H(v^k_{-i})$, $p^k_i(a^k, v^k_{-i}) < p^l_i(a^l, v^k_{-i}) + (v^l_i(a^k) - v^l_i(a^l))$ for some $l < k$. Then $v^l_i(a^k) - p^k_i(a^k, v^k_{-i}) > v^l_i(a^l) - p^l_i(a^l, v^k_{-i})$, violating (A2) and a contradiction. Now, suppose the mechanism fails (d). Then there exists $a^l \in f^H(v^k_{-i})$, $p^k_i(a^k, v^k_{-i}) > p^l_i(a^l, v^k_{-i}) + (v^k_i(a^k) - v^k_i(a^l))$ for some $l < k$. Then $v^l_i(a^k) - p^k_i(a^k, v^k_{-i}) < v^l_i(a^l) - p^l_i(a^l, v^k_{-i})$, violating (A2) and a contradiction. □

3.3 Passive Verification via Constraint Networks

An algorithm for SIMPLECHECKER can be identified by reformulating the task as that of checking for a feasible solution to a constraint network.

We gain two main advantages from formulating the problem in terms of constraint networks. First, we can leverage

standard data structures (e.g. linked-lists) and standard algorithms (e.g. all-pairs shortest path algorithms) to construct a passive verifier. Second, we can gain additional accelerations by introducing constraints *between* networks to capture additional problem structure (see Section 4.2).

For each v_{-i}, we construct a constraint network in which each node in the network is associated with an alternative and arcs in the network impose constraints on the difference in prices between pairs of alternatives. Each network contains a single node for each set of equivalent alternatives; e.g. one node for all allocations in which agent i receives bundle of goods S, irrespective of the allocation to other agents. In addition to binary constraints, each node is also annotated with an exact price, once known.[3]

Recall that every strategyproof mechanism must satisfy inequalities (3) and (4). In order to capture this requirement for some pair of alternatives $< a, b >$ given current history H, a directed arc $b \rightarrow a$ is labeled with a weight $w(b, a)$ representing the linear inequality $p_i(a, v_{-i}) - p_i(b, v_{-i}) \leq w(b, a)$, with $w(b, a) = \inf_{v_i \in v_i^H(a, v_{-i})} \{v_i(a) - v_i(b)\}$. Similarly, a directed arc $a \rightarrow b$ is created and labeled with weight $w(a, b)$. In addition, each node that represents an observed alternative can be annotated with the price.

As described, our constraint network is of the same form as a Simple Temporal Problem (STPs)(Dechter et al. [6]). Prices take the role of time and the weights on arcs are now bounds on the difference between prices. The existence of prices that form a feasible solution to this network is equivalent to the nonexistence of negative-length cycles in the constraint network.

THEOREM 4. *[6] A given STP T is consistent if and only if it has no negative-length cycles.*

Thus, we can check for the existence of prices satisfying all constraints by checking for the existence of negative-length cycles using an all-pairs-shortest-path algorithm. We also refer to this process as *tightening* the network.

Let $\mathcal{N}$ denote the current set of networks (one for each v_{-i} that has been observed). New networks are introduced dynamically (for new v_{-i} sets) and a new node is introduced on a network when an additional alternative is observed.

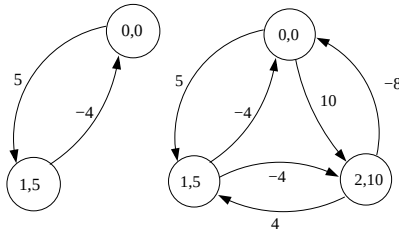
Algorithm NETWORKCHECKER implements the sound and correct rules defined in SIMPLECHECKER. If a set of feasible prices exist, by step (c) and step (e) of NETWORKCHECKER, $p_i^a - p_i^k \leq w(a^k, a) = \inf_{v_i \in v_i^H(a, v_{-i})}(v_i(a) - v_i(a^k))$ or $p_i^k - p_i^a \geq \sup_{v_i \in v_i^H(a, v_{-i})} (v_i(a^k) - v_i(a))$ for all existing nodes a. Rewriting, we get $p_i^k \geq p_i^a + \sup_{v_i \in v_i^H(a, v_{-i})}(v_i(a^k) - v_i(a))$ $\forall a \in H(v_{-i}^k)$, which is precisely the condition that is checked in step (c) of the SIMPLECHECKER. Now consider step (d), in combination with step (e), of NETWORKCHECKER: if an arc from node a to node a^k already exists, and the arc has weight $w < v_i^k(a^k) - v_i^k(a)$, then we know that $p_i^{a^k} - p_i^a \leq w$ even before observing instance k, and no constraints are updated. Otherwise, we have a new (or tighter) constraint on the set of feasible prices, namely, $p_i^k - p_i^a \leq w(a, a^k) = v_i^k(a^k) - v_i^k(a)$ for all a such that $w(a, a^k)$ is updated after observing instance k. $p_i^k \leq p_i^a + (v_i^k(a^k) - v_i^k(a))$ for all a such that $w(a, a^k)$ is updated. Combining the two cases, we get $p_i^k \leq v_(^k a^k) + \inf_{a \in f^H(v_{-i}^k)}[p_i^a - v_i^k(a)]$, which is the condition checked in step (d) of the SIMPLECHECKER.

[3]Compared with the directed graph formalism introduced in Gui et al. [11] for reasoning about strategyproof mechanisms, the constraint network of a passive verifier is typically incomplete (only containing a subset of alternatives) and associates fixed prices with alternatives that have been observed. These prices imply constraints on future prices. In comparison, Gui et al. use the constraint network to reason (in analysis) about whether *any* assignment of prices is possible.

NETWORKCHECKER. Initialize $\mathcal{N} = \emptyset$.

1. For a new instance $< v^k, a^k, p^k >$, and for each agent i, if there is a constraint graph $G \in \mathcal{N}$ for v_{-i}^k work with this graph. Otherwise, create an empty graph G and add to set $\mathcal{N}$. Then:
   (a) if $v_i^k \in v_i^H(v_{-i}^k)$ check $a^k \in E_i(f^H(v^k))$.
   (b) if node a^k (or equivalent[4]) exists in graph G then check p_i^k equals price on node.
   (c) Otherwise, add node a^k and assign price p_i^k to the node, and for every other node a in the graph: add a new arc from the new node a^k to a with weight $w = \inf_{v_i \in v_i^H(a, v_{-i})}(v_i(a) - v_i(a^k))$
   (d) add a new arc from every other node a to the new node a^k with weight $v_i^k(a^k) - v_i^k(a)$ if no arc exists. If one already exists, and $v_i^k(a^k) - v_i^k(a)$ is less than the current weight then update the weight to be $v_i^k(a^k) - v_i^k(a)$.
   (e) tighten the network and check for feasibility.
2. If any check fails, then **reject** the mechanism, else update the history as necessary.
3. Once all $v_i \in V_i$ for all $v_{-i} \in V_{-i}$ have been observed then **pass** the mechanism.

