The Communication Complexity of Payment Computation

Shahar Dobzinski
Weizmann Institute of Science
Israel
shahar.dobzinski@weizmann.ac.il

Shiri Ron
Weizmann Institute of Science
Israel
shiriron@weizmann.ac.il

ABSTRACT
Let \((f, P)\) be an incentive compatible mechanism where \(f\) is the social choice function and \(P\) is the payment function. In many important settings, \(f\) uniquely determines \(P\) (up to a constant) and therefore a common approach is to focus on the design of \(f\) and neglect the role of the payment function.

Fadel and Segal [JET, 2009] question this approach by taking the lenses of communication complexity: can it be that the communication complexity of an incentive compatible mechanism that implements \(f\) (that is, computes both the output and the payments) is much larger than the communication complexity of computing the output? I.e., can it be that \(cc_{IC}(f) >> cc(f)\)?

Fadel and Segal show that for every \(f\), \(cc_{IC}(f) \leq \exp(cc(f))\). They also show that fully computing the incentive compatible mechanism is strictly harder than computing only the output: there exists a social choice function \(f\) such that \(cc_{IC}(f) = cc(f) + 1\). In a follow-up work, Babaioff, Blumrosen, Naor, and Schapira [EC’08] provide a social choice function \(f\) such that \(cc_{IC}(f) = \Theta(n \cdot cc(f))\), where \(n\) is the number of players. The question of whether the exponential upper bound of Fadel and Segal is tight remained wide open.

In this paper we solve this question by explicitly providing a function \(f\) such that \(cc_{IC}(f) = \exp(cc(f))\). In fact, we establish this via two very different proofs.

In contrast, we show that if the players are risk-neutral and we can compromise on a randomized truthful-in-expectation implementation (and not on deterministic ex-post implementation) then gives that \(cc_{TE}(f) = poly(n, cc(f))\) for every function \(f\), as long as the domain of \(f\) is single parameter or a convex multi-parameter domain.

We also provide efficient algorithms for deterministic computation of payments in several important domains.

CCS CONCEPTS
• Theory of computation → Communication complexity.

KEYWORDS
Communication Complexity, Payment Computation, Mechanism Design

1 INTRODUCTION
In a mechanism design problem we have \(n\) players and a set \(A\) of alternatives. Each player \(i\) has a valuation function \(v_i : A \rightarrow \mathbb{R}\) that specifies his value for each alternative. We assume that each \(v_i\) belongs to some known set \(V_i\). A basic question in mechanism design asks: given a social choice function \(f : V_1 \times \cdots \times V_n \rightarrow A\), are there payment functions \(P_1, \ldots, P_n : V_1 \times \cdots \times V_n \rightarrow \mathbb{R}\) that make \(f\) incentive compatible? For now, we interpret incentive compatibility as truthful, ex-post implementation of \(f\), that is: \(P_1, \ldots, P_n\) satisfy that for every player \(i\), \(v_i, v_i' \in V_i\), and \(v_{-i}\) that specifies the values of the other players, \(v_i(f(v_i, v_{-i})) = P_i(v_i, v_{-i})\geq v_i(f(v_i', v_{-i})) - P_i(v_i', v_{-i})\).

A highly successful paradigm in mechanism design is the "prices do not matter" paradigm. One pillar of this approach is various characterization theorems that provide relatively simple conditions for the implementability of social choice functions. Examples for such conditions include cycle monotonicity for all functions [15], monotonicity for functions in "single parameter" domains [2, 13], and weak monotonicity for "rich enough" multi-parameter domains [6]. Another pillar are "uniqueness of payments" or "revenue equivalence" theorems. Those theorems state that in most domains if \(P_1, \ldots, P_n\) and \(P_1', \ldots, P_n'\) are two possible payment functions for \(f\), then for each \(i\) and \(v_{-i}\) there exists a constant \(\epsilon\) such that \(P_i(v_i, v_{-i}) - P_i'(v_i, v_{-i}) = \epsilon\) (see, e.g., [14]). The combination of the two pillars justifies the focus on the social choice function: given \(f\), one can easily determine whether it is implementable, and if so, the prices are (almost) unique.

Fadel and Segal [9] were the first to make the important observation that this paradigm breaks when computational considerations are taken into account. In other words, if computing the alternative chosen by \(f\) is computationally "easy", can it be that determining how much each player has to pay is much harder? Note that the characterization theorems discussed above guarantee the existence of "good" payment functions, but they do not guarantee an efficient way to actually compute the prices.

