Automated Mechanism Design: Complexity Results Stemming from the Single-Agent Setting[*]

Vincent Conitzer Tuomas Sandholm
Carnegie Mellon University
Computer Science Department
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
{conitzer, sandholm}@cs.cmu.edu

ABSTRACT

The aggregation of conflicting preferences is a central problem in multiagent systems. The key difficulty is that the agents may report their preferences insincerely. *Mechanism design* is the art of designing the rules of the game so that the agents are motivated to report their preferences truthfully and a (socially) desirable outcome is chosen. We propose an approach where a mechanism is automatically created for the preference aggregation setting at hand. This has several advantages, but the downside is that the mechanism design optimization problem needs to be solved anew each time. Hence the computational complexity of mechanism design becomes a key issue. In this paper we analyze the single-agent mechanism design problem, whose simplicity allows for elegant and generally applicable results.

We show that designing an optimal deterministic mechanism that does not use payments is $\mathcal{NP}$-complete even if there is only one agent whose type is private information—even when the designer's objective is social welfare. We show how this hardness result extends to settings with multiple agents with private information. We then show that if the mechanism is allowed to use randomization, the design problem is solvable by linear programming (even for general objectives) and hence in $\mathcal{P}$. This generalizes to any fixed number of agents. We then study settings where side payments are possible and the agents' preferences are quasilinear. We show that if the designer's objective is social welfare, an optimal deterministic mechanism is easy to construct; in fact, this mechanism is also *ex post* optimal. We then show that designing an optimal deterministic mechanism with side payments is $\mathcal{NP}$-complete for general objectives, and this hardness extends to settings with multiple agents. Finally, we show that an optimal randomized mechanism can be designed in polynomial time using linear programming even for general objective functions. This again generalizes to any fixed number of agents.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Economics

Keywords

Game Theory, Automated Mechanism Design

1. INTRODUCTION

In multiagent settings, agents generally have different preferences, and it is of central importance to be able to aggregate these, that is, to pick a socially desirable *outcome* from a set of outcomes. Such outcomes could be potential presidents, joint plans, allocations of goods or resources, etc. The preference aggregator generally does not know the agents' preferences *a priori*. Rather, the agents report their preferences to the coordinator. Unfortunately, an agent may have an incentive to misreport its preferences in order to mislead the mechanism into selecting an outcome that is more desirable to the agent than the outcome that would be selected if the agent revealed its preferences truthfully. Such manipulation is undesirable because preference aggregation mechanisms are tailored to aggregate preferences in a socially desirable way, and if the agents reveal their preferences insincerely, a socially undesirable outcome may be chosen.

Manipulability is a pervasive problem across preference aggregation mechanisms. A seminal negative result, the *Gibbard-Satterthwaite theorem*, shows that under *any* nondictatorial preference aggregation scheme, if there are at least 3 possible outcomes, there are preferences under which an agent is better off reporting untruthfully [7, 16]. (A preference aggregation scheme is called dictatorial if one of the agents dictates the outcome no matter how the others vote.)

What the aggregator would like to do is design a preference aggregation mechanism so that 1) the self-interested agents are motivated to report their preferences truthfully, and 2) the mechanism chooses an outcome that is desirable from the perspective of some social objective. This is the classic setting of *mechanism design* in game theory. Mechanism design provides a variety of carefully crafted definitions of what it means for a mechanism to be nonmanipulable, and objectives to pursue under this constraint (e.g., social welfare maximization). It also provides some general mechanisms which,

[*] The material in this paper is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678.

under certain assumptions, are nonmanipulable and socially desirable (among other properties). The upside of these mechanisms is that they do not rely on (even probabilistic) information about the agents' preferences (e.g., the Vickrey-Clarke-Groves mechanism [2,9,17]), or they can be easily applied to any probability distribution over the preferences (e.g., the dAGVA mechanism [1,5]). The downside is that these mechanisms only work under restrictive assumptions. It is often assumed that side payments can be used to tailor the agents' incentives, but this is not always practical. For example, in many voting settings, the use of side payments would not be politically feasible. Furthermore, among software agents, it might be more desirable to construct mechanisms that do not rely on the ability to make payments. Another common assumption is that the designer's objective is social welfare. There are many other measures of social desirability, such as fairness, that the classical mechanisms do not maximize. Furthermore, sometimes the designer's objective is not a measure of social desirability (e.g., in many auctions, the auctioneer seeks to maximize expected revenue).

In contrast, we propose that the *mechanism be designed automatically for the specific preference aggregation problem at hand*. We formulate the mechanism design problem as an optimization problem. The input is characterized by the number of agents, the agents' possible types (preferences), and the aggregator's prior probability distributions over the agents' types. The output is a nonmanipulable mechanism that is optimal with respect to some objective.

This approach has three advantages over the classical approach of designing general mechanisms. First, it can be used even in settings that do not satisfy the assumptions of the classical mechanisms (such as availability of side payments or that the objective is social welfare). Second, it may allow one to circumvent impossibility results (such as the Gibbard-Satterthwaite theorem) which state that there is no mechanism that is desirable across all preferences. When the mechanism is designed to the setting at hand, it does not matter that it would not work more generally. Third, it may yield better mechanisms (in terms of stronger nonmanipulability guarantees and/or better outcomes) than classical mechanisms because the mechanism capitalizes on the particulars of the setting (the probabilistic information that the mechanism designer has about the agents' types). Given the vast amount of information that parties have about each other today, this approach is likely to lead to tremendous savings over classical mechanisms, which largely ignore that information. For example, imagine a company automatically creating its procurement mechanism based on its statistical knowledge about its suppliers, rather than using a classical descending procurement auction.

