On the Impossibility of Black-Box Truthfulness Without Priors

Nicole Immorlica*

Brendan Lucier†

Abstract

We consider the problem of converting an arbitrary approximation algorithm for a single-parameter social welfare problem into a computationally efficient incentive compatible mechanism. We ask for reductions that are black-box, meaning that they require only oracle access to the given algorithm and (in particular) do not require explicit knowledge of the problem constraints. We demonstrate that, while such transformations are known to exist in Bayesian settings of partial information, any computationally efficient transformation that guarantees deterministic ex post incentive compatibility or universal truthfulness must sometimes incur a polynomially large loss in worst-case performance.

1 Introduction

In problems of social welfare maximization, a central authority wishes to assist a group of individuals in choosing from among a set of outcomes with the goal of maximizing the total outcome value to all participants. The classic result of Vickrey, Clarke and Groves demonstrates that for any such problem, there exists an ex post incentive compatible mechanism that maximizes social welfare. However, this construction requires that the mechanism optimize social welfare precisely, which may be computationally infeasible for large, complex problem instances. A primary open problem in algorithmic mechanism design is to understand the power of *computationally efficient* incentive compatible mechanisms to maximize social welfare.

It has recently been established that in Bayesian settings of partial information, there are general reductions that convert an arbitrary approximation algorithm into an incentive compatible mechanism with arbitrarily small loss in expected performance [8, 7, 2]. Moreover, such reductions are black-box, meaning that they require only oracle access to the prior type distributions and the algorithm, and proceed without knowledge of the feasibility constraints of the problem to be solved. Thus, in the Bayesian setting, the requirement that an approximation algorithm be incentive compatible is essentially without loss for a very broad class of social welfare problems.

While Bayesian settings are ubiquitous in the economics literature, much of the existing work in theoretical computer science is focused on an alternative goal of designing computationally efficient mechanisms that are incentive compatible *ex post* (i.e. without Bayesian priors) and achieve good worst-case approximations to the optimal social welfare. In light of the success with which algorithms can be converted into mechanisms in Bayesian settings, one might ask whether there exist reductions that transform approximation algorithms into ex post incentive compatible mechanisms without loss in worst-case approximation ratio. Note that this strengthens the demands of

*Department of Electrical Engineering and Computer Science, Northwestern University.

†Department of Computer Science, University of Toronto.

the reduction in two significant ways. First, it requires the stronger solution concept of ex post incentive compatibility, rather than Bayesian incentive compatibility. Second, it requires that the approximation factor of the original algorithm be preserved in the worst case over all possible inputs, rather than in expectation.

It is already known that such reductions are not possible for general multi-parameter social welfare problems. For some multi-parameter problems, no deterministic ex post incentive compatible mechanism can match the worst-case approximation factors achievable by algorithms without game-theoretic constraints [10]. Thus, for general social welfare problems, the relaxation from an ex post, worst-case setting to a Bayesian setting provably improves one's ability to implement algorithms as mechanisms. However, for the important special case of single-parameter problems, it is not known whether such a gap exists, and hence whether a Bayesian setting is required in order to convert algorithms to incentive compatible mechanisms without loss. We ask: does lossless reduction from mechanism design to algorithm design for single-parameter social welfare problems depend crucially on access to prior distributions?

One way to establish that the Bayesian setting is essential would be to demonstrate the existence of single-parameter social welfare problems for which there is a gap in the approximating power of arbitrary algorithms and ex post incentive compatible algorithms. This is an important open problem which has resisted much effort by the algorithmic mechanism design community, and is beyond the scope of our work.

Instead, we will focus upon the black-box nature of the reductions possible in the Bayesian setting. Does there exist a polytime transformation that implements an ex post incentive compatible mechanism, given access to an arbitrary approximation algorithm, without loss in worst-case approximation?

Results We provide a partial answer to the above question by first focusing on deterministic ex post incentive compatible mechanisms. For deterministic mechanisms, we answer our question in the negative by showing that any transformation that guarantees ex post incentive compatibility must incur a very large loss in worst-case performance for some algorithms and problem instances. Thus, any method of converting approximation algorithms into deterministic mechanisms that are ex post truthful must rely upon some extrinsically guaranteed property of the given algorithm or the problem to be solved.

We can extend our lower bound to randomized mechanisms that are *universally truthful*, meaning that each deterministic algorithm in the support of the transformed algorithm is itself ex post incentive compatible. We leave open the question of whether there exists a randomized transformation that guarantees truthfulness in expectation, which is a standard notion of ex post incentive compatibility for randomized mechanisms.

The main idea behind our impossibility result is to design a problem instance such that we can "hide" a non-monotonicity in an approximation algorithm. Roughly speaking, we consider a feasibility condition in which there are two different large allocations that can be made to certain sets of agents. When both allocations would result in high social welfare, our algorithm will choose between them in a non-incentive compatible way. In order for a transformation to fix these non-monotonicities without impacting the performance of the algorithm, it must switch the outcome from one to the other on some inputs; but to do so it must first determine what these allocations are, and hence find them by querying the original algorithm. However, our algorithm will have the property that, for many inputs, it is exponentially unlikely that the transformation would find both of the large allocations after polynomially many queries.

An implication for Bayesian mechanism design is that the reductions of [8, 7, 2] cannot be made into deterministic, prior-free reductions to Bayesian incentive compatible mechanims. In other words, it is crucial that the transformation be able to sample randomly from the distributions over agent types. It is therefore not only the relaxation of solution concept, but also the extra power of being able to sample typical inputs that makes such black-box reductions possible.

