Expressive Negotiation in Settings with Externalities*

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

Abstract

In recent years, certain formalizations of combinatorial negotiation settings, most notably combinatorial auctions, have become an important research topic in the AI community. A pervasive assumption has been that of *no externalities*: the agents deciding on a variable (such as whether a trade takes place between them) are the only ones affected by how this variable is set. To date, there has been no widely studied formalization of combinatorial negotiation settings with externalities. In this paper, we introduce such a formalization. We show that in a number of key special cases, it is NP-complete to find a feasible nontrivial solution (and therefore the maximum social welfare is completely inapproximable). However, for one important special case, we give an algorithm which converges to the solution with the maximal concession by each agent (in a linear number of rounds for utility functions that decompose into piecewise constant functions). Maximizing social welfare, however, remains NP-complete even in this setting. We also demonstrate a special case which can be solved in polynomial time by linear programming.

Introduction

One key problem in multiagent settings is that of *preference aggregation*, where the agents must collectively choose one *outcome* from a set of candidate outcomes, based on the agents' individual preferences. Oftentimes, this outcome space is highly combinatorial in nature. For instance, in a *combinatorial auction* (see, e.g., (Rothkopf, Pekeč, & Harstad 1998; Fujishima, Leyton-Brown, & Shoham 1999; Sandholm 2002)), multiple items are to be allocated to the agents, so an outcome is defined by a specification of which bundle of items each agent gets (plus, perhaps, payments to be made by or to the agents). The (usual) goal of preference aggregation is to find an outcome that is good for the agents in aggregate (for instance, one that has high *social welfare*, defined as the sum of the agents' utilities). However, in most settings, there are additional constraints that must be satisfied. Typically, there is a *participation constraint* that no agent is made worse off by participating in the preference aggregation protocol. Additionally, if only the agents themselves know their preferences, and the agents are self-interested (the setting of *mechanism design*), there may be an *incentive compatibility* constraint: no agent should be able to make itself better off by misreporting its preferences.[1] The goal is to find a preference aggregation function (mechanism), mapping (reported) preferences to outcomes, that is desirable (for instance, one that leads to high social welfare) while satisfying the desired constraints. Additionally, we should have an algorithm for *computing* this function. (Computing the most desirable preference aggregation functions tends to be very difficult in combinatorial settings, so the goal is to find a good function that can still be computed efficiently. In mechanism design, this process has been called *algorithmic mechanism design*, and this line of research has produced a number of interesting results (Nisan & Ronen 2001; 2000; Feigenbaum, Papadimitriou, & Shenker 2001; Lehmann, O'Callaghan, & Shoham 2002; Mu'alem & Nisan 2002; Archer *et al.* 2003; Bartal, Gonen, & Nisan 2003).)

Combinatorial auctions are by far the most-studied combinatorial preference aggregation setting. To a lesser extent, some variants such as combinatorial exchanges (where the agents seek to trade items among each other—see, e.g., (Parkes, Kalagnanam, & Eso 2001; Sandholm *et al.* 2002)) have also received attention. A pervasive assumption in all of this work has been that there are *no allocative externalities*: no agent cares what happens to an item unless that agent itself receives the item. This is insufficient to model situations where there are certain items (such as nuclear weapons) that are such that bidders who do not win the item still care which other bidder wins it (Jehiel & Moldovanu 1996). More generally, there are many important preference aggregation settings where decisions taken by a few agents may affect many other agents. For example, many agents may benefit from one agent taking on a task such as building a bridge (and the extent of their benefit may depend on how the bridge is built, for example, on

*This material is based upon work supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship.

[1] In mechanism design, this constraint is motivated by the *revelation principle* (e.g. (Mas-Colell, Whinston, & Green 1995)), which can be informally stated as follows. For every mechanism that achieves a certain objective in the face of strategic agents, there exists another, *truthful* mechanism (that is, a mechanism satisfying the incentive compatibility constraint) that achieves the same objective.

how heavy a load it can support). Alternatively, if a company reduces its pollution level, many individuals may benefit, even if they have nothing to do with the goods that the company produces. A decision's effect on an otherwise uninvolved agent is commonly known as an *externality* (Mas-Colell, Whinston, & Green 1995). In designing a good preference aggregation function, externalities must be taken into account, so that (potentially complex) arrangements can be made that are truly to every agent's benefit.