However, this approach requires the mechanism design optimization problem to be solved anew for each setting. Hence its computational complexity becomes a key issue. In this paper we study the computational complexity of mechanism design in the single-agent setting, for the following reasons:

- All of the many concepts of nonmanipulability for multiple agents coincide for the single agent setting, so all results in this setting rely only on the most fundamental properties of nonmanipulability (rather than specific aspects of a particular definition of nonmanipulability);
- It is the simplest version of the mechanism design problem;
- Results here easily extend to settings with multiple agents—as we will demonstrate.

The rest of this paper is organized as follows. In Section 2, we justify our focus on nonmanipulable mechanisms. In Section 3, we define the problem. We study its complexity without payments, for deterministic mechanisms (Section 4) and randomized mechanisms (Section 5). We then study its complexity with payments, for deterministic mechanisms (Section 6) and randomized mechanisms (Section 7).

The next two sections mostly review the relevant well-established definitions and results from game theory; our only contributions in these sections are the *computational* considerations and definitions.

2. JUSTIFYING THE FOCUS ON NONMANIPULABLE MECHANISMS

Before we define the computational problem of single-agent mechanism design, we should justify our focus on nonmanipulable mechanisms. After all, it is not immediately obvious that there are no manipulable mechanisms that, even when agents report their types strategically and hence sometimes untruthfully, still reach better outcomes (according to whichever objective we use) than any nonmanipulable mechanism. Additionally, given our computational focus, we should also be concerned that manipulable mechanisms that do as well as nonmanipulable ones may be easier to construct. It turns out that we need not worry about either of these points: given any mechanism, we can quickly construct a nonmanipulable mechanism whose performance is identical. For given such a mechanism, we can build an interface layer between the agent and this mechanism. The agents report their preferences (or *types*) to the interface layer; subsequently, the interface layer inputs into the original mechanism the types *that the agents would have strategically reported* to the original mechanism, if their types were as declared to the interface layer. The resulting outcome is the outcome of the new mechanism.[1] Since the interface layer acts "strategically on each agent's behalf", there is never an incentive to report falsely to the interface layer; and hence, the types reported by the interface layer are the strategic types that would have been reported without the interface layer, so the results are exactly as they would have been with the original mechanism. This argument (or at least the existential part of it, if not the constructive) is known in the mechanism design literature as the *revelation principle* [11]. Given this, we can focus on truthful mechanisms in the rest of the paper.

3. DEFINITIONS

We now formally define the setting of single-agent mechanism design.

DEFINITION 1. *In a* single-agent mechanism design setting, *we are given a finite set of outcomes O, a finite set of types Θ for the agent together with a probability distribution γ over Θ, a utility function $u : \Theta \times O \to \mathbb{R}$ for the agent,*[2] *and an objective function $g : \Theta \times O \to \mathbb{R}$.*

[1] There are computational considerations to this argument: for instance, an agent's optimization problem for a given type may be hard. Thus employing this argument may shift a hard optimization problem from the agent (where the hardness may have been to the designer's benefit because it made manipulation harder) to the designer (where it is certainly not to the designer's benefit). We study this issue elsewhere [4]. This issue, however, does not affect the results obtained in this paper, because in the representation we use here, an agent's optimization problem is always computationally easy.

[2] Though this follows standard game theory notation [11], the fact that the agent has both a utility function and types is perhaps con-

One typical objective function is the *standard social welfare function* which is simply the sum of the agents' utilities. In settings where there is only one agent, standard social welfare is simply the utility of that agent; so if our objective function is the standard social welfare function, there is no conflict of interest and our mechanism can simply select the outcome with the most utility for each reported type. On the other hand, it is possible that there are one or more *silent agents who do not report types (e.g., because their types are already known) who nevertheless have an interest in the outcome, and whose preferences the mechanism should take into account.* (If the mechanism designer itelf has an interest in the outcome, the mechanism designer can be considered to be such a silent agent.) Because these agents do not need to report their types, we can still phrase this as a single-agent mechanism design setting where the other agents are accounted for by the objective function. This leads to the following definition:

DEFINITION 2. *An objective function g is a* generalized social welfare function *if it is possible to decompose it as $g(\theta, o) = u(\theta, o) + v(o)$, where u is the given utility function for the (type-reporting) agent, and $v : O \rightarrow \mathbb{R}$ is any function (which represents the interests of other agents in the outcome selection).*

We now define the kinds of mechanisms that we will consider.

