Truthfulness Flooded Domains and the Power of Verification for Mechanism Design*

Dimitris Fotakis and Emmanouil Zampetakis

School of Electrical and Computer Engineering
National Technical University of Athens, 15780 Athens, Greece
fotakis@cs.ntua.gr, mzampet@corelab.ntua.gr

Abstract. We investigate the reasons that make symmetric partial verification essentially useless in virtually all domains. Departing from previous work, we consider any possible (finite or infinite) domain and general symmetric verification. We identify a natural property, namely that the correspondence graph of a symmetric verification M is strongly connected by finite paths along which the preferences are consistent with the preferences at the endpoints, and prove that this property is sufficient for the equivalence of truthfulness and M-truthfulness. In fact, defining appropriate versions of this property, we obtain this result for deterministic and randomized mechanisms with and without money. Moreover, we show that a slightly relaxed version of this property is also necessary for the equivalence of truthfulness and M-truthfulness. Our conditions provide a generic and convenient way of checking whether truthful implementation can take advantage of any symmetric verification scheme in any domain. Since the simplest case of symmetric verification is local verification, our results imply, as a special case, the equivalence of local truthfulness and global truthfulness in the setting without money. To complete the picture, we consider asymmetric verification, and prove that a social choice function is M-truthfully implementable by some asymmetric verification M if and only if f does not admit a cycle of profitable deviations.

1 Introduction

In mechanism design, a principal seeks to implement a social choice function that maps the private preferences of some strategic agents to a set of possible outcomes. Exploiting their power over the outcome, the agents may lie about their preferences if they find it profitable. Trying to incentivize truthfulness, the principal may offer payments to (or collect payments from) the agents or find ways of partially verifying their statements, thus restricting the false statements available to them. A social choice function is *truthfully implementable* (or implementable, in short) if there is a payment scheme under which truthtelling becomes a dominant strategy of the agents. Since many social choice functions are not implementable, a central research direction in mechanism design is

* This research was supported by the project AlgoNow, co-financed by the European Union (European Social Fund - ESF) and Greek national funds, through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES, investing in knowledge society through the European Social Fund.

to identify sufficient and necessary conditions under which large classes of functions are truthfully implementable. In this direction, we seek a deeper understanding of the power of partial verification in mechanism design, as far as truthful implementation is concerned, a question going back to the work of Green and Laffont [9].

The Model. For the purposes of this work, it is without loss of generality to consider mechanism design with a single agent, also known as the *principal-agent* setting (see e.g., [2,3] for an explanation). In this setting, the principal wants to implement a *social choice function* $f : D \to O$, where O is the set of possible *outcomes* and D is the *domain* of agent's preferences. Formally, D consists of the agent's *types*, where each type $x : O \to \mathbb{R}$ gives the utility of the agent for each outcome. The agent's type is private information. So, based on the agent's declared type x, the principal computes the outcome $o = f(x)$. A function f is (truthfully) *implementable* if for each type x, with $o = f(x)$, and any other type y, with $o' = f(y)$, $x(o) \geq x(o')$. Then, declaring her real type x is a dominant strategy of the agent. Otherwise, the agent may misreport a type y that results in a utility of $x(o') > x(o)$ under her true type x. This undesirable situation is usually corrected with a payment scheme $p : O \to \mathbb{R}$, that compensates the agent for telling the truth. Then, a function f is (truthfully) *implementable with payments* p (or, in general, implementable with money) if for each type x, with $o = f(x)$, and any other type y, with $o' = f(y)$, $x(o) + p(o) \geq x(o') + p(o')$.

Gui, Müller, and Vohra [10] cast this setting in terms of a (possibly infinite) directed graph G on vertex set D. For each ordered pair of types x and y, G has a directed edge (x, y). Given the social choice function f, we obtain an edge-weighted version of G, denoted G_f, where the weight of each edge (x, y) is $x(o) - x(o')$, with $o = f(x)$ and $o' = f(y)$. This corresponds to the gain of the agent if instead of misreporting y, she reports her true type x. Then, a social choice function f is truthfully implementable if and only if G_f does not contain any negative edges. Moreover, Rochet's theorem [14] implies that a function f is truthfully implementable with money if and only if G_f does not contain any directed negative cycles (see also [17]).

There are many classical impossibility results stating that natural social choice functions (or large classes of them) are not implementable, even with the use of money (see e.g., [12]). Virtually all such proofs seem to crucially exploit that the agent can declare any type in the domain. Hence, Nisan and Ronen [13] suggested that the class of implementable functions could be enriched if we assume *partial verification* [9], which restricts the types that the agent can misreport. Formally, we assume a *correspondence function* (or simply, a *verification*) $M : D \to 2^D$ such that if the agent's true type is x, she can only misreport a type in $M(x) \subseteq D$. As before, we can cast M as a (possibly infinite) directed *correspondence graph* G_M on D. For each ordered pair of types x and y, G_M has a directed edge (x, y) if $y \in M(x)$. Given the social choice function f, we obtain the edge-weighted version $G_{M,f}$ of G_M by letting the edge weights be as in G_f. A social choice function f is M-truthfully implementable (resp. with money) if and only if $G_{M,f}$ does not contain any negative edges (resp. directed negative cycles).

Previous Work. Every function f can be implemented by an appropriately strong verification scheme combined with payments (see also Section 5). So, the problem now is to come up with a meaningful verification M, which is either inherent in or naturally enforceable for some interesting domains and allows for a few non-implementable functions to be M-truthfully implementable. To this end, previous work has considered two kinds of verification, namely *symmetric* and *asymmetric* verification.