1.1 Related Work

Reductions from mechanism design to algorithm design in the Bayesian setting were first studied by Hartline and Lucier [8], who showed that any approximation algorithm for a single-parameter social welfare problem can be converted into a Bayesian incentive compatible mechanism with arbitrarily small loss in expected performance. This was extended to multi-parameter settings by Hartline, Kleinberg and Malekian [7] and Bei and Huang [2].

Some reductions from mechanism design to algorithm design are known for prior-free settings, for certain restricted classes of algorithms. Lavi and Swamy [9] consider mechanisms for multi-parameter packing problems and show how to construct a (randomized) β-approximation mechanism that is truthful in expectation, from any β-approximation that verifies an integrality gap. Dughmi, Roughgarden and Yan [6] extend the notion of designing mechanisms based upon randomized rounding algorithms, and obtain truthful in expectation mechanisms for a broad class of submodular combinatorial auctions. Dughmi and Roughgarden [5] give a construction that converts any FPTAS algorithm for a social welfare problem into a mechanism that is truthful in expectation, by way of a variation on smoothed analysis. Babaioff et al. [1] consider the equilibrium notion of *algorithmic implementation in undominated strategies* and give a technique for turning a β-algorithm into a $\beta(\log v_{max})$-approximation mechanism.

Many recent papers have explored limitations on the power of deterministic ex post incentive compatible mechanisms to approximate social welfare. Papadimitriou, Schapira and Singer [10] gave an example of a social welfare problem for which constant-factor approximation algorithms exist, but any polytime ex post incentive compatible mechanism attains at best a polynomial approximation factor. A similar gap for the submodular combinatorial auction problem was established by Dobzinski [4]. For the general combinatorial auction problem, such gaps have been established for the restricted class of max-in-range mechanisms by Buchfuhrer et al. [3].

2 Definitions

In a single-parameter, real-valued social welfare maximization problem, we are given an input vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, where each v_i is assumed to be drawn from a known set $V_i \subseteq \mathbb{R}$. The goal is to choose some $\mathbf{x} \in \Gamma \subseteq \mathbb{R}^n$ such that $\mathbf{v} \cdot \mathbf{x}$ is maximized, where $\Gamma \subseteq \mathbb{R}^n$ is a space of allowable outcomes. We think of the feasibility set Γ as defining an instance of the social welfare maximization problem. We will write $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where each $x_i \in \mathbb{R}$.

Given an instance Γ of the social welfare problem, we will write $OPT_\Gamma(\mathbf{v})$ for the allocation in Γ that maximizes $\mathbf{v} \cdot \mathbf{x}$. We will take the convention that the social welfare for an allocation outside Γ is $-\infty$.

An algorithm $\mathcal{A}$ defines a mapping from input vectors $\mathbf{v}$ to outcomes $\mathbf{x}$. In general an algorithm can be randomized, though in our construction we will use only deterministic algorithms. Given algorithm $\mathcal{A}$, we will write $approx_\Gamma(\mathcal{A})$ for the worst-case approximation ratio of $\mathcal{A}$ for problem Γ. That is, $approx_\Gamma(\mathcal{A}) = \min_{\mathbf{v}\in V} \frac{\mathcal{A}(\mathbf{v})}{OPT_\Gamma(\mathbf{v})}$. Note that $approx_\Gamma(\mathcal{A}) \le 1$ for all Γ and $\mathcal{A}$.

We will consider such a social welfare optimization problem in a mechanism design setting with n rational agents, where each agent possesses one value from the input vector as private information. We think of an outcome $\mathbf{x}$ as representing an *allocation* to the agents, where x_i is the allocation to agent i. A (direct-revelation) mechanism for our optimization problem then proceeds by eliciting declared values $\mathbf{b} \in \mathbb{R}^n$ from the agents, then applying an allocation rule $\mathcal{A} : \mathbb{R}^n \to \Gamma$ that maps $\mathbf{b}$ to an allocation $\mathbf{x}$, and a payment rule that maps $\mathbf{b}$ to a payment vector $\mathbf{p}$. We will write $\mathbf{x}(\mathbf{b})$ and $\mathbf{p}(\mathbf{b})$ for the allocations and payments that result on input $\mathbf{b}$. The *utility* of agent i, given that the agents declare $\mathbf{b}$ and his true private value is v_i, is taken to be $v_i x_i(\mathbf{b}) - p_i(\mathbf{b})$.

A deterministic mechanism is ex post incentive compatible (EPIC) if each agent maximizes its utility by reporting its value truthfully, regardless of the reports of the other agents. That is, $v_i x_i(v_i, \mathbf{b}_{-i}) - p_i(v_i, \mathbf{b}_{-i}) \ge v_i x_i(b_i, \mathbf{b}_{-i}) - p_i(b_i, \mathbf{b}_{-i})$ for all i, all $v_i, b_i \in V_i$, and all $\mathbf{b}_{-i} \in V_{-i}$. We say that an algorithm is EPIC if there exists a payment rule such that the resulting mechanism is EPIC. It is known that an algorithm is EPIC if and only if, for all i and all $\mathbf{v}_{-i}$, $x_i(v_i, \mathbf{v}_{-i})$ is monotone non-decreasing as a function of v_i.

We say that a randomized mechanism is *universally truthful* if every deterministic mechanism in its support is EPIC. Likewise, a randomized algorithm is universally truthful if its support is composed entirely of EPIC algorithms.