In this paper, we define a representation for combinatorial preference aggregation settings with externalities. Under various assumptions, we study the computational complexity of, given the agents' preferences, finding a good (if possible, the optimal) outcome that honors the participation constraint (no agent should be made worse off). We will mostly focus on restricted settings that cannot model e.g. fully general combinatorial auctions and exchanges (because problems in those settings are hard even without externalities). Also, in this first research, we do not consider any incentive compatibility constraints, and take the agents' reported preferences at face value. This is reasonable when the agents' preferences are common knowledge; when there are other reasons to believe that the agents' preferences are reported truthfully (for example, for ethical reasons, or because the party reporting the preferences is concerned with the global welfare rather than the agent's individual utility);[2] or when we are simply interested in finding outcomes that are good relative to the reported preferences (for example, because we are an optimization company that gets rewarded based on how good the outcomes that we produce are relative to the reported preferences). Nevertheless, mechanism design aspects of our setting are an important issue, and we will discuss them as a topic of future research.

Definitions

We formalize the problem setting as follows.

Definition 1 *In a* setting with externalities, *there are* n *agents* $1, 2, \ldots, n$*; each agent* i *controls* m_i *variables* $x_i^1, x_i^2, \ldots, x_i^{m_i} \in \mathbb{R}^{\geq 0}$*; and each agent* i *has a utility function* $u_i : \mathbb{R}^M \to \mathbb{R}$ *(where* $M = \sum_{j=1}^{n} m_j$*). (Here,* $u_i(x_1^1, \ldots, x_1^{m_1} \ldots, x_n^1, \ldots, x_n^{m_n})$ *represents agent* i*'s utility for any given setting of the variables.)*

In general, one can also impose constraints on which values for $(x_i^1, \ldots, x_i^{m_i})$ agent i can choose, but we will refrain from doing so in this paper. (We can effectively exclude certain settings by making the utilities for them very negative.) We say that the *default outcome* is the one where all the x_i^j are set to 0,[3] and we require without loss of generality that all agents' utilities are 0 at the default outcome. Thus, the participation constraint states that every agent's utility should be nonnegative.

Without any restrictions placed on it, this definition is very general. For instance, we can model a (multi-item, multi-unit) combinatorial exchange with it. In a combinatorial exchange, each agent has an initial endowment of a number of units of each item, as well as preferences over endowments (possibly including items not currently in the agent's possession). The goal is to find some reallocation of the items (possibly together with a specification of payments to be made and received) so that no agent is left worse off, and some objective is maximized under this constraint. We can model this in our framework as follows: for each agent, for each item in that agent's possession, for each other agent, let there be a variable representing how many units of that item the former agent transfers to the latter agent. (If payments are allowed, then we additionally need variables representing the payment from each agent to each other agent.) We note that this framework allows for *allocative externalities*, that is, for the expression of preferences over which of the other agents receives a particular item.

Of course, if the agents can have nonlinear preferences over bundles of items (there are complementarities or substitutabilities among the items), then (barring some special concise representation) specifying the utility functions requires an exponential number of values.[4] We need to make some assumption about the structure of the utility functions if we do not want to specify an exponential number of values. In (most of) this paper, we make the following assumption, which states that the effect of one variable on an agent's utility is independent of the effect of another variable on that agent's utility. We note that this assumption disallows the model of a combinatorial exchange that we just gave, unless there are no complementarities or substitutabilities among the items. This is not a problem insofar as our primary interest here is not so much in combinatorial exchanges as it is in more natural, simpler externality problems such as negotiation over pollution levels. We note that this restriction makes the hardness results that we present later much more interesting (without the restriction, the results would have been unsurprising given known hardness results in combinatorial exchanges). However, for some of our positive results we will actually not need the assumption, for example for convergence results for our algorithm.

Definition 2 u_i decomposes (across variables) *if* $u_i(x_1^1, \ldots, x_1^{m_1}, \ldots, x_n^1, \ldots, x_n^{m_n}) = \sum_{k=1}^{n} \sum_{j=1}^{m_k} u_i^{k,j}(x_k^j)$.