Symmetric verification naturally applies to convex domains (e.g., Combinatorial Auctions) and to domains with an inherent notion of distance (e.g., Facility Location, Voting). The idea is that every type x can only declare some type y not far from x. A typical example is M^{ε} verification where each type x can declare any type y in a ball of radius ε around x. Another typical example is M^{swap} verification, naturally applicable to Voting and to ordinal preference domains. In M^{swap} verification, each type x is as a linear order on O and can declare any type y obtained from x by swapping two adjacent outcomes. Rather surprisingly, previous work provides strong evidence that symmetric verification does not give any benefit to the principal, as far as truthful implementation is concerned. In particular, the strong and elegant result of Archer and Kleinberg [2] and its extension by Berger, Müller, and Naeemi [5] imply that M^{ε} verification does not help in convex domains. Formally, the results of [2,5,6] imply that for any convex domain, truthfulness with money is equivalent with M^{ε}-truthfulness with money. Similarly, Caragiannis, Elkind, Szegedy, and Yu [6] proved that M^{swap} verification does not help in the domain of Voting.

As far as implementation without money is concerned, the research on the power of symmetric verification is closely related to the research about sufficient and necessary conditions under which weaker properties are equivalent to global truthfulness. Even though the motivation for studying weaker properties may be more general (see e.g., [15,2,7,16]), in the absence of money, local truthfulness is essentially a special case of symmetric verification. In this research agenda, Sato [16] considered M^{swap} verification (under the name of adjacent manipulation truthfulness) for ordinal preference domains, and proved that if $G_{M^{\mathrm{swap}}}$ is strongly connected by paths satisfying the no-restoration property, then truthful implementation and M^{swap}-truthful implementation are equivalent. He also proved that the universal domain, that includes all linear orders on O, and single-peaked domains have the no-restoration property, and thus, for these domains, truthful implementation is equivalent to M^{swap}-truthful implementation. Independently, Carroll [7] obtained similar results for convex domains, for the universal domain, and for single-peaked and single-crossing domains, which also extend to randomized mechanisms. Carroll also gave a necessary condition for the equivalence of local and global truthfulness in a specific domain with cardinal preferences.

On the other hand, asymmetric verification is “one-sided”. Given a social choice function f, a typical example of asymmetric verification is when the agent can only lie either by overstating or by understating her utility. E.g., for Scheduling on related machines, the machine can only lie by overstating its speed [4], for Combinatorial Auctions, the agent can only underbid on her preferred sets [11], and for Facility Location, the agent can only understate her distance to the nearest facility [8]. The use of asymmetric verification has led to strong positive results about the truthful implementation of natural social choice functions in several important domains (see e.g., [4,11,8] and the references there in). The intuition is that the mechanism discourages one direction of lying, while the other direction of lying is forbidden by the verification.

Motivation and Contribution. Our work is motivated by the general observation, stated explicitly in [6], that even very strict symmetric verification schemes do not help in truthful implementation, while strong positive results are possible with simple asymmetric verification. So, we seek a deeper understanding of the reasons that make symmetric verification essentially useless in virtually all domains, and some formal justification behind the success of asymmetric verification.

Departing from previous work, we consider any possible (finite or infinite) domain D and very general classes of partial verification. To formalize the notions of symmetric and asymmetric verification, we say that a verification M is symmetric if the presence of a directed edge (x, y) in G_M implies the presence of the reverse edge (y, x), and asymmetric if G_M is an acyclic tournament.

Our main result is a general and unified explanation about the weakness of symmetric verification. In Section 3, we identify a natural property, namely that the correspondence graph G_M is strongly connected by finite paths along which the preferences are consistent with the preferences at the endpoints. In fact, we define three versions of this property depending on whether we consider implementation by deterministic truthful mechanisms (strict order-preserving property), by deterministic mechanisms that use payments (strict difference-preserving property), and by randomized truthful-in-expectation mechanisms (difference-convex property). Despite the slightly different definitions, the essence of the property is the same, but stronger versions of it are required as the mechanisms become more powerful. We show that for any (finite or infinite) domain D and any symmetric verification M that satisfies the corresponding version of the property, deterministic / randomized truthful implementation (resp. with money) is equivalent to deterministic / randomized M-truthful implementation (resp. with money). In all cases, the proof is simple and elegant, and only exploits an elementary combinatorial argument on the paths of G_M. With this general sufficient condition for the equivalence of truthfulness and M-truthfulness, we simplify, unify, and strengthen several known results about symmetric verification and local truthfulness without money. E.g., we obtain, as simple corollaries, the equivalence of truthful and M^{ε}-truthful implementation for any convex domain (even with money) and for Facility Location, and the equivalence of truthfulness and M^{swap}-truthfulness for Voting.

In Section 4, we identify necessary conditions for the equivalence of truthfulness and M-truthfulness, for any symmetric verification M. These are relaxed versions of the sufficient conditions, and require that the correspondence graph G_M is strongly connected by finite preference preserving paths. Otherwise, we show how to find a separator of G_M, which in turn, leads to the definition of a function that is M-truthfully implementable, but not implementable. We also observe that the necessary condition is violated by the domain of 2-Facility Location. To conclude the discussion about symmetric verification, we close the small gap between the sufficient and necessary properties, and present the first known condition that is both sufficient and necessary for the equivalence of truthful and M-truthful implementation. Overall, our conditions provide a generic and convenient way of checking whether truthful implementation can take advantage of any symmetric verification scheme in any domain.

Finally, in Section 5, we consider asymmetric verification, and prove that a social choice function f is M-truthfully implementable by some asymmetric verification M if and only if the subgraph of G_f consisting of negative edges is acyclic (Theorem 8). This result provides strong formal evidence about the power of asymmetric verification, since, as we discuss in Section 5, any reasonable social choice function f should not have a cycle in G_f that entirely consists of negative edges. Moreover, we prove that given any function f truthfully implementable by payments p, an asymmetric verification that truthfully implements f can be directly obtained by p (Proposition 1).

