

Non-optimal Mechanism Design (Online Appendix)

By JASON D. HARTLINE AND BRENDAN LUCIER *

A. PROFIT WITH IRREGULAR DISTRIBUTIONS

In Section II.E we showed that for distributions satisfying the regularity condition of Myerson (1981), the same ironed algorithm that preserves welfare and is incentive compatible also preserves profit (via an analysis of Myerson’s virtual values). For the case of irregular distributions, i.e., where the virtual values of Definition 6 are non-monotone, we give a method that preserves profit but may not simultaneously preserve welfare.

DEFINITION 9: *An ironed virtual value function is $\bar{\varphi}(\cdot)$ a weakly monotone function such that*

- 1) *for any monotone allocation rule, expected ironed virtual welfare is at least the expected payment of the agent, and*
- 2) *for any monotone allocation rule that is constant where the ironed virtual value function is constant, expected ironed virtual welfare equals expected payment of the agent.*

THEOREM 6 (Myerson, 1981): *For any value distribution F an ironed virtual value function exists and can be found by ironing (Definition 4) the virtual value function (Definition 6).*

Observe from Theorem 3 and Definition 9 that, for any distribution of values and any (possibly non-monotone) algorithm with good ironed virtual welfare, the ironed algorithm of Definition 5 is monotone and weakly improves ironed virtual welfare. However, Part 2 of Definition 9 may fail to hold for such an ironed algorithm, and consequentially expected ironed virtual welfare may not correspond to expected profit. The main question of this section, thus, refocuses on converting a monotone algorithm with good ironed virtual welfare into a monotone algorithm that also satisfies Part 2 of Definition 9.

We will resolve this question positively by again employing the approach of resampling transformations. Unlike the resampling transformation of Section II

* Hartline: Northwestern University, Evanston, IL, USA, hartline@eecs.northwestern.edu.
Lucier: Microsoft Research, Cambridge, MA, USA, brlucier@microsoft.com.

which was specific to the allocation rule $x(\cdot)$ and oblivious to the value function $\nu(\cdot)$, this new resampling transformation will be oblivious to the allocation rule and specific to the value function. If the original algorithm is non-monotone, then this transformation can be composed with the ironing method from Definition 5 to achieve both monotonicity and the equivalence of expected ironed virtual welfare and expected profit.

DEFINITION 10: *The value-ironed intervals for ν are the intervals where $\nu'(\cdot) = 0$; the value-ironing transformation σ is the interval resampling transformation corresponding to the value-ironed intervals; and the value-ironed allocation rule $\bar{x}$ for x is defined by $\bar{x}(v) = \mathbf{E}_\sigma[x(\sigma(v))]$.*

LEMMA 8: *For any monotone value function $\nu(\cdot)$, the value ironed allocation rule $\bar{y}(\cdot)$ for allocation rule $y(\cdot)$ has the same expected welfare and is constant on intervals where $\nu(\cdot)$ is constant.*

PROOF: For the second part of the lemma and by the geometric interpretation of resampling on the allocation rule as averaging, the value-ironed allocation rule is constant on any value-ironed interval. For the first part of the lemma and by the line segment interpretation of resampling on the cumulative allocation rule, the cumulative allocation rules satisfy $\bar{Y}(q) = Y(q)$ for q that is not in any ironed interval. Thus,

$$\begin{aligned} \mathbf{E}_q[\nu(q)y(q)] &= \nu(1)Y(1) + \mathbf{E}_q[-\nu'(q)Y(q)] \\ &= \nu(1)\bar{Y}(1) + \mathbf{E}_q[-\nu'(q)\bar{Y}(q)] = \mathbf{E}_q[\nu(q)\bar{y}(q)]. \end{aligned}$$

The first and third equality follow from Lemma 1. The second equality follows because $Y(q) = \bar{Y}(q)$ at all q that are not within a value-ironed interval (by the line-segment interpretation of resampling that is given in Section II.C) and $\nu'(q) = 0$ for q within any value-ironed interval. Q.E.D.

THEOREM 7: *For any distribution $\mathbf{F}$, monotone algorithm $\mathcal{A}$, and value functions given by ironed virtual values for $\mathbf{F}$, the value-ironed algorithm $\bar{\mathcal{A}}$ is BIC and has expected profit equal to the expected ironed virtual welfare of $\mathcal{A}$.*

PROOF: Bayesian incentive compatibility of $\bar{\mathcal{A}}$ follows from the assumption that $\mathcal{A}$ is monotone, the equivalence of monotonicity of allocation rule and concavity of cumulative allocation rule, and the line-segment interpretation of interval resampling on the cumulative allocation rule by which concavity is clearly preserved. Equality of expected profit of $\bar{\mathcal{A}}$ and expected ironed virtual welfare of $\mathcal{A}$ follows from the definition and existence of ironed virtual values (Definition 9; Theorem 6), Lemma 8, and the fact that expected costs are unchanged by resampling transformations. Q.E.D.

Theorem 7 assumes that algorithm $\mathcal{A}$ is monotone. If it is not then, as discussed at the beginning of this section, we can compose the allocation-rule ironing

construction (Definition 5) of Section II with the value-ironing construction (Definition 10). Let σ^x be the interval resampling transformation for allocation-rule ironing of allocation rule $x(\cdot)$ and let σ^ν be the interval resampling transformation for value ironing of $\nu(\cdot)$. The composed transformation $\sigma(v) = \sigma^x(\sigma^\nu(v))$ fixes non-monotonicities (via σ^x) and ensures that the allocation rule is constant on intervals where $\nu(\cdot)$ is constant (via σ^ν). Moreover, each of σ^x and σ^ν weakly improves expected welfare with respect to $\nu(\cdot)$, and hence the same is also true of σ . The composed transformation therefore yields, for an *arbitrary* algorithm $\mathcal{A}$, a BIC algorithm with expected profit that is no less than the expected ironed virtual welfare of $\mathcal{A}$.