DEFINITION 3. *A* deterministic single-agent mechanism without payments *consists of an outcome selection function $o : \Theta \rightarrow O$. A* randomized single-agent mechanism without payments *consists of a distribution selection function $p : \Theta \rightarrow \mathcal{P}(O)$, where $\mathcal{P}(O)$ is the set of probability distributions over O. A* deterministic single-agent mechanism with payments *consists of an outcome selection function $o : \Theta \rightarrow O$ and a payment selection function $\pi : \Theta \rightarrow \mathbb{R}$, where $\pi(\theta)$ gives the payment made to the agent when it reports θ. A* randomized single-agent mechanism with payments *consists of a distribution selection function $p : \Theta \rightarrow \mathcal{P}(O \times \mathbb{R})$, where $\mathcal{P}(O)$ is the set of (joint) probability distributions over $O \times \mathbb{R}$.*

Next, we need a definition of what it means for a mechanism to be nonmanipulable. Informally, a mechanism is nonmanipulable if agents never have incentives to misreport their type; but this definition is incomplete without a statement about what the agents may know about each others' types and behavior. Different statements about this lead to different definitions of manipulability (that is, different solution concepts from noncooperative game theory). For example, truthful implementation in *dominant strategies* means that agents have no incentive to manipulate even if they know what the other agents reported. On the other hand, truthful implementation in *Bayes-Nash equilibrium* means that no agent has incentive to manipulate as long as it does not know the other agents' types, and the other agents are reporting truthfully. We have studied the computational complexity of generating mechanisms for multiple agents for these two solution concepts (only in the setting with no payments) [3]. However, there are numerous other solution concepts in noncooperative game theory that we did not cover. Also, one can imagine new solution concepts. For example, we could require that an agent cannot manipulate unless it knows the types of at least k other agents (this is the flavor of nonmanipulability in many cryptographic applications).

Fortunately, one of the main benefits of studying single-agent mechanism design is that in this setting, all sensible notions of nonmanipulability coincide. This is because different nonmanipulability definitions correspond to different statements about what can be known about other agents' types and behavior; but in single-agent mechanism design there are no other agents with types or behaviors pertinent to how the agent should play the game. Thus, all sensible notions of nonmanipulability coincide to the following definition.

DEFINITION 4. *A single-agent mechanism is* nonmanipulable *if for no type, the agent can increase its (expected) utility by reporting another type (instead of the true type). The formal definitions for each type of mechanism are as follows. (In these definitions, the symbol $\leftarrow$ means "drawn from".) A deterministic single-agent mechanism without payments is nonmanipulable if for all $\theta, \hat{\theta} \in \Theta$, $u(\theta, o(\theta)) \geq u(\theta, o(\hat{\theta}))$. A randomized single-agent mechanism without payments is nonmanipulable if for all $\theta, \hat{\theta} \in \Theta$, $E_{o \leftarrow p(\theta)}[u(\theta, o)] \geq E_{o \leftarrow p(\hat{\theta})}[u(\theta, o)]$. In the settings with side payments, we make the common [11] assumption that the agents' utility functions are quasilinear, that is, each agent's utility is linear in money. A deterministic single-agent mechanism with payments is nonmanipulable if for all $\theta, \hat{\theta} \in \Theta$, $u(\theta, o(\theta)) + \pi(\theta) \geq u(\theta, o(\hat{\theta})) + \pi(\hat{\theta})$. A randomized single-agent mechanism with payments is nonmanipulable if for all $\theta, \hat{\theta} \in \Theta$, $E_{(o,\pi) \leftarrow p(\theta)}[u(\theta, o) + \pi] \geq E_{(o,\pi) \leftarrow p(\hat{\theta})}[u(\theta, o) + \pi]$.*

The fact that all notions of nonmanipulability coincide for single-agent mechanism design implies that all results on this topic apply to any notion of nonmanipulability. Now we define the computational problem.

DEFINITION 5. **SINGLE-AGENT-MECHANISM-DESIGN (SAMD)** *We are given a single-agent mechanism design setting,*[3] *the kind of mechanism (deterministic or randomized, with or without payments), and a threshold G. We are asked whether there exists a nonmanipulable mechanism of the given kind such that the expected value of the objective function g is at least G.*

We observe that, without the nonmanipulability constraint (that is, with an agent that always reports truthfully regardless of incentives), the SAMD problem (in any of its forms) is trivial: the optimal mechanism is to simply let the mechanism choose the objective-maximizing outcome for each type.[4] However, as we will see, the problem is harder with the nonmanipulability constraint.

In the rest of the paper, we will analyze the SAMD problem for the four kinds of mechanism (deterministic and randomized; without and with payments).

fusing. The types encode the various possible preferences that the agent may turn out to have, and the agent's type is not known by the aggregator. The utility function is common knowledge, but because the agent's type is a parameter in the agent's utility function, the aggregator cannot know what the agent's utility is without querying the agent about its type.

[3] The setting is given *explicitly*, that is, the outcome set, the type set, the probability distribution over the type set, the utility function, and the objective function all have all their elements or values listed one by one.

[4] In our representation, finding the objective-maximizing outcome for a given type is straightforward.

4. COMPLEXITY OF DESIGNING DETERMINISTIC MECHANISMS WITHOUT PAYMENTS

In this section we will show that the SAMD problem is $\mathcal{NP}$-complete for deterministic single-agent mechanisms without payments. To demonstrate $\mathcal{NP}$-hardness, we reduce from the MINSAT problem.

DEFINITION 6 (MINSAT). *We are given a formula ϕ in conjunctive normal form, represented by a set of Boolean variables V and a set of clauses C, and an integer k ($k < |C|$). We are asked whether there exists an assignment to the variables in V such that at most k clauses in ϕ are satisfied.*

MINSAT was recently shown to be $\mathcal{NP}$-complete [10]. We are now ready to present our result.

