

# Bayesian Incentive Compatibility via Matchings

Jason D. Hartline\*

Robert Kleinberg<sup>†</sup>

Azarakhsh Malekian<sup>‡</sup>

## Abstract

We give a simple reduction from Bayesian incentive compatible mechanism design to algorithm design in settings where the agents' private types are multi-dimensional. The reduction preserves performance up to an additive loss that can be made arbitrarily small in polynomial time in the number of agents and the size of the agents' type spaces.

## 1 Introduction

Motivated by Internet applications the methodologies of *mechanism design* have been adopted to consider design of algorithms and protocols for selfish agents. A central problem in this area is in merging computational constraints (from approximation algorithms) with incentive constraints (from mechanism design). Much of the recent literature on this problem has focused on mechanisms satisfying the strongest possible incentive constraints and the strongest possible notions of tractability. Namely, the focus has been on ex post incentive compatibility (i.e., truth-telling in dominant strategies) and worst-case approximation algorithms. Positive results in this area are in the form of new algorithms for paradigmatic problems that satisfy incentive constraints and match the worst-case performance bounds of the algorithmic design problem sans incentive constraints. For ex post incentive compatibility, no general reductions are known.

In this paper we consider a relaxation of the incentive constraints to *Bayesian incentive compatibility* (BIC) where truth-telling is a Bayes-Nash equilibrium, i.e., where strategies are a mutual best response to strategies of other agents when agent preferences are drawn from a known distribution. We also con-

sider a relaxation of the performance constraints of algorithms; we allow algorithms not to be worst-case approximations. In this context we give a very general reduction from Bayesian mechanism design to algorithm design that approximately preserves or improves the expected performance, i.e., social welfare, of the algorithm. Of course, our reduction can also be applied to a worst-case  $\beta$ -approximation algorithm; in such a setting it results in a mechanism that is a Bayesian  $\beta(1 + \epsilon)$ -approximation.

This approach to mechanism design sits well with a standard paradigm for algorithm design wherein a practitioner might fine-tune their algorithm to the actual workload they face rather than optimize for the worst case. Furthermore, in the Internet applications that motivate this field, protocols are executed at a large scale and distributions are likely to be learnable.

This paper is a direct followup to work of Hartline and Lucier [8] that gives a such a Bayesian reduction for the special case where the selfish agents' preferences are single-dimensional, e.g., a value for receiving service. While this work has elegant economic motivation, its sampling-based blackbox reduction requires much extra machinery to achieve Bayesian incentive compatibility; furthermore, it has no clear generalization of its ironing-based approach to more the general (and fundamental) multi-dimensional setting, e.g., where an agent might have distinct values for several services that are available. In contrast we consider multi-dimensional settings and give a very simple reduction. Essentially: incentive constraints in BIC mechanism design are as simple as solving a bipartite matching problem independently on the type space of each agent. In settings with small type spaces but many agents, this gives a general polynomial time reduction from mechanism design to algorithm design.

The main approach is similar to the one in [8]. Our reduction will accept reported types from agents. It will transform each of these types to a distribution over types with the properties: (a) the transformation applied to a random type from the distribution results in the transformed type with the same distribution, (b) the transformation weakly improves performance, and (c) the algorithm composed with the transformation is Bayesian incentive compatible. It

---

\*Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL. Email: [hartline@eecs.northwestern.edu](mailto:hartline@eecs.northwestern.edu). Supported in part by NSF Grant CCF-0830773 and NSF Career Award CCF-0846113.

<sup>†</sup>Department of Computer Science, Cornell University, Ithaca, NY. Email: [rdk@cs.cornell.edu](mailto:rdk@cs.cornell.edu). Supported by NSF grants CCF-0643934 and AF-0910940, an Alfred P. Sloan Foundation Fellowship, and a Microsoft Research New Faculty Fellowship.

<sup>‡</sup>Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL. Email: [a-malekian@northwestern.edu](mailto:a-malekian@northwestern.edu).

then inputs the transformed types into the original algorithm and returns its output. Finally, it computes appropriate payments.

The transformation we apply to each agent’s type is based on computing the maximum weighted matching in a bipartite graph between the random types from the distribution (including the agent’s real type), termed “replicas”, and the set of outcomes of the algorithm on another equal-sized random set of types drawn from the distribution, termed “surrogates”. The transformation then outputs the surrogate type to which the agent’s real type is matched. This basic *replica-surrogate-matching* approach can be further refined and improved for the special cases of single-dimensional agents and discrete explicitly specified type spaces. (The latter reduction was independently and concurrently discovered by Bei and Huang [2].)

To solve the above maximum weighted matching problem we need to be able to evaluate the value of each replica for the outcome the algorithm obtains for each of the surrogates. If a replica is matched to a surrogate the value it obtains is the expected value over outcomes the algorithm produces when run on the surrogate and random types of the other agents. We consider two computational models: an *ideal* model (Section 3) where we are able to exactly calculate the expected value an agent might have for the distribution over outcomes produced by the algorithm, and a *blackbox* model (Section 4) where we can only estimate this value by repeatedly sampling the types of the other agents and running the algorithm. Naturally, the blackbox model is more realistic, though it presents some serious challenges.

**Results.** In the ideal model the replica-surrogate matching and variants are BIC and we give bounds on the number of replicas and surrogates needed to ensure that the surplus loss of the resulting mechanism is at most a small additive  $\varepsilon$ . Naturally, these bounds degrade with a suitable notion of the complexity of the type space. In the blackbox model our results for single-dimensional settings can be extended to give BIC mechanisms. For discrete explicitly specified type spaces we can correct errors inherent in sampling to give a BIC reduction in pseudo-polynomial time. For continuous type spaces our reduction is not BIC but it is  $\varepsilon$ -BIC.

**Related Work.** There has been extensive work on designing ex post incentive compatible and tractable mechanisms that provide worst-case approximation guarantees. The paradigmatic problem of single-minded combinatorial auctions was solved by Lehmann et al. [10] and related machine scheduling (to minimize makespan) was solved by Dhang-

watnotai et al. [5], where “solved” means that the approximation factor of the mechanism matches the algorithmic lower bound. There is also a large literature on multi-dimensional combinatorial auctions and approximation that we do not cite exhaustively.

There have been a few reductions from ex post incentive compatible mechanism design to algorithm design for special classes of algorithms. Of course the VCG mechanism reduces the IC mechanism design problem to exact algorithm design [13, 4, 7]. Lavi and Swamy [9] consider IC mechanisms for multi-parameter packing problems and give a technique for constructing a (randomized)  $\beta$ -approximation mechanism from any  $\beta$ -approximation algorithm that verifies an integrality gap. For polynomial time approximation schemes (PTAS), Briest et al. [3] solve the single-dimensional case and Dugmhi and Roughgarden [6] solve the case of multi-dimensional additive valuations in downward-closed settings; both papers can be viewed as blackbox reductions from the ex post IC mechanism design problem to the PTAS algorithm design problem.

Finally, for Bayesian incentive compatibility, Hartline and Lucier [8] give a blackbox reduction from BIC mechanism design to algorithm design in single-dimensional settings. This reduction converts any algorithm into a mechanism and approximately preserves its expected (for the Bayesian prior) performance. Our approach follows their methodology closely. We improve on their results for single-dimensional settings and extend them to multi-dimensional settings.

In concurrent and independent work Bei and Huang [2] reduce  $\varepsilon$ -BIC mechanism design to algorithm design in settings with discrete and explicitly given type spaces. In contrast, our work for discrete settings uses the same basic approach but produces a BIC mechanism, albeit in pseudo-polynomial time. Importantly, Bei and Huang give a nice application of the approach to the paradigmatic problem of combinatorial auctions with subadditive bidders.

## 2 Preliminaries

We consider mechanisms for selfish agents. An agent has a private *type*  $t$  from *type space*  $\mathcal{T}$ . There are  $n$  agents. When we wish to be specific about a particular agent  $\kappa \in [n]$  we index as, e.g.,  $t^\kappa$  and  $\mathcal{T}^\kappa$ . The *type profile* of the  $n$  agents is denoted by  $\mathbf{t} = (t^1, \dots, t^n) \in \mathcal{T}^1 \times \dots \times \mathcal{T}^n$ . An algorithm  $\mathcal{A}$  maps a type profile  $\mathbf{t}$  to an *outcome*  $x$  from *outcome space*  $\mathcal{X}$ . Agent  $\kappa$  with type  $t^\kappa$  has *valuation*  $v(t^\kappa, x)$  for outcome  $x$ . We assume that agent values are non-negative and come from a bounded range that, without loss of generality, is  $[0, 1]$ . A mechanism

$\mathcal{M} = (\mathcal{A}, \mathbf{p})$  maps the type profile into an outcome via an algorithm  $\mathcal{A}$  and into a *payment profile*  $\mathbf{p} = (p^1, \dots, p^n)$ . We overload notation and refer to  $p^\kappa$  as both the payment for  $\kappa$  and, when invoked on a type profile, as the mapping from type profile to payment profile, e.g., as  $p^\kappa(\mathbf{t})$ . Agent  $\kappa$  with type  $t^\kappa$  has *utility*  $v(t^\kappa, x) - p^\kappa$  for outcome  $x$  at payment  $p^\kappa$ . Agents are *quasi-linear* utility maximizers, i.e., they aim to maximize their expected utility.

We are looking for mechanisms for maximizing the *social surplus*, i.e., the sum of the values of the agents less any cost to the designer. Denote by  $c(x)$  the designer's cost for outcome  $x$ . The optimal social surplus is  $\text{OPT}(\mathbf{t}) = \sup_{x \in \mathcal{X}} \sum_i v(t^\kappa, x) - c(x)$ . We will overload notation and use  $\mathcal{A}(\mathbf{t})$  to refer to both the outcome and its surplus; i.e., an algorithm  $\mathcal{A}$  with outcome  $x = \mathcal{A}(\mathbf{t})$  has social surplus  $\mathcal{A}(\mathbf{t}) = \sum_i v(t^\kappa, x) - c(x)$ .