B. APPROXIMATE BAYESIAN INCENTIVE COMPATIBILITY

We now prove the following restatement of Theorem 5.

THEOREM 5': *Given black-box access to algorithm $\mathcal{A}$, an ϵ -BIC algorithm $\bar{\mathcal{A}}$ with $\bar{\mathcal{A}} \geq \mathcal{A} - \epsilon$ can be constructed in $O\left(\frac{n^4}{\epsilon^3} \log\left(\frac{n}{\epsilon}\right)\right)$ basic operations and calls to $\mathcal{A}$.¹*

PROOF: Fix a parameter $\lambda \in (0, 1)$ which we will subsequently tune. Given algorithm $\mathcal{A}$ with allocation rules $\mathbf{x}$, we construct estimated allocation rules as follows. For each agent i we partition $[0, 1]$ into $1/\delta$ intervals of length δ . For each such interval I , we draw $\frac{1}{\lambda^2} \log\left(\frac{n}{\delta\lambda}\right)$ samples $\mathbf{v} \sim \mathbf{F}$ subject to $v_i \in I$, and calculate the average allocation to agent i by algorithm $\mathcal{A}$ under these samples. Chernoff-Hoeffding bounds then imply that the resulting estimation differs from the true interim allocation probability by at most λ with probability at least $1 - \frac{\lambda\delta}{n}$. The union bound over all intervals and all agents then implies that each of the estimated allocation rules is λ -close to x_i with probability at least $1 - \lambda$.

In the λ -probability event that the estimates are not λ -close, the resulting mechanism is trivially 1-BIC and reduces the total social value of $\mathcal{A}$ by at most n (an upper bound on the sum of agent values). Since this event occurs with probability at most λ , in expectation it contributes at most λ to the extent of non-monotonicity of the constructed mechanism, and at most λn to the expected loss in social welfare. We will therefore assume for the remainder that this event does not occur, and that the estimates are indeed λ -close. Write $\mathbf{x}'$ for the allocation rule that corresponds to these estimates.

Let σ' be the welfare-optimal resampling transformations for $\mathbf{x}'$, and let $\mathbf{p}$ denote the payment rule that would make $\mathbf{x}' \circ \sigma'$ BIC. Since $\mathbf{x}$ and $\mathbf{x}'$ are λ -close, applying Lemma 7 to each of the n agents implies that $\mathbf{E}_{\mathbf{v}}[\mathbf{v} \cdot \mathbf{x}(\sigma'(\mathbf{v}))] \geq \mathbf{E}_{\mathbf{v}}[\mathbf{v} \cdot \mathbf{x}(\mathbf{v})] - 2n\lambda$. Moreover, we know from Lemma 5 that $\mathbf{x} \circ \sigma'$ is λ -close to monotone allocation rule $\mathbf{x}' \circ \sigma'$. Since $\mathbf{x}' \circ \sigma'$ is BIC with payments $\mathbf{p}$, any agent's potential benefit from misreporting his value in $\mathbf{x} \circ \sigma'$, say from true value v_i to

¹In *big-oh notation*, function f is $O(g)$ if there exist constants n_0 and c such that for all $n > n_0$, $f(n) \leq cg(n)$, i.e., f grows asymptotically no faster than g .

value v'_i , is bounded by $|x_i(v_i) - x'_i(v_i)| + |x_i(v'_i) - x'_i(v'_i)|$, which is at most 2λ . Finally, recalling that $c(\mathbf{x})$ is the cost associated with allocation $\mathbf{x}$, we note that the stationarity of σ' implies $\mathbf{E}_{\mathbf{v}}[c(\mathbf{x}(\sigma'(\mathbf{v})))] = \mathbf{E}_{\mathbf{v}}[c(\mathbf{x}(\mathbf{v}))]$.

We conclude that, conditioning on the error event of probability λ not occurring, algorithm $\mathcal{A}$ composed with resampling transformation σ' is 2λ -BIC, and the application of σ' reduces the social welfare of $\mathcal{A}$ by at most $2\lambda n$. (This intermediate result may be of interest in and of itself; see the note following this proof). Since the partitioning of value space into intervals of width δ reduces social welfare by up to an additional amount of δ per agent, we conclude that the resulting mechanism has social welfare at least $\mathcal{A} - (2\lambda + \delta)n$. Thus, unconditionally, we have that the composed algorithm is 3λ -BIC, and suffers a loss in social welfare bounded by $(3\lambda + \delta)n$. The total runtime, which is dominated by sampling, is $O\left(\frac{n}{\delta\lambda^2} \log\left(\frac{n}{\delta\lambda}\right)\right)$. Choosing $\lambda = \frac{\epsilon}{4n}$ and $\delta = \frac{\epsilon}{4n}$ then leads to the desired result. Q.E.D.