When utility functions decompose, we will sometimes be interested in the special cases where the $u_i^{k,j}$ are step functions (denoted $\delta_{x \geq a}$, which evaluates to 0 if $x < a$ and to

[2] For example, in a large organization, when a representative of a department within the organization is asked what the department's needs are, it is possible that this representative acts in the organization's best interest (rather than the department's).

[3] This is without loss of generality because the variables x_i^j can be used to represent the *changes* in the real-world variables relative to the default outcome. If these changes can be both positive and negative for some real-world variable, we can model this with two variables $x_i^{j_1}, x_i^{j_2}$, the difference between which represents the change in the real-world variable.

[4] Thus, the fact that finding a feasible solution for a combinatorial exchange is NP-complete (Sandholm *et al.* 2002) does not imply that finding a feasible solution in our framework is NP-complete, because there is an exponential blowup in representation.

1 otherwise), or piecewise constant functions (linear combinations of step functions).[5]

In addition, we will focus strictly on settings where the higher an agent sets its variables, the worse it is for itself. We will call such settings *concessions settings*. So, if there is no negotiation, each agent will selfishly set all its variables to 0. This also provides a game-theoretic justification of the participation constraint that every agent's utility should be nonnegative: specifically, if any agent can block the negotiation process, then any agent that would receive negative utility from the negotiation process would do so.

Definition 3 *A* concessions setting *is a setting with externalities for which for any $(x_1^1, \ldots, x_1^{m_1}, \ldots, x_n^1, \ldots, x_n^{m_n}) \in \mathbb{R}^M$, for any $i, 1 \leq j \leq m_i$, and for any $\hat{x}_i^j > x_i^j$, we have $u_i(x_1^1, \ldots, x_1^{m_1}, \ldots, \hat{x}_i^j, \ldots, x_n^1, \ldots, x_n^{m_n}) \leq u_i(x_1^1, \ldots, x_1^{m_1}, \ldots, x_i^j, \ldots, x_n^1, \ldots, x_n^{m_n})$.*

In parts of this paper, we will be interested in the following additional assumption, which states that the higher an agent sets its variables, the better it is for the others. (For instance, the more a company reduces its pollution, the better it is for all others involved.)

Definition 4 *A concessions setting has* only negative externalities *if for any $(x_1^1, \ldots, x_1^{m_1}, \ldots, x_n^1, \ldots, x_n^{m_n}) \in \mathbb{R}^M$, for any $i, 1 \leq j \leq m_i$, for any $\hat{x}_i^j > x_i^j$, and for any $k \neq i$, $u_k(x_1^1, \ldots, x_1^{m_1}, \ldots, \hat{x}_i^j, \ldots, x_n^1, \ldots, x_n^{m_n}) \geq u_k(x_1^1, \ldots, x_1^{m_1}, \ldots, x_i^j, \ldots, x_n^1, \ldots, x_n^{m_n})$.*

We define *trivial* settings of variables as settings that are indistinguishable from setting them to 0.

Definition 5 *The value r is* trivial *for variable x_i^j if it does not matter to anyone's utility function whether x_i^j is set to r or to 0. (That is, for any $x_1^1, \ldots, x_1^{m_1}, \ldots, x_i^{j-1}, x_i^{j+1}, \ldots, x_n^1, \ldots, x_n^{m_n}$, and for any k, we have $u_k(x_1^1, \ldots, x_1^{m_1}, \ldots, x_i^{j-1}, r, x_i^{j+1}, \ldots, x_n^1, \ldots, x_n^{m_n}) = u_k(x_1^1, \ldots, x_1^{m_1}, \ldots, x_i^{j-1}, 0, x_i^{j+1}, \ldots, x_n^1, \ldots, x_n^{m_n})$. A setting of all the variables is* trivial *if each variable is set to a trivial value.*

We are now ready to define the following two computational problems that we will study. Say that an outcome is *feasible* if it honors the participation constraint, that is, no agent prefers the default outcome to it.