In Bayesian algorithm design and mechanism design the types are drawn from a distribution.  $F^\kappa$  denotes the distribution for  $t^\kappa \in \mathcal{T}^\kappa$  and  $\mathbf{F} = F^1 \times \dots \times F^n$  the joint (product) distribution. When reporting on the expected surplus of the algorithm we will often take the distribution as implicit and denote the expected surplus of an algorithm on types from the distribution as  $\mathcal{A} = \mathbf{E}_{\mathbf{t} \sim \mathbf{F}}[\mathcal{A}(\mathbf{t})]$ . We will denote by  $\mathcal{A}^\kappa = \mathbf{E}_{\mathbf{t} \sim \mathbf{F}}[v(t^\kappa, \mathcal{A}(\mathbf{t}))]$  the expected surplus contributed by agent  $\kappa$ .

From agent  $\kappa$ 's perspective, the algorithm and payment rule are composed with the other agent types which are drawn from the joint distribution  $\mathbf{F}^{-\kappa}$ . We will denote by  $\mathcal{A}(t^\kappa)$  the distribution over outcomes when  $\kappa$ 's type is  $t^\kappa$  and the other agents' types are drawn from the distribution. Agent  $\kappa$ 's value for the outcome distribution  $\mathcal{A}(t^\kappa)$  is denoted  $v(t^\kappa, \mathcal{A}(t^\kappa)) = \mathbf{E}_{\mathbf{t} \sim \mathbf{F}}[v(t^\kappa, \mathcal{A}(\mathbf{t})) \mid t^\kappa]$ . We will denote by  $p^\kappa(t^\kappa) = \mathbf{E}_{\mathbf{t} \sim \mathbf{F}}[p^\kappa(\mathbf{t}) \mid t^\kappa]$  the expected payment of agent  $\kappa$  with type  $t^\kappa$  when the other agents' types are drawn from the distribution.

A mechanism is *Bayesian incentive compatible* (BIC) if for all agents  $\kappa$ , and all types  $t^\kappa, \tau \in \mathcal{T}^\kappa$ , when all agents except  $\kappa$  truthfully report their types then  $\kappa$ 's expected utility is maximized by truthful reporting as well, i.e.,

$$v(t^\kappa, \mathcal{A}(t^\kappa)) - p^\kappa(t^\kappa) \geq v(t^\kappa, \mathcal{A}(\tau)) - p^\kappa(\tau).$$

For instance, the well known Vickrey-Clarke-Groves (VCG) mechanism [13, 4, 7] selects the outcome that maximizes the social surplus, i.e.,  $\text{OPT}(\mathbf{t})$ , and assigns each agent  $\kappa$  a payment equal to the *externality that  $\kappa$  imposes on the other agents*, i.e.,  $p^\kappa = \text{OPT}(\mathbf{t}^{-\kappa}) - \text{OPT}^{-\kappa}(\mathbf{t})$  where the first term is the surplus of the optimal outcome that excludes agent  $\kappa$  and the second term is the surplus of the

optimal outcome with agent  $\kappa$  but does not include  $\kappa$ 's value in the sum.

Unlike (ex post) IC mechanism design where the mechanism, for all profiles of bids, must have a truthful reporting best response, for BIC mechanisms all that is relevant is the outcome behavior of the composition of the mechanism with the distribution of values of the other agents. The following theorem rephrases the well-known characterization of Bayesian implementable allocation rules via cyclic monotonicity [12].

**THEOREM 2.1.** *For an algorithm  $\mathcal{A}$ , there exists  $\mathbf{p}$  such that  $\mathcal{M} = (\mathcal{A}, \mathbf{p})$  is BIC if and only if a maximum weighted matching property is satisfied on an induced graph on any finite subset of the agent types: in the weighted bipartite graph given by*

- *left-hand-side vertices: any finite subset of the agent's types  $\mathcal{T}' \subset \mathcal{T}$ ,*
- *right-hand-side vertices: the corresponding multiset of outcome distributions*

$$\mathcal{X}' = \{X = \mathcal{A}(t) \mid t \in \mathcal{T}'\}, \text{ and}$$

- *edge weights:  $v(t, X)$  between every  $t \in \mathcal{T}'$  and  $X \in \mathcal{X}'$ ,*

*the maximum weighted matching is the identity, i.e.,  $t$  matched to  $X = \mathcal{A}(t)$ .*

From this perspective the various types of an agent are competing amongst themselves for potential outcomes of the algorithm. For the algorithm to be BIC, it must resolve this competition to maximize expected social surplus. For finite type spaces, appropriate payments can be calculated from the maximum weighted matching on the full typespace  $\mathcal{T}' = \mathcal{T}$  as they would be by VCG.

We will be giving reductions from BIC mechanism design to algorithm design. In other words, we will take an algorithm  $\mathcal{A}$  and convert it into a mechanism  $\mathcal{M}$  that is BIC. Importantly, while the algorithm  $\mathcal{A}$  may not satisfy the maximum weighted matching property, the mechanism  $\mathcal{M}$  will. While our goal is to perform this transformation and preserve the expected performance of the algorithm, we will only achieve this goal approximately. We refer to  $\varepsilon$  as the *additive loss* of our reduction when  $\mathcal{M}$ 's performance satisfies  $\mathcal{M} \geq \mathcal{A} - \varepsilon$ . Our reductions will allow for arbitrarily small additive loss in time polynomial in  $1/\varepsilon$ ;  $n$ , the number of agents; and a notion of the "size" of the type space.

A further distinction that is relevant to our results concerns the structure of the agents' type

spaces. When we refer to *single-dimensional types*, we mean that  $\mathcal{X} \subseteq [0, 1]^n$ ,  $\mathcal{T}^\kappa = [0, 1]$  for all  $\kappa$ , and  $v(t^\kappa, x) = t^\kappa x^\kappa$ . This is the standard model of single-dimensional mechanism design, normalized so that types and allocations take  $[0, 1]$  values. When we refer to the setting of *discrete type spaces*, we mean that each type space  $\mathcal{T}^\kappa$  has finite cardinality, the algorithm is given a complete description of the type distribution of agent  $\kappa$ , in the form of a listing of all the points of  $\mathcal{T}^\kappa$  and their associated probabilities, and the algorithm’s running time is allowed to be polynomial in this input representation. Finally, the setting of *multi-dimensional continuous types* is the fully general setting defined at the start of this section.

We consider two computational models: an *ideal model* that makes strong (and often unrealistic) assumptions about our ability to exactly compute expected values, and a *blackbox model* in which our only access to the algorithm  $\mathcal{A}$  and distribution  $\mathbf{F}$  is by sampling and blackbox evaluation, i.e., we can draw a random type profile  $\mathbf{t}$  from  $\mathbf{F}$ , sample outcome  $x \sim \mathcal{A}(\mathbf{t})$ , and evaluate the value  $v(t, x)$  that an agent with a given type  $t$  derives from a specific outcome  $x$ . We present our reduction for the conceptually simpler ideal model on Section 3 and explain how to extend most of these results to the blackbox model in Section 4.

### 3 Ideal Model

In this section we make the simplifying, but unrealistic, assumption that we have access to a closed-form formula for calculating the expected value that an agent derives from the outcome of the algorithm on a distribution of agent types. That is, we can directly calculate  $v(t^\kappa, \mathcal{A}(t^\kappa)) = \mathbf{E}_{\mathbf{t}}[v(t^\kappa, \mathcal{A}(\mathbf{t})) \mid t^\kappa]$ .

At the heart of our reduction is a *surrogate selection rule*,  $\Gamma$ , that, for a given agent type, selects a distribution over types (a.k.a., *surrogate types*) to be input into the algorithm in place of the agent’s real type. We will typically specify a surrogate selection rule as a randomized algorithm that outputs a surrogate type. The surrogate selection rule is then employed in the following generic reduction that converts an algorithm into a mechanism (essentially from [8]).

DEFINITION 3.1. *The  $\Gamma$  reduction is:*

1. *Apply the surrogate selection rule  $\Gamma$  independently to each agent’s type to get a surrogate type profile.*
2. *Call  $\mathcal{A}$  on the profile of surrogate types.*

3. *Output the outcome of  $\mathcal{A}$  and appropriate payments.*

As in [8] this generic reduction will result in a good mechanism if the surrogate selection rule satisfies the following three properties.

1. *Distribution preservation:* When applying the surrogate selection rule to a random type from the distribution, the resulting distribution of the surrogate type is identical to the original distribution of the type.
2. *(Approximate) surplus preservation:* The expected valuation of an agent in the algorithm approximately improves by the mapping.
3. *Bayesian maximum-weight matching property:* The composition of the algorithm (with other agent types drawn from the distribution) with the surrogate selection rule satisfies the maximum-weight matching property.

Notice that the distribution preservation condition will imply that we can perform the surrogate selection rule independently to each agent and no agent notices that the other agents’ types have been transformed. The Bayesian maximum-weight matching property, then, implies that the reduction is BIC. We will not tend to take this approach; instead, we will argue that our reductions are BIC because of distribution preservation and by analogy to VCG.

It remains to describe a surrogate selection rule that satisfies the above conditions and discuss payment computation. These operations pertain to each agent independently; therefore, we proceed by considering a single agent and assume the algorithm is hard-wired with the distribution of types from other agents. For convenience, since we consider only a single agent, we drop the index  $\kappa$  from all notations.