EXAMPLE 2. *We can revisit the example in Table 1. Figure 1 illustrates the constraint network for $v_{-i} = (4, 9)$. After instance 4, the arc from node $< 1, 5 >$ to node $< 2, 10 >$ has weight 4, the arc from node $< 2, 10 >$ to node $< 1, 5 >$ has weight -4, the arc from node $< 0, 0 >$ to node $< 2, 10 >$ has weight 10, and the arc from node $< 2, 10 >$ to node $< 0, 0 >$ has weight -8. Given these weights, we see that the prices $p(0) = 0$, $p(1) = 5$, and $p(2) = 10$ are inconsistent, since $p(1) - p(2) = -5 < -4$.*



(a) After instance 2 (b) After instance 4

Figure 1: Constraint network for Multiple Identical Items with Two Bidders and Two items.

4. ACCELERATED VERIFICATION VIA STRUCTURAL REQUIREMENTS

The biggest potential shortcoming of this approach to verification is that the space complexity quickly becomes untenable: there are d^{n-1} subnetworks, for a type space of size d and n agents, and thus the space complexity is exponential in the number of agents.

In order to address this problem we propose that the passive verifier impose *structural requirements*. We consider three kinds of structural requirements: *(a) summarization, (b) natural payments, (c) envy-freeness.* These are illustrative of a more general approach which is to restrict the space of implementable mechanisms in order to allow for efficient verification. More than just reducing the space requirements for verification, these structural requirements also *accelerate* verification by allowing for stronger inference and more rapid proofs of non-strategyproofness.

A strategyproof mechanism may wish to volunteer structural requirements in order to facilitate faster verification. But, given that the concern in verification is to identify the "bad apples" then it is in general more appropriate for the verifier to require certain additional properties, especially properties that appear to hold for many plausible mechanisms. We will comment on the restrictiveness of each property as it is introduced.

4.1 Summarization

Summarization provides an exponential reduction in memory and computational requirements on the verifier and also significantly accelerates the process of verification.

DEFINITION 3 (VALID SUMMARIZATION FUNCTION). *Given reports v_{-i}, a summarization function s selects a subset $s(v_{-i})$ of reports, and is valid when $p_i(a, v_{-i}) = p'_i(a, s(v_{-i}))$ for all a, all v_{-i}, for some price function p'_i.*

EXAMPLE 3. *In a single-item Vickrey auction the price that an agent faces is determined by the highest bid among bids from other agents and $s(v_{-i}) = \{\max_{j \neq i}\{v_j\}\}$ is a valid summarization function. In this case, the induced price rule $p'_i(i \text{ wins item}, s(v_{-i})) = x$ given $s(v_{-i}) = x$ and $p'_i(i \text{ does not win item}, s(v_{-i})) = 0$.*

EXAMPLE 4. *Consider a combinatorial auction with single-minded bidders. A bidder is single-minded if there exists a set g of goods and value b such that $v(g') = b$ if $g \subseteq g'$, and $v(g') = 0$ otherwise. The* LOS mechanism*[15] consists of the greedy allocation rule and the greedy payment rule. The greedy allocation rule works as follows: bids are sorted in decreasing order according to the average amount*[5] *of a bid $v_j =< g_j, b_j >$, defined as $\frac{b_j}{|g_j|}$, where $|g_j|$ is the number of goods in the desired set g_j of agent j, and b_j is her value for g_j. Then given this sorted list L, each bid is examined in order and is granted if and only if it does not conflict with any of the bids previously granted. For each $v_j \in L$, define the set $D_j = \{i | i > j, g_i \cap g_j \neq \emptyset, \forall l < i, l \neq j, l'\text{th bid granted} \Rightarrow g_l \cap g_i = \emptyset\}$. Note that $i > j \Leftrightarrow \frac{b_i}{|g_i|} \leq \frac{b_j}{|g_j|}$. D_j is the set of indices of the bids that are denied, but would have been granted if it were not for the presence of v_j. Under the greedy payment rule, if agent j's bid is granted and $D_j \neq \emptyset$, j pays $|g_j|\frac{b_i}{|g_i|}$ where $i = \min D_j$. In other words, j pays the average amount of the first bidder whose bid was denied due to j, per good won. Otherwise, j pays 0. In the LOS mechanism, the summarization function $s(v_{-i}) = \{v_j | \forall k < j, k \neq i, g_k \cap g_j = \emptyset\}$. If VCG payments are used instead, the set $s(v_{-i})$ is given by first sorting the set v_{-i} according to the norm used in the LOS allocation rule and then removing each bid whose set of desired items is a superset of the set of desired items of any bid that appears before it in the ordered list.*

[5]The average amount can be replaced with any norm that satisfied bid-monotonicity, i.e., is nondecreasing in b_i and $norm(< g, b >) \geq norm(< g', b >)\ \forall g \subseteq g'$.

Summarization information can be defined by a mechanism incrementally, by identifying a subset of values v_{-i} that define the prices to agent i for each instance. In this case, this is done by extending the interface between the verifier and the mechanism. Summarization functions can also be defined statically by instantiating an explicit function. As long as the summarization function is *constant*, then passive verification remains sound and correct.

THEOREM 5. *A passive verifier that implements rules* SIMPLECHECKER *will continue to detect a non-strategyproof mechanism and continue to pass any strategyproof mechanism when used in combination with a fixed summarization function.*

PROOF. See appendix. □

The following example shows that summarization can facilitate faster identification of a non-strategyproof mechanism.