Comparison to Previous Work. The strict order-reserving property, which we employ as a sufficient condition for deterministic truthful implementation without money, is similar to the no-restoration property of [16]. However, the results of [16] are restricted to finite domains with ordinal preferences and to M^{swap} verification. Our results are far more general, since we manage, in Theorem 1, to extend the equivalence of truthful and M-truthful implementation, under the strict order-preserving property, to any (even infinite) domain and to any symmetric verification. Moreover, our necessary property generalizes and unifies the necessary conditions of both [7,16].

We also note that our results in case of deterministic implementation with money are not directly comparable to the strong and elegant results about local truthfulness with money in convex domains (see e.g., [2,1]). For instance, if we restrict Theorem 3 to convex domains and compare it to [2, Theorem 3.8], our result is significantly weaker, since it starts from a much stronger hypothesis (see also the discussion in Section 3.2). On the other hand, Theorem 3 is more general, in the sense that it applies to any symmetric strict difference-preserving verification and to arbitrary (even non-convex) domains.

2 Notation and Preliminaries

The basic model and most of the notation are introduced in Section 1. Next, we discuss some conventions, give some definitions, and state some useful facts.

Ordinal Preferences. We always assume that each type x is a function from O to $\mathbb{R}$. However, in case of deterministic mechanisms without money, when the preferences are ordinal, we only care about the relative order of the outcomes in each type.

Truthful Implementation. A social choice function $f : D \to O$ is *M-truthfully implementable* if for every type x and any $y \in M(x)$, $x(f(x)) \geq x(f(y))$. A social choice function f is *M-truthfully implementable with money* if there is a payment scheme $p : O \to \mathbb{R}$ such that for every type x and any $y \in M(x)$, $x(f(x)) + p(f(x)) \geq x(f(y)) + p(f(y))$. If there is no verification, i.e., if for all types x, $M(x) = D$, we say that f is *truthfully implementable* and *truthfully implementable with money*, respectively. We say that truthfulness (resp. with money) is equivalent to M-truthfulness (resp. with money) if for every function f, f is truthfully implementable (resp. with money) iff it is M-truthfully implementable (resp. with money). In what follows, we use the terms *mechanism* and *social choice function* interchangeably.

Randomized Mechanisms. A randomized mechanism $f : D \to \Delta(O)$ maps each type x to a probability distribution over O. A randomized mechanism is (resp. M-)*universally truthful* if it is a probability distribution over deterministic (resp. M-)truthful mechanisms (even with money). For truthfulness-in-expectation, we assume, for simplicity, that O is finite, and let $f_o(x)$ be the probability of the outcome o if the agent reports x. Then, a randomized mechanism f is (resp. M-)*truthful-in-expectation* if for every type x and any $y \in D$ (resp. $y \in M(x)$), $\sum_{o\in O} f_o(x)x(o) \geq \sum_{o\in O} f_o(y)x(o)$.

A randomized mechanism f is (resp. M-)*truthful-in-expectation with money* if there are payments $p : O \to \mathbb{R}$ such that for every $x \in D$ and any $y \in D$ (resp. $y \in M(x)$), $\sum_{o \in O} f_o(x)(x(o) + p(o)) \geq \sum_{o \in O} f_o(y)(x(o) + p(o))$.

Correspondence Graph. A verification M can be represented by the directed *correspondence graph* $G_M = (D, \{(x, y) : y \in M(x)\})$. Given a social choice function f, we let the edge-weighted graph

$$G_{M,f} = (D, \{(x, y) : y \in M(x)\}, w)\,, \text{ where } w(x, y) = x(f(x)) - x(f(y))$$

A k-cycle (resp. k-path) in G_M is a directed cycle (resp. path) consisting of k edges. We say that an edge (x, y) of $G_{M,f}$ is negative if $w(x, y) < 0$. We say that a cycle in $G_{M,f}$ is negative if the total weight of its edges is negative. We let $G^-_{M,f}$ denote the subgraph of $G_{M,f}$ that consists of all its negative edges. If there is no verification, we refer to $G_{D,f}$, $G^-_{D,f}$ as G_f, G^-_f. Also, given a graph G, we let $V(G)$ be its vertex set and $E(G)$ be its edge set.

A social choice function f is M-truthfully implementable iff $G_{M,f}$ does not contain any negative edges. Furthermore, Rochet [14] proved that a social choice function f is M-truthfully implementable with money if and only if the correspondence graph $G_{M,f}$ does not have any finite negative cycles.

Symmetric and Asymmetric Verification. We say that a verification M is symmetric if G_M is *symmetric*, i.e., for each directed edge $(x, y) \in E(G_M)$, $(y, x) \in E(G_M)$. We say that a verification M is *asymmetric* if G_M is an acyclic tournament.

Weak Monotonicity and Cycle Monotonicity. A social choice function f satisfies M-*weak-monotonicity* if for every $x \in D$ and any $y \in M(x)$, $x(f(x)) + y(f(y)) \geq x(f(y)) + y(f(x))$. Equivalently, f is M-weakly-monotone iff $G_{M,f}$ does not contain any negative 2-cycles. A function f satisfies M-*cycle-monotonicity* if for all $k \geq 1$, and all $x_1, \dots, x_k \in D$, such that $x_{i+1} \in M(x_i)$, $\sum_{i=1}^{k} x_i(f(x_i)) \geq \sum_{i=1}^{k} x_{i-1}(f(x_i))$, where the subscripts are modulo k. Equivalently, f is M-cyclic-monotone iff $G_{M,f}$ does not contain any finite negative cycles. If there is no verification, we simply say that f is weakly-monotone and cyclic-monotone, respectively.