Definition 6 (FEASIBLE-CONCESSIONS) *We are given a concessions setting. We are asked whether there exists a nontrivial feasible solution.*

Definition 7 (SW-MAXIMIZING-CONCESSIONS) *We are given a concessions setting. We are asked to find a feasible solution that maximizes social welfare (the sum of the agents' utilities).*

The following shows that if the first problem is hard, the second problem is hard to approximate to any ratio. (We omit many proofs in this paper due to space constraint.)

[5]For these special cases, it may be conceptually desirable to make the domains of the variables x_i^j discrete, but we will refrain from doing so in this paper for the sake of consistency.

Proposition 1 *Suppose that FEASIBLE-CONCESSIONS is NP-hard even under some constraints on the instance (but no constraint that prohibits adding another agent that derives positive utility from any nontrivial setting of the variables of the other agents). Then it is NP-hard to approximate SW-MAXIMIZING-CONCESSIONS to any positive ratio, even under the same constraints.*

Hardness with positive and negative externalities

We first show that if we do not make the assumption of only negative externalities, then finding a feasible solution is NP-complete even when each agent controls only one variable. (In all the problems that we study, membership in NP is straightforward, so we just give the hardness proof.)

Theorem 1 *FEASIBLE-CONCESSIONS is NP-complete, even when all utility functions decompose (and all the components u_i^k are step functions), and each agent controls only one variable.*

Proof: We reduce an arbitrary SAT instance (given by variables V and clauses C) to the following FEASIBLE-CONCESSIONS instance. Let the set of agents be as follows. For each variable $v \in V$, let there be an agent a_v, controlling a single variable x_{a_v}. Also, for every clause $c \in C$, let there be an agent a_c, controlling a single variable x_{a_c}. Finally, let there be a single agent a_0 controlling x_{a_0}. Let all the utility functions decompose, as follows: For any $v \in V$, $u_{a_v}^{a_v}(x_{a_v}) = -\delta_{x_{a_v} \geq 1}$. For any $v \in V$, $u_{a_v}^{a_0}(x_{a_0}) = \delta_{x_{a_0} \geq 1}$. For any $c \in C$, $u_{a_c}^{a_c}(x_{a_c}) = (n(c) - 2|V|)\delta_{x_{a_c} \geq 1}$ where $n(c)$ is the number of variables that occur in c in negated form. For any $c \in C$, $u_{a_c}^{a_0}(x_{a_0}) = (2|V| - 1)\delta_{x_{a_0} \geq 1}$. For any $c \in C$ and $v \in V$ where $+v$ occurs in c, $u_{a_c}^{a_v}(x_{a_v}) = \delta_{x_{a_v} \geq 1}$. For any $c \in C$ and $v \in V$ where $-v$ occurs in c, $u_{a_c}^{a_v}(x_{a_v}) = -\delta_{x_{a_v} \geq 1}$. $u_{a_0}^{a_0}(x_{a_0}) = -|C|\delta_{x_{a_0} \geq 1}$. For any $c \in C$, $u_{a_0}^{a_c}(x_{a_c}) = \delta_{x_{a_c} \geq 1}$. All the other functions are 0 everywhere. We proceed to show that the instances are equivalent.

First suppose there exists a solution to the SAT instance. Then, let $x_{a_v} = 1$ if v is set to *true* in the solution, and $x_{a_v} = 0$ if v is set to *false* in the solution. Let $x_{a_c} = 1$ for all $c \in C$, and let $x_{a_0} = 1$. Then, the utility of every a_v is at least $-1 + 1 = 0$. Also, the utility of a_0 is $-|C| + |C| = 0$. And, the utility of every a_c is $n(c) - 2|V| + 2|V| - 1 + pt(c) - nt(c) = n(c) - 1 + pt(c) - nt(c)$, where $pt(c)$ is the number of variables that occur positively in c and are set to *true*, and $nt(c)$ is the number of variables that occur negatively in c and are set to *true*. Of course, $pt(c) \geq 0$ and $-nt(c) \geq -n(c)$; and if at least one of the variables that occur positively in c is set to *true*, or at least one of the variables that occur negatively in c is set to *false*, then $pt(c) - nt(c) \geq -n(c) + 1$, so that the utility of a_c is at least $n(c) - 1 - n(c) + 1 = 0$. But this is always the case, because the assignment satisfies the clause. So there exists a solution to the FEASIBLE-CONCESSIONS instance.