Specifically, Fadel and Segal consider an implementable social choice function \(f\) with communication complexity \(cc(f)\). Denote by \(cc_{IC}(f)\) the communication complexity of the implementation of \(f\). The implementation of \(f\) must output both the chosen alternative and the payments, so clearly \(cc_{IC}(f) \geq cc(f)\). But can it be that computing the prices \(P_1(\cdot), \ldots, P_n(\cdot)\) makes the computational task much harder, that is \(cc_{IC}(f) >> cc(f)\)?
Fadel and Segal showed that the gap is at most exponential: 
\[ cc(f) \leq 2^{cc(f)} - 1. \] They also showed that the inequality \( cc(f) \geq cc(f) \) is strict by providing a specific function \( f \) for which \( cc(f) = cc(f) + 1. \) Although they were able to show that for bayesian implementations the gap indeed might be exponential, determining whether it is exponential for the basic setting of ex-post implementations was left as their main open question.

Babaioff, Blumrosen, Naor, and Schapira [4] managed to narrow the gap and prove that for every \( n \), there exists a function \( f \) for \( n \) players for which \( cc(f) \geq n \cdot cc(f). \) They also provided several single-parameter domains for which the gap is small: for every \( f \) in these domains, \( cc(f) = O(cc(f)). \)

### 1.1 Our Results I: Impossibilities

Our first main result (Section 3) answers the open questions of [4, 9] by showing that computing the payments might be significantly harder than computing the output:

**Theorem:** For every \( k \), there exists a function \( f \) for two players (or more, by adding players that do not affect the outcome) for which \( cc(f) = O(k) \) and \( cc(f) = \exp(k) \). Therefore, \( cc(f) = \exp(\log(\exp(k))). \)

In fact, the function \( f \) that we provide is simple enough in the sense that it is single parameter. We note that a similar result was obtained concurrently and independently by [16] (but their function is not single parameter). Roughly speaking, we construct a function \( f: \mathcal{V}_A \times \mathcal{V}_B \to \mathcal{A} \), where \( \mathcal{A} = \{a_0, \ldots, a_k\} \). The domain of valuations is single parameter, and Alice’s private information is \( r_A \in [0, 2^{k+1} - 1] \). For each alternative \( a_i \in \mathcal{A} \), let \( w_A(a_i) = \lceil \mathcal{A} \rceil i^k - 1. \) Alice’s value for alternative \( a_i \) is \( r_A \cdot w_A(a_i) \). Bob is also a single parameter player but his valuation takes a simpler form: he is indifferent to the alternative chosen and his private information \( r_B \) is also his value of each alternative. However, the number of possible values that \( r_B \) can take is doubly exponential in \( k \). The function \( f \) itself is defined by some arbitrary map that takes each possible value of \( r_B \) and projects it to a different partition of the possible values of Alice to the \( \lceil \mathcal{A} \rceil \) alternatives, making sure that each such partitioning is monotone: if \( r' \geq r \), then \( r \) is not mapped to a lower alternative than \( r' \). The function \( f \) takes the value of Alice and outputs the alternative that it belongs to according to the monotone map that is determined by Bob’s value.

Computing \( f \) is easy: Alice can send her private information \( r_A \) \((k + 2 \text{ bits})\) and Bob can then compute the output of \( f \) and announce it \((k + 1 \text{ bits})\). How about computing the payments? Bob is always indifferent to the chosen alternative, so his payment is always 0. Computing the payment of Alice is a bit more subtle.

---

1. In fact they write: "We stress that achieving a better lower bound than the linear lower bound shown in this paper may be hard. The communication cost is known to be at most linear (in the number of players) for welfare-maximization objectives and in single-parameter domains (in FS)." However, their interpretation of the results of Fadel and Segal is not accurate, since as mentioned by Fadel and Segal, their results assume that the type space is sufficiently small. As will be discussed later, we will be able to improve over this lower bound both for welfare maximization and for single parameter domains.

2. We stress that we show all truthful mechanisms for \( f \) require at least \( \exp(\log(cc(f))) \) bits, whereas the linear lower bound of [4] applies only to the the normalized mechanism of \( f \).

3. Recall that by Myerson’s formula the payment of Alice is given by \( P_A(v_A, v_B) = r_A \cdot w_A(f(v_A, v_B)) - \int_0^{r_A} w_A(f(z, w_A, v_B))dz \), where \( v_A \) and \( v_B \) are respectively the valuations of Alice and Bob. Thus, the problem of computing the payments reduces to computing the integral in the formula. The crux of the proof is showing that even if we know that the outcome is \( a_k \), the value of the integral is different for each map (this is why Alice’s value for an alternative is obtained by multiplying her private information \( r_A \) by a fast increasing function \( w_A \)). Since each \( r_B \) of Bob defines a different map and hence a different payment, the number of distinct prices for the alternative \( a_k \) is doubly exponential. Standard arguments imply that at least \( 2^k \) bits are required to specify the payments, which completes the proof.