DEFINITION 3.2. *The Replica-Surrogate Matching (RSM) surrogate selection rule is implemented by the following randomized algorithm, given agent’s type  $t$  and an integer parameter  $m$ :*

1. *Pick  $i^*$  uniformly at random from  $[m]$ .*
2. *Sample the replica type profile,  $\mathbf{r}$ , an  $m$ -tuple of types with  $r_{i^*} = t$ , the real agent’s type, and with each remaining replica type in  $\mathbf{r}_{-i^*}$  drawn independently and identically from type distribution  $F$ .*
3. *Sample the surrogate type profile,  $\mathbf{s}$ , an  $m$ -tuple of types with each surrogate type independent and identically drawn from type distribution  $F$ .*

4. Define the surrogate outcome profile as  $\mathbf{X}$  with  $X_j$  as the distribution of outcomes from  $\mathcal{A}(s_j)$ .
5. Define an edge-weighted bipartite graph between replicas (left-hand side) and surrogate outcomes (right-hand side) with the weight of the edge between replica  $i$  and surrogate outcome  $j$  equal to  $v(r_i, X_j)$ .
6. Simulate the VCG mechanism to select a matching between replicas (as agents) and surrogate outcomes (as items), i.e., compute the maximum weighted matching and appropriate payments.
7. Select the surrogate to whose outcome the real agent's type,  $i^*$ , is matched.

The appropriate payment for the replica-surrogate matching reduction is the VCG payment for the real agent's type calculated in Step 6.

We argue that this reduction is distribution preserving and Bayesian incentive compatible.

LEMMA 3.3. *In the ideal model, RSM is distribution preserving.*

*Proof.* Each surrogate is i.i.d. from  $F$ . Each surrogate is matched to some replica. Using the principle of deferred decisions, we now pick the replica  $i^*$  that corresponds to the real type. Since this choice of replica is uniform from  $[m]$ , then the selection of surrogate types is uniform from  $[m]$ , and therefore the distribution of the selected surrogate type is  $F$ .  $\square$

THEOREM 3.1. *In the ideal model, RSM is Bayesian incentive compatible.*

*Proof.* Consider an agent. VCG on the constructed bipartite matching problem is incentive compatible for any instantiation of  $\mathbf{r}$  and  $\mathbf{s}$  under the assumption that the expected value that any type  $r_i$  derives from being matched to surrogate outcome  $X_j$  is exactly  $v(r_i, X_j)$ . This property holds in the Bayesian sense, i.e., when the surrogate types of other agents are drawn from their respective distributions, as happens when the other agents bid truthfully. (Here we are applying the distribution-preservation property, Lemma 3.3.) Therefore, conditioning on the replica and surrogate type profiles,  $\mathbf{r}$  and  $\mathbf{s}$ , the mechanism is Bayesian incentive compatible. The theorem now follows by averaging over the random choice of  $\mathbf{r}$  and  $\mathbf{s}$ .  $\square$

We now turn to evaluating the social surplus. The RSM reduction is based on a weighted bipartite matching between types (replicas) and outcomes

(surrogate outcomes) where the weights are given by the agent's valuation function. Central to our proof will be a weighted bipartite matching between types (replicas) and types (surrogates) where costs on edges are given by how different the types are. Specifically, the costs are the maximum, over outcomes, of the magnitude of the difference in the value of the two types.

The following steps give the high-level proof of approximate surplus preservation.

1. The surplus of the mechanism is a  $1/m$  fraction of the total weight of the maximum replica-to-surrogate-outcome matching.
2. This maximum replica-to-surrogate-outcome matching is certainly more than the maximum value surrogate-to-surrogate-outcome matching less the minimum cost replica-to-surrogate matching.
3. The surplus of the original algorithm is a  $1/m$  fraction of the total weight of the identity surrogate-to-surrogate-outcome matching which is certainly at most the maximum surrogate-to-surrogate-outcome matching.
4. Therefore, the reduction approximately preserves surplus if the minimum cost replica-to-surrogate matching is small.

Below we give formal statements and proofs of each of these steps. By far the most technical result is that of showing that the cost of the optimal replica-to-surrogate matching is small.

LEMMA 3.4. *The expected valuation of the agent in  $\mathcal{M}$  from the RSM reduction is a  $1/m$  fraction of the maximum weighted matching between the replicas and the surrogate outcomes.*

*Proof.* The valuation of the agent is equal to the weight of the edge matched to it. Since the index of the agent  $i^*$  is uniform from  $[m]$ , this weight is a uniform draw from the  $m$  edges in the matching.  $\square$

LEMMA 3.5. *The expected valuation of the agent in  $\mathcal{A}$ , to which the RSM reduction is applied, is at most a  $1/m$  fraction of the maximum weighted matching between the surrogates and surrogate outcomes.*

*Proof.* Consider the weighted bipartite graph given by the surrogates and surrogate outcomes, i.e., the weights are the valuations of each surrogate for each outcome. Consider the matching where surrogate  $\ell$  is matched to surrogate outcome  $\ell$ , i.e., the matching that the algorithm  $\mathcal{A}$  produces. The expected weight

of each edge in this matching is equal to the expected surplus of  $\mathcal{A}$  so the expected weight of the entire matching is  $m$  times  $\mathcal{A}$ 's expected surplus. Of course the weight of the maximum matching is at least as great as this matching.  $\square$

We now get a bound on how far the replicas are from the surrogates. To do this we first formally define a distance metric on the agent's type space. The metric is given by distance function:

$$d(t_1, t_2) = \sup_{x \in \mathcal{X}} |v(t_1, x) - v(t_2, x)|.$$

We second define the notion of a covering of the type space. This is a common notion in machine learning theory.

**DEFINITION 3.6.** *Type subspace  $\mathcal{T}' \subset \mathcal{T}$  is an  $r$ -cover of the type space  $\mathcal{T}$  if every  $t \in \mathcal{T}$  is within distance  $r$  from some type  $t' \in \mathcal{T}'$ .*

Now we can express the cost from matching replicas to surrogates in terms of any  $r$  and the minimum cardinality  $r$ -cover.

**LEMMA 3.7.** *For any  $r > 0$  and any  $r$ -cover  $\mathcal{T}'$  with cardinality  $\beta = |\mathcal{T}'|$ , the expected cost of the minimum weight matching between replicas and surrogates (with weights equal to the distance function) is  $O(2mr + \sqrt{\beta m})$ .*

*Proof.* We are considering a stochastic matching problem where  $m$  red points (e.g., replicas) and  $m$  blue points (e.g., surrogates) in a metric space (as given by  $d(\cdot, \cdot)$ ) are being drawn i.i.d. from a distribution. The space has a  $r$ -cover  $\mathcal{T}'$  with cardinality  $\beta$ . We consider the heuristic matching that prefers matching points with the same covering. For such points the cost of the edge in the matching is at most  $2r$ ; for the remaining points the cost is at most one, by our assumption that values are in  $[0, 1]$ . Let  $D$  denote the number of matches of this latter type. The cost of this matching is at most  $2rm + D$ . Of course this upper bounds the minimum cost matching. It remains for us to prove that  $D$  satisfies  $\mathbf{E}[D] = O(\sqrt{\beta m})$ .

Define  $\mathbf{c} : \mathcal{T} \rightarrow \{0, 1\}^\beta$  to indicate a point in the cover for every type and let  $\mathbf{c}(\mathbf{t}) = \sum_{i=1}^m \mathbf{c}(t_i)$  for any vector of types  $\mathbf{t}$ . Notice that  $c_{t'}(\mathbf{r})$  is the number of replicas (red) covered by  $t'$  and  $c_{t'}(\mathbf{s})$  is the number of such surrogates (blue),  $\min\{c_{t'}(\mathbf{r}), c_{t'}(\mathbf{s})\}$  is the number that can be matched to each other, and the remaining  $|c_{t'}(\mathbf{r}) - c_{t'}(\mathbf{s})|$  contribute to  $D$ . The total number of points that must be matched outside their cover is thus given by the  $L_1$ -norm,  $\|\mathbf{c}(\mathbf{r}) - \mathbf{c}(\mathbf{s})\|_1$ . Of course the number of such matches is half the number

of points matched, i.e.,

$$(3.1) \quad 2 \cdot D = \left\| \sum_{i=1}^m \mathbf{c}(r_i) - \mathbf{c}(s_i) \right\|_1.$$

The terms in the sum on the right side of (3.1) are independent random vectors. Each of them has mean zero (since  $r_k$  and  $s_k$  are sampled from the same distribution) and consequently,

$$\mathbf{E} [\|\mathbf{c}(\mathbf{r}) - \mathbf{c}(\mathbf{s})\|_2^2] = m \cdot \mathbf{E} [\|\mathbf{c}(r_1) - \mathbf{c}(s_1)\|_2^2] < 2m.$$

Finally,

$$\begin{aligned} \mathbf{E}[D] &= \frac{1}{2} \mathbf{E} [\|\mathbf{c}(\mathbf{r}) - \mathbf{c}(\mathbf{s})\|_1] \\ &\leq \frac{1}{2} \sqrt{\mathbf{E} [\|\mathbf{c}(\mathbf{r}) - \mathbf{c}(\mathbf{s})\|_1^2]} \leq \sqrt{\beta m}, \end{aligned}$$

using the fact that  $\|\mathbf{z}\|_1^2 \leq \beta \|\mathbf{z}\|_2^2$  for any vector  $\mathbf{z} \in \mathbb{R}^\beta$ .  $\square$

Combining lemmas 3.4, 3.5, and 3.7 we conclude the following theorem.

