Computing Optimal Outcomes under an Expressive Representation of Settings with Externalities*

Vincent Conitzer
Department of Computer Science
Duke University
Durham, NC 27708, USA
conitzer@cs.duke.edu

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
sandholm@cs.cmu.edu

Abstract

When a decision must be made based on the preferences of multiple agents, and the space of possible outcomes is combinatorial in nature, it becomes necessary to think about how preferences should be represented, and how this affects the complexity of finding an optimal (or at least a good) outcome. We study settings with *externalities*, where each agent controls one or more variables, and how these variables are set affects not only the agent herself, but also potentially the other agents. For example, one agent may decide to reduce her pollution, which will come at a cost to herself, but will result in a benefit for all other agents.

We formalize how to represent such domains and show that in a number of key special cases, it is NP-complete to determine whether there exists a nontrivial feasible solution (and therefore the maximum social welfare is completely inapproximable). However, for one important special case, we give an algorithm that converges to the solution with the maximal concession by each agent (in a linear number of rounds for utility functions that additively decompose into piecewise constant functions). Maximizing social welfare, however, remains NP-hard even in this setting. We also demonstrate a special case that can be solved in polynomial time using linear programming.

1 Introduction

In many domains, a decision needs to be made based on the preferences of multiple agents. Often, the space of possible outcomes is combinatorial in nature, so that it becomes necessary to consider how preferences should be represented, as well as to design algorithms for finding an optimal (or at least a good) outcome. (For a recent overview of such work, see [6].)

Combinatorial auctions (for an overview, see [11]) are a common example. In such an auction, there are multiple items to be allocated among 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). Variants such as *combinatorial reverse auctions* (where the auctioneer seeks to *procure* a set of items) and *combinatorial exchanges* (where the agents trade items among themselves) have also received attention [29, 37, 28, 2, 35].

A pervasive assumption in this work (with a very recent exception [24]) has been that there are *no allocative externalities*: that is, no agent cares what happens to an item unless that agent herself receives the item. This is insufficient to model situations where a bidder who does not win an item still cares which other bidder wins it—for example, this may be the case if the item is a nuclear weapon [21]. Recently, some work in

*A short, early conference version appeared in the proceedings of the 2005 AAAI conference. This work has been supported by the National Science Foundation under grants IIS-0121678, IIS-0427858, IIS-0812113, IIS-0905390, and CAREER IIS-0953756; it has also been supported by two Sloan Fellowships.

sponsored-search auctions, where multiple advertisement slots on a page are for sale, has started to consider the role of externalities [30, 1, 14, 16, 23, 15, 17, 32, 10]. Here, the idea is that the attention that the user pays to one ad can depend on which other ads are shown simultaneously.

More generally, and more closely related to this paper, there are many important domains where actions taken by one agent affect many other agents. For example, if one agent takes on a task, such as building a bridge, many other agents may benefit from this (and the extent of their benefit in general depends on how the bridge is built, for example, on how heavy a load it can support). Similarly, if a company reduces its pollution, many individuals may benefit, even if they have nothing to do with the goods that the company produces. An action's effect on an otherwise uninvolved agent's utility is commonly known as an *externality* (for a discussion, see [25]). When making decisions based on the preferences of multiple agents, externalities must be taken into account, so that (potentially complex) arrangements can be made that are truly to every agent's benefit.

One domain in which externalities play a fundamental role, and that fits in the framework described in this paper, is the design of expressive mechanisms for donating to (say) charitable causes [8, 13]. The basic idea here is as follows. If one agent donates to a charity, then another agent who also cares about this charity benefits. For that reason, it may happen that, even if each individual agent does not care enough about the charity to give money to it by herself, it is nevertheless possible that all of the agents prefer a joint arrangement in which each agent gives a certain amount to the charity. This is because, thanks to the arrangement, each individual agent's donation is effectively multiplied by the number of agents. This opens up the possibility of mechanisms that take everyone's preferences over the charities as input, and then determine an arrangement for how much each agent should pay. Externalities play a fundamental role here: an agent giving to a charity imposes an externality on the other agents who care about this charity, and this is why the agents can benefit from a joint arrangement.

In this paper, we study whether optimal (or at least good) outcomes can be efficiently computed, under a quite general representation of settings with externalities. To our knowledge, this is the first such study of a general representation of settings with externalities. A common objective is to maximize *social welfare*, which is the sum of the agents' utilities. However, in most settings, there are constraints that must be satisfied. Typically, there are *voluntary participation* constraints, meaning that no agent is made worse off by participating in the mechanism. Additionally, if only the agents themselves know their preferences, and the agents are self-interested (the setting of *mechanism design*), then there may be *incentive compatibility* constraints, meaning that no agent should be able to make herself better off by misreporting her preferences.

After introducing our basic representation scheme for settings with externalities, we study the computational complexity of the following problem: given the agents' preferences, find a good (if possible, an optimal) outcome that satisfies the voluntary participation constraints. This problem is analogous to the *winner determination* problem in combinatorial auctions and exchanges, which consists of finding an optimal allocation of the items, given the bids. The winner determination problem in combinatorial auctions and exchanges has received a tremendous amount of previous attention (for example, [33, 12, 34, 37, 4, 7, 38, 9, 19, 18, 3]). In this paper, we will mostly focus on restricted settings that cannot model, *e.g.*, fully general combinatorial auctions and exchanges, so that we do not inherit all of the complexities from those settings (which would trivialize our results). Also, in this first research on the topic, we do not consider any incentive compatibility constraints—that is, we 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 report their preferences 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);[1] 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, we believe that incentive compatibility is an important topic for future

[1] 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.

research, and we will discuss it at the end of the paper in Section 9. As we noted, we do impose voluntary participation constraints.

The rest of this paper is organized as follows. In Section 2, we define our representation and the basic problems that we study under this representation. We show that the problem of finding a nontrivial feasible solution is hard in a number of special cases, including when each agent controls only one variable (Section 3); when there are only negative externalities and each agent controls at most two variables (Section 4); and when there are only negative externalities and there are only two agents, but there is no constraint on how many variables they control (Section 7). In Section 5, we give an algorithm for the case where there are only negative externalities and each agent controls only one variable. Under minimal assumptions, this algorithm finds or converges to the feasible outcome with the "maximal concessions" by each agent; moreover, given some additional assumptions (under which the hardness results proven in other sections still hold), the algorithm requires only a linear number of rounds. Nevertheless, in Section 6, we show that finding the social welfare maximizing outcome remains hard even in this setting. Finally, in Section 8, we show that the social welfare maximizing outcome can be found in polynomial time using linear programming if all the utility functions are piecewise linear and concave.