Our construction of ironed algorithm $\bar{\mathcal{A}}$ involves random sampling from the value distribution. Given the samples, the remainder of the construction is deterministic. In the statement of the theorem, the ϵ -BIC property holds in expectation over the randomness in the sampling procedure. In fact, our construction satisfies a stronger property: As mentioned in the proof above, the mechanism generated by the random construction procedure is 2λ -BIC, and has its welfare reduced by at most $2\lambda n$, subject to an event of probability at least $1 - \lambda$. Taking an appropriate choice of λ , we conclude that even if the samples were observed by the agents, the probability that any agent would have more than an ϵ incentive to deviate is at most ϵ .

C. COMPUTATION AND EXACT BAYESIAN INCENTIVE COMPATIBILITY

In this section we restate and prove Theorem 4. Before the formal proof, we give a high-level description of the construction.

THEOREM 4: *Given black-box access to algorithm $\mathcal{A}$, a BIC algorithm $\bar{\mathcal{A}}$ with $\bar{\mathcal{A}} \geq \mathcal{A} - \epsilon$ can be constructed in a polynomial (in n and $1/\epsilon$) number of basic operations and calls to $\mathcal{A}$.*

To implement monotone algorithm $\bar{\mathcal{A}}$, we first construct an approximately monotone algorithm, call it $\mathcal{A}'$, as described in Section III and Section B of this online appendix. Recall that $\mathcal{A}'$ is close to a monotone algorithm (Definition 8), and its allocation rules are piecewise constant on intervals of width δ . Any non-monotonicities therefore occur as discontinuities at the boundary between consecutive intervals. We will correct these non-monotonicities by taking a convex combination between $\mathcal{A}'$ and a “stair” algorithm whose allocation rules are stair functions (i.e., with monotone discontinuities at these boundary points). Implementing the stair algorithm involves choosing an agent i at random, then choosing randomly between an outcome with a high allocation for i or one with

a low allocation for i , with more weight placed on the high allocation when the agent reports a higher value.

A few details deserve mention. Implementing the stair algorithm requires, for each agent i , outcomes $\mathbf{y}$ and $\mathbf{z}$ such that $y_i < z_i$ differ sufficiently to offset any non-monotonicities in $\mathcal{A}'$. To find these outcomes, we will search the samples taken in the construction of $\mathcal{A}'$. If no appropriate pair of outcomes were observed, then we can deduce that agent i 's allocation is nearly independent of her report. In this event, we will monotinize agent i 's allocation rule, heavy-handedly, by ironing over the entire range $[0, 1]$, which effectively disregards her reported value.

To summarize, the construction of $\bar{\mathcal{A}}$ proceeds in the following steps:

- 1) Construct an approximately BIC algorithm $\mathcal{A}'$ via sampling.
- 2) Construct $\mathcal{A}''$ from $\mathcal{A}'$ by fully ironing the reported value of each agent i for which all outcomes observed in Step 1 provide nearly the same allocation to i .
- 3) Construct $\bar{\mathcal{A}}$ from $\mathcal{A}''$ and a stair algorithm: run $\mathcal{A}''$ with high probability; otherwise, run a stair algorithm using outcomes observed during Step 1.

Write $\mathbf{x}$ for the allocation rule of $\mathcal{A}$. Let $\epsilon', \rho, \pi \in (0, 1)$ be parameters that we will subsequently tune. Let $\mathcal{A}'$ be the approximately BIC algorithm from Theorem 5, with approximation parameter ϵ' . That is, $\mathcal{A}'$ is ϵ' -BIC and satisfies $\mathcal{A}' \geq \mathcal{A} - \epsilon'$. Let σ' be the resampling transformation used to construct $\mathcal{A}'$ (see Section B of this online appendix), so that $\mathbf{x} \circ \sigma'$ is the allocation rule of $\mathcal{A}'$.

We construct algorithm $\mathcal{A}''$ from $\mathcal{A}'$ as follows. For each agent i , over all samples used to construct σ' , consider the observed allocation $\mathbf{y}^{(i)}$ that minimizes $y_i^{(i)}$ and the allocation $\mathbf{z}^{(i)}$ that maximizes $z_i^{(i)}$. If $z_i^{(i)} - y_i^{(i)} < \rho$ then we will set σ_i'' to completely iron agent i 's report on the interval $[0, 1]$. Otherwise, we set $\sigma_i'' = \sigma_i'$. Algorithm $\mathcal{A}''$ is defined by allocation rule $\mathbf{x} \circ \sigma''$. Note that in the special case of binary allocations we would set $\rho = 1$.

Finally, we construct $\bar{\mathcal{A}}$ from $\mathcal{A}''$. With probability $1 - \pi$, $\bar{\mathcal{A}}$ will apply allocation rule $\mathbf{x} \circ \sigma''$. Otherwise, with probability π , $\bar{\mathcal{A}}$ will proceed as follows. Select an agent i uniformly at random from the set of n agents. Let $k \in \{1, \dots, 1/\delta\}$ be such that $v_i \in [(k-1)\delta, k\delta)$. That is, the reported value of agent i lies in the k th subinterval of $[0, 1]$ of width δ . Return allocation $\mathbf{z}^{(i)}$ with probability $k\delta$, and otherwise return allocation $\mathbf{y}^{(i)}$. Denote by $\bar{\mathbf{x}}$ the allocation rule of $\bar{\mathcal{A}}$.

LEMMA 9: *If the parameters of the construction are chosen so that $\rho\delta\pi \geq n\epsilon'$ then $\bar{\mathcal{A}}$ is BIC.*

PROOF: For any agent i , if σ_i'' irons on the interval $[0, 1]$ then the allocation rule is trivially monotone. Suppose otherwise, in which case $z_i^{(i)} - y_i^{(i)} \geq \rho$.