In a Bayesian setting, the true values of the agents are drawn independently from publicly known distributions: $v_i \sim F_i$ for each i. Write $x_i(v_i) = \mathbf{E}_{\mathbf{v}_{-i}\sim\mathbf{F}_{-i}}[x_i(v_i, \mathbf{v}_{-i})]$ and $p_i(v_i) = \mathbf{E}_{\mathbf{v}_{-i}\sim\mathbf{F}_{-i}}[p_i(v_i, \mathbf{v}_{-i})]$ to be the expected allocation and payment to agent i if he declares v_i, given that other agents report their values truthfully. We say that a mechanism is Bayesian incentive compatible (BIC) if reporting values truthfully is a Bayes-Nash equilibrium; that is, $v_i x_i(v_i) - p_i(v_i) \ge v_i x_i(v_i{}') - p_i(v_i{}')$ for all $v_i, v_i{}'$. We say allocation rule $\mathcal{A}$ is BIC if there exists a payment rule such that the resulting mechanism is BIC. It is known (Myerson, 81) that an allocation rule $\mathcal{A}$ is BIC if and only if the function $x_i(v_i)$ is monotone non-decreasing for each i.

A *polytime transformation* $\mathcal{T}$ is a social welfare algorithm that is given black-box access to an algorithm $\mathcal{A}$. We will write $\mathcal{T}(\mathcal{A}, \mathbf{v})$ for the allocation returned by $\mathcal{T}$ on input $\mathbf{v}$, given that its black-box access is to algorithm $\mathcal{A}$. Then, for any $\mathcal{A}$, we can think of $\mathcal{T}(\mathcal{A}, \cdot)$ as an algorithm for social welfare maximization; we think of this as the algorithm $\mathcal{A}$ *transformed by* $\mathcal{T}$. We write $\mathcal{T}(\mathcal{A})$ for the allocation rule that results when $\mathcal{A}$ is transformed by $\mathcal{T}$. Note that $\mathcal{T}$ is not parameterized by Γ; informally speaking, $\mathcal{T}$ has no knowledge of the feasibility constraint Γ being optimized by a given algorithm $\mathcal{A}$.

We say that transformation $\mathcal{T}$ is ex post incentive compatible if, for all $\mathcal{A}$, $\mathcal{T}(\mathcal{A})$ is an EPIC allocation rule. Similarly, we say $\mathcal{T}$ is universally truthful if $\mathcal{T}(\mathcal{A})$ is universally truthful for all $\mathcal{A}$. Note that this incentive compatibility is independent of the problem instance Γ.

3 A Lower Bound for Ex Post IC Transformations

In this section we show that, for any ex post IC transformation $\mathcal{T}$, there is some allocation problem Γ and algorithm $\mathcal{A}$ for Γ such that $\mathcal{T}$ degrades the worst-case performance of $\mathcal{A}$ by a polynomially large factor.

Theorem 3.1. *There is a constant $c > 0$ such that, for any polytime universally truthful transformation $\mathcal{T}$, there is an algorithm $\mathcal{A}$ and problem instance Γ such that $\frac{approx_\Gamma(\mathcal{A})}{approx_\Gamma(\mathcal{T}(\mathcal{A}))} \ge n^c$.*

The high-level idea behind our proof of Theorem 3.1 is as follows. We will construct an algorithm $\mathcal{A}$ and input vectors $\mathbf{v}$ and $\mathbf{v}'$ such that, for each agent i in some large subset of the players, ${v_i}' > v_i$ but $\mathcal{A}_i(\mathbf{v}') < \mathcal{A}_i(\mathbf{v})$. This does not immediately imply that $\mathcal{A}$ is non-truthful, but we will show that it does imply non-truthfulness under a certain feasibility condition Γ. Thus, any ex post IC transformation $\mathcal{T}$ must alter the allocation of $\mathcal{A}$ either on input $\mathbf{v}$ or on input $\mathbf{v}'$. However, we will craft our algorithm in such a way that, on input $\mathbf{v}$, the only allocations that the transformation will observe given polynomially many queries of $\mathcal{A}$ will be $\mathcal{A}(\mathbf{v})$, plus allocations that have significantly worse social welfare than $\mathcal{A}(\mathbf{v})$, with high probability. Similarly, on input $\mathbf{v}'$, with high probability the transformation will only observe allocation $\mathcal{A}(\mathbf{v}')$ plus allocations that have significantly worse social welfare than $\mathcal{A}(\mathbf{v}')$. Thus, in order to guarantee that it generates an ex post IC allocation rule, the transformation will be forced to choose an outcome with poor social welfare for either input $\mathbf{v}$ or $\mathbf{v}'$, reducing the worst-case performance of the algorithm $\mathcal{A}$.

We now turn to a formal proof of Theorem 3.1 by first describing a family of feasibility constraints and algorithms, then showing that for each $\mathcal{T}$ there is a feasibility constraint and algorithm from this family that satisfies the requirements of the theorem. This proof will be limited to deterministic transformations; at the end of the section we discuss extensions to randomized transformations that generate universally truthful mechanisms.

3.1 Construction

In the problems we consider, each private value v_i is chosen from $\{0, 1\}$. That is, we will set $V_i = \{0, 1\}$ for all $i \in [n]$. We can therefore interpret an input vector as a subset $y \subseteq [n]$, corresponding to those agents with value 1. We can therefore define $\mathcal{A}(y)$, $OPT_\Gamma(y)$, etc., for a given subset $y \subseteq [n]$. Also, for $a \geq 0$ and $y \subseteq [n]$, we will write $\mathbf{x}_y^a$ for the allocation in which each agent $i \in y$ is allocated a, and each agent $i \notin y$ is allocated 0.