2 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 all 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 here. (We can effectively exclude certain values 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,[2] and we require without loss of generality that all agents' utilities are 0 at the default outcome. Thus, the voluntary participation constraint states that every agent's utility should be nonnegative.

Definition 2 *An outcome $(x_1^1, \ldots, x_1^{m_1}, \ldots, x_n^1, \ldots, x_n^{m_n})$ is* feasible *(aka. satisfies* voluntary participation*) if for every i, we have $u_i(x_1^1, \ldots, x_1^{m_1}, \ldots, x_n^1, \ldots, x_n^{m_n}) \geq 0$.*

Without any restrictions placed on it, this setup is very general. For instance, we can model a (multi-item, multi-unit) combinatorial exchange with it. We recall that 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

[2]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.

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.[3] 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. For 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 aggregating preferences 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 for combinatorial exchanges). However, for some of our positive results, we will not actually need this assumption—for example, for convergence results for an algorithm that we will present.

Definition 3 u_i additively decomposes (across variables) *if* $u_i(x_1^1, \dots, x_1^{m_1}, \dots, x_n^1, \dots, x_n^{m_n}) = \sum_{k=1}^{n} \sum_{j=1}^{m_k} u_i^{k,j}(x_k^j)$.

That is, the agent has a separate component utility function for each variable, and the agent's overall utility is the sum of these components. When utility functions additively 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 1 otherwise), or piecewise constant functions (linear combinations of step functions).[4]

In addition, we will focus strictly on settings where the higher an agent sets her own variables, the worse it is for herself. We will call such settings *concessions settings*. So, if the agents were to act independently, then each agent would selfishly set all her variables to 0 (the default outcome).

Definition 4 *A* concessions setting *is a setting with externalities where for any* $(x_1^1, \dots, x_1^{m_1}, \dots, x_n^1, \dots, x_n^{m_n}) \in \mathbb{R}^M$*, for any* $i, 1 \leq j \leq m_i$*, and for any* $\hat{x}_i^j > x_i^j$*, we have* $u_i(x_1^1, \dots, x_1^{m_1}, \dots, \hat{x}_i^j, \dots, x_n^1, \dots, x_n^{m_n}) \leq u_i(x_1^1, \dots, x_1^{m_1}, \dots, x_i^j, \dots, x_n^1, \dots, x_n^{m_n})$.

Thus, in a concessions setting, an agent's utility is monotonically weakly decreasing in that agent's own variables.

In parts of this paper, we will be interested in the following additional assumption, which states that the higher an agent sets her 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 5 *A concessions setting has* only negative externalities *if for any* $(x_1^1, \dots, x_1^{m_1}, \dots, x_n^1, \dots, x_n^{m_n}) \in \mathbb{R}^M$*, for any* $i, 1 \leq j \leq m_i$*, for any* $\hat{x}_i^j > x_i^j$*, and for any* $k \neq i$*,* $u_k(x_1^1, \dots, x_1^{m_1}, \dots, \hat{x}_i^j, \dots, x_n^1, \dots, x_n^{m_n}) \geq u_k(x_1^1, \dots, x_1^{m_1}, \dots, x_i^j, \dots, x_n^1, \dots, x_n^{m_n})$.

Thus, when there are only negative externalities, an agent's utility is monotonically weakly increasing in the other agents' variables.

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

[3] Thus, the fact that determining the existence of a nontrivial feasible solution for a combinatorial exchange is NP-complete [37] does not imply that determining the existence of a nontrivial feasible solution in our framework is NP-complete, because there is an exponential blowup in representation size.

[4] 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.

Definition 6 *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.

Definition 7 (FEASIBLE-CONCESSIONS) *We are given a concessions setting. We are asked whether there exists a nontrivial feasible solution.*

Definition 8 (SW-MAXIMIZING-CONCESSIONS) *We are given a concessions setting. We are asked to find a feasible solution that maximizes social welfare (among feasible solutions).*

The following simple proposition shows that if the first problem is hard, then the second problem is hard to approximate to any ratio.

Proposition 1 *Suppose that FEASIBLE-CONCESSIONS is NP-hard even under some constraints on the instance (but no constraint that prohibits adding another "dummy" 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.*

Proof: We reduce an arbitrary FEASIBLE-CONCESSIONS instance to a SW-MAXIMIZING-CONCESSIONS instance that is identical, except that a single additional agent has been added that derives positive utility from any nontrivial setting of the variable(s) of the other agents, and to whose variables all agents are completely indifferent (they cannot derive any utility from the new agent's variable(s)). If the original instance has no nontrivial feasible solution, then neither does the new instance, and the maximum social welfare that can be obtained is 0. On the other hand, if the original instance has a nontrivial feasible solution, then the new instance has a feasible solution with positive social welfare: the exact same solution is still feasible, and the new agent will get positive utility (and the others, nonnegative utility). It follows that any algorithm that approximates SW-MAXIMIZING-CONCESSIONS to some positive ratio will return a social welfare of 0 if there is no solution to the FEASIBLE-CONCESSIONS problem instance, and positive social welfare if there is a solution—and thus the algorithm could be used to solve an NP-hard problem. ■

It would appear that these problems (or, in the case of SW-MAXIMIZING-CONCESSIONS, its decision variant) should naturally at least fall in the class NP, because an outcome should be able to serve as a certificate. Nevertheless, we cannot say this without making some assumption about how the utility functions are represented. We now give a weak sufficient condition.

Definition 9 *A family of concessions instances has the* outcomes-are-certificates (OAC) *property if*

- *we can without loss of generality restrict our attention to a set of outcomes such that these outcomes can be represented in polynomial space,*
- *given an outcome, we can compute each agent's utility in polynomial time, and*
- *given an outcome, we can compute in polynomial time whether this outcome corresponds to a trivial setting of the variables.*

For example, when the agents' utility functions additively decompose, and their utility functions for individual variables are piecewise constant functions (where the points of discontinuity and the corresponding values are explicitly given, say, as rational numbers), the OAC property holds: we can without loss of generality restrict our attention to outcomes where each variable is set to a value at which a discontinuity occurs for some agent (or to 0)—because any other value will be equivalent to some such value—and because these values are given explicitly in the input, these outcomes can be represented in polynomial space. Moreover, given an outcome, a simple lookup suffices to compute each agent's utility for each variable. Finally, to determine whether an outcome is trivial, it suffices to check whether there exists an agent that receives nonzero utility from at least one of the variables' values. We note that we can approximate any continuous function with a piecewise constant function, with the caveat that we may need an infinite number of pieces to approximate the tail end of the function (in practice, we can simply ignore values of the variables that are too large to occur in practical solutions).