EXAMPLE 5. *Consider an auction for multiple identical items, and suppose three instances are observed for the case of three bidders and two items. The auction implements a VCG mechanism for the first two instances but deviates in instance 3. Now, consider the summarization function $s(v_{-i}) = \{v_j | \forall j \neq i, j \text{ wins at least 1 item in the allocation without agent } i\}$. Note that $s(v^1_{-1}) = s(v^3_{-1}) = s(v^2_{-3}) = s(v^3_{-3}) = \{(4, 9)\}$. Consider the constraints when $s(v_{-i}) = \{(4, 9)\}$. Following the notation from Example 1, after instance 1, we get $p_i(0) = 0$. After the second instance, the verifier learns that $p_i(1) = 5$, and checks that the constraints (c) $p_i(1) \geq p_i(0) - (v^1_i(0) - v^1_i(1)) = 4$ and (d) $p_i(1) \leq p_i(0) + v^2_i(1) - v^2_i(0) = 5$ are satisfied. After instance 3, agent i wins one item and pays the price of 5, so the verifier checks that (b) $p^3_i(1) = p_i(1) = 5$ holds. After instance 3, the verifier also learns $p_i(2) = 10$, and checks that (c) $p_i(2) \geq \max\{p_i(0) - (v^1_i(0) - v^1_i(2)), p_i(1) - (v^2_i(1) - v^2_i(2))\} = 9$. However, the verifier also checks for the constraint (d) $p_i(2) \leq p_i(1) + v^3_i(2) - v^3_i(1) = 9$, which is violated (the price is too high). Hence, this mechanism is rejected.*

instance	values	allocation	payments
1	((4, 8), (4, 9), (3, 6))	(0, 2, 0)	(0, 8, 0)
2	((0, 8), (4, 9), (5, 9))	(0, 1, 1)	(0, 3, 5)
3	((4, 9), (2, 8), (6, 10))	(0, 0, 2)	(0, 0, 10)

Table 2: Sequence of Instances: 3 Agents and 2 Identical Items

Without summarization none of the v_{-i} subnetworks have more than one node and the mechanism would not have been rejected until more instances were seen.

Summarization also provides an exponential improvement in space complexity. Let $d = max_i|V_i|$, $m = |A|$ and let n denote the maximal number of agents. Without summarization, we need d^{n-1} subnetworks, each with at most m nodes and thus $O(md^{n-1})$ nodes.

THEOREM 6. *With summarization there are $O(md^w)$ nodes across all subnetworks, where w is the maximal number of agents required in a summarization, i.e. $w = \max_{i,v_i} |s(v_{-i})|$.*

Thus, the space complexity is exponential in the worst-case number of agents required for summarization instead of the worst-case number of agents.

EXAMPLE 6. *Consider the single item Vickrey auction with 4 agents, where $|V_i| = 10$. In this case, only the highest bid among bids of other agents matters. Without summarization, we would need to keep 1000 subnetworks, while with summarization, we only need to keep 10 subnetworks.*

4.2 Natural Payment Functions

So far we have focused on the constraints within a single v_{-i} subnetwork. The additional structure provided by *natural payments*, allows internetwork constraints which improve the speed with which a mechanism can be validated as strategyproof, or proved not to be strategyproof.

Consider again inequalities (3) and (4). Following Lavi et al. [14], define $\delta_{ab}(v_{-i}) = \inf\{v_i(a) - v_i(b) | v_i \in V_i, a \in E_i(f(v_i, v_{-i}))\}$. Then, we have:

$$-\delta_{ba}(v_{-i}) \leq p_i(a, v_{-i}) - p_i(b, v_{-i}) \leq \delta_{ab}(v_{-i}) \qquad (8)$$

DEFINITION 4 (NATURAL PAYMENT FUNCTIONS). *A mechanism has a natural payment function if the agent-independent price function satisfies*

$$p_i(a, v_{-i}) - p_i(b, v_{-i}) = \delta_{ab}(v_{-i}) = -\delta_{ba}(v_{-i}), \qquad (9)$$

for all pairs $< a, b >$.

The natural payment function is often satisfied by known mechanisms, for instance, it is typically satisfied by the VCG mechanism (this depends on the value domain) and the LOS mechanism. Natural payments permit the introduction of internetwork constraints, as demonstrated by the following lemma.

LEMMA 1. *Consider a strategyproof mechanism with natural payment functions. Suppose for an alternative a, v_{-i} and v'_{-i} are such that $\forall j \neq i$, either $v'_j = v_j$ or $v'_j(a) - v'_j(b) > v_j(a) - v_j(b)$ for all $b \neq a$. Then $p_i(a, v'_{-i}) - p_i(b, v'_{-i}) \leq p_i(a, v_{-i}) - p_i(b, v_{-i})$, for all $b \neq a$.*

PROOF. See appendix. □

Thus, given a constraint in the subnetwork for v_{-i} and some alternate reports v'_{-i} by all agents except one that satisfies the condition of Lemma 1, then we can add a corresponding constraint to the subnetwork for v'_{-i}, or vice versa. The additional constraint provides the following kind of inference: given $p_i(a, v_{-i}) \in [c, d]$ for some constants c and d, then $p_i(a, v'_{-i}) \leq d$. To illustrate the condition of Lemma 1, consider the case of single-minded bidders. Here, the condition is satisfied if each agent $j \neq i$ bids for the same bundle in v'_{-i} as in v_{-i}, and all agents $j \neq i$ who win in v_{-i} submit the same or larger value in v' while losing agents do not increase their bid in v'.

Modified NETWORKCHECKER. Add a new step (d1) between steps (d) and (e):

(d1) For each v_{-i} such that $G(v_{-i}) \in N$ and $a^k \in f^H(v'_{-i})$:

- if $\mathbb{I}(a^k, v^k_{-i}, v_{-i}) = 1$, then
  - In $G(v^k_{-i})$: update the weight on the arc from a^k to each $a \in f^H(v^k_{-i}) \cap f^H(v_{-i})$ to be $\min\{w^{v^k_{-i}}(a^k, a), w^{v_{-i}}(a^k, a)\}$.
  - In $G(v_{-i})$: update the weight on the arc from $a \in f^H(v^k_{-i}) \cap f^H(v_{-i})$ to a^k to be $\min\{w^{v^k_{-i}}(a, a^k), w^{v_{-i}}(a, a^k)\}$.
- If $\mathbb{I}(a^k, v_{-i}, v^k_{-i}) = 1$, then reverse the role of v^k_{-i} and v_{-i}, and repeat previous step.

