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.

Y. Chen and N. Immorlica (Eds.): WINE 2013, LNCS 8289, pp. 202–215, 2013.

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