For any $v' < v$, either v' and v lie in the same interval of width δ or they do not. If they do lie in the same interval, then monotonicity follows immediately as

$\bar{x}_i$ is constant on such intervals. If not, then

$$\begin{aligned}\bar{x}_i(v) - \bar{x}_i(v') &\geq (1 - \pi) [(x_i \circ \sigma'_i)(v) - (x_i \circ \sigma'_i)(v')] + \pi \left[\frac{1}{n} \cdot \delta \cdot (z_i^{(i)} - y_i^{(i)}) \right] \\ &\geq (1 - \pi)(-\epsilon') + \pi \left[\frac{1}{n} \cdot \delta \cdot \rho \right] \\ &\geq -\epsilon' + \frac{\pi \delta \rho}{n} \geq 0.\end{aligned}$$

The first inequality uses the fact that v' and v lie in different intervals, and hence the expected allocation from the stair algorithm on these two values would differ by at least $\delta \cdot (z_i^{(i)} - y_i^{(i)})$ if agent i were selected (which occurs with probability $\frac{1}{n}$). The second inequality follows from the fact that $\mathbf{x} \circ \sigma'$ is ϵ' -approximately monotone and that $z_i^{(i)} - y_i^{(i)} \geq \rho$, and the final inequality is the assumption $\rho \geq \frac{n\epsilon'}{\delta\pi}$. Q.E.D.

It remains to bound the welfare of $\bar{\mathcal{A}}$ relative to $\mathcal{A}$. We can assume without loss of generality that $c(\mathbf{x}) \in [0, n]$ for any $\mathbf{x}$ returned by $\mathcal{A}$.² From Theorem 5 we know

$$(C1) \quad \mathcal{A} \leq \mathcal{A}' + \epsilon'.$$

Next, we claim that ironing the report of each agent i whose allocations are not distinguished by more than ρ generates a loss of at most $n \cdot (\rho + 2\epsilon')$.

LEMMA 10: $\mathcal{A}' \leq \mathcal{A}'' + n \cdot (\rho + 2\epsilon')$.

PROOF: As in the proof of Theorem 3, the expected costs of $\mathcal{A}'$ and $\mathcal{A}''$ are identical, so it suffices to compare the expected agent welfare. Let S be the set of agents for which $z_i^{(i)} - y_i^{(i)} < \rho$; that is, for which $\sigma_i'' \neq \sigma_i'$. Then

$$\begin{aligned}\mathbf{E}_{\mathbf{v}}[\mathbf{v} \cdot (\mathbf{x} \circ \sigma')(\mathbf{v})] - \mathbf{E}_{\mathbf{v}}[\mathbf{v} \cdot (\mathbf{x} \circ \sigma'')(\mathbf{v})] &\leq \sum_i \max_{v_i} \{ (x_i \circ \sigma'_i)(v_i) - (x_i \circ \sigma''_i)(v_i) \} \\ &= \sum_{i \in S} \max_{v_i} \{ (x_i \circ \sigma'_i)(v_i) - (x_i \circ \sigma''_i)(v_i) \} \\ &\leq \sum_{i \in S} \left[\max_{v_i} \{ x_i(v_i) \} - \min_{v_i} \{ x_i(v_i) \} \right] \\ &\leq \sum_{i \in S} \left[(z_i^{(i)} + \epsilon') - (y_i^{(i)} - \epsilon') \right] \\ &\leq n \cdot (\rho + 2\epsilon')\end{aligned}$$

where the first inequality uses linearity of expectation and the fact that agent values lie in $[0, 1]$, the second equality is the fact that σ'_i and σ''_i differ only for

²Normalize the cost function so that the lowest-cost allocation, say $\mathbf{x}_0$, has $c(\mathbf{x}_0) = 0$ and modify the original algorithm to, on any input that would produce an outcome $\mathbf{x}$ with cost $c(\mathbf{x}) > n$, give outcome $\mathbf{x}_0$. This modification is welfare improving and the modified algorithm has cost in $[0, n]$.

agents in S , and the third is from the definition of σ_i'' . The fourth inequality uses the fact (established in Section B of this online appendix) that the estimated allocation rules can differ from the true allocation rules, in expectation, by at most ϵ' pointwise, and hence the maximum expected allocation can be greater than the observed maximum by at most this much (and similarly for the observed minimum). The final inequality follows from the definition of S . Q.E.D.

Finally, we claim that executing the stair algorithm with probability π does not generate too much loss:

$$(C2) \quad \mathcal{A}'' - \bar{\mathcal{A}} \leq 2n\pi.$$

This is because $\bar{\mathcal{A}}$ uses allocation rule $\mathbf{x} \circ \sigma''$ with probability $1 - \pi$, so the allocations of $\bar{\mathcal{A}}$ and $\mathcal{A}''$ differ with probability at most π . In the event that the allocations differ, the change in welfare is bounded by the difference between n (the maximum possible welfare of any outcome) and $-n$ (the minimum possible welfare, recalling that $\mathcal{A}$ never returns allocations of cost greater than n).

Combining (C1), (C2), and Lemma 10, and choosing $\pi = \frac{\epsilon}{8n}$, $\delta = \frac{\epsilon}{8n}$, $\epsilon' = \frac{\epsilon^3}{8^3 n^4}$, and $\rho = \frac{n\epsilon'}{\delta\pi} = \frac{\epsilon}{8n}$, we conclude that $\bar{\mathcal{A}} \geq \mathcal{A} - \epsilon$ as desired.