EXAMPLE 7. *Consider a strategyproof mechanism with 2 agents and two items s and t, and suppose $a = \{$agent 1 wins s, agent 2 wins t $\}$. If v'_2 is such that agent 2 values alternative a higher relative to all other alternatives, then if the alternative is a for both v and v', agent 1 must not pay a higher price under v'.*

We can implement the internetwork constraints as follows. First, define the indicator function:

$$\mathbb{I}(a, v_{-i}, v'_{-i}) = \begin{cases} 1, & \text{if } v_j = v'_j \text{ or} \\ & v'_j(a) - v'_j(b) > v_j(a) - v_j(b)\ \forall b \neq a; \\ 0, & \text{otherwise.} \end{cases}$$

Then, for each alternative a and valuations v_i, v_{-i} such that $\mathbb{I}(a, v_{-i}, v'_{-i}) = 1$, we add the following constraints: $p_i(a, v'_{-i}) - p_i(b, v'_{-i}) \leq \max_{p_i \in \mathbb{F}_i(v_{-i})} [p_i(a, v_{-i}) - p_i(b, v_{-i})]$ to subnetwork v'_{-i} and $p_i(a, v_{-i}) - p_i(b, v_{-i}) \geq \min_{p_i \in \mathbb{F}_i(v'_{-i})} [p_i(a, v'_{-i}) - p_i(b, v'_{-i})]$ to subnetwork v_{-i}, where $\mathbb{F}_i(v_{-i})$ denotes the space of feasible prices defined by the constraint network. Note that $\max_{p_i \in \mathbb{F}_i(v_{-i})} [p_i(a, v_{-i}) - p_i(b, v_{-i})] = w^{v_{-i}}(b, a)$ and $\min_{p_i \in \mathbb{F}_i(v'_{-i})} [p_i(a, v'_{-i}) - p_i(b, v'_{-i})] = -w^{v'_{-i}}(a, b)$, where $w^{(v^{-i})}(a, b)$ denotes the weight on the arc from a to b in subnetwork v^{-i}.

4.3 Envy-freeness

Our third example to illustrate the power of structural requirements is the property of *envy-freeness* [9].

Let $f_i(v)$ denote the allocation of goods to agent i defined by social choice function f.

DEFINITION 5. *Mechanism $M =< f, \tilde{p} >$ is envy-free if $v_i(f_i(v)) - \tilde{p}_i(v) \geq v_i(f_j(v)) - \tilde{p}_j(v), \forall i, \forall j, \forall v \in V$.*

This property is easily verified by checking that there is no agent that prefers an outcome-price pair that was received by another agent. Thus, this can be checked by adding an additional set of internetwork constraints.

Many strategyproof mechanisms are also envy-free. For example, VCG mechanisms in domains with superadditive valuations satisfy the property of envy-freeness [19]. As another example, it is easy to show that envy-freeness holds for any strategyproof mechanism in the domain of single-minded combinatorial auctions that is *fair*, which requires

that for any set w and any two agents i and j such that the bids are $b_i = (w, v_i)$ and $b_j = (w, v_j)$ where $v_i > v_j$, then agent j should not win. The LOS family of mechanisms satisfy fairness and therefore all LOS mechanisms are envy-free.

5. PROVIDING INTERMEDIATE FEEDBACK TO PARTICIPANTS

We describe in this section some intermediate feedback that can be provided to participants to guide their behavior even before a mechanism's strategyproofness (or lack thereof) is completely established. This is the main technical achievement of our work.

5.1 Partial Strategyproofness

First, recall that the verifier is able to check the outcome of a mechanism *before* the outcome is implemented and before payments are collected from agents and *veto* the result if the input-output instance provides proof that the mechanism is not strategyproof. This *veto* power is important in establishing the following useful result.

THEOREM 7. *Given history H, and consider an instance in which the reports of agents except i are $\hat{v}_{-i}$ and instance $(v_i, \hat{v}_{-i})$ has been observed by the mechanism for agent i's true valuation v_i. In this case, and conditioned on the mechanism passing the verifier in this instance, then agent i's best-response is to report her true valuation.*

PROOF. Fix $\hat{v}_{-i}$. Condition on the case that all checks in the verifier pass. First, if the agent reports v_i then the mechanism must make the same decision as previously, by checks (a) and (b) in SIMPLECHECKER. Now suppose the agent reports some $v_i' \neq v_i$. We argue that this cannot improve the agent's utility. Case 1. The mechanism selects an alternative a' already observed. Now we argue that the agent must actually prefer the outcome, a, that would have been selected given truthful report v_i. Either a' was observed after a in which case check (c) would have ensured that a was preferred to a' by an agent with type v_i, or a was observed after a' and check (d) would have ensured that a was preferred to a' by an agent with type v'. Case 2. The mechanism selects a new alternative, not previously observed for $\hat{v}_{-i}$. In this case, because v_i was already observed (with outcome a) then to pass check (c) it must be the case that outcome a would preferred by an agent with true type v_i. This concludes the proof. □

Note that the above theorem continues to hold even if the mechanism is ultimately not strategyproof: it is still rational for an agent to bid her true type if v_i and $\hat{v}_{-i}$ (reports by the other agents) have already been observed. Moreover, the observation leads to the following two corollaries. Here, when truthful reporting is an *ex post* Nash equilibrium given knowledge $\tilde{V} \subset V$ about joint types, then every agent must maximize its utility by reporting its true type in equilibrium as long as every other agent is truthful and whatever the actual joint valuation $v \in \tilde{V}$.

COROLLARY 1. *Given history H, let $\tilde{V} \subset V$ denote some subspace of joint type space V for which all joint inputs $v \in V$ have been observed. Given knowledge that $v \in \tilde{V}$, and conditioned on the mechanism passing the verifier in this instance, then truthful reporting is an ex post Nash equilibrium.*

COROLLARY 2. *Given history H, and considering agent i with true type v_i, then if (v_i, v_{-i}) has been observed by the verifier for all v_{-i}, then conditioned on the mechanism passing the verifier in this instance truthful bidding is a dominant strategy for bidder i.*

We see that there are reasonable conditions under which a partially informed agent (e.g. with information about the space of possible types of other agents) should report its type truthfully even before the strategyproofness of a mechanism is fully verified. Moreover, these observations suggest that it will be useful for a verifier to *publish* the set of types for which the mechanism is provably strategyproof. Note also that these results can all be rephrased in terms of *summarizations* of reports from other agents when summarization is used.

A somewhat weaker guarantee is also available to a bidder in the case that report $\hat{v}_{-i}$ has been observed but not in combination with the agent's type v_i. This guarantee is provided in the context of an *adversarial* mechanism, which *seeks to minimize the utility to agent i while passing the verifier in this instance.* The earlier guarantees are provided for any mechanism.