Convex Domains. A domain D is *convex* if for every $x, y \in D$ and any $\lambda \in [0, 1]$, the function $z : O \to \mathbb{R}$, with $z(a) = \lambda x(a) + (1 - \lambda) y(a)$, for each $a \in O$, is also in D.

Strategic Voting. We have k candidates and select one of them based on the preferences of n agents. Hence, $O = \{o_1, \dots, o_k\}$ is the set of candidates, $V = \{v_1, \dots, v_n\}$ is the set of voters, and the type of each voter is a linear order over O.

k-Facility Location. In k-Facility Location, we place $k \geq 1$ facilities on the real line based on the preferences of n agents. The type of each agent i is determined by $x_i \in \mathbb{R}$, and the set of outcomes is $O = \mathbb{R}^k$. The utility of agent i from an outcome $(y_1, \dots, y_k) \in O$ is $-\min_j |x_i - y_j|$. If $k = 1$, we simply refer to Facility Location.

M^ε and M^{swap} Verification. In case of a convex domain or Facility Location, given an $\varepsilon > 0$, we let $M^\varepsilon(x) = \{y \in D : ||x - y|| \leq \varepsilon\}$, for all x, where $|| \cdot ||$ is the l_2 distance in $\mathbb{R}^O$ for convex domains and $|x - y|$ for Facility Location. If we have a domain D where the agent's types are linear orders on O, for any type $x \in D$, $M^{\text{swap}}(x)$ is the set of all linear orders on O obtained from x by swapping two adjacent outcomes in x.

3 Sufficient Conditions for Truthful Implementation

Without any assumptions on the domain, symmetric verification is not sufficient for the equivalence of truthfulness and M-truthfulness. Next, we assume that the correspondence graph G_M is symmetric and strongly connected by finite paths along which the preferences are consistent with the preferences at the endpoints. We prove that this suffices for the equivalence of truthfulness and M-truthfulness, even for infinite domains.

3.1 Deterministic Mechanisms

We start with a sufficient condition for a symmetric verification M (and its correspondence graph) under which any deterministic M-truthful mechanism is also truthful.

Definition 1 (Order-Preserving Path). *Given a verification M, an $x-y$ path p in G_M is* order-preserving *if for all outcomes $a, b \in O$, with $x(a) > x(b)$ and $y(a) \geq y(b)$, and for any intermediate type w in p, $w(a) > w(b)$. A $x - y$ path p in G_M is* strict order-preserving *if for every type w in p, the subpath of p from x to w is order-preserving.*

Intuitively, if the endpoints x and y of an order-preserving path p agree that outcome a is preferable to outcome b, any intermediate type w in p should also agree on this. Following Definition 1, we say that a verification M is *symmetric* (resp. *strict*) *order-preserving* if M is symmetric and for any types $x, y \in D$, there is a *finite* (resp. strict) order-preserving $x - y$ path in the correspondence graph G_M. Next, we show that:

Theorem 1. *Let M be a symmetric strict order-preserving verification. Then, truthfulness is equivalent to M-truthfulness.*

Proof. If a social function is truthfully implementable, it is also M-truthfully implementable. For the converse, we use induction on the length of the strict order-preserving paths in G_M. Technically, for sake of contradiction, we assume that there is a function f that is M-truthfully implementable, but not implementable. Therefore, all edges in $G_{M,f}$ are non-negative, but there is a negative edge $(x, z) \in E(G_f)$.

Since M is symmetric strict order-preserving, there is a finite strict order-preserving $x - z$ path p in $G_{M,f}$. In particular, we let $p = (x = v_0, v_1, v_2, \ldots, v_k = z)$, and let i, $2 \leq i \leq k$, be the smallest index such that the edge $(x, v_i) \in E(G_f)$ is negative. For convenience, we let $y = v_i$ and $w = v_{i-1}$. We note that by the definition of i, the edge $(x, w) \in E(G_f)$ is non-negative, and also since f is M-truthfully implementable, the edges $(w, y), (y, w) \in E(G_{M,f})$ are non-negative (see also Fig. 1.i).

For convenience, we let $a = f(x)$, $b = f(w)$, $c = f(y)$ denote the outcome of f at x, y, and w, respectively. Since the edge (x, y) is negative, $a \neq c$. Moreover, by the definition of i (and of y), $b \neq c$. By the discussion above, we have that $x(c) > x(a) \geq x(b)$ and $y(c) \geq y(b)$. Therefore, since the $x - z$ path is strict order-preserving, and thus its $x - y$ subpath is order-preserving, we obtain that $w(c) > w(b)$, a contradiction to the hypothesis that the edge $(w, y) \in E(G_{M,f})$ is non-negative. Therefore there is no negative edge in G_f, which implies that f is truthfully implementable. □

If D is finite, we can show that for a symmetric verification, the strict order-preserving property is equivalent to the order-preserving property. Thus, we obtain that:

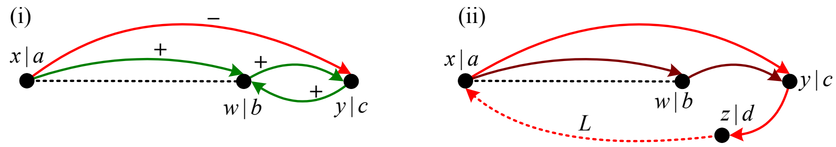

Fig. 1. (i) The part of G_f considered in the proof of Theorem 1. (ii) The part of G_f considered in the proof of Theorem 3. The label of each node consists of the type and the outcome of f.