D. PAYMENT COMPUTATION

For our ϵ -BIC reduction, described in Section III, it was sufficient to precompute payments from the allocation rule estimated in the construction. Since this estimated allocation rule is close to the actual allocation rule, these payments are approximately correct. Precomputed payments require no additional computation at execution time. On the other hand, the algorithm constructed in Theorem 4 is (exactly) monotone but the functional form of its allocation rules cannot be derived from the estimated allocation rules in its construction.

We now describe two black-box approaches for computing the appropriate payments for an exactly monotone algorithm. The first approach, analogously to the VCG payment scheme, requires an additional n calls to the algorithm.³ This approach is a simplification of one given by Archer et al. (2003). The second approach slightly changes the algorithm so that payments can be calculated implicitly with only a single call to the algorithm. This change to the algorithm results in an additional small loss in the social welfare, but results in a mechanism that can be executed in a linear number of basic operations (a constant number per agent) and a single invocation of the algorithm. This latter approach is an adaptation and simplification of one given by Babaioff, Kleinberg and Slivkins (2010). These approaches can be used to calculate payments for any monotone algorithm $\mathcal{A}$ (including $\bar{\mathcal{A}}$ from our construction) and, therefore, we describe them in general.

³For the remainder of this section, “the algorithm” will refer to a monotone algorithm whose accompanying payments are to be calculated; not to be confused with the arbitrary algorithm $\mathcal{A}$ from the statement of Theorem 4.

The main idea behind the Archer et al. (2003) payment construction is that an unbiased estimator for the payment of an agent can be calculated by an additional call to the algorithm that replaces the agent's reported value v_i with a new uniform value that is at most v_i . In the following discussion, in addition to our standard notation where $x_i(v_i)$ denotes the probability of allocation to agent i with value i , we will denote by x_i the random variable for a realized outcome of algorithm $\mathcal{A}$ on an implicit input $\mathbf{v}$. Thus, $\mathbf{E}_{\mathbf{v}_{-i}}[x_i] = x_i(v_i)$.

DEFINITION 11: *The unbiased payment mechanism for monotone algorithm $\mathcal{A}$ on values $\mathbf{v}$ works as follows:*

- 1) Calculate outcome $\mathbf{x}$ of $\mathcal{A}(\mathbf{v})$.
- 2) For each agent i :
 - a) draw $w_i \sim U[0, v_i]$,
 - b) let z_i be an indicator for agent i being served by $\mathcal{A}(w_i, \mathbf{v}_{-i})$, and
 - c) calculate payment $p_i = v_i(x_i - z_i)$.
- 3) Output $(\mathbf{x}, \mathbf{p})$.

Just like the VCG approach to calculating payments that satisfy the payment identity of Theorem 1, the unbiased payment mechanism makes a total of $n + 1$ calls to the algorithm. The first call is to compute the algorithm's outcome, the remaining n calls are to compute each of the n agents' payments. Like the VCG approach, these payments are ex post individually rational. Agent i 's utility is $v_i x_i - p_i = v_i z_i$ where v_i has been assumed to be non-negative and z_i is in $\{0, 1\}$. While the VCG mechanism has no positive transfers to the agent, the unbiased payment mechanism has no positive transfers only in an interim sense. When $x_i = 0$ and $z_i = 1$ a positive transfer is made to a losing agent; however, monotonicity of $\mathcal{A}$ implies that $\mathbf{E}_{\mathbf{v}_{-i}}[x_i] \geq \mathbf{E}_{\mathbf{v}_{-i}}[z_i]$ so in expectation an agent is not paid for participating.⁴

THEOREM 8: *If $\mathcal{A}$ is monotone for distribution $\mathbf{F}$ then the unbiased payment mechanism for $\mathcal{A}$ is BIC.*

PROOF: Monotonicity is given by assumption; we must therefore only show that for each agent i the payment p_i satisfies, in expectation, the payment identity of Theorem 1, i.e., $\mathbf{E}_{\mathbf{v}_{-i}, \mathcal{A}}[p_i] = v_i x_i(v_i) - \int_0^{v_i} x_i(w) dw$ where $x_i(v_i)$ is the interim allocation rule of algorithm $\mathcal{A}$. The unbiased payment is $v_i x_i - v_i z_i$. The first term is equal to the first term of the payment identity in expectation. For the second term, $\mathbf{E}_{\mathbf{v}_{-i}, \mathcal{A}}[z_i] = \int_0^{v_i} x_i(w_i) (1/v_i) dw_i$ because the density function of the

⁴Of course, for deterministic algorithms that are monotone for all $\mathbf{v}_{-i}$, the unbiased payment satisfies no positive transfers ex post as $x_i \geq z_i$ holds pointwise.

random variable w_i is $1/v_i$. Therefore, $\mathbf{E}_{\mathbf{v}_{-i}, \mathcal{A}}[v_i z_i]$ is equal to the second term of the payment identity. Q.E.D.

If we are willing to slightly modify the algorithm at a small expense to the social welfare, appropriate payments for the modified algorithm can be implicitly calculated from a single call to the original algorithm. As described above, our method of calculating an agent's payment will sometimes input to the algorithm something other than her declared value. We must be careful that this change affects the allocation rule in a controlled manner and does not affect the Bayesian monotonicity or expected welfare of the algorithm for the other agents. We do this by sometimes replacing an agent's value with a random value drawn from the same distribution. The detailed mechanism is given as follows; it is an adaptation of the implicit payment mechanism of Babaioff, Kleinberg and Slivkins (2010).