THEOREM 8. *Given history H, then agent i maximizes her worst-case utility against an adversarial mechanism by reporting truthfully whenever report $\hat{v}_{-i}$ has been observed, contingent on the mechanism passing the verifier in this instance.*

PROOF. Worst-case utility refers to the minimum utility achieved over all possible choices that can be made by the mechanism in responding to the input, while still passing the verifier. Fix v_i and $\hat{v}_{-i} \in H$. First suppose the agent is truthful. In this case the best the adversary can do, in terms of minimizing the utility to the agent, and still pass the checks (e.g. in SIMPLECHECKER) is to select the preferred alternative for the agent in $a \in f^H(\hat{v}_{-i})$. Choosing some less-preferred alternative in $f^H(\hat{v}_{-i})$, or some less-preferred alternative outside of $f^H(\hat{v}_{-i})$ would violate check (d) in SIMPLECHECKER. Now suppose the agent is untruthful and reports $v_i' \neq v_i$. In this case, the mechanism can always choose one of the current alternatives and pass since check (c) would not be triggered and check (d) is satisfied by choosing $a' \in f^H(\hat{v}_{-i})$, the most-preferred alternative with respect to v_i'. Note that the alternative selected when reporting v_i is at least this good. Also, if the mechanism chooses to select some new alternative then this must necessarily be even worse for the agent (since the mechanism is adversarial). This completes the proof. □

5.2 Metrics for Partial Verification

We introduce two quantitative metrics for the degree to which a mechanism's strategyproofness has been partially verified. First, Corollary 1 can be used to define a lower-bound on the probability that truthful reporting is an *ex post* Nash equilibrium, given history H and given a distribution on types of bidders.

COROLLARY 3. *Given history H and probability distribution g on agent types, then truthful bidding is an ex post Nash equilibrium conditioned on the mechanism passing the verifier in this instance, with probability:*

$$Pr(SP|H) = \sum_{v \in V^N} g(v) \mathbb{I}^H(v) \tag{10}$$

where $g(v)$ is the probability that agents have types v and $\mathbb{I}^H(v)$ is an indicator function, and equal to 1 if and only if $v \in v^H$.

We refer to this metric as the *probability-of-strategyproofness.* In combination with summarization, this becomes:

$$Pr(SP|H) = \sum_{v \in V^N} g(v)\mathbb{I}^H(s(v_{-1}), \ldots, s(v_{-N})) \qquad (11)$$

where $\mathbb{I}^H(s(v_{-1}), \ldots, s(v_{-N}))$ is an indicator function, and equal to 1 if and only if summarizations $s(v_{-i})$ for all i have been observed by the mechanism given history H. A probability-of-strategyproofness metric can also be defined for a single agent i, where the average is computed with respect to the marginal distribution on the reports of agents $\neq i$, given agent i's type v_i.

A complementary measure is provided by the *price-flexibility* metric. This metric takes into account the range of prices still available to a mechanism in setting prices on alternatives not yet observed. Let $Feas^H(v) \subseteq A$ denote the set of alternatives that can be selected without failing the verifier given input v and history H. If v has been observed this is the singleton containing the alternative $f^H(v)$ that was previously selected. But, even when v has not been observed we have restrictions on alternatives that can be selected. There can be alternatives for which, if selected, there is no price that can be assigned to satisfy checks (c) and (d) in the verifier for all agents.

For $a \in Feas^H(v)$, we define the price flexibility on a, denoted $flex^H(v, a)$ as:

$$v_i(a) + \inf_{b \in f^H(v_{-i})} \{p_i^H(b, v_{-i}) - v_i(b)\} - \left[\sup_{b \in f^H(v_{-i})} \left\{p_i^H(b, v_{-i}) + \sup_{v_i' \in v_i^H(b, v_{-i})} (v_i'(a) - v_i'(b))\right\}\right] \qquad (12)$$

This represents the range of prices available to the mechanism without violating the checks in the verifier, and is non-empty since a is in the feasibility set. Now, for valuation profile $v = (v_1, \ldots, v_N)$, define the *price-flexibility*, as:

$$PF^H(v) = \sum_{v \in V^N} g(v) \left[\frac{1}{N^H(v)} \sum_{a \in Feas^H(v)} flex^H(v, a)\right], \qquad (13)$$

where $N^H(v) = |Feas^H(v)|$. Notice that if all inputs v_{-i} have been observed for v_i then $PF^H(v_i) = 0$. This is as we would expect: a fully-verified mechanism has no flexibility and must continually make the same decisions and assign the same prices as in the past.

6. EXPERIMENTS

We present experimental results to demonstrate that the verifier can impose useful restrictions on the mechanism design space (via summarization, natural payments, and envy-freeness) and increase the confidence in truthful mechanisms and accelerate the detection of failure in manipulable mechanisms.

For illustrative purposes, the experimental results are presented for a mixture of known and invented mechanisms. We consider both Vickrey and first-price auctions for the allocation of multiple identical items. In addition, we consider single-minded bidders in a combinatorial auction (with multiple non-identical items) and the LOS mechanism [15] as well as the LOS greedy allocation rule in combination with VCG payments (this mechanism is called *greedyVCG*, and is manipulable). We also find it useful to construct an artificial, manipulable mechanism for single-minded bidders using local search techniques combined with VCG payments. We call this mechanism *localVCG*.

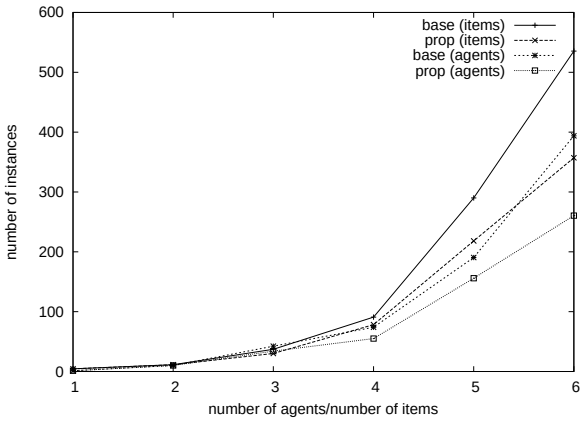


Figure 2: Number of instances before failure in a first-price multi-item auction.

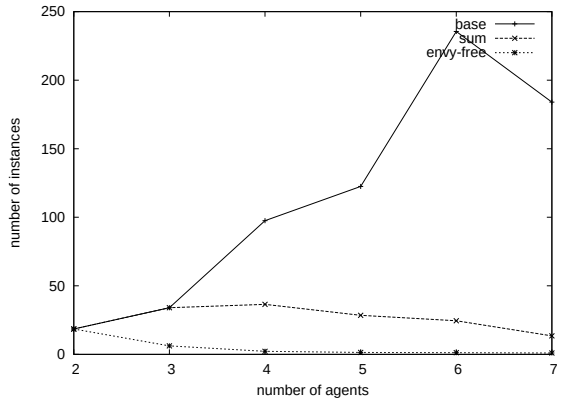
To provide a baseline we compare with an algorithm that simply checks for violations of (A1). This algorithm is sound but not complete. Actually, it completely fails to reject the greedyVCG mechanism, which only makes errors of type (A2). On the other hand, it is sound against first-price auctions, which only violate (A1). The localVCG mechanism is useful because it violates both (A1) and (A2), permitting a comparison between the baseline and our verification methods.