We note that \( f \) is not a very natural function, but we can build on it to show that sometimes even welfare maximization can be hard: there exists a multi-unit auction such that computing the welfare maximizing solution requires \( k \) bits, but computing the payments requires \( \exp(k) \) bits. This result has one additional interesting feature: it provides an example of an auction domain where the approximation ratio to the social welfare achievable by non-truthful algorithms that use polynomial communication (in our case, the approximation ratio is 1) is strictly better than the approximation ratio that can be achieved by truthful mechanisms that use only polynomial communication (in our case, we show that exponential communication is needed for a truthful mechanism to achieve an approximation ratio of 1). This is only the second such example, following [3] (other separations exist but in non-auction domains or in auctions with restrictions). Unlike all previous separations, in which the hardness is based on the hardness of computing the allocation, here computing the allocation is easy so the hardness stems from the additional overhead of computing the prices.

Quite remarkably, the function \( f \) demonstrates that even if computing the output requires only \( k \) bits, the number of possible payments in the truthful implementation might be as large as \( \exp(\exp(k)) \). In fact, one can see that the possible number of distinct prices was the decisive factor in determining the communication complexity of a mechanism for \( f \). This is no coincidence. Denote by \( P_f \) the maximum number of possible payments for a single alternative that any player might face. Then, the communication complexity of truthfully implementing an implementable function \( f \) for two players can be determined up to a constant multiplicative factor:

\[ cc(f) + \log P_f \leq cc(f) \leq cc(f) + 2\log P_f \]

The left inequality holds since obviously \( cc(f) \geq cc(f) \) and since \( cc(f) \geq \log P_f \), because \( \log P_f \) bits are needed to specify which price the player has to pay out of the possible \( P_f \) prices. The right inequality holds since we can use \( cc(f) \) bits to compute the output of \( f \), and then each of the two players uses (at most) \( \log P_f \) bits to specify the price of the other player (recall that by the taxation principle, the price of an alternative for a player depends only on the valuations of the other players).

We thus have that for any two-player function \( cc(f) = \text{poly}(cc(f), \log P_f) \). Note that this characterization is tight in the sense that it is easy to come up with examples where
We proceed with developing algorithms for payment computation. For every $k$ (or more, by adding players that do not affect the outcome) and a bit is just the disjointness function in disguise, which implies that $1$ is significantly harder than computing $f$ alone.

**Theorem:** For every $k$, there exists a function $f$ for three players (or more, by adding players that do not affect the outcome) and a payment scheme for which this (and all other) payment schemes for implementing $f$ is polynomially harder to compute than computing $f$ alone.

The proof is very different than the previous proof. Rather than basing the hardness on the number of parameters, the hardness stems from carefully constructing the function so that determining the prices for Alice requires to decide whether the bit representations of the types of Bob and Charlie share a common 1 bit, whereas computing $f$ requires to decide whether a single specific bit intersects. Of course, determining whether Bob and Charlie share a common 1 bit is just the disjointness function in disguise, which implies that computing the payments is indeed hard.

### 1.2 Our Results II: Algorithms for Payment Computation

We proceed with developing algorithms for payment computation. Our algorithms come in three different flavours: truthful in expectation implementations of deterministic social choice functions, deterministic implementations of single parameter functions, and deterministic implementations of multi-parameter functions that satisfy uniqueness of payments.

We observe that if $f$ is an implementable social choice function, then although $cc_{IC}(f)$ might be exponential in $cc(f)$, if we compromise on truthful in expectation implementation, the payment might be computed very efficiently. In fact, if we let $cc_{TE}(f)$ be the communication complexity of implementing $f$ as a truthful in expectation mechanism, we prove that $cc_{TE}(f) = poly(n, cc(f))$ for single parameter domains and for multi-parameter domains that are convex. For single parameter domains, we rely on the (well known) observation that the expected value of the integral in Myerson’s formula can be estimated by measuring the height of the integral at a random point. For convex multi-parameter domains, we develop another algorithm relying on a characterization of the payments in scalable domains by Babaioff, Kleinberg, and Slivkins [5], by showing that convex domains are essentially scalable.

In the rest of the paper, we return to consider the fundamental question of payment computation in (deterministic) ex-post equilibrium. Babaioff et al. [4] consider several simple single parameter problems. All of their problems are binary: for each player $i$, the set of alternatives is divided into a set of “winning” alternatives for which his value is his private information $r_i$, and a set of “losing” alternatives for which his value is 0. Babaioff et al. [4] provide algorithms for payment computation for some specific settings. Our first algorithm provides a general polynomial upper bound for all binary problems:

**Theorem:** Let $f$ be an implementable social choice function for $n$ players in a binary single parameter domain. Then, $cc_{IC}(f) \leq O(n \cdot cc^2(f))$.

In fact, our algorithm extends to a much more general single parameter setting that may have many alternatives, like the setting of our impossibility result discussed above. In this case we show that $cc_{IC}(f) \leq O(n \cdot cc^2(f) \cdot |\mathcal{A}|)$, where $|\mathcal{A}|$ is the set of alternatives. This bound is tight in the sense that if we omit $cc(f)$ or $|\mathcal{A}|$ from the RHS, $cc_{IC}(f)$ might become much bigger than the RHS.