DEFINITION 12: *The implicit payment mechanism for monotone algorithm $\mathcal{A}$ on values $\mathbf{v}$ works as follows:*

- 1) *Pick an agent k uniformly at random and draw $w_k \sim F_k$.*
- 2) *Calculate outcome $\mathbf{x}'$ of $\mathcal{A}(w_k, \mathbf{v}_{-k})$.*
- 3) *For each agent $i \neq k$, set $p'_i = v_i x'_i$.*
- 4) *For agent k , set $p'_k = -(n-1)x'_k/f_k(w_k)$ if $w_k \leq v_k$ and $p'_k = 0$ otherwise.⁵*
- 5) *Output $(\mathbf{x}', \mathbf{p}')$.*

We give two theorems to show that the implicit payment mechanism is Bayesian incentive compatible and has only a small loss in welfare relative to the algorithm; further discussion of practical aspects of the mechanism is given after these theorems.

THEOREM 9: *If $\mathcal{A}$ is monotone for distribution $\mathbf{F}$ then the implicit payment mechanism for $\mathcal{A}$ is BIC.*

PROOF: Consider an agent i , the allocation rule of the implicit payment mechanism is $x'_i(\cdot)$ while the allocation rule of the original algorithm is $x_i(\cdot)$. The relationship of these two allocation rules can easily be seen to be

$$(D1) \quad x'_i(v_i) = \left(1 - \frac{1}{n}\right) \cdot x_i(v_i) + \frac{1}{n} \cdot \mathbf{E}_{w_i \sim F_i}[x_i(w_i)];$$

i.e., the new allocation rule is multiplicatively scaled and additively offset from the original. Monotonicity of $\mathcal{A}$ implies monotonicity of $\mathcal{A}'$; therefore, we must only argue that the payments satisfy the payment identity.

⁵If we do not have access to the density function $f_k(\cdot)$, we can follow the discretizing approach where valuation space is divided into intervals of width δ within which values are always resampled, and then $f_k(w_k)$ can be replaced with $F_k(\delta\ell + \delta) - F_k(\delta\ell)$ when $w_k \in [\delta\ell, \delta\ell + \delta]$.

Additive offsets in allocation rules do not affect payments, so the requisite payment rule (in expectation) must satisfy

$$\begin{aligned}
 p'_i(v_i) &= \left(1 - \frac{1}{n}\right) \cdot p_i(v_i) \\
 (D2) \quad &= \left(1 - \frac{1}{n}\right) \cdot v_i x_i(v_i) - \left(1 - \frac{1}{n}\right) \cdot \int_0^{v_i} x_i(w) dw
 \end{aligned}$$

where $p_i(\cdot)$ is the payment rule for $x_i(\cdot)$ that follows from the payment identity of Theorem 1. We will show that an agent $i \neq k$ has expected payment equal to the first term in the payment identity of (D2) and the agent k has expected payment equal to the second term of the payment identity (which is negative). Conditioning on $i \neq k$, $\mathbf{E}[x'_i] = x_i(v_i)$. The probability of $i \neq k$ is $1 - 1/n$. Therefore, the expected payment from the case $i \neq k$ is $(1 - 1/n)v_i x_i(v_i)$ as desired for the first term of (D2). Conditioning on $i = k$, the expected payment is $-F_i(v_i) \cdot (n - 1) \cdot \mathbf{E}[x_i(w_i)/f_i(w_i)]$ where the expectation in the last term taken over $w_i \sim F_i[0, v_i]$. For such w_i ,

$$\mathbf{E}_{w_i \sim F_i[0, v_i]}[x_i(w_i)/f_i(w_i)] = \frac{1}{F_i(v_i)} \int_0^{v_i} x_i(w_i) dw_i.$$

The probability of $i = k$ is $1/n$ and therefore this expected payment is equal to the second term of (D2). Q.E.D.

THEOREM 10: *The implicit payment mechanism $\mathcal{A}'$ for any $\mathcal{A}$ and distribution $\mathbf{F}$ has expected welfare $\mathcal{A}' \geq \mathcal{A} - 1$.*

PROOF: The welfare of any agent i is $v_i x'_i(v_i)$ which is at least $(1 - \frac{1}{n}) \cdot v_i x_i(v_i)$ by (D1). Of course, $\mathbf{E}_{v_i}[v_i x_i(v_i)] \leq 1$ by the assumption that $v_i \leq 1$ and the fact that $x_i(v_i)$ is a probability. Therefore, the welfare of agent i is at least $v_i x_i(v_i) - 1/n$ in expectation. Moreover, the expected cost of the allocation is unaffected by the construction. Taking expectations and summing over all agents gives the theorem. Q.E.D.

Some additional discussion of the implicit payment mechanism is warranted. It is easy to see that the implicit payment mechanism is ex post individually rational. The only agents that are required to pay their values are ones that are served by the algorithm; otherwise, payments are zero or negative (a rebate). There may, however, be a large positive transfer (a rebate) made to agent k .

While we suffered an additive loss of one in Theorem 10, this loss can be made arbitrarily small by choosing $k = 0$ (no agent) with some large probability and scaling the payment (rebate) in the case that $i = k$ to offset this probability. Since the agents are risk neutral, the model permits this probability to go to one, the rebate to go to infinity, and, therefore, the performance loss to go to zero. Of course, such an approach is not relevant in practice as it relies too strongly on the assumed linearity of agent utilities.