THEOREM 1. *SAMD with deterministic mechanisms without payments is $\mathcal{NP}$-complete, even when the objective function is a generalized social welfare function and the probability distribution over Θ is uniform.*

PROOF. The problem is in $\mathcal{NP}$ because we can nondeterministically generate an outcome selection function, and subsequently verify in polynomial time whether it is nonmanipulable, and whether the expectation of the objective function achieves the threshold. To show that the problem is $\mathcal{NP}$-hard, we reduce an arbitrary MINSAT instance to a SAMD instance as follows.

Let the outcomes O be as follows. For every clause $c \in C$, there is an outcome o_c. For every variable $v \in V$, there is an outcome o_v and an outcome o_{-v}. Finally, there is a single additional outcome o_b.

Let L be the set of literals, that is, $L = \{v : v \in V\} \cup \{-v : v \in V\}$. Then, let the type space Θ be as follows. For every clause $c \in C$, there is a type θ_c. For every variable $v \in V$, there is a type θ_v. The probability distribution over Θ is uniform.

Let the utility function be as follows:

- $u(\theta_v, o_v) = u(\theta_v, o_{-v}) = |C| + 3$ for all $v \in V$;
- $u(\theta_c, o_l) = 1$ for all $c \in C$ and $l \in c$ (that is, l is a literal that occurs in c);
- $u(\theta_c, o_c) = 1$ for all $c \in C$;
- u is 0 everywhere else.

Let $g(\theta, o) = u(\theta, o) + v(o)$, where $v(o_b) = 2$ and v is 0 everywhere else. (Note that g is a generalized social welfare function.) Finally, let $G = \frac{|V|(|C|+3)+2|C|-k}{|V|+|C|}$ (k is the threshold of the MINSAT instance). We claim that the SAMD instance has a solution if and only if the MINSAT instance has a solution.

First suppose that the MINSAT instance has a solution, that is, an assignment to the variables that satisfies at most k clauses. Then consider the following mechanism. If $v \in V$ is set to *true* in the assignment, then set $o(\theta_v) = o_v$; if it is set to *false*, then set $o(\theta_v) = o_{-v}$. If $c \in C$ is satisfied by the assignment, then set $o(\theta_c) = o_c$; if it is not satisfied, then set $o(\theta_c) = o_b$. First we show that this mechanism is nonmanipulable. If the agent's type is either any one of the θ_v or one of the θ_c corresponding to a satisfied clause c, then the mechanism gives the agent the maximum utility it can possibly get with that type, so there is no incentive for the agent to misreport. On the other hand, if the agent's type is one of the θ_c corresponding to a nonsatisfied clause c, then any outcome o_l corresponding to a literal l in c, or o_c, would give utility 1, as opposed to o_b (which the mechanism actually chooses for θ_c) which gives the agent utility 0. It follows that the mechanism is nonmanipulable if and only if there is no other θ such that $o(\theta)$ is any outcome o_l corresponding to a literal l in c, or o_c. It is easy to see that there is indeed no θ such that $o(\theta) = o_c$. There is also no θ such that $o(\theta)$ is any outcome o_l corresponding to a literal l in c: this is because the only type that could possibly give the outcome o_l is θ_v, where v is the variable corresponding to l; but because c is not satisfied in the assignment to the variables, we know that actually, $o(\theta_v) = o_{-l}$ (that is, the outcome corresponding to the opposite literal is chosen). It follows that the mechanism is indeed nonmanipulable. All that is left to show is that the expected value of $g(\theta, o(\theta))$ reaches G. For any θ_v we have $g(\theta_v, o(\theta_v)) = |C| + 3$. For any θ_c where c is a satisfied clause, we have $g(\theta_c, o(\theta_c)) = 1$. Finally, for any θ_c where c is an unsatisfied clause, we have $g(\theta_c, o(\theta_c)) = 2$. If s is the number of satisfied clauses, then, using the facts that the probability distribution over Θ is uniform and that $s \leq k$, we have $E[g(\theta, o(\theta))] = \frac{|V|(|C|+3)+s+2(|C|-s)}{|V|+|C|} \geq \frac{|V|(|C|+3)+2|C|-k}{|V|+|C|} = G$. So there is a solution to the SAMD instance.