We then proceed to considering multi-parameter settings. These turn out to be quite challenging. However, we do provide an algorithm for those domains as well, assuming “uniqueness of payments”, i.e., that the payment functions are uniquely determined by the allocation function (up to a constant). Most interesting domains (combinatorial auctions, scheduling, etc.) satisfy uniqueness of payments.

**Theorem:** Let $f$ be an implementable social choice function for $n$ players that satisfies uniqueness of payments. Then, $cc_{IC}(f) \leq poly(n, cc(f), |\mathcal{A}|)$.

To prove this theorem, we first prove that there exists a non-deterministic algorithm that computes the payment of player $i$. We then leverage this result and the fact that non-deterministic and deterministic communication complexity are polynomially related to establish our upper bound. We show that for every player $i$ and every price, the prover can send $O(|\mathcal{A}|^2)$ types in $V_i$ that serve as a non-deterministic witness. The proof of the theorem consists in explicitly describing those types, showing that they suffice and that they can be described succinctly.

### 1.3 Our Results III: The Hardness of Computing the Payments in a Menu

We now change gears and consider a slightly different but very related problem. Up until now we assumed that we are given an instance $(\alpha_1, \ldots, \alpha_n)$ and we want to compute the payment of each player. However, the taxation principle asserts that each truthful mechanism can be seen as follows: each player $i$ faces a menu that specifies a price for each alternative. The output of a truthful mechanism is an alternative that maximizes the profit, i.e., maximizes $value(a) - price(a)$ for each player. The taxation principle leads to a definition of taxation complexity, which was shown to characterize the communication complexity of truthful mechanisms in many settings [7]. The notion of taxation complexity was crucial in establishing a lower bound on the communication complexity of truthful approximation mechanisms in the recent breakthrough of Assadi et al. [3].

Consider the notion of a “constructive taxation principle” or “menu reconstruction” [7]: an algorithm that efficiently finds the menu that $a \cdot i$ presents to player $i$. The basic building block of this algorithm is a subroutine $price(\cdot)$ that assumes that the input of

\[Recall that, roughy speaking, in a truthful in expectation mechanism each player has a strategy that maximizes his expected profit regardless of the strategies of the other players, where the expectation is taken over the random coins of the mechanism.\[The taxation complexity of a mechanism is $log(\max_i |M_i|)$, where $M_i$ is the set of possible menus player $i$ might face.\]
each player \( i' \neq i \) is \( v_{i'} \), gets an alternative \( a \) and returns the price of \( a \) in the menu induced by the truthful mechanism \( M \). We have efficient and constructive taxation principle whenever \( cc(price) = poly(cc(M)) \).

We pinpoint the hardness of \( price() \) on deciding whether an alternative \( a \) is reachable, i.e., whether there exists \( v \) such that \( f(v, v_{i'}) = a \). We denote this function with \( reach() \). We show that if \( reach() \) is easy, \( price() \) is also easy, i.e.: \( cc(price) \leq poly(cc(reach), cc(M), n) \). We use this observation to show that for all the mechanisms of player decisional functions, \( cc(price) \leq poly(n, cc(M)) \). Furthermore, we show an instance \( M = (f, P) \) where \( cc(reach) = exp(cc(M)) \) and prove that this gap is tight.

Due to lack of space, we exclude those results in this paper, and they can be found in the full version.

2 PRELIMINARIES

Truthfulness. We consider settings with \( n \) players. Each player \( i \) has a valuation function \( v_i : \mathcal{A} \rightarrow \mathbb{R} \) which is his private information. Let \( V_i \) be the set of all possible valuations of player \( i \). A mechanism \( M \) consists of a social choice function \( f : V_1 \times \cdots \times V_n \rightarrow \mathcal{A}, \) where \( \mathcal{A} \) is the set of possible alternatives, and a payment function \( P_i : V_1 \times \cdots \times V_n \rightarrow \mathbb{R} \) for each player \( i \). A mechanism is ex-post incentive compatible (or truthful) if for each player \( i \), every valuations profile of the other players \( v_{\neq i} \in V_{\neq i} \) and every \( v_i, v'_i \in V_i \), it holds that:

\[
v_i(f(v_i, v_{\neq i})) - P_i(f(v_i, v_{\neq i})) \geq v_i(f(v'_i, v_{\neq i})) - P_i(f(v'_i, v_{\neq i}))
\]

is called implementable if for some \( P_1, \ldots, P_n \) the resulting mechanism is ex-post incentive compatible. We say that a truthful mechanism \( M = (f, \mathcal{P}) \) implements the social choice function \( f \). Throughout the paper, we denote the image of a payment function \( P_i \) with \( Im(P_i) \).