Now suppose there exists a solution to the FEASIBLE-CONCESSIONS instance. If it were the case that $x_{a_0} < 1$, then for all the a_v we would have $x_{a_v} < 1$ (or a_v would have a negative utility), and for all the a_c we would have $x_{a_c} < 1$ (because otherwise the highest utility possible for a_c is $n(c) - 2|V| < 0$, because all the x_{a_0} are below 1). So the solution would be trivial. It follows that $x_{a_0} \geq 1$. Thus, in order for a_0 to have nonnegative utility, it follows that for all $c \in C$, $x_{a_c} \geq 1$. Now, let v be set to *true* if $x_{a_v} = 1$, and to *false* if $x_{a_v} = 0$. So the utility of every a_c is $n(c) - 2|V| + 2|V| - 1 + pt(c) - nt(c) = n(c) - 1 + pt(c) - nt(c)$. In order for this to be nonnegative, we must have (for any c) that either $nt(c) < n(c)$ (at least one variable that occurs negatively in c is set to *false*) or $pt(c) > 0$ (at least one variable that occurs positively in c is set to *true*). So we have a satisfying assignment. ■

Hardness with only negative externalities

Next, we show that even if we do make the assumption of only negative externalities, then finding a feasible solution is still NP-complete, even when each agent controls at most two variables.

Theorem 2 *FEASIBLE-CONCESSIONS is NP-complete, even when there are only negative externalities, all utility functions decompose (and all the components are step functions), and each agent controls at most two variables.*

An algorithm for the case of only negative externalities and one variable per agent

We have shown that with both positive and negative externalities, finding a feasible solution is hard even when each agent controls only one variable; and with only negative externalities, finding a feasible solution is hard even when each agent controls at most two variables. In this section we show that these results are, in a sense, tight, by giving an algorithm for the case where there are only negative externalities and each agent controls only one variable. Under some minimal assumptions, this algorithm will return (or converge to) the maximal feasible solution, that is, the solution in which the variables are set to values that are as large as possible. Although the setting for this algorithm may appear very restricted, it still allows for the solution of interesting problems. For example, consider governments negotiating over by how much to reduce their countries' carbon dioxide emissions, for the purpose of reducing global warming.

We will not require the assumption of decomposing utility functions in this section (except where stated). The following claim shows the sense in which the maximal solution is well-defined in the setting under discussion (there cannot be multiple maximal solutions, and under a continuity assumption, a maximal solution exists).

Theorem 3 *In a concessions setting with only negative externalities and in which each agent controls only one variable, let $x_1, x_2, \ldots, x_n$ and $x'_1, x'_2, \ldots, x'_n$ be two feasible solutions. Then $\max\{x_1, x'_1\}, \max\{x_2, x'_2\}, \ldots, \max\{x_n, x'_n\}$ is also a feasible solution. Moreover, if all the utility functions are continuous, then, letting X_i be the set of values for x_i that occur in some feasible solution, $\sup(X_1), \sup(X_2), \ldots, \sup(X_n)$ is also a feasible solution.*

We are now ready to present the algorithm. First, we give an informal description. The algorithm proceeds in stages: in each stage, for each agent, it eliminates all the values for that agent's variable that would result in a negative utility for that agent regardless of how the other agents set their variables (given that they use values that have not yet been eliminated).

ALGORITHM 1
1. **for** $i := 1$ **to** n {
2. $X_i^0 := \mathbb{R}^{\geq 0}$ (alternatively, $X_i^0 := [0, M]$ where M is some upper bound) }
3. $t := 0$
4. **repeat until** $((\forall i)\ X_i^t = X_i^{t-1})$ {
5. $t := t + 1$
6. **for** $i := 1$ **to** n {
7. $X_i^t := \{x_i \in X_i^{t-1} : \exists x_1 \in X_1^{t-1}, x_2 \in X_2^{t-1}, \ldots, x_{i-1} \in X_{i-1}^{t-1}, x_{i+1} \in X_{i+1}^{t-1}, \ldots, x_n \in X_n^{t-1} : u_i(x_1, x_2, \ldots, x_i, \ldots, x_n) \geq 0\}$ } }