E. COMPARISON WITH IRONED VIRTUAL VALUATIONS

The ironing procedure described in this paper is reminiscent of the ironing procedure used by Myerson (1981) for maximizing revenue. Myerson's optimal mechanism allocates to maximize virtual value (defined by $\phi_i(v_i) = v_i - \frac{1-F_i(v_i)}{f_i(v_i)}$ for agent i). When the virtual valuation functions are non-monotone, this mechanism first irons the virtual valuation functions, then allocates to maximize ironed virtual value. A natural question is whether this is identical to ironing the non-monotone allocation rule that would result from maximizing non-ironed virtual values. We now give an example demonstrating that these two mechanisms are distinct.

Consider an auction of a single indivisible item to multiple bidders with values drawn independently from distribution F . Suppose $\mathcal{A}$ assigns the item to the agent with highest virtual value. Let $\bar{\mathcal{A}}$ be the ironed algorithm corresponding to $\mathcal{A}$. Let $\mathcal{A}'$ be Myerson's algorithm, which assigns the item to the agent with the highest *ironed* virtual value. Let $x(\cdot)$, $x'(\cdot)$, and $\bar{x}(\cdot)$ denote the allocation rules of $\mathcal{A}$, $\mathcal{A}'$, and $\bar{\mathcal{A}}$, respectively (by symmetry, allocation rules are the same for each player). We claim that, in general, $\bar{x}(\cdot) \neq x'(\cdot)$.

Consider the following distribution F : with probability $1/2$, the value is drawn uniformly from $[10, 11]$; otherwise, it is drawn uniformly from $[11, 15]$. The virtual valuation function corresponding to this distribution is

$$\phi(v) = \begin{cases} 2v - 12 & v \in [10, 11] \\ 2v - 15 & v \in (11, 15]. \end{cases}$$

Let $\phi'(\cdot)$ be the ironed function corresponding to $\phi(\cdot)$. It can be verified that $\phi'(10) < \phi'(v)$ for all $v \in (10, 15]$. This implies that $x'(10) = 0$.

On the other hand, $x(10) > 0$ (since $\phi(v) < \phi(10)$ for $v \in (11, 11.5)$), and furthermore $x(v) \geq 0$ for all $v \in [10, 15]$. Thus, from the definition of the ironed algorithm $\bar{\mathcal{A}}$, there is some $z \in [10, 15]$ such that $\bar{x}(10) = \mathbf{E}_v[x(v) \mid v \leq z] > 0$. We therefore conclude $x'(\cdot) \neq \bar{x}(\cdot)$, as claimed.

F. EQUILIBRIA OF NON-MONOTONE ALGORITHMS

We have shown how to transform an arbitrary algorithm into an approximation mechanism that is BIC, without loss to the social welfare. We assert that this intervention is necessary, in that the agents cannot, in general, implement such an outcome as an equilibrium of the original algorithm. In other words, there are non-monotone algorithms that, when coupled with a payment rule that is individually-rational (IR) and has no positive transfers (NPT),⁶ do not have any BNE with expected welfare close to the expected performance of the original

⁶IR and NPT together imply that losers have zero payment.

algorithm (on the true valuation profile).⁷ We now give an example illustrating this phenomenon.

Consider an auction of a single indivisible item to n bidders with values drawn independently from the following distribution, F . With probability $1 - 1/n$ the value is drawn uniformly from $[0, \frac{1}{n^2}]$, with the remaining probability the value is drawn uniformly from $[\frac{1}{n^2}, 1]$. Let $\mathcal{A}$ be the algorithm that allocates to the bidder with the largest value in $[0, \frac{1}{n^4}]$, if any; otherwise it allocates to the bidder with the largest value.

Consider the expected welfare of $\mathcal{A}$. With probability $1 - (1 - \frac{1}{n})^n > 1 - \frac{1}{e}$ at least one player has value drawn from $[\frac{1}{n^2}, 1]$, and with probability at least $n \cdot \frac{1}{n^2}$ no agent has value in $[0, \frac{1}{n^4}]$. Thus, with probability at least $(1 - \frac{1}{e} - \frac{1}{n})$, $\mathcal{A}$ will allocate to a player with value drawn from $[\frac{1}{n^2}, 1]$, and hence the expected welfare of $\mathcal{A}$ is greater than $\frac{1}{2}(1 - \frac{1}{e} - \frac{1}{n}) > \frac{1}{4}$.

Consider now any BNE of $\mathcal{A}$, under any payment scheme that is ex post IR and satisfies NPT. First note that, under this equilibrium, each agent declares value in $[0, \frac{1}{n^4}]$ with probability 1; this is because any declared value greater than $\frac{1}{n^4}$ is strictly dominated by a declaration of 0, which yields a higher allocation probability and payment 0. We can therefore conclude that, in this equilibrium, the winner will pay at most $\frac{1}{n^4}$ with probability 1.