Proposition 2 *For any family of concessions instances that satisfies the OAC property, FEASIBLE-CONCESSIONS and the decision variant of SW-MAXIMIZING-CONCESSIONS (does there exist a feasible solution with social welfare $\geq K$?) are in NP.*

Proof: In each case, the outcomes to which we can restrict our attention will serve as the certificates; by assumption, these certificates have polynomial length. Also, by assumption, we can determine in polynomial time whether it is a trivial outcome. To determine whether an outcome is feasible, we compute each agent's utility (which, by assumption, we can do in polynomial time), and check whether it is at least 0. This shows that FEASIBLE-CONCESSIONS is in NP. Moreover, because we can compute the agents' utilities efficiently, we can also compute the social welfare efficiently. This shows that the decision variant of SW-MAXIMIZING-CONCESSIONS is in NP. ■

3 Hardness with positive and negative externalities

We first show that if we do not make the assumption that there are only negative externalities, then determining whether a nontrivial feasible solution exists is NP-complete even when each agent controls only one variable. In this paper, when membership in NP is straightforward, we just give the hardness proof. Also, when each agent controls only one variable, no superscript j on the variables or the component utility functions is necessary.

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

Proof: We reduce an arbitrary satisfiability 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 additional agent a_0 controlling x_{a_0}. Let all the utility functions decompose additively, as follows. (We recall that the notation $\delta_{x \geq a}$ evaluates to 0 if $x < a$, and to 1 otherwise.) 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 satisfiability 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 $n(c)-2|V|<0$). 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 $x_{a_v}\geq 1$, and to *false* if $x_{a_v}<1$. 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*). Therefore, we have a satisfying assignment. ■

4 Hardness with only negative externalities

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

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

Proof: We reduce an arbitrary satisfiability instance 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 variables $x^+_{a_v}$ and $x^-_{a_v}$. Also, for every clause $c\in C$, let there be an agent a_c, controlling a single variable x_{a_c}. Let all the utility functions decompose additively, as follows: For any $v\in V$, $u^{a_v,+}_{a_v}(x^+_{a_v})=-|C|\delta_{x^+_{a_v}\geq 1}$, and $u^{a_v,-}_{a_v}(x^-_{a_v})=-|C|\delta_{x^-_{a_v}\geq 1}$. For any $v\in V$ and $c\in C$, $u^{a_c}_{a_v}(x_{a_c})=\delta_{x_{a_c}\geq 1}$. For any $c\in C$, $u^{a_c}_{a_c}(x_{a_c})=-\delta_{x_{a_c}\geq 1}$. For any $c\in C$ and $v\in V$ where $+v$ occurs in c, $u^{a_v,+}_{a_c}(x^+_{a_v})=\delta_{x^+_{a_v}\geq 1}$; and for any $c\in C$ and $v\in V$ where $-v$ occurs in c, $u^{a_v,-}_{a_c}(x^-_{a_v})=\delta_{x^-_{a_v}\geq 1}$. All the other functions are 0 everywhere. We proceed to show that the instances are equivalent.

First suppose there exists a solution to the satisfiability instance. Then, let $x^+_{a_v}=1$ if v is set to *true* in the solution, and $x^+_{a_v}=0$ otherwise; and, let $x^-_{a_v}=1$ if v is set to *false* in the solution, and $x^-_{a_v}=0$ otherwise. Let $x_{a_c}=1$ for all $c\in C$. Then, the utility of every a_v is $-|C|+|C|=0$. Also, the utility of every a_c is at least $-1+1$ (because all clauses are satisfied in the solution, there is at least one $+v\in c$ with $x^+_{a_v}=1$, or at least one $-v\in c$ with $x^-_{a_v}=1$). So there exists a solution to the FEASIBLE-CONCESSIONS instance.

Now suppose there exists a solution to the FEASIBLE-CONCESSIONS instance. At least one of the $x^+_{a_v}$ or at least one of the $x^-_{a_v}$ must be set nontrivially (≥ 1), because otherwise no x_{a_c} can be set nontrivially. But this implies that for any clause $c\in C$, $x_{a_c}\geq 1$ (for otherwise the a_v with a nontrivial setting of her variables would have negative utility). So that none of the a_c have nonnegative utility, it must be the case that for any $c\in C$, either there is at least one $+v\in c$ with $x^+_{a_v}\geq 1$, or at least one $-v\in c$ with $x^-_{a_v}\geq 1$. Also, for no variable $v \in V$ can it be the case that both $x^+_{a_v} \geq 1$ and $x^-_{a_v} \geq 1$, as this would leave a_v with negative utility. But then, letting v be set to *true* if $x^+_{a_v} \geq 1$, and to *false* otherwise, must satisfy every clause. So there exists a solution to the satisfiability instance. ∎

5 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 nontrivial feasible solution is hard even when each agent controls only one variable; and with only negative externalities, finding a nontrivial 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.[5] 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. Moreover, in the case of piecewise constant functions, it will return this solution in a linear number of iterations and hence in polynomial time. (Since our hardness results so far were for piecewise constant functions, this implies that those results are tight.) Although the setting for this algorithm may appear very restricted, it still allows for the solution of many interesting problems. For example, consider governments negotiating over how much to reduce their countries' carbon dioxide emissions, for the purpose of reducing global warming. As another example, consider agents negotiating over how much to reduce their use of a common resource such as a communication network (where heavy use slows down the network).

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.

Lemma 1 *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.*

Proof: For the first claim, we need to show that every agent i receives nonnegative utility in the proposed solution. Suppose without loss of generality that $x_i \geq x'_i$. Then, we have $u_i(\max\{x_1, x'_1\}, \max\{x_2, x'_2\}, \ldots, \max\{x_i, x'_i\}, \ldots, \max\{x_n, x'_n\}) = u_i(\max\{x_1, x'_1\}, \max\{x_2, x'_2\}, \ldots, x_i, \ldots, \max\{x_n, x'_n\}) \geq u_i(x_1, x_2, \ldots, x_i, \ldots, x_n)$, where the inequality stems from the fact that there are only negative externalities. But the last expression is nonnegative because the first solution is feasible.