We note that the set updates in step 7 of the algorithm are simple to perform, because all the X_i^t always take the form $[0, r]$, $[0, r)$, or $\mathbb{R}^{\geq 0}$ (because we are in a concessions setting), and in step 7 it never hurts to choose values for $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ that are as large as possible (because we have only negative externalities). Roughly, the goal of the algorithm is for $\sup(X_1^t), \sup(X_2^t), \ldots, \sup(X_n^t)$ to converge to the maximal feasible solution (that is, the feasible solution such that all of the variables are set to values at least as large as in any other feasible solution). We now show that the algorithm is sound, in the sense that it does not eliminate values of the x_i that occur in feasible solutions.

Theorem 4 *Suppose we are running Algorithm 1 in a concessions setting with only negative externalities where each agent controls only one variable. If for some t, $r \notin X_i^t$, then there is no feasible solution with x_i set to r.*

However, the algorithm is not complete, in the sense that (for some "unnatural" functions) it does not eliminate all the values of the x_i that do not occur in feasible solutions.

Proposition 2 *Suppose we are running Algorithm 1 in a concessions setting with only negative externalities where each agent controls only one variable. For some (discontinuous) utility functions (even ones that decompose), the algorithm will terminate with nontrivial X_i^t even though the only feasible solution is the zero solution.*

However, if we make some reasonable assumptions on the utility functions (specifically, that they are either continuous or piecewise constant), then the algorithm is complete, in the sense that it will (eventually) remove any values of the x_i that are too large to occur in any feasible solution. Thus, the algorithm converges to the solution. We will present the case of continuous utility functions first.

Theorem 5 *Suppose we are running Algorithm 1 in a concessions setting with only negative externalities where each* *agent controls only one variable. Suppose that all the utility functions are continuous. Also, suppose that all the X_i^0 are initialized to $[0, M]$. Then, all the X_i^t are closed sets. Moreover, if the algorithm terminates after the tth iteration of the* **repeat** *loop, then $\sup(X_1^t), \sup(X_2^t), \ldots, \sup(X_n^t)$ is feasible, and it is the maximal solution. If the algorithm does not terminate, then $\lim_{t\to\infty} \sup(X_1^t), \lim_{t\to\infty} \sup(X_2^t), \ldots,$ $\lim_{t\to\infty} \sup(X_n^t)$ is feasible, and it is the maximal solution.*

We observe that piecewise constant functions are not continuous, and thus Theorem 5 does not apply to the case where the utility functions are piecewise constant. Nevertheless, the algorithm works on such utility functions, and we can even prove that the number of iterations is linear in the number of pieces. There is one caveat: the way we have defined piecewise constant functions (as linear combinations of step functions $\delta_{x\geq a}$), the maximal solution is not well defined (the set of feasible points is never closed on the right, i.e. it does not include its least upper bound). To remedy this, call a feasible solution *quasi-maximal* if there is no feasible solution that is larger (that is, all the x_i are set to values that are at least as large) and that gives some agent a different utility (so it is maximal for all intents and purposes).

Theorem 6 *Suppose we are running Algorithm 1 in a concessions setting with only negative externalities where each agent controls only one variable. If all the utility functions decompose and all the components u_i^k are piecewise constant with finitely many steps (the range of the u_i^k is finite), then the algorithm will terminate after at most T iterations of the* **repeat** *loop, where T is the total number of steps in all the self-components u_i^i (i.e. the sum of the sizes of the ranges of these functions). Moreover, if the algorithm terminates after the tth iteration of the* **repeat** *loop, then any solution $(x_1, x_2, \ldots, x_n)$ with for all i, $x_i \in \arg\max_{x_i \in X_i^t} \sum_{j\neq i} u_j^i(x_i)$, is feasible and quasi-maximal.*