Write x_i for the interim allocation rule for agent i under this BNE. We will now argue that the bound on payments will imply that $x_i(1)$ is small for many agents, which will lead us to conclude that the expected social welfare at this equilibrium is low. We have from Theorem 1 that x_i is monotone non-decreasing, and moreover

$$\frac{1}{n^4} \geq x_i(1) - \int_0^1 x_i(v) dv$$

since $p_i(1) \leq \frac{1}{n^4}$. But since x_i is non-decreasing, the right-hand side is at least $\frac{1}{2n^2}(x_i(1) - x_i(\frac{1}{2n^2}))$. This implies in particular that

$$x_i(1) \leq x_i\left(\frac{2}{n^2}\right) + \frac{2}{n^2}.$$

Since each player has value at least $\frac{1}{2n^2}$ with probability greater than $1/2$, and only a single item is sold on any valuation profile realization, it must be that

$$\sum_i \frac{1}{2} x_i\left(\frac{2}{n^2}\right) < 1$$

and we can therefore conclude that

$$\sum_i x_i(1) \leq \sum_i \left(x_i\left(\frac{2}{n^2}\right) + \frac{2}{n^2}\right) < 2 + \frac{2}{n} < 3.$$

⁷The IR restriction is motivated by our interest in outcomes that arise without the intervention of a mechanism designer. We discuss this restriction further in Section G of this online appendix.

This leads us to the following loose upper bound on the expected social welfare of this equilibrium:

$$\sum_i \left(\frac{1}{n^2} + \Pr \left[v_i > \frac{1}{n^2} \right] \cdot 1 \cdot x_i(1) \right) < \frac{1}{n} + \frac{3}{n} = \frac{4}{n}.$$

Thus, while $\mathcal{A}$ has expected welfare greater than $\frac{1}{4}$, the expected welfare at any BNE is at most $\frac{4}{n}$. In other words, any BNE of the mechanism that results if we do not modify the allocation rule of $\mathcal{A}$ extracts at most a linear fraction of the social welfare of the original algorithm.

G. MATCHING-BASED TRANSFORMATIONS AND INDIVIDUAL RATIONALITY OF PAYMENTS

The main result of the paper is a transformation that maps reported values to inputs to a non-monotone algorithm so that the resulting allocation rules are monotone. While the transformation described in this paper is welfare-optimal among interval resampling transformations, it is not welfare-optimal among all stationary transformations. The welfare-optimal transformation, described in the follow-up works of Hartline, Kleinberg and Malekian (2015) and Bei and Huang (2011), can be obtained by matching values of an agent to outcomes of the algorithm to maximize the cumulative welfare of the matching. In our single-dimensional environment this matching is easy to interpret: sort the outcomes of the algorithm by expected service level, sort the possible values of the agent, and assign values to outcomes in this sorted order.

In Section F of this online appendix, we argued that an intervention by a mechanism designer is necessary to guarantee good outcomes for a given non-monotone algorithm. In particular, there is a non-monotone algorithm with high welfare for which any “natural” (formally: individually rational) payment rule has only low-welfare equilibria. However, as we will show, the matching transformation described above can be implemented by specifying a particular (non-individually-rational) payment rule, instead of directly altering the allocation rule. With this payment rule, there is a Bayes-Nash equilibrium where the agents implement the transformation on their own, i.e., for transformation σ_i , value v_i , and allocation rule x_i , agent i will have an equilibrium bid of $\sigma_i(v)$ and the equilibrium allocation rule will be $x_i^{\sigma_i}(\cdot) = x_i(\sigma_i(\cdot))$. Arguably, the approach of transforming the allocation rule is more direct than this payment-rule-based approach.

We will illustrate this observation for the matching-based transformation from Hartline, Kleinberg and Malekian (2015) in the case that the agents’ values are uniformly distributed on a finite set of distinct values; the construction for the continuous case is analogous. For any agent, calculate the payment $p(v)$ that the algorithm should assign to v as follows. Sort the agent’s possible values, sort the (interim) service probabilities given by the algorithm for each of these values, and let x^* be the allocation that assigns these values to these outcomes in their sorted order. Denote the resampling transformation that would implement this

assignment of values to allocation probabilities in this sorted order by σ , i.e., satisfying $x^*(v) = x(\sigma(v))$. Note that σ is one-to-one. Calculate the payment $p^*(v)$ for v from the payment identity for $x^*(v)$. Now, define the payment rule for $x(\cdot)$ as $p(v) = p^*(\sigma^{-1}(v))$. It is an equilibrium for the agent to bid $\sigma(v)$. For such a report, the allocation is $x(\sigma(v)) = x^*(v)$ (which is monotone) and the payment is $p(\sigma(v)) = p^*(v)$ (which satisfies the payment identity by definition of p^*).

Notice that it is possible that $p(v) > v$ in this construction and therefore the payment rule is not individually rational. It is for this reason that there is no contradiction between this result and the example presented in Section F of this online appendix.

REFERENCES

- Archer, Aaron, Christos Papadimitriou, Kunal Talwar, and Éva Tardos.** 2003. “An Approximate Truthful Mechanism for Combinatorial Auctions with Single Parameter Agents.” In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 205–214. Society for Industrial and Applied Mathematics.
- Babaioff, Moshe, Robert Kleinberg, and Alex Slivkins.** 2010. “Truthful Mechanisms with Implicit Payment Computation.” In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 43–52. ACM.
- Bei, Xiaohui, and Zhiyi Huang.** 2011. “Bayesian incentive compatibility via fractional assignments.” In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, 720–733, SIAM.
- Hartline, Jason, Robert Kleinberg, and Azarakhsh Malekian.** 2015. “Bayesian incentive compatibility via matchings.” *Games and Economic Behavior*. In press.
- Myerson, Roger.** 1981. “Optimal Auction Design.” *Mathematics of Operations Research*, 6: 58–73.