**THEOREM 3.2.** *For any  $r > 0$ , any  $r$ -cover  $\mathcal{T}'$  with cardinality  $\beta = |\mathcal{T}'|$ , and any agent  $\kappa$ , the mechanism  $\mathcal{M}$  from the RSM reduction on algorithm  $\mathcal{A}$  has expected surplus satisfying*

$$\mathcal{M}^\kappa \geq \mathcal{A}^\kappa - O(r + \sqrt{\beta/m}).$$

**COROLLARY 3.1.** *For any  $\mathcal{T}$  with doubling dimension  $\Delta$ , any  $\varepsilon > 0$ , and any agent  $\kappa$ , the RSM reduction with  $m = \Omega(\varepsilon^{-\Delta-2})$  has an expected surplus satisfying*

$$\mathcal{M}^\kappa \geq \mathcal{A}^\kappa - O(\varepsilon).$$

*Proof.* The assumption that  $\mathcal{T}$  has doubling dimension  $\Delta$  implies that there is an  $r$ -cover with cardinality  $\beta = O(r^{-\Delta})$ . The corollary now follows by substituting  $r = \varepsilon$  in Theorem 3.2.  $\square$

In fact, when  $\mathcal{T}$  has doubling dimension  $\Delta > 2$ , one can replace  $\varepsilon^{-\Delta-2}$  with  $\varepsilon^{-\Delta-1}$  in Corollary 3.1 using a more delicate analysis: one partitions  $\mathcal{T}$  into sets of radius  $1/2, 1/4, \dots$ , each partition refining the preceding one, and one constructs the matching recursively based on this hierarchical partition. The details are left to the full version of this paper.

**3.1 Single-dimensional Settings.** Let us now specialize to single-dimensional settings where  $\mathcal{X} \subseteq [0, 1]^n$ ,  $\mathcal{T}^\kappa = [0, 1]$ , and  $v(t^\kappa, x) = t^\kappa x^\kappa$ . Again, we will take the perspective of a single agent and drop the agent index  $\kappa$ . What is special about single-dimensional settings is that a maximum matching

between replicas and surrogate outcomes can be calculated simply by sorting the replica types and the surrogate outcomes and matching in this sorted order.

Notice that for single-dimensional settings Theorem 3.2 with  $r = m^{-1/3}$  implies that the RSM reduction has loss  $O(m^{-1/3})$  per agent. A sharper analysis establishes a bound of  $O(m^{-1/2})$  in this setting; for details see the full version of this paper.

In this section we improve on these bounds by giving a single-dimensional variant of RSM with loss  $\Theta(m^{-1})$ . Informally this variant defines the surrogates to be equal to the replicas, except with the real type  $r_{i^*}$  replaced by a new type drawn from the distribution. This approach gives the best known reduction for single-dimensional settings.

**DEFINITION 3.8.** *The Replica-Replica Matching (RRM) surrogate selection rule is equivalent to RSM with  $j^*$  drawn uniformly from  $[m]$  and Steps 3 and 5 replaced with:*

- 3'. Define surrogates equal to replicas, i.e.,  $\mathbf{s} = \mathbf{r}$ .
- 5'. Define a weighted bipartite graph between replicas  $\mathbf{r}_{-j^*}$  (left-hand side) and surrogate outcomes  $\mathbf{X}_{-i^*}$  (right-hand side).

We now argue that in single-dimensional settings RRM is distribution preserving and approximately surplus preserving. Importantly, neither of these results for RRM generalize to multi-dimensional settings.

**LEMMA 3.9.** *The RRM surrogate selection rule is distribution preserving.*

*Proof.* Sort the replicas  $\mathbf{r}$  in decreasing order of value and the surrogate outcomes  $\mathbf{X}$  in decreasing order of expected value. Since we are in the single-dimensional setting, we know that the maximum weight matching selected in Step 6 of the reduction is the unique order-preserving matching between the subsets  $\mathbf{r}_{-j^*}$  and  $\mathbf{X}_{-i^*}$  of these two ordered lists. As in the proof of Lemma 3.3, we must show that the surrogate outcome to which  $i^*$  is matched is uniformly distributed among the  $m - 1$  elements of  $\mathbf{X}_{-i^*}$ . For the surrogate outcome occurring  $k^{\text{th}}$  in the sorted list, it is matched to  $i^*$  if and only if either: **(a)**  $i^*$  occurs  $k^{\text{th}}$  in the list of replicas and  $j^*$  occurs after it, or **(b)**  $i^*$  occurs  $(k+1)^{\text{th}}$  in the list of replicas and  $j^*$  occurs before it. The two events are mutually exclusive. The probability of the first one is  $\frac{1}{m} \cdot \frac{m-k}{m-1}$ , and the probability of the second one is  $\frac{1}{m} \cdot \frac{k}{m-1}$ , so their combined probability is exactly  $\frac{1}{m-1}$  as claimed.  $\square$

The proof of Theorem 3.1 can be applied as is to RRM and, together with Lemma 3.9, implies that it is BIC.

**THEOREM 3.3.** *For any agent  $\kappa$ , the mechanism  $\mathcal{M}$  from the RRM reduction on algorithm  $\mathcal{A}$  has expected surplus satisfying*

$$\mathcal{M}^\kappa \geq \mathcal{A}^\kappa - 1/m.$$

*Proof.* Sort the replicas  $\mathbf{r}$  in decreasing order of value and the surrogate outcomes  $\mathbf{X}$  in decreasing order of expected value. Notate the  $k^{\text{th}}$  ranked in each ordering as  $r_{(k)}$  and  $X_{(k)}$ , respectively. The maximum weighted matching between  $\mathbf{r}$  and  $\mathbf{X}$  would assign replica to surrogate outcome by this rank order.

Compared to the above matching, the matching between  $\mathbf{r}_{-j^*}$  and  $\mathbf{X}_{-i^*}$ , at worst, matches the  $k^{\text{th}}$  highest replica to the  $(k+1)^{\text{th}}$  highest surrogate outcome. Since the real agent, corresponding to replica  $i^*$ , is uniformly at random from these  $m$  replicas, the expected surplus of the real agent is at least (for convenience defining  $X_{(m+1)} = 0$ ),

$$\begin{aligned} \mathcal{M}^\kappa &\geq \frac{1}{m} \sum_{k=1}^m r_{(k)} X_{(k+1)} \\ &\geq \frac{1}{m} \sum_{k=2}^m r_{(k)} X_{(k)} \\ &\geq \frac{1}{m} \left[ \sum_{k=1}^m r_{(k)} X_{(k)} - 1 \right] \\ &= \mathcal{A}^\kappa - \frac{1}{m}. \end{aligned}$$

$\square$

**3.2 Discrete Type Spaces.** In the ideal model, settings with discrete type spaces can be solved with no loss by a reduction with running time that is polynomial in the combined cardinality of the type spaces. The mechanism is a variant on the RSM mechanism, though we will not make the connection precise. This reduction is based on network flow, and for convenience we will adopt the terminology of replica and surrogate even though neither the replicas nor the surrogates are random draws from the distribution.

**DEFINITION 3.10.** *The Replica-Surrogate Flow (RSF) surrogate selection rule for type space with cardinality  $m$  is:*

1. For each distinct type  $t \in [m]$ , let replica  $r_t = t$ , surrogate  $s_t = t$ , and surrogate outcome,  $X_t = \mathcal{A}(s_t)$ .
2. Define a minimum cost network flow instance on the replicas and surrogate outcomes:

- Connect the source to each replica  $r_t$  with capacity  $\pi_t$  (equal to the probability that  $t$  is drawn from  $F$ ).
  - Connect the sink to each surrogate outcome  $X_t$  with capacity  $\pi_t$  (equal to the probability that  $t$  is drawn from  $F$ ).
  - Connect all replicas  $i$  to surrogates outcomes  $j$  with capacity 1 and cost  $-v(r_i, X_j)$ .
  - Let  $f(i, j)$  denote the flow from replica  $i$  to surrogate  $j$  in the minimum cost flow.
3. Run the VCG mechanism to find the minimum cost flow and appropriate payments.
  4. Replace the real type  $t$  with a convex combination of the surrogates as suggested by the flow, i.e., surrogate type  $s_j$  with probability  $f(t, j)/\pi_t$ .

The appropriate payment for the replica-surrogate flow reduction is the VCG payment for the real agent as calculated in Step 3.

The proofs of Lemma 3.3 and Theorem 3.1 can be adapted to RSF and imply that it is BIC. Furthermore, the expected surplus of the resulting mechanism is equal to the magnitude of the minimum cost flow which is at least the magnitude of the cost of the identity flow, i.e., with  $f(i, i) = \pi_i$ , which is the expected surplus of the original algorithm.

**THEOREM 3.4.** *In the ideal model in discrete settings with finite type spaces, the mechanism  $\mathcal{M}$  from the RSF reduction on algorithm  $\mathcal{A}$  has expected surplus satisfying  $\mathcal{M} \geq \mathcal{A}$  and runs in polynomial time in  $m$ , the number of distinct types, and  $n$ , the number of agents.*

#### 4 The blackbox model

In this section we explain how to modify RSM reductions to achieve incentive compatibility in the blackbox model, in which we can only access the algorithm  $\mathcal{A}$  by evaluating it at a given type profile. In particular, we cannot directly evaluate the expectation of  $v(t^\kappa, \mathcal{A}(t^\kappa, \mathbf{t}^{-\kappa}))$  over a random type profile  $\mathbf{t}^{-\kappa}$ , as is allowed in the ideal model. Instead, such quantities must be estimated by sampling. The main difficulty, then, lies in designing a mechanism that satisfies Bayesian incentive compatibility despite the inevitability of small sampling error.

As before, we focus on a single agent  $\kappa$  and omit its index;  $\mathcal{A}(t)$  denotes the distribution of outcomes obtained by evaluating the algorithm  $\mathcal{A}$  on the agent with type  $t$  assuming all other agent types are random from the distribution.

Consider the following definitions.