Now suppose there is a solution to the SAMD instance, that is, a nonmanipulable mechanism given by an outcome function $o : \Theta \rightarrow O$, which leads to an expected value of $g(\theta, o(\theta))$ of at least G. We observe that the maximum value that we can get for $g(\theta, o(\theta))$ is $|C|+3$ when θ is one of the θ_v, and 2 otherwise. Thus, if for some v it were the case that $o(\theta_v) \notin \{o_v, o_{-v}\}$ and hence $g(\theta, o(\theta)) \leq 2$, it would follow that $E[g(\theta, o(\theta))]$ can be at most $\frac{(|V|-1)(|C|+3)+2(|C|+1)}{|V|+|C|} < \frac{(|V|)(|C|+3)+|C|}{|V|+|C|} < \frac{|V|(|C|+3)+2|C|-k}{|V|+|C|} = G$ (because $k < |C|$). (Contradiction.) It follows that for all v, $o(\theta_v) \in \{o_v, o_{-v}\}$. From this we can derive an assignment to the variables: set v to *true* if $o(\theta_v) = o_v$, and to *false* if $o(\theta_v) = o_{-v}$. We claim this assignment is a solution to the MINSAT instance for the following reason. If a clause c is satisfied by this assignment, there is some literal l such that $l \in c$ and $o(\theta_v) = o_l$ for the corresponding variable v. But then $o(\theta_c)$ cannot be o_b, because if it were, the agent would be motivated to report θ_v when its true type is θ_c, to get a utility of 1 as opposed to the 0 it would get for reporting truthfully. Hence $g(\theta_c, o(\theta_c))$ can be at most 1 for a satisfied clause c. It follows that $E[g(\theta, o(\theta))]$ can be at most $\frac{|V|(|C|+3)+s+2(|C|-s)}{|V|+|C|}$ where s is the number of satisfied clauses. But because $E[g(\theta, o(\theta))] \geq G$, we can conclude $\frac{|V|(|C|+3)+s+2(|C|-s)}{|V|+|C|} \geq G = \frac{|V|(|C|+3)+2|C|-k}{|V|+|C|}$, which is equivalent to $s \leq k$. So there is a solution to the MINSAT instance. □

In an earlier paper we showed that designing an optimal deterministic mechanism for 2 agents is $\mathcal{NP}$-complete even when the objective function is the *standard* social welfare function [3]. We showed this both for implementation in dominant strategies and for implementation in Bayes-Nash equilibrium. We will conclude this section by demonstrating the power of hardness results for single-agent mechanism design, by showing that Theorem 1 immediately implies (the hardness parts of) both of these earlier results.

We will not formally redefine either the 2-agent mechanism design problem or implementation in dominant strategies/Bayes-Nash equilibrium. All that is necessary to know is that both solution concepts coincide to the nonmanipulability concept for the single-agent mechanism design problem.

COROLLARY 1. *Designing a deterministic mechanism without payments for 2 (or more) agents is $\mathcal{NP}$-hard for implementation in dominant strategies and for implementation in Bayes-Nash equilibrium (and in fact for any solution concept that coincides with nonmanipulability in the single-agent case), even when the objective function is the standard social welfare function (where there are no silent agents).*

PROOF. There are two reasons why the SAMD problem we analyzed is not immediately a special case of 2-agent mechanism design with the standard social welfare function as an objective. First, we have one agent too few. Second, we allowed for more general social welfare functions with an outside component corresponding to silent agents' interests. We solve these problems by reducing an arbitrary SAMD instance with a generalized social welfare function as objective function, to a 2-agent mechanism design instance with the standard social welfare function, as follows. We introduce a dummy agent that has only one type; its utility (given this type) is simply the outside component of the SAMD-instance's generalized social welfare function. Because an outcome function here cannot depend on the dummy agent's type (because it is constant), it corresponds naturally to an outcome function for the SAMD-instance. An outcome function in the 2-agent mechanism design instance is nonmanipulable (for any of the nonmanipulability concepts) if and only if the corresponding outcome function for the SAMD-instance is nonmanipulable, because the dummy agent can never manipulate, and the nonmanipulability concept coincides with the SAMD concept for the original agent. Furthermore, the social welfare is the same in both cases because the outside component in the SAMD-instance has been incorporated into the dummy agent. Thus the problem instances are equivalent.

For the case of more than two agents, we simply add more dummy agents that have one type each, and utility zero for all outcomes. □

5. COMPLEXITY OF DESIGNING RANDOMIZED MECHANISMS WITHOUT PAYMENTS

In this section we show that if we allow the mechanism to select the outcome randomly on the basis of the reported type, the SAMD problem without payments becomes easy.

THEOREM 2. *SAMD with randomized mechanisms without payments can be done in polynomial time using linear programming (even for general objective functions).*

PROOF. We can generate an optimal mechanism using linear programming as follows. For each $\theta \in \Theta$, we need to choose a probability distribution $p(\theta)$ over the outcomes in O. Such a probability function is defined by a probability $(p(\theta))(o)$ for each $o \in O$. These will be the variables in the linear program. In addition to the constraints necessary to enforce that each $p(\theta)$ is a probability distribution, we need the following constraints to ensure nonmanipulability: for each $\theta, \hat{\theta} \in \Theta$, we must have $\sum\limits_{o \in O} (p(\theta))(o)u(\theta, o) \geq \sum\limits_{o \in O} (p(\hat{\theta}))(o)u(\theta, o)$. We seek to maximize the expected value of g, which is $\sum\limits_{\theta \in \Theta} \gamma(\theta) \sum\limits_{o \in O} (p(\theta))(o)g(\theta, o)$. Observing that all the constraints and the objective are linear in the variables $(p(\theta))(o)$, we conclude that this is a linear program. Because there is a polynomial number of constraints and variables, we conclude that this program can be solved in polynomial time. □

For any specific solution concept, this linear program can easily be generalized to multiple agents. The size of the program is exponential in the number of agents, but for any constant number of agents, the problem is polynomial in size.[5]

6. COMPLEXITY OF DESIGNING DETERMINISTIC MECHANISMS WITH PAYMENTS

We first show that when the objective is generalized social welfare, allowing for payments makes the SAMD problem easy even when randomization in the mechanism is not possible.