Theorem 2. *Let M be a symmetric order-preserving verification in a finite domain D. Then, truthfulness is equivalent to M-truthfulness.*

Applications. Theorems 1 and 2 provide a generic and convenient way of checking whether truthful implementation can take any advantage of symmetric verification. E.g., one can verify that for any convex domain D, M^ε verification is strict order-preserving, and that for Strategic Voting, $M^{\rm swap}$ verification is order-preserving. Thus, we obtain alternative (and very simple) proofs of [6, Theorems 3.1 and 3.3]. Moreover, our corollary about $M^{\rm swap}$ verification implies the main result of [16]. Similarly, we can show that for the Facility Location domain, which is non-convex, M^ε verification is strict order-preserving. Thus, for Facility Location, a mechanism is truthful iff it is M^ε-truthful.

3.2 Deterministic Mechanisms with Money

Next, we extend the notion of order-preserving paths to mechanisms with money. Since utilities are not ordinal anymore, we use the notion of difference-preserving paths, which takes into account the difference between the utility of different outcomes. Formally, given a verification M, an $x-y$ path p in G_M is *difference-preserving* if for any intermediate type w in p and for all outcomes $a, b \in O$, if $x(a)-x(b) \neq y(a)-y(b)$,

- $w(a)-w(b) \in (\min\{x(a)-x(b), y(a)-y(b)\}, \max\{x(a)-x(b), y(a)-y(b)\})$
- $w(a)-w(b) = x(a)-x(b)$, if $x(a)-x(b) = y(a)-y(b)$.

As for order-preserving paths, if both endpoints x and y of a difference-preserving path p prefer a to b, any type w in p should also prefer a to b. Moreover, the strength of w's reference for a, i.e., $w(a)-w(b)$, should lie between the strength of x's and of y's preference for a. In fact, the difference-preserving property is a stronger version of the increasing difference property in [5, Definition 5]. Similarly, an $x-y$ path p in G_M is *strict difference-preserving* if for every type w in p, the subpath of p from x to w is also difference-preserving. A verification M is *symmetric* (resp. *strict*) *difference-preserving* if M is symmetric and for any $x, y \in D$, there is a *finite* (resp. strict) difference-preserving $x-y$ path in G_M.

We proceed to show that the symmetric strict difference-preserving property is sufficient for the equivalence of M-truthfulness with money and truthfulness with money. The proof is based on the equivalence of cycle monotonicity and truthful implementation with money. As a first step, we employ a proof similar to that of Theorem 1, and show that under the symmetric strict difference-preserving property, for any function f, $G_{M,f}$ does not have any negative 2-cycles iff G_f does not have any negative 2-cycles.

Lemma 1. *Let M be a symmetric strict difference-preserving verification. Then for any social choice function f, f is M-weakly monotone if and only if f is weakly-monotone.*

Using Lemma 1, we next show that under the symmetric strict difference-preserving property, M-cycle monotonicity is equivalent to cycle monotonicity.

Theorem 3. *Let M be a symmetric strict difference-preserving verification. Then for any social choice function f, f is M-truthfully implementable with money if and only if f is truthfully implementable with money.*

Proof. If f is truthfully implementable with money, it is also M-truthfully implementable with money. For the converse, we show that if $G_{M,f}$ does not have any negative cycles, then G_f does not have any negative cycles as well. In what follows, we assume that G_f does not have any negative 2-cycles, since otherwise, by Lemma 1, f is not M-weakly monotone, and thus, not truthfully implementable with money.

For sake of contradiction, we assume that G_f includes some negative cycle with more than 2 (and a finite number of) edges. In particular, we let $C = (x, y, z, \dots, x)$ be any such cycle. The existence of such a cycle C is guaranteed by Rochet's theorem. Moreover, C contains at least one edge $(x, y) \in E(G_f) \setminus E(G_{M,f})$, because C is not present in $G_{M,f}$. Since M is a symmetric strict difference-preserving verification, there is a finite strict difference-preserving $x - y$ path $p = (v_0 = x, v_1, \dots, v_k = y)$. For convenience, we let $w = v_{k-1}$ be the last node before y in p, let $a = f(x)$, $b = f(w)$, $c = f(y)$, and $d = f(z)$ be the outcome of f at x, w, y, and z, respectively, and let L be the total length of the $z - x$ path used by C (see also Fig. 1.ii).

Since the cycle C is negative, $x(a)-x(c)+y(c)-y(d)+L < 0$. Moreover, since G_f does not contain any negative 2-cycles, $x(c) - x(b) \leq y(c) - y(b)$. Otherwise, since w belongs to a difference-preserving $x-y$ path, we would have that $y(c)-y(b) < w(c)-w(b)$, which implies that the 2-cycle (w, y, w) is negative. Hence, since w belongs to a difference-preserving $x - y$ path, $x(c) - x(b) \leq w(c) - w(b)$. Therefore,

$$x(a) - x(b) + w(b) - w(c) + y(c) - y(d) + L \leq x(a) - x(c) + y(c) - y(d) + L < 0$$

So, we have that the cycle $C_1 = (x, w = v_{k-1}, y, \dots, z)$ is also negative.

Since p is strict difference-preserving, the path $p' = (x = v_0, v_1, \dots, v_{k-1} = w)$ is also difference-preserving. Therefore, using the same argument, we can prove that the cycle $C_2 = (x, v_{k-2}, v_{k-1}, y, \dots, z)$ is also negative. Repeating the same process $k - 1$ times, we obtain that the cycle $C_{k-1} = (x = v_0, v_1, \dots, v_{k-1}, y, \dots, z)$ is also negative. However, all the edges (v_i, v_{i+1}), $i = 0, \dots, k-1$, of the strict difference-preserving $x - y$ path p belong to G_M. Hence, the edge $(x, y) \in E(G_f) \setminus E(G_{M,f})$ in C is replaced by k edges of $E(G_{M,f})$ in C_{k-1}. Therefore, the negative cycle C_{k-1} has one edge not in $E(G_{M,f})$ less than the original negative cycle C. Repeating the same process for every edge of C not in $E(G_{M,f})$, we obtain a negative cycle C' with all edges in $E(G_{M,f})$. This is a contradiction, since it implies that f is not M-truthfully implementable with money. □