Feasible Allocations We will first define a family of feasibility constraints. Roughly speaking, we will choose some $\alpha \in (0, 1)$ and sets $S, T \subseteq [n]$ of agents. The feasible allocations will then be $\mathbf{x}_{[n]}^{\alpha^2}$, $\mathbf{x}_S^1$, $\mathbf{x}_T^\alpha$, and $\mathbf{x}_W^\alpha$ for all sufficiently small sets W. That is, we can allocate α^2 to every agent, 1 to all agents in S, α to all agents in T, or α to each of any small number of agents. We will also require that S and T satisfy certain properties, which essentially state that S and T are sufficiently large and have a sufficiently large intersection.

More formally, define parameters $\alpha \in (0, 1)$, $r \geq 1$, and $t \geq 1$ (which we will fix later), such that $r^5 t \leq n$ and $r \geq \max\{2\alpha^{-1}, 6\alpha^{-2/3}\}$. We think of t as a bound on the size of "small" sets, and we think of r as a ratio between the sizes of "small" and "large" sets. Write $\Psi = \{x_W^\alpha \mid |W| \leq t\}$ to be all subsets of agents of size at most t.

Suppose that V, S, and T are subsets of $[n]$. We say that the triple V, S, T is *admissible* if the following conditions hold:

1. $V \subset S \cap T$,
2. $|V| = rt$,
3. $|S \cap T| = r^2 t$, and
4. $|S| = |T| = r^3 t$.

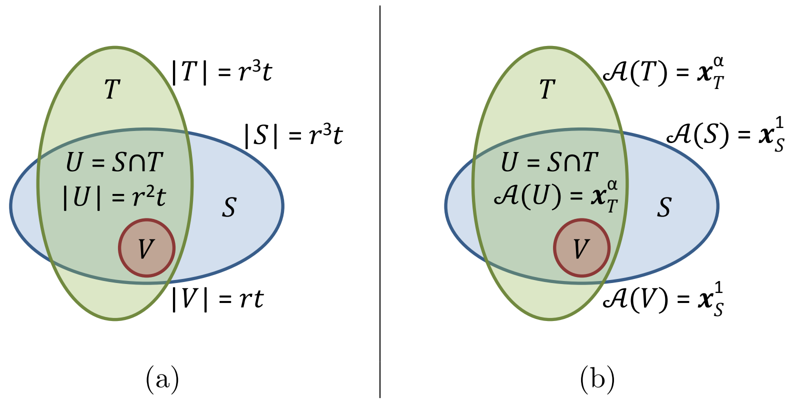

Figure 1: (a) Visualization of typical admissible sets of bidders V, S, and T, given size parameters r and t, and (b) the corresponding allocations of algorithm $\mathcal{A} = \mathcal{A}_{V,S,T}$.

In general, for a given admissible V, S, and T, we will tend to write $U = S \cap T$ for notational convenience. See Figure 1(a) for an illustration of the relationship between the sets in an admissible triple. For each admissible tuple V, S, T, we define a corresponding feasibility constraint

$$\Gamma_{V,S,T} = \{x_S^1, x_T^{\alpha}, x_{[n]}^{\alpha^2}\} \cup \Psi.$$

Note that $\Gamma_{V,S,T}$ does not depend on V; we include set V purely for notational convenience.

The Algorithm We now define the algorithm $\mathcal{A}_{V,S,T}$ corresponding to an admissible tuple V, S, T. We think of $\mathcal{A}_{V,S,T}$ as an approximation algorithm for the social welfare problem $\Gamma_{V,S,T}$ and later show that there is no EPIC transformation of $\mathcal{A}_{V,S,T}$ without a significant loss in worst-case approximation. Given $y \subset [n]$, let $W(y)$ denote the first (up to) t agents in set y, in order of agent index. The algorithm $\mathcal{A}_{V,S,T}$ is then described as Algorithm 1.

Algorithm 1: Allocation Algorithm $\mathcal{A}_{V,S,T}$

Input: Subset $y \in [n]$ of agents with value 1
Output: An allocation $\mathbf{x} \in \Gamma_{V,S,T}$

if $|y| < \alpha^{-1/3}t$ **then**
  return $\mathbf{x}_{W(y)}^{\alpha}$
else if $|y \cap V| > \alpha|y|$ **then**
  return $\mathbf{x}_S^1$
else if $|y \cap T| > \alpha^{2/3}|y|$ **then**
  return $\mathbf{x}_T^{\alpha}$
else if $|y \cap S| > \alpha^{1/3}|y|$ **then**
  return $\mathbf{x}_S^1$
else
  return $\mathbf{x}_{[n]}^{\alpha^2}$
end

Note that $\mathcal{A}_{V,S,T}$ always returns an outcome from $\Gamma_{V,S,T}$. For example, if we let $U = S \cap T$, we have that $\mathcal{A}(S) = \mathbf{x}_S^1$, $\mathcal{A}(T) = \mathbf{x}_T^\alpha$, $\mathcal{A}(U) = \mathbf{x}_T^\alpha$, and $\mathcal{A}(V) = \mathbf{x}_S^1$. See Figure 1(b) for an illustration of the relationship between these sets. The idea underlying the algorithm is that $\mathcal{A}_{V,S,T}$ returns $\mathbf{x}_S^1$ if y is an approximate subset of S, and returns $\mathbf{x}_T^\alpha$ if y is an approximate subset of T. Otherwise, $\mathcal{A}_{V,S,T}$ allocates α^2 to all players. However, this allocation rule is ambiguous whenever y is approximately a subset of both S and T; in this case, the algorithm will "break the tie" by returning $\mathbf{x}_T^\alpha$, except when y is an approximate subset of V, in which case the algorithm returns $\mathbf{x}_S^1$. Finally, in the special case that y is very small, $\mathcal{A}_{V,S,T}$ will return an allocation from Ψ.