THEOREM 3. *When g is a generalized social welfare function and the agents' preferences are quasilinear, there exists a nonmanipulable single-agent mechanism (with payments) that, for any θ, selects an outcome $o(\theta)$ that maximizes $g(\theta, o(\theta))$, and hence achieves the maximum possible expectation of this function. Such a mechanism can be constructed in polynomial time. So, SAMD with deterministic mechanisms with payments is in $\mathcal{P}$ when g is a generalized social welfare function.*

PROOF. Let $o(\theta) = \arg\max_{o \in O} g(\theta, o)$ (if multiple $o \in O$ maximize this expression, choose one arbitrarily). Because g is a generalized social welfare function, we know it can be decomposed as $g(\theta, o) = u(\theta, o) + v(o)$. Let $\pi(\theta) = v(o(\theta))$. Clearly, this mechanism can be constructed in $O(|\Theta||O|)$ time. All we need to show is that it is indeed nonmanipulable. If the agent has type θ, it will report $\hat{\theta}$ to maximize $u(\theta, o(\hat{\theta})) + \pi(\hat{\theta}) = u(\theta, o(\hat{\theta})) + v(o(\hat{\theta})) = g(\theta, o(\hat{\theta}))$. But $o(\theta)$ is chosen to maximize $g(\theta, o)$ over all $o \in O$, and hence reporting θ is optimal for the agent. □

The mechanism constructed in the proof belongs to the more general class of *Groves mechanisms* [9], which are designed (even in multiagent settings) to allow the mechanism to choose the social welfare maximizing outcome while still guaranteeing implementation in dominant strategies. (In fact, over general quasilinear preferences, they are the only mechanisms with this property [8].) However, the mechanism is not a *Clarke mechanism* [2], the most common Groves mechanism. In the Clarke mechanism, even the silent agents may have to make payments (and the payments collected cannot be redistributed among the agents because this would compromise the incentives), while in our mechanism there are no transfers from the silent agents. For the type-reporting agent, our mechanism can be made to coincide with the Clarke mechanism by subtracting a constant payment $\max_{o \in O} v(o)$ from that agent.

[5] In our earlier paper, we provide linear programs for solving 2-agent mechanism design without payments with randomization, for implementation in dominant strategies and in Bayes-Nash equilibrium [3]. The linear program presented above can be derived from either of these programs, again by adding a dummy agent into the SAMD instance.

Unfortunately, SAMD turns out to be $\mathcal{NP}$-complete for general objective functions. To demonstrate $\mathcal{NP}$-hardness, we reduce from the INDEPENDENT-SET problem which is $\mathcal{NP}$-complete [14].

DEFINITION 7 (INDEPENDENT-SET). *We are given an undirected graph (V, E) (with no self-loops, that is, edges that begin and end at the same node) and an integer k. We are asked whether there is some $I \subseteq V$ of size at least k such that no two elements of I have an edge between them.*

We are now ready to show hardness of mechanism design even with quasilinear preferences.

THEOREM 4. *SAMD with deterministic mechanisms with payments is $\mathcal{NP}$-complete (with general objective functions), even when the probability distribution over Θ is uniform.*

PROOF. First we show that the problem is in $\mathcal{NP}$. We can nondeterministically generate an outcome function o. We then check whether the payment function π can be set so as to make the mechanism nonmanipulable. Because we have already generated o, we can phrase this problem as a linear program with the following constraints: for all $\theta, \hat{\theta} \in \Theta$, $u(\theta, o(\theta)) + \pi(\theta) \geq u(\theta, o(\hat{\theta})) + \pi(\hat{\theta})$. If the linear program has a solution, we subsequently check if the corresponding mechanism achieves the threshold G for $E[g(\theta, o(\theta))]$.

To show that the problem is $\mathcal{NP}$-hard, we reduce an arbitrary INDEPENDENT-SET instance to a SAMD instance as follows. For every vertex $v \in V$, let there be outcomes o_v^1 and o_v^2, and a type θ_v. The probability distribution over Θ is uniform. Let the utility function be as follows:

- $u(\theta_v, o_w^1) = 1$ for all $v, w \in V$ with $(v, w) \in E$;
- $u(\theta_v, o_w^1) = 0$ for all $v, w \in V$ with $(v, w) \notin E$ (this includes all cases where $v = w$ as there are no self-loops in the graph);
- $u(\theta_v, o_v^2) = 1$ for all $v \in V$;
- $u(\theta_v, o_w^2) = 0$ for all $w \in V$ with $v \neq w$.

Let the objective function be $g(\theta_v, o_v^1) = 1$ for all $v \in V$, and $g() = 0$ everywhere else. Finally, let $G = \frac{k}{|V|}$ (where k is the threshold of the INDEPENDENT-SET instance). We claim that the SAMD instance has a solution if and only if the INDEPENDENT-SET instance has a solution.