**DEFINITION 4.1.** *For a given positive integer  $L$ , the estimation procedure Est for outcome distribution  $X$  is the following. Draw  $L$  outcomes  $x_1, \dots, x_L$  independently and identically from  $X$ . The outcome estimate  $\tilde{X} = \text{Est}(X)$  is the uniform distribution over these  $L$  outcomes. The value estimate for type  $t$  and outcome estimate  $\tilde{X}$  is  $v(t, \tilde{X}) = \frac{1}{L} \sum_{j=1}^L v(t, x_j)$ .*

**DEFINITION 4.2.** *The Estimated Replica-Surrogate Matching (ERSM), surrogate selection rule is identical to RSM except with outcome distribution  $X_j = \mathcal{A}(s_j)$  replaced with outcome estimate  $\tilde{X}_j = \text{Est}(\mathcal{A}(s_j))$  for all  $j \in [m]$ . Similarly, for discrete settings define ERSF for RSF using outcome estimates in place of outcome distributions.*

Unfortunately ERSM and ERSF are not generally BIC. We first show that in single-dimensional settings a variant of ERSM with (to be defined) *binomial estimates* is BIC. We then turn to discrete settings and show that the convex combination of ERSF with a (to be defined) blatantly monotone algorithm is BIC. Finally, for general continuous settings, as we know of no general BIC reduction, we show that ERSM itself is in fact  $\epsilon$ -BIC for a suitable choice of  $L$ .

**4.1 Single-dimensional types.** Consider single-dimensional settings, i.e., where  $\mathcal{X} \subseteq [0, 1]^n$ ,  $\mathcal{T}^\kappa = [0, 1]$ ,  $v(t^\kappa, x) = t^\kappa x^\kappa$ .

Observe that ERSM is not BIC via the following simple counterexample. Assume that the agent's type distribution is uniform on  $[0, 1]$ , that  $L = 1$ ,<sup>1</sup> and that the function  $\mathcal{A}(t)$  has the following structure. When  $t > 1/2$ ,  $\mathcal{A}(t) = 4/m$  with probability 1. When  $t \leq 1/2$ ,  $\mathcal{A}(t) = 1$  with probability  $2/m$  and  $\mathcal{A}(t) = 0$  otherwise. Note that since  $L = 1$ , each  $\tilde{X}_j$  is obtained by simply sampling a random type  $s_j \in [0, 1]$  and drawing one sample from the distribution  $\mathcal{A}(s_j)$ . We have that for all  $j$ ,  $\Pr(\tilde{X}_j = 1) = (1/2) \cdot (2/m) = 1/m$ , and hence  $\Pr(\max_{1 \leq j \leq m} \tilde{X}_j < 1) = (1 - 1/m)^m \sim 1/e$ . Accordingly, if an agent with type  $t = 1$  bids truthfully and is assigned to  $s_\ell$  where  $\ell = \text{argmax}_j \{\tilde{X}_j\}$ , then  $\Pr(s_\ell \geq 1/2) \approx 1/e$  and  $\mathbf{E}[v(t, \mathcal{A}(s_\ell))] \approx (1/e) \cdot (2/m) + (1 - 1/e) \cdot (4/m)$ . If the agent instead bids  $3/4$  and is matched to some other surrogate  $s_s$ , then  $s_s > 1/2$  unless the set  $\{s_1, \dots, s_m\}$  contains more than  $3m/4$  samples from  $[0, 1/2]$  or the multiset  $\{\tilde{X}_1, \dots, \tilde{X}_m\}$  contains more than  $m/4$  instances of the number 1. Both of these events have exponentially small probability (in  $m$ ), so

<sup>1</sup>Similar counterexamples can be constructed for any value of  $L$ , by modifying the parameters in this counterexample to have the appropriate dependence on  $L$ .



$\mathbf{E}[v(t, \mathcal{A}(s_s))] \approx 4/m$ . Thus we see that the agent's expected allocation is higher when bidding  $3/4$  than when bidding  $1$ , violating monotonicity.

The problem with ERSM is that the head-to-head comparison between random values from different distributions may not be consistent with the expected values of these distributions. To address this we explicitly turn each estimate distribution into the binomial distribution.

**DEFINITION 4.3.** *The binomial estimation procedure BinomEst for outcome distribution  $X$  is the following. Draw  $L$  outcomes  $x_1, \dots, x_L$  independently and identically from  $X$  (note:  $x_j \in [0, 1]$ ). Draw Bernoulli trials  $y_1, \dots, y_L$  independently, setting  $y_j = 1$  with probability  $x_j$  and  $0$  otherwise. The binomial outcome estimate  $\tilde{X} = \text{BinomEst}(X)$  is the average of these  $L$  binary outcomes, i.e.,  $\tilde{X} = \frac{1}{L} \sum_{j=1}^L y_j$ .*

Notice, in the definition of the binomial estimation procedure, that without conditioning on the  $x_j$ 's, the  $y_j$ 's are i.i.d. Bernoulli trials with success probability equal to the expectation of  $X$ . Therefore,  $\tilde{X} = \text{BinomEst}(X)$  is a binomial random variable.

**DEFINITION 4.4.** *The Binomial Estimated Replica-Surrogate Matching (BERSM) surrogate selection rule is identical to RSM except with outcome distribution  $X_j = \mathcal{A}(s_j)$  replaced with outcome estimate  $\tilde{X}_j = \text{BinomEst}(\mathcal{A}(s_j))$  for all  $j \in [m]$ . Similarly, define BERRM for RRM using binomial outcome estimates in place of outcome distributions.*

The surplus of the blackbox model BERSM (and BERRM) reduction is close that of the ideal model RSM reduction.

**LEMMA 4.5.** *The expected surplus of the BERSM mechanism  $\tilde{\mathcal{M}}$  relative to the RSM mechanism  $\mathcal{M}$  satisfies  $\tilde{\mathcal{M}}^\kappa \geq \mathcal{M}^\kappa - O(1/m)$  when surrogate outcomes are estimated with  $L > m^2 \ln(m^2)$  samples. The same bound applies to the expected surplus of the BERRM mechanism relative to the RRM mechanism.*

*Proof.* The Chernoff-Hoeffding bound implies that when  $L > 2\varepsilon^{-2} \ln(m/\varepsilon)$ , the event that there exists an estimate  $\tilde{X}_k$  that differs from the actual value  $y_k$  by more than  $\varepsilon$  has probability no greater than  $\varepsilon$ . When this event does not occur, the matching with the greatest estimated value has an actual value that differs by at most  $2\varepsilon m$  from the actual value of the maximum matching. Rescaling the matching weight by  $\frac{1}{m}$  in accordance with Lemma 3.4, we find that the RSM reduction's expected welfare loss in the blackbox model differs from its expected welfare loss in the ideal model by at most  $3\varepsilon$ .  $\square$

To prove that the reduction is BIC, we use the following property of binomial distributions that we call *posterior monotonicity*.

**LEMMA 4.6. (POSTERIOR MONOTONICITY)**  
*Suppose  $p, q$  are real numbers with  $0 \leq p \leq q \leq 1$ ,  $X, Y$  are random variables with binomial distributions  $B(n, p)$  and  $B(n, q)$ , respectively, and  $a, b$  are integers with  $0 \leq a \leq b$ , then*

$$\Pr(X = a) \Pr(Y = b) \geq \Pr(X = b) \Pr(Y = a).$$

*Proof.* The lemma is trivial if either  $p$  or  $q$  is equal to  $0$  or  $1$ . Consequently, let us assume henceforth that  $p(1-p)q(1-q) \neq 0$ . Then it is easy to see that

$$\begin{aligned} \frac{\Pr(X = a) \Pr(Y = b)}{\Pr(X = b) \Pr(Y = a)} &= \frac{p^a (1-p)^{n-a} q^b (1-q)^{n-b}}{p^b (1-p)^{n-b} q^a (1-q)^{n-a}} \\ &= \left( \frac{q(1-p)}{p(1-q)} \right)^{b-a} = \left( 1 + \frac{q-p}{p(1-q)} \right)^{b-a}, \end{aligned}$$

which is greater than or equal to  $1$  by our hypotheses that  $p \leq q, a \leq b$ .  $\square$

**THEOREM 4.1.** *In the blackbox model and single-dimensional settings, BERSM and BERRM are BIC.*

*Proof.* [Proof sketch; full proof is in Appendix A.] Let  $\mathcal{M}$  be the mechanism that results from applying the BERSM reduction to algorithm  $\mathcal{A}$ . Let  $t_1 < t_2$  be any pair of types for an agent. To show that  $\mathbf{E}[\mathcal{M}(t_1)] \leq \mathbf{E}[\mathcal{M}(t_2)]$ , we show that this inequality holds when we condition on all the relevant data about an execution of the BERSM reduction *except* the identities of the surrogates to whom  $t_1, t_2$  would be matched. (Thus, we condition on the types of all replicas and surrogates, the estimates  $\tilde{X}_j$  for all but two surrogates  $\alpha, \beta$ , and the unordered pair of values  $\{\tilde{X}_\alpha, \tilde{X}_\beta\}$ .) Posterior monotonicity (Lemma 4.6) implies that the probability of generating this data is maximized when the higher of the two estimates  $\tilde{X}_\alpha, \tilde{X}_\beta$  is generated by the surrogate who has the higher expected allocation. Using Bayes' Law, this in turn implies that the agent's expected allocation is maximized when matched to the surrogate with the higher estimate. The RSM matches the highest bidders to the surrogates with the highest estimates, implying monotonicity.  $\square$

Finally, we describe how to compute payments so as to satisfy BIC when using the RSM reduction combined with the binomial estimator. According to the well-known characterization of truthful single-parameter mechanisms [11, 1], a mechanism is Bayesian incentive compatible as long each agent  $\kappa$ 's

expected allocation  $\mathcal{M}^\kappa(t)$  is monotone in  $t$ , and their expected payment when bidding  $t$  satisfies

$$(4.1) \quad \begin{aligned} \mathbf{E}[p^\kappa(t)] &= t^\kappa \cdot \mathcal{M}^\kappa(t) - \int_0^t \mathcal{M}^\kappa(z) dz \\ &= t \int_0^1 \mathcal{M}^\kappa(t) - \mathcal{M}^\kappa(yt) dy. \end{aligned}$$

Thus, to compute the payment for an agent, the mechanism samples a uniformly random  $y \in [0, 1]$ , runs the RSM reduction again replacing  $t^\kappa$  with  $yt^\kappa$ , and charges a price of  $t^\kappa \cdot (\mathcal{M}(t^\kappa) - \mathcal{M}(yt^\kappa))$ . The expectation of this price is equal to the right side of (4.1), thus ensuring Bayesian incentive compatibility.