For the second claim, we will find a sequence of feasible solutions that converges to the proposed solution. By continuity, any agent's utility at the limit point must be the limit of that agent's utility in the sequence of feasible solutions; and because these solutions are all feasible, this limit must be nonnegative. For each agent i, let $\{(x_1^{i,j}, x_2^{i,j}, \ldots, x_n^{i,j})\}_{j \in \mathbb{N}}$ be a sequence of feasible solutions with $\lim_{j \to \infty} x_i^{i,j} = \sup(X_i)$. By repeated application of the first claim, we have that (for any j) $\max_i\{x_1^{i,j}\}, \max_i\{x_2^{i,j}\}, \ldots, \max_i\{x_n^{i,j}\}$ is a feasible solution, giving us a new sequence of feasible solutions. Moreover, because this new sequence dominates every one of the original sequences, and for each agent i there is at least one original sequence where the

[5] After the conference version of this paper, Ghosh and Mahdian, who were at that point not aware of this work, independently discovered effectively the same algorithm in their more specific framework for mechanisms for donations to charities [13].

ith element converges to $\sup(X_i)$, the sequence converges to the solution $\sup(X_1), \sup(X_2), \ldots, \sup(X_n)$. ■

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. $X_i^t := \{x_i \in X_i^{t-1} : \exists x_1 \in X_1^{t-1}, x_2 \in X_2^{t-1}, \ldots, x_{i-1} \in X_{i-1}^{t-1}, x_{i+1} \in X_{i+1}^{t-1}, \ldots, x_n \in X_n^{t-1} : u_i(x_1, x_2, \ldots, x_i, \ldots, x_n) \geq 0\}$ } }

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.

Lemma 2 *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.*

Proof: We will prove this by induction on t. For $t = 0$ this is vacuously true. Now suppose we have proved it true for $t = k$; we will prove it true for $t = k + 1$. By the induction assumption, all feasible solutions lie within $X_1^k \times \ldots \times X_n^k$. But if $r \neq X_i^{k+1}$, this means exactly that there is no feasible solution in $X_1^k \times \ldots \times X_n^k$ with $x_i = r$. It follows there is no feasible solution with $x_i = r$ at all. ■

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 3 *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 additively), the algorithm will terminate with nontrivial X_i^t even though the only feasible solution is the zero solution.*

Proof: Consider the following symmetric example:

- $u_1^1(x_1) = -x_1$ for $x_1 < 1$, $u_1^1(x_1) = -2$ otherwise;
- $u_1^2(x_2) = (x_2)^2$ for $x_2 < 1$, $u_1^2(x_2) = 1$ otherwise;
- $u_2^1(x_1) = (x_1)^2$ for $x_1 < 1$, $u_2^1(x_1) = 1$ otherwise;

- $u_2^2(x_2) = -x_2$ for $x_2 < 1$, $u_2^2(x_2) = -2$ otherwise.

There is no feasible solution with $x_1 \geq 1$ or $x_2 \geq 1$, because the corresponding agent's utility would definitely be negative. In order for agent 1 to have nonnegative utility we must have $(x_2)^2 \geq x_1$. Unless they are both zero, this implies $x_2 > x_1$. Similarly, in order for agent 2 to have nonnegative utility we must have $(x_1)^2 \geq x_2$, and unless they are both zero, this implies $x_1 > x_2$. It follows that the only feasible solution is the zero solution. Unfortunately, in the algorithm, we first get $X_1^1 = X_2^1 = [0, 1)$; then also, we get $X_1^2 = X_2^2 = [0, 1)$ (for any $x_1 < 1$, we can set $x_2 = \sqrt{x_1} < 1$ and agent 1 will get utility 0, and similarly for agent 2). So the algorithm terminates with nontrivial X_i^t. ■

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 maximal feasible solution. (This does not mean that it necessarily *terminates*, and as a result we cannot give a runtime for the algorithm in the continuous case.) We will present the case of continuous utility functions first.

Theorem 3 *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 the maximal feasible 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 the maximal feasible solution.*

Proof: First we show that all the X_i^t are closed sets, by induction on t. For $t = 0$, the claim is true, because $[0, M]$ is a closed set. Now suppose they are all closed for $t = k$; we will show them to be closed for $t = k + 1$. In the step in the algorithm in which we set X_i^{k+1}, in the choice of $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, we may as well always set each of these x_j to $\sup(X_j^k)$ (which is inside X_j^k because X_j^k is closed by the induction assumption), because this will maximize agent i's utility. It follows that $X_i^{k+1} = \{x_i : u_i(\sup(X_1^k), \ldots, \sup(X_{i-1}^k), x_i, \sup(X_{i+1}^k), \ldots, \sup(X_n^k)) \geq 0\}$. But because u_i is continuous, this set must be closed by elementary results from analysis.

Now we proceed to show the second claim. Because each X_i^t is closed, it follows that $\sup(X_i^t) \in X_i^t$. This implies that, for every agent i, there exist $x_1 \in X_1^{t-1}, x_2 \in X_2^{t-1}, \ldots, x_{i-1} \in X_{i-1}^{t-1}, x_{i+1} \in X_{i+1}^{t-1}, \ldots, x_n \in X_n^{t-1}$ such that $u_i(x_1, x_2, \ldots, \sup(X_i^t), \ldots, x_n) \geq 0$. Because for every agent i', $X_{i'}^t = X_{i'}^{t-1}$ (the algorithm terminated), this is equivalent to saying that there exist $x_1 \in X_1^t, x_2 \in X_2^t, \ldots, x_{i-1} \in X_{i-1}^t, x_{i+1} \in X_{i+1}^t, \ldots, x_n \in X_n^t$ such that $u_i(x_1, x_2, \ldots, \sup(X_i^t), \ldots, x_n) \geq 0$. Of course, for each of these $x_{i'}$, we have $x_{i'} \leq \sup(X_{i'}^t)$. Because there are only negative externalities, it follows that $u_i(\sup(X_1^t), \sup(X_2^t), \ldots, \sup(X_i^t), \ldots, \sup(X_n^t)) \geq u_i(x_1, x_2, \ldots, \sup(X_i^t), \ldots, x_n) \geq 0$. Thus, $\sup(X_1^t), \sup(X_2^t), \ldots, \sup(X_n^t)$ is feasible. It is also maximal by Lemma 2.