Since M^{ε} verification is symmetric and strict difference-preserving for any convex domain, Theorem 3 implies that for convex domains, M^{ε}-truthful implementation with money is equivalent to truthful implementation with money. This result is also a corollary of [2, Theorem 3.8], but here we obtain it through a completely different approach. In particular, Archer and Kleinberg [2] proved that if there is no "local" negative cycle C in G_f, where "local" means that C can fit in a small area of the convex domain D, then G_f does not contain any negative cycles, and thus, f is truthfully implementable with money. On the other hand, we prove here that if G_f does not contain any negative cycles consisting of "local" edges, then G_f does not contain any negative cycles. So, in our case, the hypothesis is much stronger, since it excludes the existence of negative cycles that consist of "local" edges, but may cover an arbitrarily large area of the convex domain D. In this sense, if we restrict Theorem 3 to convex domains, our result is different in nature and weaker than [2, Theorem 3.8]. Nevertheless, Theorem 3 is quite more general, in the sense that it applies to any symmetric strict difference-preserving verification and to arbitrary (even non-convex) domains.

3.3 Randomized Truthful-in-Expectation Mechanisms

A general condition is sufficient and/or necessary for the equivalence between universal truthfulness and M-universal truthfulness in randomized mechanisms, iff it is sufficient and/or necessary for the equivalence between truthfulness and M-truthfulness in deterministic mechanisms. Hence, all the results of Sections 3.1, 3.2, and 4 directly apply to randomized universally-truthful mechanisms (also with money).

A similar, but more interesting, correspondence holds for the case of randomized truthful-in-expectation mechanisms. For simplicity, we assume here that the set of outcomes $O = \{o_1, \dots, o_m\}$ is finite. With each type $x : O \mapsto \mathbb{R}$, we associate a new type $X : \Delta(O) \mapsto \mathbb{R}$, such that for each probability distribution $\boldsymbol{q}$ over outcomes, the utility $X(\boldsymbol{q})$ is the expected utility of x wrt. $\boldsymbol{q}$. Formally, $X(\boldsymbol{q}) = \sum_{i=1}^{m} q_i x(o_i)$. We let D' be the set of these new types. By definition, there is an one-to-one correspondence between types in D and types in D'. Hence, a social choice function $f : D \to \Delta(O)$ corresponds to a (deterministic) social choice function $f' : D' \to \Delta(O)$. Moreover, (resp. given a verification M) f is (resp. M-)truthful-in-expectation iff f' is (resp. M-)truthful.

As before, we seek a general condition under which truthfulness-in-expectation is equivalent to M-truthfulness-in-expectation. For each type $X \in D'$, corresponding to type $x \in D$, we define $M'(X) = \{Y \in D' : y \in M(x)\}$. Now, the results of Sections 3.1, 3.2, and 4 directly apply to the new domain D' with verification M'. We note that if M is symmetric, then M' is symmetric as well. Hence, for a result that directly applies to the original verification M and domain D, we need a property of the paths in G_M that guarantees that the corresponding paths in $G_{M'}$ are order-preserving.

An $x - y$ path p in G_M is *difference-convex* if for any type w in p, there is a $\lambda \in (0, 1)$, such that for all $a, b \in O$, $w(a) - w(b) = \lambda(x(a) - x(b)) + (1 - \lambda)(y(a) - y(b))$. Similarly, an $x - y$ path p in G_M is *strict difference-convex* if for every type w in p, the subpath of p from x to w is also difference-convex. A verification M is called *symmetric* (resp. *strict*) *difference-convex* if M is symmetric and for any $x, y \in D$, there is a *finite* (resp. strict) difference-convex $x - y$ path in G_M. For truthfulness-in-expectation, we quantify the utility of each type x for each outcome. Hence, the difference-convex property is a stronger version of the difference-preserving property, which in turn, is a stronger version of the order-preserving property.

Lemma 2. *If an $x - y$ path p in G_M is (resp. strict) difference-convex, then the corresponding $X - Y$ path p' in $G_{M'}$ is (resp. strict) difference-preserving, and thus, (resp. strict) order-preserving.*

Although the difference-convex property seems quite strong, a slight deviation from it results in paths in $G_{M'}$ that are not difference-preserving. In this sense, the difference-convex property and Lemma 2 are tight.

By the discussion above, Lemma 2, Theorem 1, and Theorem 3 imply that:

Theorem 4. *Let M be a symmetric strict difference-convex verification. Then, truthfulness-in-expectation (resp. with money) is equivalent to M-truthfulness-in-expectation (resp. with money).*

4 Necessary Conditions for Truthful Implementation

Next, we study relaxed versions of the sufficient conditions in Section 3, and show that they are necessary conditions for the equivalence of truthfulness and M-truthfulness.

Deterministic Mechanisms. Given an outcome $a \in O$, we say that an $x - y$ path p in G_M is *a-preserving* if for all outcomes $b \in O$, with $x(a) > x(b)$ and $y(a) \geq y(b)$, and for any intermediate type w in p, $w(a) > w(b)$. Namely, if the endpoints x and y of p agree that a is preferable to b, any intermediate type w in p should also prefer a to b. A verification M is called *symmetric outcome-preserving* if M is symmetric and for all types $x, y \in D$ and all outcomes $a \in O$, there is a *finite* a-preserving $x - y$ path p in G_M. Though quite close to each other, the order-preserving property implies the outcome-preserving property, but not vice versa. Specifically, an a-preserving path p may not be order-preserving, because the relative preference order of some outcomes, other than a, may change in the intermediate nodes of p.