In this paper we give special treatment to single parameter domains. A domain of a player \( V_i \) is single parameter if there exists a public function \( w_i : \mathcal{A} \rightarrow \mathbb{R} \) and a set of real numbers \( R_i \subseteq \mathbb{R} \) such that \( V_i = \{ r \cdot w_i(\cdot) \mid r \in R_i \} \). A single parameter domain \( V_i \) is binary if its public function \( w_i \) satisfies that its range, \( Im(w_i) \), is equal to \( \{0, 1\} \). If \( V_1, \ldots, V_n \) are all single parameter domains, we say that \( f : V_1 \times \cdots \times V_n \rightarrow \mathcal{A} \) is single parameter. In particular, since we can assume that the private information of player \( i \) is \( r_{i0} \), we often identify \( v_i \) with \( r_{i0} \) and slightly abuse notation by writing, e.g., \( v_i > v'_i \) where we mean \( r_{i0} > r_{i0}' \).

A social choice function \( f \) is monotone with respect to player \( i \), if \( V_i \) is a single parameter domain and for every \( v_{\neq i} \in V_{\neq i} \):

\[
r_{i0} > r_{i0}' \implies w_i(f(v'_i, v_{\neq i})) \geq w_i(f(v_i, v_{\neq i}))
\]

is monotone if it is monotone with respect to each of its players.

Let \( M = (f, \mathcal{P}) \) be a mechanism over a domain, where each \( V_i \) is single parameter and \( 0 \in R_i \). \( M \) is normalized if for each player \( i \) and every \( v_{\neq i} \in V_{\neq i}, r_{i0} = 0 \implies P_i(v_i, v_{\neq i}) = 0 \).

The following proposition is well known [13]:

**Proposition 2.1 (Monotonicity and Myerson’s Payment Formula).** Let \( V = V_1 \times \cdots \times V_n \) be a single parameter domain. Then, a social choice function \( f : V \rightarrow \mathcal{A} \) is implementable if and only if it is monotone. If \( R_i = [0, b_i] \), then the unique payment rule of player \( i \) that satisfies normalization is given by:

\[
P_i(v_i, v_{\neq i}) = b_i \cdot w_i(f(v_i, v_{\neq i})) - \int_0^{v_i} w_i(f(z, v_{i0}, v_{\neq i})) dz
\]  

(1)

Communication Complexity. In this paper communication complexity refers to the number-in-hand model where each player \( i \) denotes \( Im(\cdot) \) to denote the image of a function.

3 THE COST OF PAYMENT COMPUTATION IS EXPONENTIAL

Recall that Fadel and Segal [9] showed that for every social choice function \( f \) we have that \( cc_{IC}(f) \leq exp(cc(f)) \). In this section we solve their main open question and show that their bound is tight, that is, there exists a social choice function \( f \) such that \( cc_{IC}(f) = exp(cc(f)) \). In fact, we provide two proofs by constructing two social choice functions, each function highlights a different source of hardness of payment computation.

The first source of hardness is the fact that the number of prices that a player might see in an implementation of a social choice function \( f \) with \( cc(f) = O(k) \) is doubly exponential in \( k \). Therefore, just specifying the payments requires \( exp(k) \) bits, which immediately implies that \( cc_{IC}(f_k) = exp(k) \).

If there are only two players, we show that this is the only source of hardness in the sense that payment computation becomes easy when the number of payments is not huge. However, when there are more than two players we show that even when the number of payments is small, payment computation might be hard because of the interaction between the players.

One possible criticism about those results is that the functions that we construct are quite contrived. Thus, we conclude by showing a welfare maximizer in a multi-unit auction that satisfies that \( cc_{IC}(f) = exp(cc(f)) \).

3.1 Proof I: Hardness via the Number of Payments

In the two player case, we are able to fully characterize the relationship between \( cc_{IC}(f) \) and \( cc(f) \). For every implementable social choice function \( f \), let \( P : V_1 \times \cdots \times V_n \rightarrow \mathbb{R}^n \) be the payment scheme of the most efficient mechanism for \( f \), i.e. the one that satisfies \( cc(f, P) = cc_{IC}(f) \). Let \( P_f \) be the maximum number of prices

\[ b_i \text{ might be equal to } \infty. \]

\[ The \ definition \ of [1] \ is slightly different: \ there, \ cc_{IC}(f) \ is \ the \ cost \ of \ the \ most \ efficient \ normalized \ mechanism \ for \ it. \]
for an alternative when using $P$. Formally:

$$P_f = \max_{i \in [N]} \max_{a \in A_i} |\{p \mid \exists v \in S \text{ s.t. } f(a) = a, P_i(v) = p\}|$$

**Proposition 3.1.** Let $f$ be an implementable social choice function for two players. Then:

$$\frac{cc(f) + \log(P_f)}{2} \leq cc_{IC}(f) \leq cc(f) + 2 \log(P_f) \quad (2)$$

**Proof.** Obviously, $cc_{IC}(f) \geq cc(f)$. Also, $cc_{IC}(f) \geq \log(P_f)$, because $cc_{IC}(f) \geq \log |\mathcal{M}(f, P)| \geq \log(P_f)$. For the RHS, denote the payment functions of players 1 and 2 with $P_1$ and with $P_2$, respectively. By the taxation principle, $P_1$ can be reformulated as a function of the alternative chosen and of $V_2$, and analogously $P_2$ is a function of the alternative chosen and of $V_1$.