Proof: If for some i and t, $X_i^t \neq X_i^{t-1}$, it must be the case that for some value r in the range of u_i^i, the preimage of this value is in $X_i^{t-1} - X_i^t$ (it has just been eliminated from consideration). Informally, one of the steps of the function u_i^i has been eliminated from consideration. Because this must occur for at least one agent in every iteration of the **repeat** loop before termination, it follows that there can be at most T iterations before termination. Now, if the algorithm terminates after the tth iteration of the **repeat** loop, and a solution $(x_1, x_2, \ldots, x_n)$ with for all i, $x_i \in \arg\max_{x_i \in X_i^t} \sum_{j\neq i} u_j^i(x_i)$ is chosen, it follows that each agent derives as much utility from the other agents' variables as is possible with the sets X_i^t (because of the assumption of only negative externalities, any setting of a variable that maximizes the total utility for the other agents also maximizes the utility for each individual other agent). We know that for each agent i, there is at least some setting of the other agents' variables within the X_j^t that will give agent i enough utility to compensate for the setting of its own variable (by the definition of X_i^t and using the fact that $X_j^t = X_j^{t-1}$, as the algorithm has terminated); and thus it follows that the utility maximizing setting is also enough to make i's utility nonnegative. So the solution is feasible. It is also quasi-maximal by Theorem 4. ■

Algorithm 1 can be extended to cases where some agents control multiple variables, by interpreting x_i in the algorithm as the *vector* of agent i's variables (and initializing the X_i^0 as cross products of sets). However, the next proposition shows how this extension of Algorithm 1 fails.

Proposition 3 *Suppose we are running the extension of Algorithm 1 just described in a concessions setting with only negative externalities. When some agents control more than one variable, the algorithm may terminate with nontrivial X_i^t even though the only feasible solution is the zero solution (all variables set to 0), even when all of the utility functions decompose and all of the components $u_i^{k,j}$ are step functions (or continuous functions).*

In the next section, we discuss *maximizing social welfare* under the conditions under which we showed Algorithm 1 to be successful in finding the maximal solution.

Maximizing social welfare remains hard

In a concessions setting with only negative externalities where each agent controls only one variable, the algorithm we provided in the previous section returns the *maximal* feasible solution, in a linear number of rounds for utility functions that decompose into piecewise constant functions. However, this may not be the most desirable solution. For instance, we may be interested in the feasible solution with the highest social welfare (that is, the highest sum of the agents' utilities). In this section we show that finding this solution remains hard, even in the setting in which Algorithm 1 finds the maximal solution fast.

Theorem 7 *The decision variant of SW-MAXIMIZING-CONCESSIONS (does there exist a feasible solution with social welfare $\geq K$?) is NP-complete, even when there are only negative externalities, all utility functions decompose (and all the components u_i^k are step functions), and each agent controls only one variable.*

Hardness with only two agents

So far, we have not assumed any bound on the number of agents. A natural question to ask is whether such a bound makes the problem easier to solve. In this section, we show that the problem of finding a feasible solution in a concessions setting with only negative externalities remains NP-complete even with only two agents (when there is no restriction on how many variables each agent controls).

Theorem 8 *FEASIBLE-CONCESSIONS is NP-complete, even when there are only two agents, there are only negative externalities, and all utility functions decompose (and all the components $u_i^{k,j}$ are step functions).*

A special case that can be solved to optimality using linear programming

Finally, in this section, we demonstrate a special case in which we can find the feasible outcome that maximizes social welfare (or any other linear objective) in polynomial time, using linear programming. (Linear programs can be solved in polynomial time (Khachiyan 1979).) The special case is the one in which all the utility functions decompose into piecewise linear, concave components. For this result we will need no additional assumptions (no bounds on the number of agents or variables per agent, *etc.*).

Theorem 9 *If all of the utility functions decompose, and all of the components $u_i^{k,j}$ are piecewise linear and concave, then SW-MAXIMIZING-CONCESSIONS can be solved in polynomial time using linear programming.*

Conclusions

In recent years, certain formalizations of combinatorial negotiation settings, most notably combinatorial auctions, have become one of the most-studied research topics in multiagent systems. A pervasive assumption has been that of *no externalities*: the agents deciding on a variable (such as whether a trade takes place between them) are the only ones affected by how this variable is set. This does not capture significant aspects of many important negotiation settings, leading to a loss in welfare. For instance, when an agent is deciding whether to build a *public good* such as a bridge, many other agents may be affected by this decision, as they could make use of the bridge. As another example, a company setting its pollution level may affect the health and safety of many. To date, there has been no widely studied formalization of combinatorial negotiation settings with externalities.[6] In this paper, we introduced such a formalization. The following table gives a summary of our results.