First suppose that the INDEPENDENT-SET instance has a solution, that is, some $I \subseteq V$ of size at least k such that no two elements of I have an edge between them. Then consider the following mechanism. For all $v \in I$, let $o(\theta_v) = o_v^1$. For all $v \notin V$, let $o(\theta_v) = o_v^2$. Let π be zero everywhere (no payments are made). First we show that this mechanism is indeed nonmanipulable. If $v \in I$ and $w \in I$, then (because I is an independent set) $(v, w) \notin I$, and thus $u(\theta_v, o(\theta_v)) + \pi(\theta_v) = u(\theta_v, o_v^1) = 0 = u(\theta_v, o_w^1) = u(\theta_v, o(\theta_w)) + \pi(\theta_w)$. If $v \in I$ and $w \notin I$, then $u(\theta_v, o(\theta_v)) + \pi(\theta_v) = u(\theta_v, o_v^1) = 0 = u(\theta_v, o_w^2) = u(\theta_v, o(\theta_w)) + \pi(\theta_w)$. Finally, if $v \notin I$, then $u(\theta_v, o(\theta_v)) + \pi(\theta_v) = u(\theta_v, o_v^2) = 1$, which is the highest possible value the agent can attain. So there is no incentive for the agent to misreport anywhere. All that is left to show is that the expected value of $g(\theta, o(\theta))$ reaches G. For $v \in I$, $g(\theta, o(\theta)) = g(\theta, o_v^1) = 1$, and for $v \notin I$, $g(\theta, o(\theta)) = g(\theta, o_v^2) = 0$. Because the distribution over Θ is uniform, it follows that $E[g(\theta, o(\theta))] = \frac{|I|}{|V|} \geq \frac{k}{|V|} = G$. So there is a solution to the SAMD instance.

Now suppose there is a solution to the SAMD instance, that is, a nonmanipulable mechanism given by an outcome function $o : \Theta \rightarrow O$ and a payment function $\pi : \Theta \rightarrow \mathbb{R}$, which leads to an expected value of $g(\theta, o(\theta))$ of at least G. Let $I = \{v : o(\theta) = o_v^1\}$. We claim I is a solution to the INDEPENDENT-SET instance. First, because $g(\theta_v, o(\theta_v))$ is 1 only for $v \in I$, we know that $\frac{k}{|V|} = G \leq E[g(\theta, o(\theta))] = \frac{|I|}{|V|}$, or equivalently, $|I| \geq k$. All that is left to show is that there are no edges between elements of I. Suppose there were an edge between $v, w \in I$. Without loss of generality, say $\pi(\theta_v) \leq \pi(\theta_w)$. Then, $u(\theta_v, o(\theta_v)) + \pi(\theta_v) = u(\theta_v, o_v^1) + \pi(\theta_v) = \pi(\theta_v) \leq \pi(\theta_w) < 1 + \pi(\theta_w) = u(\theta_v, o_w^1) + \pi(\theta_w) = u(\theta_v, o(\theta_w)) + \pi(\theta_w)$. So the agent has an incentive to misreport when its type is θ_v, which contradicts the nonmanipulability of the mechanism. It follows that there are no edges between elements of I. So there is a solution to the INDEPENDENT-SET instance. □

Again, this generalizes to multiple agents:

COROLLARY 2. *Designing a deterministic mechanism with payments for 2 (or more) agents is $\mathcal{NP}$-hard for implementation in dominant strategies and for implementation in Bayes-Nash equilibrium (and in fact for any solution concept that coincides with nonmanipulability in the single-agent case) for general objective functions, even if the agents' preferences are quasilinear.*

PROOF. The proof introduces one (or more) dummy agent(s) and is analogous to the proof of Corollary 1 (though slightly simpler because here we do not need to make the social welfare functions coincide). □

We observe that in the proof of Theorem 4, the objective still depends on the agent's type. If it does not—that is, the objective only represents the designer's own preferences over outcomes—we are simply in the setting of a general social welfare function again, which we can solve in polynomial time as pointed out above.

7. COMPLEXITY OF DESIGNING RANDOMIZED MECHANISMS WITH PAYMENTS

In this section we show that if we allow the mechanism to select the outcome randomly on the basis of the reported type, the SAMD problem with payments becomes easy for general objective functions. (In Section 6 we showed that when the objective function is a generalized social welfare function, we can quickly generate a nonmanipulable mechanism with payments that selects an objective-maximizing outcome, so in that case randomization is not necessary.)

THEOREM 5. *SAMD with randomized mechanisms with payments can be done in polynomial time using linear programming (even for general objective functions).*

PROOF. We can generate an optimal mechanism using linear programming as follows. We first observe that as far as payments are concerned, by linearity of expectation, the agent only cares about the expected payment it gets given that it reports a given type; so there is no reason to randomize over payments at all. Thus, for each $\theta \in \Theta$, we need to choose a probability distribution $p(\theta)$ over the outcomes in O, defined by a probability $(p(\theta))(o)$ for each $o \in O$; and a payment $\pi(\theta)$. These will be the variables in the linear program. In addition to the constraints necessary to enforce that each $p(\theta)$ is a probability distribution, we need the following constraints to ensure nonmanipulability: for each $\theta, \hat{\theta} \in \Theta$, we must have $(\sum\limits_{o \in O} (p(\theta))(o)u(\theta, o)) + \pi(\theta) \geq (\sum\limits_{o \in O} (p(\hat{\theta}))(o)u(\theta, o)) + \pi(\hat{\theta})$. Then, we seek to maximize the expected value of g, which is $\sum\limits_{\theta \in \Theta} \gamma(\theta) \sum\limits_{o \in O} (p(\theta))(o)g(\theta, o)$. Observing that all the constraints and the objective are linear in the $(p(\theta))(o)$ and the $\pi(\theta)$, we conclude that this is a linear program. Because there is a polynomial number of constraints and variables, we conclude that this program can be solved in polynomial time. □

The linear program in the proof can easily be generalized to settings with multiple agents, for various concepts of nonmanipulability (e.g., implementation in dominant strategies, or in Bayes-Nash equilibrium).