Finally, we prove the third claim. For any agent i, for any t, we have $u_i(\sup(X_1^{t-1}), \sup(X_2^{t-1}), \ldots, \lim_{t'\to\infty} \sup(X_i^{t'}), \ldots, \sup(X_n^{t-1})) \geq u_i(\sup(X_1^{t-1}), \sup(X_2^{t-1}), \ldots, \sup(X_i^t), \ldots, \sup(X_n^{t-1}))$ (because the X_i^t are decreasing in t, and we are in a concessions setting). The last expression evaluates to a nonnegative quantity, using the same reasoning as in the proof of the second claim with the fact that $\sup(X_i^t) \in X_i^t$. But then, by continuity, $0 \leq \lim_{t\to\infty}(u_i(\sup(X_1^{t-1}), \sup(X_2^{t-1}), \ldots, \lim_{t'\to\infty} \sup(X_i^{t'}), \ldots, \sup(X_n^{t-1}))) = u_i(\lim_{t\to\infty} \sup(X_1^{t-1}), \lim_{t\to\infty} \sup(X_2^{t-1}), \ldots, \lim_{t'\to\infty} \sup(X_i^{t'}), \ldots, \lim_{t\to\infty} \sup(X_n^{t-1})) = u_i(\lim_{t\to\infty} \sup(X_1^t), \lim_{t\to\infty} \sup(X_2^t), \ldots, \lim_{t\to\infty} \sup(X_i^t), \ldots, \lim_{t\to\infty} \sup(X_n^t))$. It follows that $\lim_{t\to\infty} \sup(X_1^t), \lim_{t\to\infty} \sup(X_2^t), \ldots, \lim_{t\to\infty} \sup(X_n^t)$ is feasible. It is also maximal by Lemma 2. ■

We observe that piecewise constant functions are not continuous, and thus Theorem 3 does not apply to the case where the utility functions decompose additively and the component 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 feasible solution is not well defined (the set of feasible points is never closed on the right, that is, 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 larger utility. Hence, a quasi-maximal feasible solution is maximal for all intents and purposes.

Theorem 4 *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 additively 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 (that is, 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 $(x_1, x_2, \dots, x_n)$ 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 a quasi-maximal feasible solution.*

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 $(x_1, x_2, \dots, x_n)$ 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 Lemma 2. ■

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 Cartesian products of sets). However, the next proposition shows that this extension of Algorithm 1 fails. (This is perhaps not surprising in light of our earlier hardness result, Theorem 2, but it is still instructive to see exactly *how* it fails.)

Proposition 4 *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 solutions are trivial solutions, even when all of the utility functions decompose additively and all of the components $u_i^{k,j}$ are step functions (or continuous functions).*

Proof: Let each of three agents control two variables, with utility functions as follows. (We recall once again that the notation $\delta_{x \geq a}$ evaluates to 0 if $x < a$, and to 1 otherwise.)

- $u_1^{1,1}(x_1^1) = -3\delta_{x_1^1 \geq 1}$

- $u_1^{1,2}(x_1^2) = -3\delta_{x_1^2 \geq 1}$
- $u_2^{2,1}(x_2^1) = -3\delta_{x_2^1 \geq 1}$
- $u_2^{2,2}(x_2^2) = -3\delta_{x_2^2 \geq 1}$
- $u_3^{3,1}(x_3^1) = -3\delta_{x_3^1 \geq 1}$
- $u_3^{3,2}(x_3^2) = -3\delta_{x_3^2 \geq 1}$
- $u_1^{2,1}(x_2^1) = 2\delta_{x_2^1 \geq 1}$
- $u_1^{3,1}(x_3^1) = 2\delta_{x_3^1 \geq 1}$
- $u_2^{1,1}(x_1^1) = 2\delta_{x_1^1 \geq 1}$
- $u_2^{3,2}(x_3^2) = 2\delta_{x_3^2 \geq 1}$
- $u_3^{1,2}(x_1^2) = 2\delta_{x_1^2 \geq 1}$
- $u_3^{2,2}(x_2^2) = 2\delta_{x_2^2 \geq 1}$

Increasing any one of the variables to a value of at least 1 will decrease the corresponding agent's utility by 3, and will raise only one other agent's utility, by 2. It follows that there is no nontrivial feasible solution, because any nontrivial solution will have negative social welfare (total utility), and hence at least one agent must have negative utility.

In the algorithm, after the first iteration, it becomes clear that no agent can set both her variables to values of at least 1 (because each agent can derive at most $4 < 6$ utility from the other agents' variables). Nevertheless, for any agent, it still appears possible at this stage to set either (but not both) of her variables to a value of at least 1. Unfortunately, in the next iteration, this still appears possible (because each of the other agents could set the variable that is beneficial to this agent to a value of at least 1, leading to a utility of $4 > 3$ for this agent). It follows that the algorithm gets stuck with nontrivial X_i^t.

These component utility functions are easily made continuous, while changing neither the algorithm's behavior on them nor the set of feasible solutions—for instance, by making each function linear on the interval $[0, 1]$. ■

In the next section, we discuss *maximizing social welfare*, under the same conditions under which we showed Algorithm 1 to be successful in finding the maximal feasible solution (which does not necessarily maximize welfare).

6 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 additively 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 5 *The decision variant of SW-MAXIMIZING-CONCESSIONS (does there exist a feasible solution with social welfare $\geq K$?) is NP-complete (assuming OAC for NP membership), even when there are only negative externalities, all utility functions decompose additively (and all the components u_i^k are step functions), and each agent controls only one variable.*

Proof: We reduce an arbitrary EXACT-COVER-BY-3-SETS instance (given by a set S and subsets $S_1, S_2, \ldots, S_q$ ($|S_i| = 3$) with which to cover S, without any overlap) to the following SW-MAXIMIZING-CONCESSIONS instance. Let the set of agents be as follows. For every S_i there is an agent a_{S_i}. Also, for every element $s \in S$ there is an agent a_s. Every agent a controls a single variable x_a. Let all the utility functions decompose additively, as follows: For any S_i, $u_{a_{S_i}}^{a_{S_i}}(x_{a_{S_i}}) = -7\delta_{x_{a_{S_i}} \geq 1}$. For any S_i and for any s, $u_{a_{S_i}}^{a_s}(x_{a_s}) = 7\delta_{x_{a_s} \geq 1}$. For any s, $u_{a_s}^{a_s}(x_{a_s}) = -\delta_{x_{a_s} \geq 1}$. For any s and for any S_i with $s \in S_i$, $u_{a_s}^{a_{S_i}}(x_{a_{S_i}}) = \frac{1}{c(s)-1}\delta_{x_{a_{S_i}} \geq 1}$, where $c(s)$ is the number of sets S_i with $s \in S_i$. All the other functions are 0 everywhere. (We may assume without loss of generality that $c(s) \geq 2$: if $c(s) = 0$ then the original problem instance is clearly infeasible, and if $c(s) = 1$ then it is clear that the one set S_i that contains s must be used in the cover, and so we can reduce it to a simpler problem instance.) Let the target social welfare be $K = 7q(|S| - 1) + 7\frac{|S|}{3}$. We proceed to show that the two instances are equivalent.