6.1 Multi-item Auctions

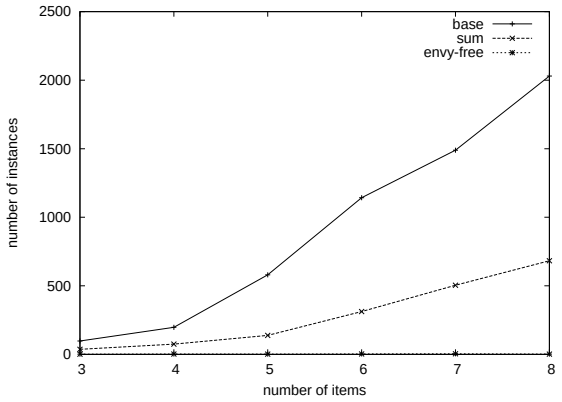
We consider a first-price multi-item auction with g identical goods and construct agent valuations by defining the marginal value of the kth item for agent i as independently drawn from a uniform distribution over $\{0, 1, \ldots, z-1\}$. Here, z denotes the max number of values in the value domain for any one of these items. With g items the size of the type space for each agent is $m = z^g$. We consider the verifier both with and without the internetwork constraints that arise from the requirement of natural payment functions (which hold for the VCG mechanism in this auction environment), but do not consider the use of summarization.

Figure 2 illustrates the number of instances observed before failure. Fixing $z = 5$, we first vary the number of items g from 1 to 10 for $n = 3$ agents (i.e. with $m = z^g = 5^g$ types per agent, m^3 different instances, and at most m^2 networks). Second, we vary the number of agents n from 1 to 10 for $g = 3$ items (i.e. with $m = 5^3 = 125$ types per agent, 125^n instances and at most 125^{n-1} networks). For each experiment each point represents an average over 10 runs. The NetworkChecker is tested with and without internetwork constraints due to natural payments, and labeled *base* and *prop* accordingly.

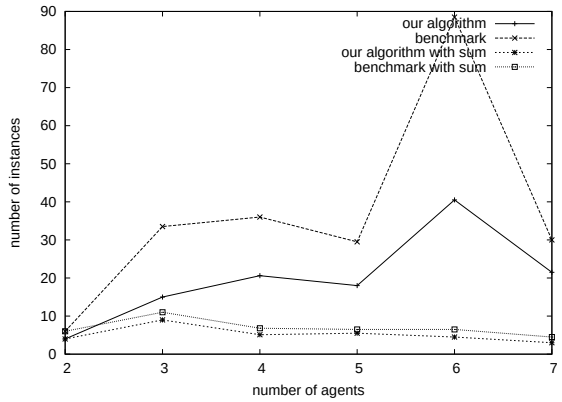
Observe that internetwork constraints considerably speed



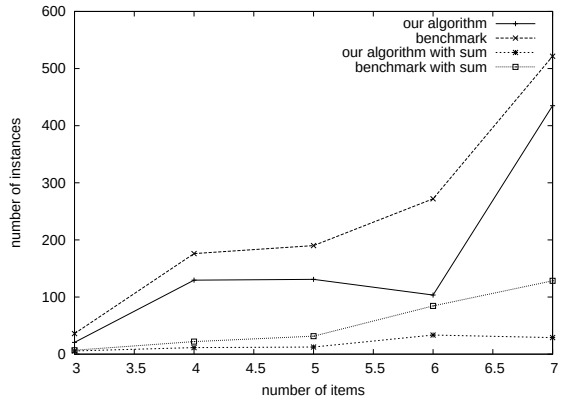
(a) Number of instances before failure in greedyVCG: Agents.



(b) Number of instances before failure in greedyVCG: Items.



(c) Number of instances before failure in localVCG: Agents.



(d) Number of instances before failure in localVCG: Items.

Figure 3: Exploring Verification with Single-Minded Bidders.

up the verification process. In addition, since at most n new nodes are created for each instance, the number of nodes in the graph is roughly proportional to a multiple of n times the number of instances observed before failure and the running time and space complexity of the verifier is roughly linear in the number of instances observed. Thus, imposing natural payments and enabling internetwork constraints improves space and time complexity.

6.2 Single-Minded Bidders

For single-minded bidders we generated a distribution of instances by using the *L4 Legacy distribution* of the Combinatorial Auctions Test Suite (CATS) [16]. We discretized the type space by rounding the values to the nearest 250 (the values fell into a range of [0, 3000]). There are g distinct items, and each agent can choose any subset of the g items, and have a value $\in \{0, 250, \ldots, 2750, 3000\}$ for the desired bundle. The results are averaged over 10 trials.

In this experiment we impose both summarization and envy-freeness, which is a reasonable structural requirement in this environment (as noted in Section 4). As a summarization function we adopt the summarization function defined in Example 4 for the VCG mechanism, which is also valid for the LOS mechanism and therefore reasonable to impose. We also compare with the benchmark algorithm, which fails to catch the non-strategyproofness of greedyVCG but is effective for localVCG.

Figures 3 (a) and (b) illustrate the results for NETWORK CHECKER ("base"), and also with summarization ("sum") and with the additional structure provided by envy-freeness ("envy-free"), which is used here without summarization [6] In Figure 3 (a) we vary the number of agents n while fixing the number of items at $g = 3$ ($m = 13^7$ types, since there are $2^3 - 1$ bundles of 3 items). In Figure 3 (b) we vary the number of items g while fixing the number of agents at $n = 4$ ($m = 13^{2^g - 1}$). The use of summarization provides a significant speed-up, and the imposition of envy-freeness makes the verification of a non-strategyproof mechanism almost immediate. As expected, summarization becomes more important as the number of agents increases, and its benefit increases, because without summarization it gets less and less likely that the history contains observations relevant to a new instance. Figures 3 (c) and (d) illustrate the results for NETWORK CHECKER with and without summarization and the benchmark algorithm with and without summarization, this time with the localVCG mechanism. The same summarization function is adopted as above.[7] We again vary the

[6]For single minded bidders, the internetwork constraints from natural payments used in NETWORKCHECKER are trivially satisfied, and do not improve the checking.

[7]Local search is modeled after the stochastic local search algorithm by Hoos and Boutilier [12]. The only difference is that our search is non-stochastic: we do greedy hill climbing with their neighborhood definition and improvement crite-

number of agents, and then the number of items. In comparison with the benchmark algorithm, the verifier detects non-strategyproofness more quickly both with and without summarization, with the benefit over the benchmark with summarization most noticeable as the number of items increases.