We use this reformulation to explicitly provide a protocol for a truthful implementation for $f$. Let $\pi$ be the most efficient communication protocol of $f$. Fix types $v_1 \in V_1, v_2 \in V_2$. The players first run $\pi(v_1, v_2)$, so they both know $f(v_1, v_2) = a$. By the above, once the alternative is known, player 1 knows $P_1(a, v_1)$ and player 2 knows $P_2(a, v_2)$. The players can now send to each other those payments, using at most $2 \log(P_f)$ bits.

As a direct corollary, to prove an exponential gap between $cc(f)$ and $cc_{IC}(f)$ when there are only two players, we must construct a function $f$ in which the number of possible payments $P_f$ is doubly exponential in $cc(f)$. We now construct such a function $f$, which gives us the first proof of our main result. We note that the $f$ that we construct is quite contrived. However, in Section 3.3 we use the same $f$ to prove that payment computation is hard even if we want to maximize the welfare in a multi-unit auction setting.

Fix some integer $k \geq 1$. In our setting there are two players, Alice and Bob, and $2^k + 1$ alternatives: $\mathcal{A} = \{a_0, a_1, \ldots, a_{2^k}\}$. The domains of Alice and of Bob are single parameter: $r_A \in [0, 2^k + 1]$ and $r_B \in [0, 1, \ldots, l-1]$, where $l = (2^k-1)$. Notice that the domain of Bob’s valuations, $V_B$, is of size $l$. The value of Alice for alternative $a_i$ is $r_A \cdot |\mathcal{A}^{[2k+1]} - 1|$, i.e., $w_A(a_i) = |\mathcal{A}^{[2k+1]} - 1|$. Bob’s value for all alternatives is identical and equal to his private information $r_B$ ($w_B \equiv 1$). Let $C$ be the following set of $(2^k + 1)$-dimensional vectors:

$$C = \left\{(c_0, \ldots, c_{|\mathcal{A}|-1}) \mid \forall i, c_i \geq 1 \text{ and } c_i \in \mathbb{N}, \sum_{i=0}^{|\mathcal{A}|-1} c_i = 2^{k+1}\right\} \quad (3)$$

Each vector $c \in C$ uniquely defines a function $\chi_c : \{0, 1, \ldots, 2^k + 1\} \to \mathcal{A}$, where $c_i$ is the number of integers that $\chi_c$ assigns to alternative $a_i$. For example, the function $\chi_c$ that corresponds to the vector $c = (c_0, \ldots, c_{|\mathcal{A}|-1})$ maps the integers in $\{0, 1, \ldots, 2^k + 1\}$ to $a_0$, the integers in $\{0, c_0 + 1, \ldots, c_0 + c_1 - 1\}$ to $a_1$ and so on. For every $c \in C$, $\chi_c$ is monotonically increasing in the sense that it maps larger integers to alternatives with smaller index. We define a set of functions which corresponds to the set of vectors: $X = \{\chi_c | c \in C\}$. Note that:

$$|X| = |C| = \binom{2^{k+1}}{\mathcal{A} + |\mathcal{A}|-1} = \frac{(2^{k+1}-1)}{2^k} = |V_B| \quad (4)$$

Since this is the number of ways to match the the integers in $\{0, 1, \ldots, 2^k + 1\}$ to alternatives in a monotone way, which is compatible with the constraint that each alternative is matched with at least one integer, i.e. that for all $i, c_i \geq 1$. It follows that there exists a bijective function between $V_B$ and the set $X$. Let $map : V_B \to X$ be such bijection. We define $f_k(a_B, v_B) = map(v_B)(\chi_{c_k})$. In words, computing $f_k(a_B, v_B)$ is done by first computing $map(v_B)$ which returns a function $\chi_{c_k}$ in $X$, which corresponds to the vector $c_k$. Afterwards, we apply $\chi_{c_k}$ to the integer $\lfloor r_{a_k} \rfloor$, which outputs an alternative.