8. RELATED RESEARCH ON COMPUTATIONAL COMPLEXITY IN MECHANISM DESIGN

There has been considerable recent interest in mechanism design in computer science. Some of it has focused on issues of computational complexity, but most of that work has strived toward designing mechanisms that are easy to *execute* (e.g. [6,12,13]), rather than studying the complexity of *designing* the mechanism. The closest piece of earlier work is our paper which studied the complexity of *multi*agent mechanism design *without side payments* and *only for two specific notions of nonmanipulability* [3]. Roughgarden has studied the complexity of designing a good network topology for agents that selfishly choose the links they use [15]. This is related to mechanism design, but differs significantly in that the designer only has restricted control over the rules of the game because there is no party that can impose the outcome (or side payments). Also, there is no explicit reporting of preferences.

9. CONCLUSIONS AND FUTURE RESEARCH

The aggregation of conflicting preferences is a central problem in multiagent systems. The key difficulty is that the agents may report their preferences insincerely. Mechanism design is the art of designing the rules of the game so that the agents are motivated to report their preferences truthfully and a (socially) desirable outcome is chosen.

We propose an approach where a mechanism is automatically created for the preference aggregation setting at hand. This approach can be used even in settings that do not satisfy the assumptions of classical mechanisms. It may also yield better mechanisms (in terms of stronger nonmanipulability guarantees and/or better outcomes) than classical mechanisms. Finally, it may allow one to circumvent impossibility results (such as the Gibbard-Satterthwaite theorem) which state that there is no mechanism that is desirable across a whole class of preferences.

The downside is that the mechanism design problem needs to be solved anew each time. Hence, the computational complexity of mechanism design becomes a key issue. In this paper, we analyzed the single-agent mechanism design problem, for the following reasons: 1) All of the many concepts of nonmanipulability for multiple agents coincide for the single agent setting, so all results in this setting rely only on the most fundamental properties of nonmanipulability; 2) It is the simplest version of the mechanism design problem; 3) Results here easily extend to settings with multiple agents—as we demonstrated.

We showed that designing an optimal deterministic mechanism that does not use payments is $\mathcal{NP}$-complete even if there is only one agent whose type is private information—even when the designer's objective is social welfare. We showed how this hardness result extends to settings with multiple agents with private information. We then showed that if the mechanism is allowed to use randomization, the design problem is solvable by linear programming (even for general objectives) and hence in $\mathcal{P}$. This generalizes to any fixed number of agents.

We then studied settings where side payments are possible and the agents' preferences are quasilinear. We showed that if the designer's objective is social welfare, an optimal deterministic mechanism is easy to construct; in fact, this mechanism is also *ex post* optimal. We then showed that designing an optimal deterministic mechanism with side payments is $\mathcal{NP}$-complete for general objectives, and this hardness extends to settings with multiple agents. Finally, we showed that an optimal randomized mechanism can be designed in polynomial time using linear programming even for general objective functions. This again generalizes to any fixed number of agents.

Future research includes extending the approach of automated mechanism design to other settings, for example, settings where side payments are possible but the agents' preferences are not quasilinear, and settings where each agent's type space is very large or infinite (but concisely representable). Another interesting use of automated mechanism design is to solve for mechanisms for a variety of settings (real or artificially generated), and to see whether general mechanisms (or mechanism design principles) can be inferred. Finally, this approach could be used to generate counterexamples or corroborate that a postulated mechanism is optimal for a given class of settings.

10. REFERENCES

[1] Kenneth Arrow. The property rights doctrine and demand revelation under incomplete information. In M Boskin, editor, *Economics and human welfare*. New York Academic Press, 1979.

[2] E H Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[3] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 103–110, Edmonton, Canada, 2002.

[4] Vincent Conitzer and Tuomas Sandholm. Computational criticisms of the revelation principle. In *the AAMAS-03 5th Workshop on Agent Mediated Electronic Commerce (AMEC V)*, Melbourne, Australia, 2003.

[5] C d’Aspremont and L A Gérard-Varet. Incentives and incomplete information. *Journal of Public Economics*, 11:25–45, 1979.

[6] Joan Feigenbaum, Christos Papadimitriou, and Scott Shenker. Sharing the cost of mulিticast transmissions. *Journal of Computer and System Sciences*, 63:21–41, 2001. Early version in STOC-00.

[7] A Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[8] J Green and J-J Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.

[9] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[10] R Kohli, R Krishnamurthi, and P Mirchandani. The minimum satisfiability problem. *SIAM Journal of Discrete Mathematics*, 7(2):275–283, 1994.

[11] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[12] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 242–252, Minneapolis, MN, 2000.

[13] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001. Early version in STOC-99.

[14] Christos H Papadimitriou. *Computational Complexity*. Addison-Wesley, 1995.

[15] Tim Roughgarden. Designing networks for selfish users is hard. In *FOCS*, 2001.

[16] M A Satterthwaite. Strategy-proofness and Arrow’s conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[17] W Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.