6.3 Computing Metrics

We have also experimented with the *probability of strategyproofness* (PrSP) and *price flexibility* (PF) metrics defined in Section 5. These metrics allow the verifier to provide feedback to agents before finally verifying that a mechanism is strategyproof (or not). We present results for single-minded combinatorial auctions and the (strategyproof) LOS mechanism. Monte Carlo analysis is used to estimate PrSP and PF over all possible valuations $v \in V$, where the valuations are drawn according to the CATS Legacy distribution.

Figure 4 (a) displays the estimated PrSP for LOS in a domain with 4 agents and 3 items. The results are averaged over 100 trials. We consider NetworkChecker without summarization and then with the two summarization functions described in Example 4 (the first is the more general VCG rule, the second is the rule designed for LOS.) Summarization greatly improves the probability that truthful reporting is a dominant strategy at any given point in the verification process, especially the second summarization function which is designed for LOS. Thus, in deploying the LOS mechanism it would be useful to instruct the verifier to adopt this strict summarization function in order to accelerate verification.

Figure 4 (b) displays the estimated PF for LOS in a domain with 3 agents and 3 items. The results are averaged over 300 trials. We compare NetworkChecker with and without the imposition of natural payments and thus with and without the availability of internetwork constraints. We find that internetwork constraints are somewhat effective in reducing the price flexibility, although the effect is not as pronounced as that of summarization on the probability of strategyproofness.

7. CONCLUSIONS AND FUTURE WORK

We have introduced the problem of verifying whether a mechanism is strategyproof and provided a constraint-network based algorithm for verification. The verifier is able to reject mechanisms that are not strategyproof based on the violation of constraints imposed by strategyproofness on the price space. Experimental results demonstrate the potential for accelerated checking when there is additional structure to exploit and also suggests metrics that can be informative in guiding bidders before a mechanism is fully verified. Useful intermediate guarantees can also be provided to participants.

In practice, we believe that the most benefit from passive verification (as defined here) will be realized in combination with *approximations* that allow the verifier to forget some of its history and further reduce the space complexity of verifiers. In future work we will explore limited memory checking, where we let the verifier forget part of the history, and allow false positives; e.g. using data structures such as Bloom filters [4] to allow for fast checking. Another important avenue for future work is to introduce methodologies that can allow for a continuous type space.

ria. Local search is used first to approximate a solution with all agents. During this phase we also track the best solution found for each marginal economy. For payments, we then run local search with a smaller number of search steps on each of the marginal economies and adopt for the economy without i the best allocation discovered in the main search and in the explicit search in the marginal economy. In addition to failing (A2), localVCG can also fail (A1) because the price to a buyer can now depend on the search path adopted in the main economy, which can on the buyer's own bid.

8. REFERENCES

[1] A Archer and E Tardos. Truthful mechanisms for one-parameter agents. In *Proc. 42nd IEEE Symp. on Foundations of Computer Science*, 2001.

[2] Moshe Babaioff, Noam Nisan, and Elan Pavlov. Mechanisms for a spatially distributed market. In *Proc. 5th ACM Conf. on Electronic Commerce*, pages 9–20. ACM Press, 2004.

[3] Y Bartal, R Gonen, and N Nisan. Incentive compatible multi unit combinatorial auctions. In *In Proc. Theoretical Aspect of Rationality and Knowledge*, 2003.

[4] A. Broder and M.Mitzenmacher. Network applications of bloom filters: A survey. *Internet Math*, 1, no. 4, 2003.

[5] D.C.Parkes, M.O.Rabin, S.M.Shieber, and C.A.Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. In *Proc. 8th Int. Conf. on Electronic Commerce (ICEC'06)*, 2006.

[6] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence Journal*, 49:61–95, 1991.

[7] Amos Fiat, Andrew Goldberg, Jason Hartline, and Anna Karlin. Competitive generalized auctions. In *Proc. 34th ACM Symposium on Theory of Computing (STOC'02)*, 2002.

[8] A. Garcia-Camino, P. Noriega, and J. A. Rodriguez-Aguilar. Implementing norms in electronic institutions. In *Proceedings of the 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-05)*, 2005.

[9] Andrew Goldberg and Jason Hartline. Envy-free auctions for digital goods. In *Proc. ACM E'Commerce*, 2003.

[10] F. Guerin and J. V. Pitt. Guaranteeing properties for e-commerce systems. In O. Shehory and N. Sadeh J. Padget, D. Parkes, editor, *LNAI 2531: Agent-Mediated Electronic Commerce IV, pages 253-272. Springer-Verlag*, 2002.

[11] Honwei Gui, Rudolf Muller, and Rakesh V Vohra. Dominant strategy mechanisms with multidimensional types. Technical report, Northwestern Univ., 2004.

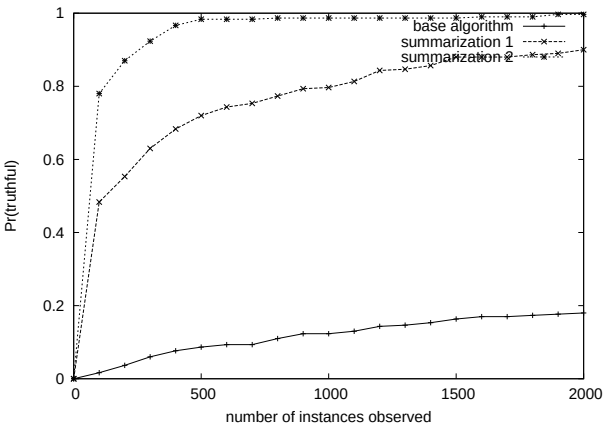
[12] Holger H Hoos and Craig Boutilier. Solving combinatorial auctions with stochastic local search. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, July 2000.

[13] Matthew O. Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.

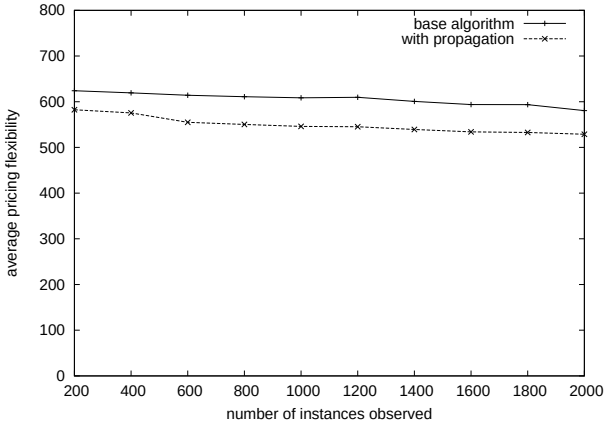
[14] R Lavi, A Mu'alem, and N Nisan. Towards a characterization of truthful combinatorial auctions. In *Proc. 44th Annual Symposium on Foundations of Computer Science*, 2003.