Theorem 5. *Let M be a symmetric verification that is not outcome-preserving. Then, there exists a function g which is M-truthfully implementable, but not implementable.*

Proof. Since M is not outcome-preserving, there exists a pair of types $x, y \in D$ and an outcome $a \in O$, such that any finite $x - y$ path in G_M violates the a-preserving property. Thus, all $x - y$ paths in G_M consist of at least 2 edges (a single edge is trivially order-preserving). Then, we construct a certificate that M is not outcome-preserving, which is a separator of x and y in G_M, and based on this, we define a function g that is M-truthfully implementable, but not truthfully implementable.

For every finite $x - y$ path p in G_M, we let t_p denote the first intermediate type in p and o_p denote an outcome, such that $x(a) > x(o_p) \wedge y(a) \geq y(o_p) \wedge t_p(o_p) \geq t_p(a)$. Namely, for every finite $x - y$ path p, t_p and o_p provide a certificate that p violates the a-preserving property. We let $O_{xy} = \{o_p \in O : p \text{ is a finite } x - y \text{ path}\}$ be the set of outcomes in these certificates, and let $C_{xy} = \{z \in D \setminus \{y\} : \exists b \in O_{xy} \text{ with } z(b) \geq z(a)\}$ be a set of types that can be used as certificates along with the outcomes in O_{xy}. For convenience, we simply use C instead of C_{xy}. The crucial observation is that for every finite $x - y$ path p in G_M, $t_p \in C$, and thus, C is a separator of x and y in G_M.

Let A be the set of types in the connected component[1] that contains x, obtained from G_M after we remove C, and let $B = D \setminus (A \cup C)$. Since $y \notin C$, by definition, and for every finite $x - y$ path p, $t_p \in C$, y is in B. We consider the following function:

$$g(z) = \begin{cases} \arg\max_{b \in O_{xy}} \{z(b)\} & z \in A \cup C \\ a & z \in B \end{cases}$$

By the definition of C, every type in $A \cup B$ prefers a to any outcome in O_{xy}. However, by the definition of A and B, no type $z \in A$ has a neighbor in B, since otherwise, we could find a finite path from x to G_M^y. Therefore, for any $z \in A$, all z's neighbors G_M are in $A \cup C$, and thus $g(z)$ is z's best outcome in its G_M neighborhood. Similarly, every type $z \in C$ prefers any type in O_{xy} to a, and every type $z \in B$ prefers a to any outcome in O_{xy}, by the definition of C. Hence, g is M-truthfully implementable. On the other hand, g is not truthfully implementable, because x prefers a to any outcome in O_{xy}, and thus has an incentive to misreport y, if we do not have any verification. □

Theorem 5 provides a convenient way of checking whether truthful implementation cannot take any advantage of symmetric verification. E.g., we can show that for the domain of 2-Facility Location, M^ε verification is not outcome-preserving, and thus, there are such social choice functions that become truthful with M^ε verification.

Deterministic Mechanisms with Money. We obtain here a necessary condition for the equivalence of weak and M-weak monotonicity. Given a verification M and $a, b \in O$, an $x - y$ path p in G_M, with $x(a) - x(b) \neq y(a) - y(b)$, is *difference (a, b)-preserving* if for any type w in p, $w(a) - w(b) \in (\min\{x(a) - x(b), y(a) - y(b)\}, \max\{x(a) - x(b), y(a) - y(b)\})$. A verification M is *symmetric difference outcome-preserving* if M is symmetric and for any types $x, y \in D$ and all outcomes $a, b \in O$, there is a *finite* difference (a, b)-preserving $x - y$ path p in G_M. As before, the difference-preserving property implies the difference outcome-preserving property, but not vice versa. By a proof similar to that of Theorem 5, we can show that:

Theorem 6. *Let M be a symmetric verification which is not difference outcome-preserving. Then, there is a social choice function g which is M-weakly monotone, but not weakly monotone.*

Sufficient and Necessary Condition. Closing the small gap between the order-preserving and outcome-preserving properties, we present a condition that is both sufficient and necessary for the equivalence of truthful and M-truthful implementation. Given a social choice function f, a $x - y$ path $p = (x = v_0, v_1, \ldots, v_k, v_{k+1} = y)$ in G_M is *f-preserving* if for any type v_i, $1 \leq i \leq k + 1$ in p, and for all outcomes $a \in O$, with $x(f(v_i)) > x(a)$ and $v_i(f(v_i)) \geq v_i(a)$, $v_{i-1}(f(v_i)) > v_{i-1}(a)$. A verification M is *symmetric function-preserving* if M is symmetric and for any M-truthfully implementable function f and all types $x, y \in D$, there is a *finite* f-preserving $x - y$ path in G_M. Using the techniques in the proofs of Theorems 1 and 5, we can show that:

Theorem 7. *Let M be a symmetric verification. Then, truthful implementation is equivalent to M-truthful implementation if and only if M is function-preserving.*

[1] If D is finite, we use the standard graph-theoretic definition of connected components. If D is infinite, A includes x and all types $w \in D$ reachable from x through a finite path.

5 On the Power of Asymmetric Verification

Intuitively, one should expect that asymmetric verification is powerful due to requirement that the correspondence graph should be acyclic. In fact, if we consider any asymmetric verification M, since G_M does not have any negative cycles, Rochet's theorem implies that any social choice function f is M-truthfully implementable with money. We next show a natural characterization of the social choice functions that can be M-truthfully implemented (without money), for some asymmetric verification M.