First, suppose there exists a solution to the EXACT-COVER-BY-3-SETS instance. Then, let $x_{a_{S_i}} = 0$ if S_i is in the cover, and $x_{a_{S_i}} = 1$ otherwise. For all s, let $x_{a_s} = 1$. Then a_{S_i} receives a utility of $7|S|$ if S_i is in the cover, and $7(|S|-1)$ otherwise. Furthermore, for all $s \in S$, a_s receives a utility of $(c(s)-1)\frac{1}{c(s)-1} - 1 = 0$ (because for exactly $c(s) - 1$ of the $c(s)$ subsets S_i with s in it, the corresponding agent has her variable set to 1: the only exception is the subset S_i that contains s and is in the cover). It follows that all the agents receive nonnegative utility, and the total utility (social welfare) is $7q(|S| - 1) + 7\frac{|S|}{3}$. So there exists a solution to the SW-MAXIMIZING-CONCESSIONS instance.

Now, suppose that there exists a solution to the SW-MAXIMIZING-CONCESSIONS instance. We first observe that if for some $s \in S$, $x_{a_s} < 1$, the total utility (social welfare) can be at most $7q(|S| - 1) + 2|S| < 7q(|S|-1)+7\frac{|S|}{3}$ (because each a_{S_i} can receive at most $7(|S|-1)$, and each a_s can receive at most $c(s)\frac{1}{c(s)-1}$, and because $c(s) \geq 2$ this can be at most 2). So it must be the case that $x_{a_s} \geq 1$ for all $s \in S$. It follows that, in order for none of these a_s to have nonnegative utility, for every $s \in S$, there are at least $c(s) - 1$ subsets S_i with $x_{a_{S_i}} \geq 1$ and $s \in S_i$. In other words, for every $s \in S$, there is at most one subset S_i with $s \in S_i$ with $x_{a_{S_i}} < 1$. In other words again, the subsets S_i with $x_{a_{S_i}} < 1$ are disjoint (and so there are at most $\frac{|S|}{3}$ of them). However, if there were only $k \leq \frac{|S|}{3} - 1$ subsets S_i with $x_{a_{S_i}} < 1$, then the total utility (social welfare) can be at most $7q(|S|-1)+7k+|S|-3k$ (each a_{S_i} receives at least $7(|S|-1)$, and they receive no more unless they are among the k, in which case they receive an additional 7; and every a_s receives 0 unless s is in none of the k disjoint subsets S_i, in which case the agent will receive at most 1; but of course there can be at most $|S|-3k$ such agents). But $7q(|S|-1)+7k+|S|-3k \leq 7q(|S|-1)+|S|+4(\frac{|S|}{3}-1) = 7q(|S|-1)+7\frac{|S|}{3}-4$, which is less than the target. It follows there are exactly $\frac{|S|}{3}$ disjoint subsets S_i with $x_{a_{S_i}} < 1$—an exact cover. So there exists a solution to the EXACT-COVER-BY-3-SETS instance. ■

7 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 determining the existence of a nontrivial 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 6 *FEASIBLE-CONCESSIONS is NP-complete (assuming OAC for NP membership), even when there are only two agents, there are only negative externalities, and all utility functions decompose additively (and all the components $u_i^{k,j}$ are step functions).*

Proof: We reduce an arbitrary KNAPSACK instance (given by r pairs (c_i, v_i), where all the c_i and v_i are positive; a cost constraint C; and a value objective V) to the following FEASIBLE-CONCESSIONS instance with two agents. Agent 1 controls only one variable, x_1^1. Agent 2 controls r variables, $x_2^1, x_2^2, \ldots, x_2^r$. Agent 1's utility function is $u_1(x_1^1, x_2^1, x_2^2, \ldots, x_2^r) = -V\delta_{x_1^1 \geq 1} + \sum_{j=1}^{r} v_j \delta_{x_2^j \geq 1}$. Agent 2's utility function is $u_2(x_1^1, x_2^1, x_2^2, \ldots, x_2^r) = C\delta_{x_1^1 \geq 1} - \sum_{j=1}^{r} c_j \delta_{x_2^j \geq 1}$. We proceed to show that the instances are equivalent.

Suppose there is a solution to the KNAPSACK instance, that is, a subset S such that $\sum_{j \in S} c_i \leq C$ and $\sum_{j \in S} v_i \geq V$. Then, let $x_1^1 = 1$, and for any $1 \leq j \leq r$, let $x_2^j = \delta_{j \in S}$ (that is, $x_2^j = 1$ if $j \in S$, and $x_2^j = 0$ otherwise). Then $u_1(x_1^1, x_2^1, x_2^2, \ldots, x_2^r) = -V + \sum_{j \in S} v_j \geq 0$. Also, $u_2(x_1^1, x_2^1, x_2^2, \ldots, x_2^r) = C - \sum_{j \in S} c_j \geq 0$. So there is a solution to the FEASIBLE-CONCESSIONS instance.

Now suppose there is a solution to the FEASIBLE-CONCESSIONS instance, that is, a nontrivial setting of the variables $(x_1^1, x_2^1, x_2^2, \ldots, x_2^r)$ such that $u_1(x_1^1, x_2^1, x_2^2, \ldots, x_2^r) \geq 0$ and $u_2(x_1^1, x_2^1, x_2^2, \ldots, x_2^r) \geq 0$. If it were the case that $x_1^1 < 1$, then either all of agent 2's variables are set to values smaller than 1 (in which case we have a trivial solution), or at least one of agent 2's variables is set to a nontrivial value (in which case agent 2 gets negative utility because the setting of x_1^1 is worthless to him). It follows that $x_1^1 \geq 1$. Thus, in order for agent 1 to get nonnegative utility, we must have $\sum_{j=1}^{r} v_j \delta_{x_2^j \geq 1} \geq V$. Let $S = \{j : x_2^j \geq 1\}$. Then it follows that $\sum_{j \in S} v_j \geq V$. Also, in order for agent 2 to get nonnegative utility, we must have $\sum_{j \in S} c_j = \sum_{j=1}^{r} c_j \delta_{x_2^j \geq 1} \leq C$. So there is a solution to the KNAPSACK instance. ■

8 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 other linear objectives) in polynomial time, using linear programming. For the sake of making things definite and simple, we assume that all the components of the utility functions are represented as piecewise linear functions (consisting of a finite number of segments). Crucially, we assume that these functions are concave. For this result we will need no additional assumptions (no bounds on the number of agents or variables per agent, *etc.*).