3.2 Analysis

For the remainder of this section we will write $\beta = \alpha^{1/3}$ for notational convenience. We begin by bounding the approximation ratio of $\mathcal{A}_{V,S,T}$ for problem $\Gamma_{V,S,T}$, for any admissible tuple V, S, T.

Lemma 3.2. *$approx_{\Gamma_{V,S,T}}(\mathcal{A}_{V,S,T}) \geq \alpha^{5/3}$.*

Proof. Choose any admissible tuple V, S, T and fix some $y \subseteq [n]$. We will show $\frac{\mathcal{A}_{V,S,T}(y)}{OPT_{\Gamma_{V,S,T}}(y)} \geq \alpha^{5/3}$ by considering various cases for y.

Case 1: $|y| < \beta^{-1}t$. Then $\mathcal{A}$ obtains social welfare $\alpha \min\{|y|, t\} \geq \alpha\beta|y|$, whereas $OPT(y) \leq |y|$, so $\frac{\mathcal{A}(y)}{OPT(y)} \geq \alpha\beta > \alpha^{5/3}$.

Case 2: $|y \cap V| \geq \beta^3|y|$. Then $\mathcal{A}$ gets social welfare $|y \cap S| \geq |y \cap V| \geq \beta^3|y|$, whereas $OPT(y) \leq |y|$, so $\frac{\mathcal{A}(y)}{OPT(y)} \geq \beta^3 > \alpha^{5/3}$.

Case 3: $|y \cap T| \geq \beta^2|y|$, but $|y \cap V| < \beta^3|y|$. Then $\mathcal{A}$ gets social welfare $\alpha|y \cap T| \geq \alpha\beta^2|y|$, whereas $OPT(y) \leq |y|$, so $\frac{\mathcal{A}(y)}{OPT(y)} \geq \alpha\beta^2 = \alpha^{5/3}$.

Case 4: $|y \cap S| \geq \beta|y|$. Then $\mathcal{A}$ gets social welfare $|y \cap S| \geq \beta|y|$, whereas $OPT(y) \leq |y|$, so $\frac{\mathcal{A}(y)}{OPT(y)} \geq \beta > \alpha^{5/3}$.

Case 5: $|y \cap S| < \beta|y|$, $|y \cap T| < \beta^2|y|$, and $|y| \geq \beta^{-1}t$. Then $\mathcal{A}$ gets social welfare $\alpha^2|y|$. The alternative $\mathbf{x}_S^1$ receives welfare at most $|y \cap S| < \beta|y|$ and the alternative $\mathbf{x}_T^\alpha$ receives welfare at most $\alpha|y \cap T| < \alpha\beta^2|y| < \beta|y|$. Any alternative x_W^α for $|W| \leq t$ receives welfare at most $\alpha t \leq \alpha\beta|y| < \beta|y|$. We conclude $OPT(y) < \beta|y|$, so $\frac{\mathcal{A}(y)}{OPT(y)} \geq \alpha^2/\beta = \alpha^{5/3}$. □

Suppose now that $\mathcal{A}'$ is any algorithm for problem $\Gamma_{V,S,T}$, and suppose that $\mathcal{A}'$ has approximation ratio better than α^2. We will show that $\mathcal{A}'$ is then very restricted in the allocations it can return on input $y = V$. We also show that if $\mathcal{A}'$ is EPIC, then the allocations on inputs $y = V$ and $y = U$ are restricted further still. As any EPIC transformation of $\mathcal{A}$ is itself an algorithm for problem $\Gamma_{V,S,T}$, these claims will later play a key role in our impossibility result.

Claim 3.3. *If algorithm $\mathcal{A}'$ is such that $approx_{\Gamma_{V,S,T}}(\mathcal{A}') > \alpha^2$, then $\mathcal{A}'(V) \in \{\mathbf{x}_S^1, \mathbf{x}_T^\alpha\}$.*

Proof. For $y = V$, the best possible allocation is $\mathbf{x}_S^1$, for a social welfare of $|y|$. Allocation $\mathbf{x}_{[n]}^{\alpha^2}$ is off by a factor of α^2. Also, for any W with $|W| \leq t$, we have $|W| \leq |V|/r \leq |y|/r$, so allocation $\mathbf{x}_W^\alpha$ receives social welfare at most $\alpha|y|/r \leq \alpha^2|y|$ by our choice of r which is also off by a factor of α^2. So the required two allocations are the only ones the lead to an approximation factor better than α^2. □

Claim 3.4. *Suppose $\mathcal{A}'$ is an algorithm for problem $\Gamma_{V,S,T}$. If $\mathcal{A}'(V) = \mathbf{x}_S^1$ and $\mathcal{A}'(U) \neq \mathbf{x}_S^1$ then $\mathcal{A}'$ is not ex post IC.*

Proof. Take any set W with $V \subseteq W \subseteq U$, $|W| = |V| + 1$. Then, on input W, $\mathcal{A}'$ must return allocation $\mathbf{x}_S^1$ by monotonicity: from input V to input W a single agent's declaration increased, and that agent received an allocation of 1, so since his allocation cannot decrease it must remain 1, and the only allocation in which any agent receives 1 is $\mathbf{x}_S^1$. By the same argument, $\mathcal{A}'$ returns allocation $\mathbf{x}_S^1$ for all W such that $V \subseteq W \subseteq U$, and in particular for $W = U$. This contradicts the fact that $\mathcal{A}'$ does not return $\mathbf{x}_S^1$ on input U. $\square$