Theorem 8. *Let f be any social choice function. There is an asymmetric verification M such that f is M-truthfully implementable iff G_f^- is a directed acyclic graph.*

Proof. Let M be an asymmetric verification that truthfully implements f. Hence, G_M is an acyclic tournament and $G_{M,f}$ does not contain any any edges of G_f^-. Therefore, if we arrange the vertices of G_f on the line according to the (unique) topological ordering of $G_{M,f}$, all edges of G_f not included in $G_{M,f}$ are directed from right to left. Therefore, the edges of G_f^- cannot form a cycle. For the converse, let f be a social choice function with an acyclic G_f^-. We consider a topological ordering of G_f^- and remove any edge of G_f directed from left to right. This removes all edges of G_f^- and leaves an acyclic subgraph G'_f, since all its edges are directed from right to left. Moreover, for every pair of types x, y, we remove one of the edges (x, y) and (y, x). Hence, G'_f is an acyclic tournament without any negative edges. Therefore, f is M-truthfully implementable for the asymmetric verification M corresponding to G'_f. □

Reasonable social choice functions should have an acyclic G_f^-. This is true for all functions maximizing the social welfare and all functions truthfully implementable with money. Although one may construct examples of functions f where G_f^- contains cycles, such functions (and such cycles) are hardly natural. For instance, a 2-cycle (x, y, x) in G_f^- indicates that type x prefers outcome $f(y)$ to $f(x)$, while type y prefers outcome $f(x)$ to $f(y)$. But then, one may change f to f', with $f'(x) = f(y)$, $f'(y) = f(x)$, and $f'(z) = f(z)$ for any other type z. Thus, one eliminates the cycle (x, y, x) and the social welfare is strictly greater using f' allocation.

We can extend the construction in the proof of Theorem 8 to a *universal asymmetric verification*, which can truthfully implement any social choice function with acyclic G_f^-. Applying this, we can show that in the Facility Location domain, the function $F_{\max}(\boldsymbol{x}) = (\min \boldsymbol{x} + \max \boldsymbol{x})/2$, that minimizes the maximum distance of the agents to the facility, can be truthfully implemented with verification $M_{\max}(x_i) = \{y : |y - F_{\max}(\boldsymbol{x}_{-i})| \leq |x_i - F_{\max}(\boldsymbol{x}_{-i})|\}$. Similarly, we can show that in the domain of Strategic Voting, Plurality can be truthfully implemented by an asymmetric verification where the voters are not allowed to misreport a higher preference for the winner of the election. Moreover, we can show that Borda Count can be truthfully implemented by an asymmetric verification where the voters are not allowed to misreport either a higher preference for the winner of the election or a lower preference for some of the remaining candidates.

Asymmetric Verification and Payments. The absence of negative cycles in G_f implies the absence of cycles in G_f^-. Thus, Theorem 8, combined with Rochet's theorem, shows that for any function f truthfully implementable with money, there is an asymmetric verification M that truthfully implements f. Extending the proof of Theorem 8, can can show that such an asymmetric verification M can be directly obtained from any payment scheme that implements f.

Proposition 1. *Let f be a social choice function truthfully implementable by payments $p : D \mapsto \mathbb{R}$. Then, removing all edges $(x, y) \in E(G_f)$ with $p(f(x)) > p(f(y))$ results in an asymmetric verification M that truthfully implements f (without money).*

References

1. A. Archer and R. Kleinberg. Characterizing truthful mechanisms with convex type spaces. *ACM SIGecom Exchanges*, 7(3), 2008.
2. A. Archer and R. Kleinberg. Truthful germs are contagious: A local-to-global characterization of truthfulness. In *Proc. of the 9th ACM Conference on Electronic Commerce (EC '08)*, pp. 21-30, 2008.
3. V. Auletta, P. Penna, G. Persiano, and C. Ventre. Alternatives to truthfulness are hard to recognize. *Autonomous Agents and Multi-Agent Systems*, 22(1):200-216, 2011.
4. V. Auletta, R. De Prisco, P. Penna, and G. Persiano. The power of verification for one-parameter agents. *Journal of Computer and System Sciences*, 75:190-211, 2009.
5. A. Berger, R. Müller, and S.H. Naeemi. Characterizing incentive compatibility for convex valuations. In *Proc. of the 2nd International Symposium on Algorithmic Game Theory (SAGT '09)*, LNCS 5814, pp. 24-35, 2009. Updated version as Research Memoranda 035, Maastricht Research School of Economics of Technology and Organization.
6. I. Caragiannis, E. Elkind, M. Szegedy, and L. Yu. Mechanism design: from partial to probabilistic verification. In *Proc. of the 13th ACM Conference on Electronic Commerce (EC '12)*, pp. 266-283, 2012.
7. G. Carroll. When are local incentive constraints sufficient? *Econometrica*, 80(2):661-686, 2012.
8. D. Fotakis and C. Tzamos. Winner-imposing strategyproof mechanisms for multiple Facility Location games. *Theoretical Computer Science*, 472:90-103, 2013.
9. J. Green and J. Laffont. Partially verifiable information and mechanism design. *Review of Economic Studies*, 53(3):447-456, 1986.
10. H. Gui, R. Müller, and R.V. Vohra. Dominant strategy mechanisms with multi-dimensional types. *Discussion Paper 1392*, Northwestern University, 2004.
11. P. Krysta and C. Ventre. Combinatorial auctions with verification are tractable. In *Proc. of the 18th European Symposium on Algorithms (ESA '10)*, LNCS 6347, pp. 39-50, 2010.
12. N. Nisan. Introduction to Mechanism Design (for Computer Scientists). *Algorithmic Game Theory*, Chapter 9:209-241, 2007.
13. N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166-196, 2001.
14. J.C. Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16(2):191-200, 1987.
15. M.E. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proc. of the 6th ACM Conference on Electronic Commerce (EC '05)*, pp. 286-293, 2005.
16. S. Sato. A sufficient condition for the equivalence of strategy-proofness and nonmanipulability by preferences adjacent to the sincere one. *J. Economic Theory*, 148:259-278, 2013.
17. R.V. Vohra. *Mechanism Design: A Linear Programming Approach*. Cambridge University Press, 2011.