Theorem 7 *If all of the utility functions decompose additively, and all of the components $u_i^{k,j}$ are piecewise linear (consisting of a finite number of segments) and concave, then SW-MAXIMIZING-CONCESSIONS can be solved in polynomial time using linear programming.*

Proof: Let the variables of the linear program be the $\mathbf{x}_k^j$ and the $\mathbf{u}_i^{k,j}$, each of which is a single-dimensional real-valued variable. (We write them in bold font to indicate that they are variables in the linear program; in particular, it is important to distinguish the variable $\mathbf{u}_i^{k,j}$ from the function $u_i^{k,j}(\cdot)$: the latter is part of the input.) Of course, to obtain a sensible solution, the values of the $\mathbf{u}_i^{k,j}$ should be determined by the values of the $\mathbf{x}_k^j$, namely, $\mathbf{u}_i^{k,j} = u_i^{k,j}(\mathbf{x}_k^j)$ should hold. This is not a linear constraint, but we can capture it with a collection of linear constraints, using a standard linear programming trick. Before we do so, we note that the objective (social welfare) can be written as $\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{m_k}\mathbf{u}_i^{k,j}$, which is linear in the variables. Also, the feasibility (voluntary participation) constraints can be written as: for any i, we require $\sum_{k=1}^{n}\sum_{j=1}^{m_k}\mathbf{u}_i^{k,j} \geq 0$. Again, these constraints are linear in the variables.

All that remains to do is to make sure that the $\mathbf{u}_i^{k,j}$ variables take the correct values with respect to the values of the $\mathbf{x}_k^j$ variables. Due to the objective of maximizing social welfare, a solver will always set the $\mathbf{u}_i^{k,j}$ variables to values that are as high as possible, so we only need to make sure that they are not set too high. That is, we only need to add constraints to ensure that $\mathbf{u}_i^{k,j} \leq u_i^{k,j}(\mathbf{x}_k^j)$. Here, we will use the fact that the functions $u_i^{k,j}(\cdot)$ are piecewise linear and concave. Consider a linear segment of the function $u_i^{k,j}(\cdot)$; let $l(\cdot)$ be the linear function that coincides with this segment. Because $u_i^{k,j}(\cdot)$ is concave, the nonlinear constraint $\mathbf{u}_i^{k,j} \leq u_i^{k,j}(\mathbf{x}_k^j)$ implies the linear constraint $\mathbf{u}_i^{k,j} \leq l(\mathbf{x}_k^j)$. Conversely, if we add all such linear constraints (one for each of the finitely many segments), collectively they will imply the nonlinear constraint $\mathbf{u}_i^{k,j} \leq u_i^{k,j}(\mathbf{x}_k^j)$. Hence, we can replace each of the nonlinear constraints with an equivalent collection of linear constraints. These constraints complete the linear program. ■

9 Conclusions and future research

In combinatorial auctions and similar settings, a no-externalities assumption is commonly made. However, more recently, externalities have been receiving more attention. For example, in sponsored-search auctions, the attention that one winning advertiser gets from the user in general depends on which other advertisers have won a slot. In other settings, externalities play an even greater role. Novel mechanisms for determining how much each agent should give to certain charitable causes [8, 13] rely on the fact that an agent may be willing to give more if this induces others to give more as well. Thus, these mechanisms fundamentally rely on an externality—namely, that one agent derives utility from seeing another agent give money to a charity. More generally, it is clear that failing to model externalities comes at a significant cost in welfare in many settings. 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. This paper, to our knowledge, is the first that considers a general representation of settings with externalities and studies the problem of computing good or even optimal outcomes within this framework.

We showed that when both positive and negative externalities occur, determining whether a nontrivial feasible solution exists is NP-complete even when each agent controls only one variable and all the utility functions decompose additively into step functions. We then showed that with only negative externalities, determining whether a nontrivial feasible solution exists is NP-complete even when each agent controls at most two variables and all the utility functions decompose additively into step functions. We then gave an algorithm for the case where there are only negative externalities and each agent controls only one variable, intended to find the feasible solution with the variables set to values that are as large as possible. We showed that, although the algorithm may fail with certain discontinuous utility functions, it either terminates at or converges to the maximal solution with continuous utility functions; and for the case where the utility functions decompose additively into piecewise constant functions, it always terminates correctly, in a linear number of rounds. We also showed why the natural generalization of the algorithm to cases where some agents control more than one variable may fail even when all the utility functions decompose additively into step functions or continuous functions. We then showed that the decision variant of the problem of maximizing social welfare remains NP-complete even when there are only negative externalities, each agent controls only one variable, and all the utility functions decompose additively into step functions. We also showed that even when there are only two agents, only negative externalities, and all the utility functions decompose additively into step functions, the problem of determining whether a nontrivial feasible solution exists remains NP-complete. Finally, we also demonstrated that if the utility functions decompose additively into functions that are piecewise linear and concave, then the optimization problem can be solved to optimality (in polynomial time) using linear programming.

The following table gives a summary of our results.

Restriction	**Complexity**
one variable per agent	NP-complete to determine existence of nontrivial feasible solution
negative externalities; two variables per agent	NP-complete to determine existence of nontrivial feasible solution
negative externalities; one variable per agent	Algorithm 1 finds maximal feasible solution (linear #steps for utilities that decompose additively into piecewise constant functions); NP-hard to find social-welfare maximizing solution
negative externalities; two agents	NP-complete to determine existence of nontrivial feasible solution
utilities decompose additively; 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 additively into step functions.

In our opinion, the most important direction for future research is to study mechanism design aspects, in particular incentive compatibility constraints, in our setting—that is, how to incentivize agents to report their preferences truthfully. It should be noted that other work on specific settings with externalities, such as the work on mechanisms for charitable donations [8, 13], also makes only very limited progress on questions of incentive compatibility; however, in the context of auctions, the study of mechanism design for settings with externalities has perhaps been more successful [21, 22].

If it is possible for the mechanism to specify payments that should be made by or to the agents (possibly to or by an external party), and utilities are quasilinear, then incentive compatibility can be obtained using VCG payments [39, 5, 20], as long as we always choose a social welfare maximizing outcome. However, if payments to or by an external party are not allowed (that is, we enforce *budget balance*), there are results that prove that it is impossible to obtain incentive compatibility while always choosing an optimal outcome; for example, the well-known Myerson-Satterthwaite impossibility result [26] can be embedded in our domain to prove such an impossibility.[6] However, we may still be able to choose outcomes that are quite good, especially under certain assumptions. Such research may fall under the research agenda of *approximate mechanism design without money* [31].