In light of these claims, our strategy for proving Theorem 3.1 will be to show that a polytime transformation $\mathcal{T}$ is unlikely to encounter the allocation $\mathbf{x}_T^\alpha$ during its sampling when the input is V, given that the sets V, S, and T are chosen uniformly at random over all admissible tuples. Similarly, a transformation is unlikely to encounter the allocation $\mathbf{x}_S^1$ during its sampling on input U. Since $\mathcal{T}$ is EPIC, Claim 3.4 will imply that $\mathcal{T}(\mathcal{A})$ returns an allocation outside $\{\mathbf{x}_S^1, \mathbf{x}_T^\alpha\}$ on input V. Claim 3.3 then implies that $\mathcal{T}(\mathcal{A})$ has approximation ratio less than α^2.

The following technical lemmas bound the likelihood that a polytime transformation will encounter the relevant allocations during its sampling, given a uniformly random choice of admissible tuples V, S, T.

Lemma 3.5. *Fix V and S satsifying the requirements of admissibility, and choose any y such that $|y| > \beta^{-1}t$ and $|y \cap V| < \beta^3|y|$. Then $\Pr_T[|y \cap T| > \beta^2|y|] \leq e^{-\frac{1}{r+1}(1-\beta^3)\beta^{-1}t}$, with probability taken over all choices of T that are admissible given V and S.*

Proof. Since $|y \cap V| < \beta^3|y|$, at least $(1-\beta^3)|y|$ elements of y lie outside V. For the event $|y \cap T| > \beta^2|y|$ to occur, it must be that at least $(\beta^2 - \beta^3)|y|$ of these elements lie in T. Thus, of the elements of y chosen outside V, the fraction that fall in T must be $\frac{\beta^2-\beta^3}{1-\beta^3} = \frac{\beta^2}{1+\beta+\beta^2} > \frac{\beta^2}{3}$.

Any element chosen from $S - V$ has probability $\frac{|U|-|V|}{|S|-|V|} = \frac{1}{r+1}$ of being in T. Any element chosen from $[n] - S$ has probability $\frac{|T|-|U|}{[n]-|S|} \leq \frac{|U|(r-1)}{|S|(r^2-1)} < \frac{1}{r+1}$ of being in T (recalling that $n \geq r^2|S|$). Thus, each element of y outside V has chance at most $\frac{1}{r+1}$ of being in T.

Thus, if we think of the elements of y lying outside V as $(1-\beta^3)|y|$ independent events, each with success (i.e. lying in T) at most $\frac{1}{r+1}$, and we write X for the random variable denoting the number of such events that are successful, Chernoff bounds imply

$$\Pr\left[X > \frac{\beta^2}{3}|y|(1-\beta^3)\right] \leq \Pr\left[X > 2\frac{1}{r+1}|y|(1-\beta^3)\right] < e^{-\frac{1}{r+1}|y|(1-\beta^3)} < e^{-\frac{1}{r+1}(1-\beta^3)\beta^{-1}t}.$$

Note that the first inequality holds because of our assumption that $r > 6\alpha^{-2/3} = 6\beta^{-2}$. Since the event that $X > \frac{\beta^2}{3}|y|(1-\beta^3)$ dominates the event $|y \cap T| > \beta^2|y|$, we obtain the desired result. $\square$

Lemma 3.6. *Fix T and U satisfying the requirements of admissibility (i.e. so that there are admissible tuples such that $U = S \cap T$), and choose any y such that $|y| > \beta^{-1}t$ and $|y \cap T| < \beta^2|y|$. Then $\Pr_S[|y \cap S| > \beta|y|] \leq e^{-\frac{1}{r+1}(1-\beta^2)\beta^{-1}t}$, with probability taken over all choices of S consistent with U and T.*

Proof. Since $|y \cap T| < \beta^2|y|$, at least $(1-\beta^2)|y|$ elements of y lie outside T. For the event $|y \cap S| > \beta|y|$ to occur, it must be that at least $(\beta - \beta^2)|y|$ of these elements lie in S. Thus, of the elements of y chosen outside T, the fraction that fall in S must be at least $\frac{\beta-\beta^2}{1-\beta^2} > \frac{\beta}{2}$.

Any element chosen from $[n]-T$ has probability $\frac{|S|-|U|}{[n]-|T|} \leq \frac{|U|(r-1)}{|T|(r^2-1)} < \frac{1}{r+1}$ of being in S (recalling that $n \geq r^2|T|$). Thus, each element of y outside T has chance at most $\frac{1}{r+1}$ of being in S.