**4.2 Discrete types.** Consider settings with discrete, finite type spaces. Without loss of generality the types are  $\mathcal{T} = [m]$  and type  $t$  occurs with probability  $\pi_t$ . We assume oracle access to the agent valuation function, i.e., for any type  $t$  and outcome  $x$  we can evaluate  $v(t_i, x_j)$ . Define  $\mu$  to be the minimum granularity of the utility function, e.g., if utilities are given in  $k$ -bit binary then  $\mu = 2^{-k}$ . We will allow our BIC reduction in this section to be polynomial in  $m$  and  $1/\mu$ , i.e., it is a pseudo-polynomial time reduction.

We restrict attention to general feasibility settings, i.e., where the designer’s cost function is  $c(\cdot) \in \{0, \infty\}$ . We assume, without loss of generality for these settings, that there is a special “status quo” outcome 0, e.g., corresponding to not running the mechanism at all. All types  $t \in \mathcal{T}$  are assumed to satisfy  $v(t, 0) = 0$ .<sup>2</sup>

The reduction we give exploits the following two aspects of discrete settings for mechanism design. First, in discrete settings if we make the estimated outcomes under distinct reports of an agent different enough, then even small errors in estimation are not enough to violate the discrete incentive constraints. Second, while with continuous type spaces the payments for a given allocation rule are uniquely determined up to an additive shift, for discrete type spaces

<sup>2</sup>In settings where there is no “status quo” outcome, we can artificially create one by defining outcome 0 to be a random outcome, i.e. the distribution over outcomes defined by sampling a random type profile from the agents’ type distributions and evaluating algorithm  $\mathcal{A}$  on this type profile. In that case the assumption  $\forall t v(t, 0) = 0$  will not be satisfied, but we can adjust each type by adding a constant to its valuation function so that  $v(t, 0) = 0$  is satisfied. All of our RSM and RSF reductions have the property that their behavior is unaffected when an agent shifts its valuation function by an additive constant, so this transformation doesn’t affect our mechanisms’ behavior.

there is some “wiggle room”. For example, if the values are  $\{0, 1, 2\}$  in a single-dimensional setting, and we wish to only allocate to the agent with value 2 then any price in  $(1, 2]$  is sufficient for guaranteeing incentive compatibility.

Given the above minimum-granularity assumption, two types  $t$  and  $t'$  are either the same or they differ on some outcome  $x$  by at least  $\mu$ . Given any set of outcomes, there is a *minimal distinguishing subset*, of size at most  $M = \binom{m}{2}$ , that distinguishes between any two types that are distinguished by the original set.

**DEFINITION 4.7.** *Parameterized by a distinguishing subset of  $M$  outcomes  $\{x_1, \dots, x_M\}$ , the blatantly monotone algorithm for an agent  $\kappa$ , is denoted by  $\text{BM}^\kappa$ . On type  $t$ , for each  $j \in [M]$  it outputs outcome  $x_j$  with probability  $\frac{1}{M}v(t, x_j)$ , and with the remaining probability, i.e.,  $1 - \frac{1}{M}\sum_{j \in [M]}v(t, x_j)$ , it outputs outcome 0. The payment charged to  $\kappa$  is*

$$p^\kappa(\mathbf{t}) = \frac{1}{2M} \sum_{j=1}^M (v(t, x_j))^2,$$

and the payment charged to all other agents is zero.

Note that our assumption that all agent values are in  $[0, 1]$  ensures that the probabilities defined in Definition 4.7 constitute a valid probability distribution.

Surrogate selection rule ERSF draws  $L$  outcomes for each of  $m$  surrogates for a total of  $Lm$  outcomes. Two types that have the same values across all  $Lm$  of these outcomes are indistinguishable and ERSF treats them identically. Two types that are treated identically do not permit strategic manipulation as an agent with either type receives the same outcome for either report. The blatantly monotone mechanism then only needs to correct for estimation error in outcomes for types that are distinguishable. However, in order to use the blatantly monotone mechanism  $\text{BM}^\kappa$  to restore Bayesian incentive compatibility to the ERSF, we will see that the probability of distinguishing two types using  $\text{BM}^\kappa$  must be greater than the probability of distinguishing them using ERSF. For this reason, we allow  $\text{BM}^\kappa$  to distinguish types using  $Km > Lm$  sampled outcomes.

**DEFINITION 4.8.** *The Blackbox Replica-Surrogate Flow (BRSF) reduction with parameters  $K, L$  and  $\delta$  is the following:*

1. *With probability  $1 - \delta$  run the ERSF reduction, drawing  $L$  samples of the algorithm’s outcome for each of the  $m$  types of each of the  $n$  agents.*

2. With probability  $\delta$  pick an agent  $\kappa$  uniformly at random. Draw  $K$  samples of the algorithm's outcome for each of the  $m$  types of  $\kappa$ , and run the blatantly monotone mechanism as defined by  $\text{BM}^\kappa$  for a distinguishing subset of the  $Km$  outcomes sampled for  $\kappa$ .

We now give three lemmas. First, we present two lemmas to show that the benefit an agent can gain in ERSF by misreporting its type is not too large. Then, in Lemma 4.12, we show that agents who misreport their type in the blatantly monotone mechanism suffer a strictly positive penalty. Combining the lemmas, Theorem 4.2 will show how to set the parameters  $K, L, \delta$  such that the penalty of misreporting in  $\text{BM}^\kappa$  is large enough to cancel out the potential benefit from misreporting in ERSF.

LEMMA 4.9. Fix two types  $t, t' \in \mathcal{T}$  and a real number  $\gamma > 0$ . If  $L > 2\gamma^{-2} \ln(\gamma/2m)$  then an agent of type  $t$  who bids  $t'$  in ERSF instead of bidding truthfully cannot increase its utility by more than  $3\gamma$ .

*Proof.* For any specific surrogate outcome  $X_j$ , the Chernoff-Hoeffding inequality implies that

$$\Pr\left(|v(t, X_j) - v(t, \tilde{X}_j)| > \gamma\right) < 2 \exp\left(-\frac{L\gamma^2}{2}\right) = \frac{\gamma}{m}.$$

Taking the union bound over all surrogate outcomes  $X_j$  we now see that with probability at least  $1 - \gamma$ , the estimate

$$(4.2) \quad |v(t, X_j) - v(t, \tilde{X}_j)| \leq \gamma$$

holds for all  $j$ , and consequently the same estimate also holds for any convex combination of surrogate outcomes, for example the convex combinations to which types  $t, t'$  are assigned in the minimum cost flow computed by the ERSF mechanism. Denote these convex combinations of surrogate outcomes by  $X(t), X(t')$ , respectively, and denote the corresponding convex combinations of outcome estimates by  $\tilde{X}(t), \tilde{X}(t')$ , respectively.

Recall that ERSF uses the VCG mechanism on the minimum-cost flow instance defined by costs  $-v(t, \tilde{X}_j)$ . We know that type  $t$  gains nothing by misreporting its type as  $t'$  in this VCG mechanism, i.e.

$$v(t, \tilde{X}(t)) - p^\kappa(t, \mathbf{t}^{-\kappa}) \geq v(t, \tilde{X}(t')) - p^\kappa(t', \mathbf{t}^{-\kappa}).$$

Assuming that (4.2) holds for all  $j$ , it implies that  $|v(t, X(t)) - v(t, \tilde{X}(t))|$  and  $|v(t, X(t')) - v(t, \tilde{X}(t'))|$  are both bounded above by  $\gamma$ , hence

$$v(t, X(t)) - p^\kappa(t, \mathbf{t}^{-\kappa}) + 2\gamma \geq v(t, X(t')) - p^\kappa(t', \mathbf{t}^{-\kappa}),$$

i.e., type  $t$  gains no more than  $2\gamma$  by misreporting her type as  $t'$ . Finally, the event that (4.2) fails to hold for all  $j$  has probability at most  $\gamma$  and in that case the benefit of misreporting one's type cannot be greater than 1. Combining these two cases, we see that the expected benefit of misreporting type  $t$  as  $t'$  is no more than  $3\gamma$ .  $\square$

We will need to use a different upper bound on the benefit of misreporting  $t$  as  $t'$  in case  $t$  and  $t'$  are very hard to distinguish, meaning that the outcomes distinguishing  $t$  from  $t'$  are very rarely sampled when evaluating  $\mathcal{A}$  on randomly sampled type profiles. To quantify this, we need the following definition.

DEFINITION 4.10. The indistinguishability parameter of two types  $t, t' \in \mathcal{T}$ , denoted by  $\iota(t, t')$ , is defined to be the probability that we fail to observe an outcome that distinguishes  $t$  from  $t'$  when sampling one outcome independently for each of the  $m$  elements of  $\mathcal{T}$ .

LEMMA 4.11. For any two types  $t, t' \in \mathcal{T}$ , an agent of type  $t$  who bids  $t'$  in ERSF instead of bidding truthfully cannot increase its utility by more than  $1 - \iota(t, t')^L$ .

*Proof.* Consider the  $L$  independent outcomes drawn by the ERSF reduction in defining the outcome estimates  $\tilde{X}_j$ . If this set of  $Lm$  outcomes fails to contain an outcome distinguishing  $t$  from  $t'$  then the ERSF mechanism will output exactly the same outcome and payment for agent  $\kappa$  regardless of whether its bid is  $t$  or  $t'$ . The probability of this event is  $\iota(t, t')^L$ . The lemma now follows from the observation that an agent can never gain a benefit greater than 1 from misreporting its type.  $\square$

We turn now to analyzing the blatantly monotone mechanism.