**Theorem 3.2.** For the $f_k$ defined above, $cc(f_k) = O(k)$, whereas $cc_{IC}(f_k) \geq \exp(k)$.

**Proof.** Observe that $cc(f_k) = O(k)$ since $f_k$ can be computed by a simple protocol where Alice sends to Bob $\lfloor r_{a_k} \rfloor$ using $k + 2$ bits, and then Bob computes $f_k$ by applying $\chi = map(v_B)$ to $\lfloor r_{a_k} \rfloor$ and sends the outcome to her, using $\log(|\mathcal{A}|)$ bits. The reason for that is that by definition the alternative that $f_k$ outputs for a given scalar $r_{a_k}$ is determined solely by the type of Bob, so no communication is involved in its computation. We now show that $f_k$ can be truthfully implemented, then we will analyze $cc_{IC}(f_k)$.

**Lemma 3.3.** $f_k$ is monotone and hence implementable.

**Proof.** $w_B(\cdot)$ is constant so $f_k$ is obviously monotone with respect to Bob. In order to show monotonicity with respect to Alice as well, fix $v_B \in V_B$ and two types $r, r' \in [0, 2^{k+1} - 1]$ such that $r > r'$. Denote the valuations $r \cdot w_A(\cdot)$ and $r' \cdot w_A(\cdot)$ with $v$ and with $v'$, respectively. Denote $map(v_B)$ with $\chi$ and define $index : \mathcal{A} \to \mathbb{N}$ as $index(a_i) = i$. We wish to prove that $w_A(f_k(v_B, v)) \geq w_A(f_k(v_B, v'))$. $r > r'$ clearly implies $|r| \geq |r'|$. By definition, $\chi = map(v_B)$ is monotonically increasing with respect to the index of alternative, so $index(\chi(r)) \geq index(\chi(r'))$. $w_A$ assigns greater values to alternatives with higher index, so $w_A(\chi(r)) \geq w_A(\chi(r'))$. By the definition of $f_k$, we get that $w_A(f_k(v_B, v)) \geq w_A(f_k(v_B, v'))$.}

We now analyze the hardness of computing the payments of Alice. By Proposition 2.1, her unique normalized payment scheme is:

$$P_A(a_B, v_B) = r_{a_B} \cdot w_A(f_k(a_B, v_B)) - \int_0^{r_{a_B}} w_A(f_k(z, w_A(a_B), v_B)) dz \quad (5)$$

**Claim 3.4.** $cc(P_A) \leq 2 \cdot cc_{IC}(f_k)$.

**Proof.** Let $M^*$ $(f_k, P_A^*)$ be the most efficient mechanism for $f_k$, i.e. $cc(M^*) = cc_{IC}(f_k)$. Denote Alice’s valuation when $r_A = 0$ with $v_0$, i.e., $v_0 \equiv 0 \cdot w_A$. Run $M^*$ on the instances $(v_A, v_B)$ and $(v_0, v_B)$ to obtain $P_A^*(v_A, v_B)$ and $P_A^*(v_0, v_B)$. Note that the payment $P_A^*(v_A, v_B) - P_A^*(v_0, v_B)$ is normalized and truthful (because it is a translation of $P^*$ by a constant). Recall that the domain of private values of Alice is an interval, so by Proposition 2.1, Alice has a unique normalized payment scheme. Thus, $P_A^*(v_A, v_B) = P_A(v_A, v_B) - P_A^*(v_0, v_B)$, so computing the normalized payment is at most twice harder than computing the communication-wise optimal mechanism.

We now move to the main part of the proof which is showing that the image of $P_A$ is “large”. We start with the following lemma. Recall that $(b, c)$ stands for the dot product of the vectors $b$ and $c$.

**Lemma 3.5.** Let $w$ be the vector $(w_A(a_1), w_A(a_1), \ldots, w_A(a_{2k}))$. For every two vectors $c \neq c' \in C$, $(w, c) \neq (w, c')$.  

937
Proof. Denote with $j$ the largest index where $c$ and $c'$ differ. Assume without loss of generality that $c_j > c'_j$. If $j = 0$, it means that $c$ and $c'$ differ in only one coordinate, so the dot product of each of them with the vector $w$ cannot be equal. Hence, we can assume from now on that $j \geq 1$. We will show that $(w, c) > (w, c')$:

$$\langle w, c \rangle - \langle w, c' \rangle = \sum_{i=0}^{\left|\mathcal{A}\right|-1} w_i \cdot (c_i - c'_i)$$

$$= \sum_{i=0, c_i > c'_i} w_i \cdot (c_i - c'_i) + \sum_{i=0, c_i' > c_i} w_i \cdot (c_i - c'_i)$$

$$\geq w_j \cdot (c_j - c'_j) + \sum_{i=0, c_i' > c_i} w_i \cdot (c_i - c'_i)$$

(6)

$$\geq w_j \cdot (c_j - c'_j) + \sum_{i=0, c_i' > c_i} w_i \cdot (c_i - c'_i)$$

since $c_j, c'_j \in \mathbb{N}$ and $c_j > c'_j$.

$$\geq w_j + \sum_{i=0, c_i' > c_i} w_i \cdot (c_i - c'_i)$$

$$= |\mathcal{A}|^{j+1} - 1 - |\mathcal{A}|^{j} + 1 = 0$$

(7)

which completes the proof. (6) holds because there are only positive summands in $\sum_{i=0, c_i > c'_i} w_i \cdot (c_i - c'_i)$, one of them is $w_j \cdot (c_j - c'_j)$.