Another important direction for future research is to investigate how preferences should be *elicited* in this domain. In this paper, we assumed that the agents' preferences are completely known to us, and we focused on the computational problem of finding an optimal (or at least good) outcome based on these preferences. However, in sufficiently rich environments, it can be impractical for an agent to reveal her preferences in

[6]This impossibility result considers the context where both a buyer and a seller have private information about their valuations for the good for sale. The result states that there is no mechanism that is efficient, does not run a deficit, is incentive compatible, and satisfies voluntary participation. We can embed this setting in our framework, by letting the seller control a variable representing whether she transfers the item to the buyer, and letting the buyer control a variable representing how much money he transfers to the seller. In both cases, there is a negative externality in the language of this paper.

their entirety up front. Alternatively, we can employ an *elicitation algorithm*, which sequentially asks agents to reveal parts of their preferences. This has the potential to greatly reduce the total amount of preference information communicated, because the algorithm need not ask the agents about parts of their preferences that have already been established to be irrelevant. Several overviews of preference elicitation in combinatorial auctions are available [36, 27].

References

[1] Gagan Aggarwal, Jon Feldman, S. Muthukrishnan, and Martin Pál. Sponsored search auctions with Markovian users. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pages 621–628, Shanghai, China, 2008.

[2] Ruggiero Cavallo, David Parkes, Adam Juda, Adam Kirsch, Alex Kulesza, Sébastien Lahaie, Benjamin Lubin, Loizos Michael, and Jeffrey Shneidman. TBBL: A tree-based bidding language for iterative combinatorial exchanges. In *Multidisciplinary Workshop on Advances in Preference Handling (IJCAI)*, Edinburgh, Scotland, UK, 2005.

[3] Jesús Cerquides, Ulle Endriss, Andrea Giovannucci, and Juan A. Rodríguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1221–1226, Hyderabad, India, 2007.

[4] Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. Multiagent resource allocation with k-additive utility functions. In *Workshop on Computer Science and Decision Theory*, 2004.

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

[6] Vincent Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.

[7] Vincent Conitzer, Jonathan Derryberry, and Tuomas Sandholm. Combinatorial auctions with structured item graphs. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 212–218, San Jose, CA, USA, 2004.

[8] Vincent Conitzer and Tuomas Sandholm. Expressive negotiation over donations to charities. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 51–60, New York, NY, USA, 2004.

[9] Vincent Conitzer, Tuomas Sandholm, and Paolo Santi. Combinatorial auctions with k-wise dependent valuations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 248–254, Pittsburgh, PA, USA, 2005.

[10] Florin Constantin, Malvika Rao, David C. Parkes, and Chien-Chung Huang. On expressing value externalities in position auctions. In *Sixth Workshop on Ad Auctions (at EC-10)*, 2010.

[11] Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[12] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 548–553, Stockholm, Sweden, August 1999.

[13] Arpita Ghosh and Mohammad Mahdian. Charity auctions on social networks. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1019–1028, 2008.

[14] Arpita Ghosh and Mohammad Mahdian. Externalities in online advertising. In *Proceedings of the 17th International World Wide Web Conference (WWW)*, pages 161–168, Beijing, China, 2008.

[15] Arpita Ghosh and Amin Sayedi. Expressive auctions for externalities in online advertising. In *Fifth Workshop on Ad Auctions (at EC-09)*, 2009.

[16] Ioannis Giotis and Anna R. Karlin. On the equilibria and efficiency of the GSP mechanism in keyword auctions with externalities. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pages 629–638, Shanghai, China, 2008.

[17] Renato Gomes, Nicole Immorlica, and Vangelis Markakis. Externalities in keyword auctions: An empirical and theoretical assessment. In *Fifth Workshop on Ad Auctions (at EC-09)*, 2009.

[18] Georg Gottlob and Gianluigi Greco. Combinatorial auctions with tractable winner determination. *ACM SIGecom Exchanges*, 7(1):15–18, 2007.

[19] Georg Gottlob and Gianluigi Greco. On the complexity of combinatorial auctions: Structured item graphs and hypertree decompositions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 152–161, San Diego, CA, USA, 2007.

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

[21] Philippe Jehiel and Benny Moldovanu. How (not) to sell nuclear weapons. *American Economic Review*, 86(4):814–829, 1996.

[22] Philippe Jehiel, Benny Moldovanu, and Ennio Stacchetti. Multidimensional mechanism design for auctions with externalities. *Journal of Economic Theory*, 85(2):258–293, 1999.

[23] David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pages 585–596, Shanghai, China, 2008.

[24] Piotr Krysta, Tomasz Michalak, Tuomas Sandholm, and Michael Wooldridge. Combinatorial auctions with externalities (extended abstract). In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Toronto, Canada, 2010.

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

[26] Roger Myerson and Mark Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 28:265–281, 1983.

[27] David Parkes. Iterative combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 2, pages 41–77. MIT Press, 2006.

[28] David Parkes, Ruggiero Cavallo, Nick Elprin, Adam Juda, Sébastien Lahaie, Benjamin Lubin, Loizos Michael, Jeffrey Shneidman, and Hassan Sultan. ICE: An iterative combinatorial exchange. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, Vancouver, BC, Canada, 2005.

[29] David Parkes, Jayant Kalagnanam, and Marta Eso. Achieving budget-balance with Vickrey-based payment schemes in exchanges. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1161–1168, Seattle, WA, USA, 2001.

[30] David Parkes and Tuomas Sandholm. Optimize-and-dispatch architecture for expressive ad auctions. In *First Workshop on Sponsored Search Auctions, at the ACM Conference on Electronic Commerce*, Vancouver, BC, Canada, June 2005.

[31] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 177–186, Stanford, CA, USA, 2009.

[32] David H. Reiley, Sai-Ming Li, and Randall A. Lewis. Northern exposure: A field experiment measuring externalities between search advertisements. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 297–304, Cambridge, MA, USA, 2010.

[33] Michael Rothkopf, Aleksandar Pekeč, and Ronald Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[34] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, January 2002.

[35] Tuomas Sandholm. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3):45–58, 2007.

[36] Tuomas Sandholm and Craig Boutilier. Preference elicitation in combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 10, pages 233–263. MIT Press, 2006.

[37] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 69–76, Bologna, Italy, July 2002.

[38] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for winner determination in combinatorial auctions. *Management Science*, 51(3):374–390, 2005. Special issue on Electronic Markets.

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