LEMMA 4.12. Consider the blatantly monotone mechanism  $\text{BM}^\kappa$  with distinguishing outcome set  $\{x_1, \dots, x_M\}$ . For any two types  $t, t' \in \mathcal{T}$ , an agent of type  $t$  is indifferent between reporting its type as  $t$  or  $t'$  unless  $\{x_1, \dots, x_M\}$  contains an outcome distinguishing  $t$  from  $t'$ . In that case, reporting type  $t'$  rather than  $t$  decreases the agent's utility by at least  $\frac{\mu^2}{2M}$ .

*Proof.* Let  $\mathbf{v}, \mathbf{w} \in \mathbb{R}^M$  denote the vectors whose components are specified by  $v_j = v(t, x_j)$  and  $w_j = v(t', x_j)$ , respectively. When we run  $\text{BM}^\kappa$  on inputs  $t, t'$  to obtain two outcome distributions, we find that type  $t$  assigns value  $\frac{1}{M} \mathbf{v} \cdot \mathbf{v}$  and  $\frac{1}{M} \mathbf{v} \cdot \mathbf{w}$ , respectively, to these two outcome distributions. The payment is

$\frac{1}{2M}\|\mathbf{v}\|^2$  when bidding  $t$  and  $\frac{1}{2M}\|\mathbf{w}\|^2$  when bidding  $t'$ , where  $\|\cdot\|$  denotes the  $L^2$  norm on  $\mathbb{R}^M$ . Hence, the decrease in utility when bidding  $t'$  rather than  $t$  is given by

$$\begin{aligned} & \frac{1}{M} \left( \mathbf{v} \cdot \mathbf{v} - \frac{1}{2}\|\mathbf{v}\|^2 - \mathbf{v} \cdot \mathbf{w} + \frac{1}{2}\|\mathbf{w}\|^2 \right) \\ &= \frac{1}{2M} (\|\mathbf{v}\|^2 - 2\mathbf{v} \cdot \mathbf{w} + \|\mathbf{w}\|^2) \\ &= \frac{1}{2M} \|\mathbf{v} - \mathbf{w}\|^2. \end{aligned}$$

If the set  $\{x_1, \dots, x_M\}$  contains no outcome distinguishing  $t$  from  $t'$ , then  $\mathbf{v} = \mathbf{w} = 0$  and the right side is zero. Otherwise,  $\|\mathbf{v} - \mathbf{w}\|^2$  is bounded below by  $\mu^2$ , so the decrease in utility is at least  $\frac{\mu^2}{2M}$ .  $\square$

The preceding lemmas established upper bounds on the expected benefit that an agent of type  $t$  can obtain by misreporting  $t'$  in ERSF and a lower bound on the expected penalty that the agent suffers by misreporting  $t'$  in  $\text{BM}^\kappa$ . Our next goal is to prove that for suitable choices of the parameters  $\delta, L, K$ , the penalty outweighs the benefit, resulting in a BIC mechanism. To do so, we need the following simple inequality, whose proof appears in Appendix A.

LEMMA 4.13. *For any real numbers  $a, b, z$  in the interval  $(0, 1)$  such that  $a \leq b/2$ , if  $q$  is any integer greater than  $1/a$ , then*

$$(4.3) \quad \min\{a, 1 - z\} < b(1 - z^q).$$

THEOREM 4.2. *If  $\delta, \gamma, L, K$  satisfy*

$$\begin{aligned} \delta &< \varepsilon/n \\ \gamma &< \frac{\mu^2 \delta}{6m^2 n} \\ L &> 2\gamma^{-2} \ln(2m/\gamma) \\ K &> L\lceil \gamma^{-1} \rceil, \end{aligned}$$

*then the BRSF reduction yields a mechanism  $\mathcal{M}$  satisfying BIC, and its social surplus satisfies the bound  $\mathcal{M} \geq \mathcal{A} - O(\varepsilon)$ .*

*Proof.* To verify BIC, consider agent  $\kappa$  of type  $t$  misreporting  $t'$ . Let  $z = \iota(t, t')^L$ . With probability  $\delta/n$ , we run the mechanism  $\text{BM}^\kappa$ . In that case, the probability that the  $K$  sampled outcomes fail to distinguish  $t$  from  $t'$  is at most  $z^q$ , where  $q = \lceil \gamma^{-1} \rceil$ . So, with probability at least  $(\delta/n)(1 - z^q)$ , we run  $\text{BM}^\kappa$  using an outcome set  $\{x_1, \dots, x_M\}$  that distinguishes  $t$  from  $t'$ . Applying Lemma 4.12, we find that the possibility of running  $\text{BM}^\kappa$  yields a contribution of

$$-\frac{\mu^2}{2M} \cdot \frac{\delta}{n} \cdot (1 - z^q)$$

to the agent's expected utility. Meanwhile, Lemmas 4.9 and 4.11 ensure that the possibility of running ERSF contributes at most

$$\min\{3\gamma, 1 - z\}$$

to the agent's expected utility. To see that the negative contribution from  $\text{BM}^\kappa$  outweighs the positive contribution from ERSF, it merely suffices to apply Lemma 4.13 with  $a = 3\gamma$  and  $b = \frac{\mu^2 \delta}{2Mn}$ . The inequality  $q > 1/a$  is obvious because  $q = \lceil \gamma^{-1} \rceil$ . To see that  $a \leq b/2$ , note first that  $2M < m^2$ , hence

$$b > (\mu^2 \delta)/(m^2 n) > 6\gamma = 2a.$$

We have verified all of the conditions for Lemma 4.13, and the lemma now implies that

$$\min\{3\gamma, 1 - z\} - \frac{\mu^2}{2M} \cdot \frac{\delta}{n} \cdot (1 - z^q) \leq 0,$$

from which it follows immediately that  $\mathcal{M}$  is BIC.

The following observations guarantee that the welfare loss of  $\mathcal{M}$  is bounded by  $O(\varepsilon)$ . First, let us exclude two bad events: the event that we run the blatantly monotone mechanism for some agent (combined probability  $\delta$ ), and the event that we run the ERSF reduction but there exists an agent  $\kappa$  and a type  $j \in \mathcal{T}$  such that (4.2) is violated (combined probability  $\gamma n$ , by applying the union bound over agents). As  $\delta + \gamma n < 2\delta < 2\varepsilon/n$ , and the combined welfare loss of all agents in this case is at most  $n$ , the two bad events contribute at most  $2\varepsilon$  to the expected loss in social surplus. Excluding the two bad events, mechanism  $\mathcal{M}$  solves a minimum-cost flow problem to maximize *estimated* utility for each agent, and each agent's estimated utility differs from its actual utility by at most  $2\gamma$ . Consequently, the actual social surplus achieved by  $\mathcal{M}$  differs from that which is achieved by  $\mathcal{A}$  by at most  $2\gamma n$ , which is bounded above by  $2\varepsilon/n$ .  $\square$

**4.3 Multi-dimensional continuous types.** Unfortunately, when agents have multi-dimensional continuous types and we only have blackbox access to the algorithm  $\mathcal{A}$ , we do not know of any way to use the RSM reduction to translate it into a perfectly BIC mechanism. As in the single-dimensional case, there are examples to illustrate that one cannot simply define  $\tilde{X}_k$  to be the average of  $L$  samples from the outcome distribution of  $s_k$ . One might think that this difficulty can be overcome by estimating  $v(v_j, s_k)$  using a binomial estimator, as we did in the single-dimensional case, but this idea fails even in dimension 2.<sup>3</sup>

<sup>3</sup>A counterexample when  $m = 2, L = 1$  can be obtained by taking agent  $i$ 's type distribution to be uniformly ran-

On the other hand, achieving  $\varepsilon$ -Bayesian incentive compatibility is easy using the RSM reduction, where we define  $\tilde{X}_k$  to be the average of  $L = \Omega(\varepsilon^{-2} \log(m/\varepsilon))$  samples from the outcome distribution  $\mathcal{A}(s_k)$ . As before, Chernoff-Hoeffding bounds imply now that with probability  $1 - \varepsilon$ , there is no edge in the bipartite graph whose estimated weight differs from its true weight by more than  $\varepsilon/m$ . Since the prices are defined using shortest augmenting paths in the bipartite graph, and an augmenting path has fewer than  $2m$  edges, our prices differ from the prices in the ideal model by at most  $2\varepsilon$  with probability at least  $1 - \varepsilon$ , and the price difference is  $O(1)$  regardless. Consequently, an agent can gain at most  $O(\varepsilon)$  in expectation by bidding untruthfully.

## References

- [1] A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. 42nd IEEE Symp. on Foundations of Computer Science*, 2001.
- [2] X. Bei and Z. Huang. Towards optimal Bayesian algorithmic mechanism design. *CoRR*, abs/1005.4244, 2010.
- [3] P. Briest, P. Krysta, and B. Vocking. Approximation techniques for utilitarian mechanism design. In *Proc. 36th ACM Symp. on Theory of Computing*, 2005.
- [4] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
- [5] P. Dhangwatnotai, S. Dobzinski, S. Dughmi, and T. Roughgarden. Truthful approximation schemes for single-parameter agents. In *Proc. 49th IEEE Symp. on Foundations of Computer Science*, 2008.
- [6] S. Dughmi and T. Roughgarden. Black-box randomized reductions in algorithmic mechanism design. In *Proc. 51th IEEE Symp. on Foundations of Computer Science*, 2010.
- [7] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
- [8] J. Hartline and B. Lucier. Bayesian algorithmic mechanism design. In *Proc. 41th ACM Symp. on Theory of Computing*, pages 301–310, 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] D. Lehmann, L. I. O’Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. In *Proc. 1st ACM Conf. on Electronic Commerce*, pages 96–102. ACM Press, 1999.
- [11] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.
- [12] J. Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16(2):191–200, 1987.
- [13] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *J. of Finance*, 16:8–37, 1961.