Thus, if we think of the elements of y lying outside T as $(1-\beta^2)|y|$ independent events, each of which occurs with probability at most $\frac{1}{r+1}$, and we write X for the random variable denoting the number of such events that are successful, Chernoff bounds imply

$$\Pr\left[X > \frac{\beta}{2}|y|(1-\beta^2)\right] \leq \Pr\left[X > 2\frac{1}{r+1}|y|(1-\beta^2)\right] < e^{-\frac{1}{r+1}|y|(1-\beta^2)} < e^{-\frac{1}{r+1}(1-\beta^2)\beta^{-1}t}.$$

Note that the first inequality holds because of the assumption that $r \geq 6\beta^{-2} \geq 4\beta^{-1}$. Since event $X > \frac{\beta^2}{3}|y|(1-\beta^2)$ is only more likely than the event $|y \cap S| > \beta|y|$, we obtain the desired result. □

Lemma 3.7. *Fix T and U satisfying the requirements of admissibility, and choose any y such that $|y| > \beta^{-1}t$. Then $\Pr_V[|y \cap V| > \beta^3|y|] \leq e^{-\frac{1}{r}\beta^{-1}t}$.*

Proof. For the event $|y \cap V| > \beta^3|y|$ to occur, it must be that $\beta^3|y|$ of the elements in y lie in V. Thus, the fraction of elements of y that fall in V must be at least β^3. Any element chosen from U has probability $\frac{1}{r}$ of being in V, and any element chosen from outside U has probability 0 of lying in V. Thus, each element of y has chance at most $\frac{1}{r}$ of being in V.

Thus, if we think of the elements of y as $|y|$ independent events, each with success (i.e. lying in S) at most $\frac{1}{r}$, and we write X for the random variable denoting the number of such events that are successful, Chernoff bounds imply

$$\Pr\left[X > \beta^3|y|\right] \leq \Pr\left[X > 2\frac{1}{r}|y|\right] < e^{-\frac{1}{r}|y|} < e^{-\frac{1}{r}\beta^{-1}t}.$$

Note that the first inequality holds because of the assumption that $r \geq 2\alpha^{-1} = 2\beta^{-3}$. Since event $X > \beta^3|y|$ is only more likely than the event $|y \cap V| > \beta^3|y|$, we obtain the desired result. □

3.3 Proof of Main Theorem

We can now set our parameters t, r, and α. We will choose $t = n^{1/5}$, $r = 2n^{3/20}$, and $\alpha = n^{-3/20}$. Note that $\beta = n^{-1/20}$. We then note that, for sufficiently large n,

- $r^5 t \leq n$,
- $r \geq \max\{2\alpha^{-1}, 6\alpha^{-2/3}\}$, and
- $e^{-\frac{1}{r+1}\beta^{-1}(1-\beta^2)t} < e^{-\frac{1}{r}\beta^{-1}t} < e^{-n^{1/20}}$.

All of the restrictions we required of our parameters are therefore satisfied, and the probabilities from Lemmas 3.5, 3.6 and 3.7 are exponentially small.

Proof of Theorem 3.1 : For each admissible V, S, T, write $\mathcal{A}'_{V,S,T}$ for $\mathcal{T}(\mathcal{A}_{V,S,T})$. Suppose for contradiction that, for every V, S, T, $\mathcal{A}'_{V,S,T}$ has approximation ratio better than α^2. By Claim 3.3, $\mathcal{A}'_{V,S,T}(V) \in \{\mathbf{x}_T^{\alpha}, \mathbf{x}_S^{1}\}$.

For any given V, S, T, consider the sequence of queries of algorithm $\mathcal{A}$ performed by $\mathcal{T}$ on input $y_1 = V$. Suppose $z \subseteq [n]$ is one such query. Then if $|z| < \beta^{-1}t$, the transformation will observe an allocation in ψ. If $|z \cap V| \geq \beta^3 |z|$, the transformation will observe allocation $\mathbf{x}_S^1$. For any other query, Lemma 3.5 implies that if $\mathcal{T}$ has no knowledge of T, then it is exponentially unlikely (over all choices of T) that $\mathcal{T}$ will observe an allocation other than $\mathbf{x}_{[n]}^{\alpha^2}$. Thus, by the union bound, it is exponentially unlikely that any of a polynomial number of queries reveals any information about the set T, and in particular it is exponentially unlikely that $\mathcal{T}$ will observe allocation $\mathbf{x}_T^\alpha$ after polynomially many queries.

Next consider the sequence of queries of algorithm $\mathcal{A}$ performed by $\mathcal{T}$ on input $y_2 = U$. Again, if $z \subseteq [n]$ is one such query, then if $|z| < \beta^{-1}t$ the transformation will observe an allocation in ψ. If $|z \cap U| \geq \beta^2 |z|$, Lemma 3.7 implies that if $\mathcal{T}$ has no knowledge of V, then it is exponentially unlikely (over all choices of V) that $\mathcal{T}$ will observe an allocation other than $\mathbf{x}_T^\alpha$. Likewise, if $|z \cap U| < \beta^2 |z|$, Lemma 3.6 implies that it is exponentially unlikely that $\mathcal{T}$ will observe an allocation other than $\mathbf{x}_{[n]}^{\alpha^2}$, over all choices of S. It is therefore exponentially unlikely that any query reveals any information about the set V or S beyond the identity of set U, and therefore by the union bound it is exponentially unlikely that $\mathcal{T}$ will observe allocation $\mathbf{x}_S^1$ after polynomially many queries.

We conclude that, by the union bound and the probabilistic method, there is some choice of V, S, T such that $\mathcal{T}(\mathcal{A}_{V,S,T})$ does not encounter allocation $\mathbf{x}_S^1$ on input U and does not encounter allocation $\mathbf{x}_T^\alpha$ on input V. Therefore $\mathcal{A}'_{V,S,T}(U) = x_T^\alpha$ and $\mathcal{A}'_{V,S,T}(V) = x_S^1$. However, Claim 3.4 then implies that $\mathcal{A}'_{V,S,T}$ is not ex post IC, contradicting the fact that $\mathcal{T}$ is an EPIC transformation.