Restriction	Complexity
one variable per agent	NP-complete to find nontrivial feasible solution
negative externalities; two variables per agent	NP-complete to find nontrivial feasible solution
negative externalities; one variable per agent	Algorithm 1 finds maximal feasible solution (linear time for utilities that decompose into piecewise constant functions); NP-complete to find social-welfare maximizing solution
negative externalities; two agents	NP-complete to find nontrivial feasible solution
utilities decompose; components piecewise linear, concave	linear programming finds social-welfare maximizing solution

Complexity of finding solutions in concessions settings. All of the hardness results hold even if the utility functions decompose into step functions.

[6]Recent work on negotiation over donations (Conitzer & Sandholm 2004) may be seen as a restricted combinatorial negotiation setting in which externalities occur (whose restriction is different from restrictions studied in this paper).

The most important direction for future research is to study incentive compatibility aspects of our setting—that is, how to give agents incentives to report their preferences truthfully. If payments can be made, the agents can be made to report their true preferences using (for example) VCG payments (Vickrey 1961; Clarke 1971; Groves 1973), if we always choose the social welfare maximizing outcome. Without payments, though, there are results that prove that it is impossible to make the agents report their true preferences while always choosing a good outcome (Myerson & Satterthwaite 1983). Given the computational hardness results that we established in this paper, this may be an interesting setting for *algorithmic mechanism design* (the design of mechanisms that can be executed in polynomial time).

References

Archer, A.; Papadimitriou, C.; Tawar, K.; and Tardos, E. 2003. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *SODA*.

Bartal, Y.; Gonen, R.; and Nisan, N. 2003. Incentive compatible multi-unit combinatorial auctions. In *TARK*.

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

Conitzer, V., and Sandholm, T. 2004. Expressive negotiation over donations to charities. In *ACM-EC*, 51–60.

Feigenbaum, J.; Papadimitriou, C.; and Shenker, S. 2001. Sharing the cost of muliticast transmissions. *Journal of Computer and System Sciences* 63:21–41.

Fujishima, Y.; Leyton-Brown, K.; and Shoham, Y. 1999. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *IJCAI*, 548–553.

Groves, T. 1973. Incentives in teams. *Econometrica* 41:617–631.

Jehiel, P., and Moldovanu, B. 1996. How (not) to sell nuclear weapons. *American Economic Review* 86(4):814–829.

Khachiyan, L. 1979. A polynomial algorithm in linear programming. *Soviet Math. Doklady* 20:191–194.

Lehmann, D.; O'Callaghan, L. I.; and Shoham, Y. 2002. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM* 49(5):577–602.

Mas-Colell, A.; Whinston, M.; and Green, J. R. 1995. *Microeconomic Theory*. Oxford University Press.

Mu'alem, A., and Nisan, N. 2002. Truthful approximate mechanisms for restricted combinatorial auctions. In *AAAI*, 379–384.

Myerson, R., and Satterthwaite, M. 1983. Efficient mechanisms for bilateral trading. *Journal of Economic Theory* 28:265–281.

Nisan, N., and Ronen, A. 2000. Computationally feasible VCG mechanisms. In *ACM-EC*, 242–252.

Nisan, N., and Ronen, A. 2001. Algorithmic mechanism design. *Games and Economic Behavior* 35:166–196.

Parkes, D.; Kalagnanam, J.; and Eso, M. 2001. Achieving budget-balance with Vickrey-based payment schemes in exchanges. In *IJCAI*, 1161–1168.

Rothkopf, M.; Pekeč, A.; and Harstad, R. 1998. Computationally manageable combinatorial auctions. *Management Science* 44(8):1131–1147.

Sandholm, T.; Suri, S.; Gilpin, A.; and Levine, D. 2002. Winner determination in combinatorial auction generalizations. In *AAMAS*, 69–76.

Sandholm, T. 2002. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence* 135:1–54.

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