## A Proofs omitted from Section 4

**A.1 Proof of Theorem 4.1.** In this section we restate and prove Theorem 4.1.

**THEOREM A.1.** *In the single-dimensional setting, the RSM reduction using the binomial estimator is monotone.*

*Proof.* Let  $t_1 < t_2$  be any pair of types for agent  $i$ . We need to show that  $\mathbf{E}[\tilde{\mathcal{A}}(t_1)] \leq \mathbf{E}[\tilde{\mathcal{A}}(t_2)]$ , and we will do this by defining an appropriate random variable  $W$  and showing that the relation  $\mathbf{E}[\tilde{\mathcal{A}}(t_1) \mid W = w] \leq \mathbf{E}[\tilde{\mathcal{A}}(t_2) \mid W = w]$  holds for every possible value  $w$  of  $W$ . To begin with, note that when we run the RSM reduction, the maximum matching of  $\{v_1, \dots, v_m\}$  to  $\{\tilde{X}_1, \dots, \tilde{X}_m\}$  will be order-preserving; we may assume the algorithm breaks ties randomly if one or both of these multisets contains a repeated element. Let  $a$  be the position of  $v_1$  when  $v_1, \dots, v_m$  are placed in increasing order (breaking ties randomly) and  $v_1 = t_1$ . Define  $b$  similarly but using  $v_1 = t_2$ . Note that  $a \leq b$ . Similarly, let  $\sigma : [m] \rightarrow [m]$  be a permutation that places  $\{\tilde{X}_1, \dots, \tilde{X}_m\}$  in increasing order, breaking ties at random. Letting  $\alpha = \sigma^{-1}(a)$ ,  $\beta = \sigma^{-1}(b)$ , we see that the RSM reduction assigns  $v_1$  to  $s_\alpha$  if  $v_1 = t_1$  and to  $s_\beta$  if  $v_1 = t_2$ . Let  $\mathbf{s} = (s_1, s_2, \dots, s_m)$  denote the sequence defined by  $s_i = L \cdot \tilde{X}_{\sigma(i)}$ . Finally, define the random variable  $W$  to be the ordered tuple consisting of the types  $s_1, \dots, s_m$ , the sequence  $\mathbf{s}$ , the random numbers  $a, b$ , and the values of  $\sigma^{-1}(k)$  for every  $k \in [m] \setminus \{a, b\}$ . Thus,  $W$  encodes all the relevant information about one execution of the RSM reduction except for one crucial missing piece: we know the value of the unordered pair  $\{\alpha, \beta\}$  but we don’t know which element of this pair is  $\alpha$  and which one is  $\beta$ ; this information is crucial because the reduction assigns  $t_1$  to  $s_\alpha$  and  $t_2$  to  $s_\beta$ .

We now express the expected allocations of  $t_1$  and  $t_2$  in terms of the values of  $W, \beta$ . Let  $k = \operatorname{argmax}_{j \in \{\alpha, \beta\}}(y_j)$ ,  $\ell = \operatorname{argmin}_{j \in \{\alpha, \beta\}}(y_j)$ . Note that the value of  $W$  determines the pair of indices  $\alpha, \beta$  and it also determines the tuple  $\mathbf{s} = (s_1, \dots, s_m)$ , which in turn determines  $\mathbf{y} = (y_1, \dots, y_m)$ . Consequently the values  $k, \ell$  are determined by  $W$ . In the following equations, when we condition on the event “ $W = w$ ”, we abbreviate this by writing “ $w$ ”.

$\overline{\operatorname{dom}}$  over a two-element set  $\{t, t'\}$  satisfying  $v(t, \mathcal{A}(t)) = 0.51, v(t, \mathcal{A}(t')) = 0.75, v(t, \mathcal{A}(t)) = 0.75, v(t, \mathcal{A}(t')) = 1.0$ .

$$\begin{aligned}\mathbf{E}[\bar{\mathcal{A}}(t_1) \| w] &= \mathbf{E}[\bar{\mathcal{A}}(t_1) \| w, \beta = k] \cdot \Pr[\beta = k \| w] \\ &\quad + \mathbf{E}[\bar{\mathcal{A}}(t_1) \| w, \beta = \ell] \cdot \Pr[\beta = \ell \| w] \\ &= y_\ell \Pr[\beta = k \| w] + y_k \Pr[\beta = \ell \| w]\end{aligned}$$

$$\begin{aligned}\mathbf{E}[\bar{\mathcal{A}}(t_2) \| w] &= \mathbf{E}[\bar{\mathcal{A}}(t_2) \| w, \beta = k] \cdot \Pr[\beta = k \| w] \\ &\quad + \mathbf{E}[\bar{\mathcal{A}}(t_2) \| w, \beta = \ell] \cdot \Pr[\beta = \ell \| w] \\ &= y_k \Pr[\beta = k \| w] + y_\ell \Pr[\beta = \ell \| w]\end{aligned}$$

$$\begin{aligned}\mathbf{E}[\bar{\mathcal{A}}(t_2) - \bar{\mathcal{A}}(t_1) \| w] &= (y_k - y_\ell) \cdot (\Pr[\beta = k \| w] - \Pr[\beta = \ell \| w]) \\ &= \frac{y_k - y_\ell}{\Pr(W = w)} \cdot \left( \Pr \left[ \begin{array}{c} \beta = k, \\ W = w \end{array} \right] - \Pr \left[ \begin{array}{c} \beta = \ell, \\ W = w \end{array} \right] \right).\end{aligned}$$

Every factor in the last line is non-negative, except possibly the probability difference  $\Pr[\beta = k, W = w] - \Pr[\beta = \ell, W = w]$ . To prove that this difference is in fact positive, we will in fact prove that

$$\Pr[\beta = k, W = w \| a, b, \mathbf{s}] \geq \Pr[\beta = \ell, W = w \| a, b, \mathbf{s}].$$

for all possible values of the random variables  $a, b$ , and  $\mathbf{s}$ . Note that when we condition on  $a, b, \mathbf{s}$ , the values of  $\beta, W$  determine the value of the vector  $\tilde{\mathbf{X}} = (\tilde{X}_1, \dots, \tilde{X}_m)$  and vice-versa. Specifically, since  $\tilde{\mathbf{X}}$  is the vector obtained from  $\mathbf{s}/L$  by rearranging its entries using  $\sigma^{-1}$ ,  $W$  constrains  $\tilde{\mathbf{X}}$  to be one of two possible vectors  $\mathbf{z}, \mathbf{z}'$  that differ by interchanging their  $k^{\text{th}}$  and  $\ell^{\text{th}}$  components. Assume without loss of generality that  $z_k \geq z_\ell$ . (Otherwise, simply rename  $\mathbf{z}$  to  $\mathbf{z}'$  and vice-versa.) Then

$$\begin{aligned}\Pr[\beta = k, W = w \| a, b, \mathbf{s}] &= \Pr[\tilde{\mathbf{X}} = \mathbf{z} \| a, b, \mathbf{s}] \\ \text{(A.1)} \quad &= \prod_{j=1}^m \Pr[L \cdot \tilde{X}_j = z_j]\end{aligned}$$

$$\begin{aligned}\Pr[\beta = \ell, W = w \| a, b, \mathbf{s}] &= \Pr[V \tilde{\mathbf{X}} = \mathbf{z}' \| a, b, \mathbf{s}] \\ \text{(A.2)} \quad &= \prod_{j=1}^m \Pr[L \cdot \tilde{X}_j = z'_j],\end{aligned}$$

where we have used the fact that the random variables  $\tilde{X}_j$  are conditionally independent given  $a, b, \mathbf{s}$ . Finally, combining (A.1) and (A.2) we obtain

$$\begin{aligned}&\frac{\Pr[\beta = k, W = w \| a, b, \mathbf{s}]}{\Pr[\beta = \ell, W = w \| a, b, \mathbf{s}]} \\ &= \frac{\Pr[L \cdot \tilde{X}_k = z_k] \Pr[L \cdot \tilde{X}_\ell = z_\ell]}{\Pr[L \cdot \tilde{X}_k = z_\ell] \Pr[L \cdot \tilde{X}_\ell = z_k]}.\end{aligned}$$

The right side is greater than or equal to 1, by Lemma 4.6, since  $L \cdot \tilde{X}_k, L \cdot \tilde{X}_\ell$  are binomial random variables with expectations  $L \cdot y_k \geq L \cdot y_\ell$ , and  $z_k \geq z_\ell$ .  $\square$

**A.2 Proof of Lemma 4.13.** In this section we restate and prove Lemma 4.13.

LEMMA A.1. *For any real numbers  $a, b, z$  in the interval  $(0, 1)$  such that  $a \leq b/2$ , if  $q$  is any integer greater than  $1/a$ , then*

$$\min\{a, 1 - z\} < b(1 - z^q).$$

*Proof.* As  $q > 1/a$  we have

$$(1 - a)^q < e^{-aq} < e^{-1} < \frac{1}{2}.$$

The proof consists of applying this inequality in two cases.

**Case 1:**  $a < 1 - z$ . In this case we have  $z < 1 - a$ , hence

$$1 - z^q > 1 - (1 - a)^q > \frac{1}{2}$$

$$\min\{a, 1 - z\} = a \leq \frac{b}{2} < b(1 - z^q),$$

as claimed.

**Case 2:**  $a \geq 1 - z$ . The equation

$$\frac{1 - z^q}{1 - z} = 1 + z + \dots + z^{q-1}$$

reveals that  $\frac{1 - z^q}{1 - z}$  is an increasing function of  $z \in (0, 1)$ . As  $z \geq 1 - a$ , we may conclude that

$$\begin{aligned}b \left( \frac{1 - z^q}{1 - z} \right) &\geq b \left( \frac{1 - (1 - a)^q}{1 - (1 - a)} \right) \\ &= \frac{b}{a} (1 - (1 - a)^q) > \frac{b}{a} \cdot \frac{1}{2} \geq 1,\end{aligned}$$

whence  $b(1 - z^q) \geq 1 - z = \min\{a, 1 - z\}$ , as desired.  $\square$