We therefore conclude that, for any EPIC transformation $\mathcal{T}$, there is some admissible V, S, T such that $approx_{\Gamma_{V,S,T}}(\mathcal{A}'_{V,S,T}) \leq \alpha^2$. Since $approx_{\Gamma_{V,S,T}}(\mathcal{A}_{V,S,T}) = \alpha^{5/3}$ for all admissible V, S, T, we have that $\frac{approx_{\Gamma_{V,S,T}}(\mathcal{A}_{V,S,T})}{approx_{\Gamma_{V,S,T}}(\mathcal{T}(\mathcal{A}_{V,S,T}))} \geq \alpha^{-1/3} = n^{1/20}$. □

3.4 Discussion

We now make note of a few simple extensions of Theorem 3.1. First, while we proved Theorem 3.1 for deterministic transformations, the result extends to randomized transformations that generate universally truthful allocation rules. Indeed, suppose that transformation $\mathcal{T}$ is such that $\mathcal{A}'_{V,S,T} = \mathcal{T}(\mathcal{A}_{V,S,T})$ is a universally truthful algorithm with $\mathbf{E}[\mathcal{A}'_{V,S,T}(V)] > \alpha^2$. Then there is a deterministic ex post IC algorithm $\mathcal{A}''$ in the support of $\mathcal{A}'_{V,S,T}$ such that $\mathcal{A}''_{V,S,T}(V) > \alpha^2$. This then leads to a contradiction in precisely the same manner as the proof of Theorem 3.1. We must therefore conclude that any such transformation satisfies $\mathbf{E}[\mathcal{A}'_{V,S,T}(V)] \leq \alpha^2$, and hence degrades the worst-case approximation factor of the algorithm $\mathcal{A}_{V,S,T}$ by a polynomially large factor.

Second, a corollary of Theorem 3.1 is that any deterministic transformation that converts approximation algorithms into BIC mechanisms *without access to the prior distribution* must degrade the expected performance of the approximation algorithm by a polynomially large factor for some distributions. This follows from two observations: first, if a transformation is Bayesian IC for every distribution $\mathbf{F}$ then it must be ex post IC; and second, if a transformation degrades the expected social welfare of an algorithm by a factor of at most c for every distribution of inputs then it must degrade the worst-case approximation ratio by a factor of at most c.

It is therefore the case that no deterministic prior-free reduction can convert algorithms to BIC mechanisms without loss. The restriction to deterministic transformations is admittedly strong, though we note that existing black-box reductions for Bayesian mechanisms use randomization only to sample from the distributions of agent values, and are otherwise deterministic [8, 7].

4 Conclusions and Future Directions

We have demonstrated that any prior-free polytime method for converting approximation algorithms into universally truthful mechanisms must incur a large loss in social welfare for some algorithms. We conclude that a relaxation of the solution concept of dominant strategy truthfulness (e.g. to Bayesian incentive compatibility) is a crucial requirement for general black-box reductions.

Our impossibility result applies to randomized mechanisms that are universally truthful, but not to transformations that guarantee only truthfulness in expectation. We leave open the question of whether there exist lossless black-box transformations from algorithms to mechanisms that are truthful in expectation.

While we have shown that general reductions are not possible when one's goals are ex post incentive compatibility and worst-case approximation, there are numerous relaxations of these requirements that may allow black-box transformations to exist. For example, one might have access to a prior distribution on agent types, and wish to generate a mechanism that is ex post incentive compatible but only preserves expected performance over the type distribution. Alternatively, one may wish to generate a mechanism that is Bayesian incentive compatible, but preserves worst-case approximation.

Our construction required that we allow for very general feasibility constraints. It may be possible to obtain transformations for natural classes of social welfare problems, such as the important subclass of downard-closed feasibility constraints.

References

[1] M. Babaioff, R. Lavi, and E. Pavlov. Single-value combinatorial auctions and algorithmic implementation in undominated strategies. *Journal of the ACM*, 2009.

[2] X. Bei and Z. Huang. Bayesian incentive compatibility via fractional assignments. In *Proc. 22nd ACM Symp. on Discrete Algorithms*, 2011.

[3] D. Buchfuhrer, S. Dughmi, H. Fu, R. Kleinberg, E. Mossel, C. Papadimitriou, M. Schapira, Y. Singer, and C. Umans. Inapproximability for vcg-based combinatorial auctions. In *Proc. 21st ACM Symp. on Discrete Algorithms*, 2010.

[4] S. Dobzinski. An impossibility result for truthful combinatorial auctions with submodular valuations. In *Proc. 42nd ACM Symp. on Theory of Computing*, 2011.

[5] S. Dughmi and T. Roughgarden. Black-box randomized reductions in algorithmic mechanism design. In *Proc. 51st IEEE Symp. on Foundations of Computer Science*, 2010.

[6] S. Dughmi, T. Roughgarden, and Q. Yan. From convex optimization to randomized mechanisms: Toward optimal combinatorial auctions. In *Proc. 42nd ACM Symp. on Theory of Computing*, 2011.

[7] J. Hartline, R. Kleinberg, and A. Malekian. Bayesian incentive compatibility via matchings. In *Proc. 22nd ACM Symp. on Discrete Algorithms*, 2011.

[8] J. Hartline and B. Lucier. Bayesian algorithmic mechanism design. In *Proc. 41st ACM Symp. on Theory of Computing*, 2010.

[9] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. In *Proc. 46th IEEE Symp. on Foundations of Computer Science*, 2005.

[10] C. Papadimitriou, M. Schapira, and Y. Singer. On the hardness of being truthful. In *Proc. 49th IEEE Symp. on Foundations of Computer Science*, 2008.