[15] Daniel Lehmann, Liadan Ita O'Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, September 2002.

[16] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, 2000.



(a) Estimated probability of strategyproofness: Effect of summarization



(b) Estimated pricing flexibility: Effect of internetwork constraints (from natural payments)

Figure 4: Evaluating metrics for partial strategyproofness in verification of the LOS mechanism in single-minded combinatorial auctions.

[17] Robert B Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981.

[18] M Naor, B Pinkas, and O Reingold. Privacy preserving auctions and mechanism design. In *Proc. 1st ACM Conf. on Electronic Commerce (EC-99)*, pages 129–139, 1999.

[19] S. Papai. Groves sealed bid auctions of heterogeneous objects with fair prices. *Social Choice and Welfare*, 20:371–385, March 2003.

[20] M. Pauly. Programming and verifying subgame perfect mechanisms. *Journal of Logic and Computation*, 15(3):295-316, 2005.

[21] M. Pauly and M. Wooldridge. Logic for mechanism design - a manifesto. *Proceedings of the Game Theoretic and Decision Theoretic Agents Workshop*, 2003.

[22] W. van der Hoek, A. Lomuscio, and M. Wooldridge. On the complexity of practical atl model checking. In *Proceedings of the 5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS-06)*, 2006.

[23] Makoto Yokoo. The characterization of strategy/false-name proof combinatorial auction protocols: Price-oriented, rationing-free protocol. In *Proc. 18th Int. Joint Conf. on Art. Intell.*, 2003.

APPENDIX

A. PROOFS

PROOF OF THEOREM 5. Correctness is immediate. To show soundness, suppose a mechanism is demonstrated to be non-strategyproof without summarization after k instances. We argue that the mechanism will still be demonstrated to be non-strategyproof with summarization. First, note that the use of summarization function does not affect step (a) of SIMPLECHECKER. We now have the following 3 cases:

1. Check fails in (b) of SIMPLECHECKER: $a^k \in f^H(v^k_{-i})$ and $p^k_i(a^k, v^k_{-i}) \neq p^l_i(a^k, v^k - i) = p^H_i(a^k, v^k_{-i})$ for some $l < k$. By definition of the summarization function, $p^k_i(a^k, v^k_{-i}) = p^k_i(a^k, s(v^k_{-i})$ and $p^l_i(a^k, v^k - i) = p^l_i(a^k, s(v^k - i))$. It follows that $p^k_i(a^k, s(v^k_{-i}) \neq p^l_i(a^k, s(v^k_{-i}))$.
2. Check fails in (c) of SIMPLECHECKER: there exists $a^l \in f^H(v^k_{-i})$, $p^k_i(a^k, v^k_{-i}) < p^l_i(a^l, v^k_{-i}) + (v^l_i(a^k) - v^l_i(a^l))$ for some $l < k$. Since $p^k_i(a^l, v^k_{-i}) = p^k_i(a^l, s(v^k_{-i}))$ and $p^l_i(a^l, v^k_{-i}) = p^k_i(a^l, s(v^k_{-i}))$, $p^k_i(a^k, s(v^k_{-i})) < p^l_i(a^l, s(v^k_{-i})) + (v^l_i(a^k) - v^l_i(a^l))$.
3. Check fails in (d) of SIMPLECHECKER: there exists $a^l \in f^H(v^k_{-i})$, $p^k_i(a^k, v^k_{-i}) > p^l_i(a^l, v^k_{-i}) + (v^k_i(a^k) - v^k_i(a^l))$ for some $l < k$. Then $p^k_i(a^k, s(v^k_{-i})) > p^l_i(a^l, s(v^k_{-i})) + (v^k_i(a^k) - v^k_i(a^l))$.

Hence, if the mechanism fails the check in one of the 3 steps above without summarization, then it will fail the check in the same step with summarization. □

LEMMA 2. *Consider a truthful social choice function f and some valuations v for which $f(v) = a$. Suppose v' is such that for each j, either $v'_j = v_j$ or $v'_j(a) - v'_j(b) > v_j(a) - v_j(b)$ $\forall b \neq a$. Then $f(v') = a$.*

PROOF. The proof follows from W-MON [14], which is a necessary condition for truthfulness.

DEFINITION 6. *A social choice function f satisfies* W-MON *if $\forall\ v \in V, i$, and $v_i \in V_i$: $f(v) = a$ and $f(v'_i, v_{-i}) = b \Rightarrow v'_i(b) - v_i(b) \geq v'_i(a) - v_i(a)$.*

Without loss of generality, suppose the first k agents are such that $v'_j = v_j$ and the rest of the agents are such that $v'_j(a) - v'_j(b) > v_j(a) - v_j(b)$ $\forall b \neq a$. By W-MON, $f(v'_1, \ldots, v'_k, v'_{k+1}, v_{k+2}, \ldots, v_N) = f(v_1, \ldots, v_k, v'_{k+1}, v_{k+2}, \ldots, v_n) = a$. Now, applying W-MON again with agent $k+2$, $f(v'_1, \ldots, v'_k, v'_{k+1}, v'_{k+2}, v_{k+3} \ldots, v_n) = a$. We can repeatedly apply W-MON up to agent n to get $f(v_1, v'_2, \ldots, v'_n) = a$. □

PROOF OF LEMMA 1. First, note that for any v_i such that $f(v_i, v_{-i}) = a$, $f(v_i, v'_{-i}) = a$ by Lemma 2. Consider $\bar{v}_i$ such that for a fixed $b \in A$, $b \neq a$,

$$\bar{v}_i = \arg_{v_i} \inf\{v_i(a) - v_i(b) | f(v_i, v_{-i}) = a\}.$$

Then $\bar{v}_i(a) - \bar{v}_i(b) = \delta_{ab}(v_{-i})$. Now, since $f(\bar{v}_i, v'_{-i}) = a$, $\delta_{ab}(v_{-i}) \in \{v_i(a) - v_i(b) | f(v_i, v'_{-i}) = a\}$. So, $\delta_{ab}(v_{-i}) \geq \inf\{v_i(a) - v_i(b) | f(v_i, v'_{-i}) = a\} = \delta_{ab}(v'_{-i})$. Hence, $\delta_{ab}(v'_{-i}) \leq \delta_{ab}(v_{-i})$. Since $\delta_{ab}(v_{-i}) = p_i(a, v_{-i}) - p_i(b, v_{-i})$ (by natural payments), it follows that $p_i(a, v'_{-i}) - p_i(b, v'_{-i}) \leq p_i(a, v_{-i}) - p_i(b, v_{-i})$. □