We now explain (7), by proving that $\sum_{i=0, c_i' > c_i} w_i \cdot (c_i - c'_i) < |\mathcal{A}|^{|j|} - 1$.

$$-(\sum_{i=0, c_i' > c_i} w_i \cdot (c_i - c'_i)) = \sum_{i=0, c_i' > c_i} w_i \cdot (c_i' - c_i) \cdot (c'_i - c_i)$$

$$\leq |\mathcal{A}| \cdot \sum_{i=0, c_i' > c_i} w_i \cdot (c_i' - c_i)$$

$$= |\mathcal{A}| \cdot (|\mathcal{A}|^{j-1} - 1) \cdot 2^{k+1}$$

$$< |\mathcal{A}|^{|j|} \cdot 2^{k+3} - |\mathcal{A}|^{|j|}$$

$$< |\mathcal{A}|^{|j|} - 1$$

(8)

holds since $j$ is the maximum coordinate where $c, c'$ differ and it is known that $c_j > c'_j$. Thus, $c_j > c'_j$ implies that the largest index in the summation is $j - 1$. Recall that by definition $w_j \geq w_i$ for every $i \leq j - 1$. Note that we assumed $j \geq 1$, so $w_{j-1}$ is well defined.

Claim 3.6. Let $v_A \in V_A$ be a valuation such that $r_{v_A} = 2^{k+1} - 1$. Then, for all $v_1, v_2 \in V_B$:

$$v_1 \neq v_2 \implies P_A(v_A, v_1) \neq P_A(v_A, v_2)$$

(9)

As a corollary, if we reformulate $P_A$ as a function of the alternative and of Bob's value, we get that:

$$v_1 \neq v_2 \implies P_A(a_2, v_1) \neq P_A(a_2, v_2)$$

(10)

We will use the corollary in the proof of Theorem 3.14 in Subsection 3.3.

Proof. Recall that every $\chi_c \in X$ is monotone, and its corresponding vector $c \in C$ satisfies that $c_k \geq 1$. Thus, for every $v_B \in V_B$, if $r_{v_B} = 2^{k+1} - 1$, then $\int f_k(z, v_B) \, dz = \int f_k(z, v_B) \, dz'$. In words, Alice always gets alternative $a_k$ when bidding her highest value. Combining (9) with the payment formula in (5), we get the following logical equivalences:

$$P_A(v_A, v_1) \geq P_A(v_A, v_2) \iff$$

$$\left(2^{k+1} - 1 \cdot w_A(a_k)\right) - \int_0^{2^{k+1} - 1} w_A(f_k(z, v_A, v_1)) \, dz \geq$$

$$\left(2^{k+1} - 1 \cdot w_A(a_k)\right) - \int_0^{2^{k+1} - 1} w_A(f_k(z, v_A, v_2)) \, dz \iff$$

$$\int_0^{2^{k+1} - 1} w_A(f_k(z, v_A, v_1)) \, dz \geq$$

$$\int_0^{2^{k+1} - 1} w_A(f_k(z, v_A, v_2)) \, dz \iff$$

$$\int_0^{2^{k+1} - 1} w_A(\text{map}(v_1)[z]) \, dz \geq$$

$$\int_0^{2^{k+1} - 1} w_A(\text{map}(v_2)[z]) \, dz \iff$$

$$\langle c_1, w \rangle \geq \langle c_2, w \rangle$$

The last transition holds since the integral of $w_A(\chi_1(\{\cdot\}))$ over the interval $[0, 2^{k+1} - 1]$ equals to $(w, c_1)$ where $w = \{w_A(a_0), w_A(a_1), \ldots, w_A(a_{2^k})\}$, and the same clearly applies also to the RHS. Recall that $\text{map}(\cdot)$ is one-to-one, so $v_1 \neq v_2$ means that $\chi_1 \neq \chi_2$ and $c_1 \neq c_2$. By Lemma 3.5, $c_1 \neq c_2$ implies that $(c_1, w) \neq (c_2, w)$. Therefore, $P_A(v_A, v_1) \neq P_A(v_A, v_2)$, as required.

As for the proof of the corollary of Claim 3.6, bidding the valuation $v_A$ such that $r_{v_A} = 2^{k+1} - 1$ guarantees alternative $a_k$, so it is immediate from the taxation principle that:

$$P_A(a_2, v_1) = P_A(v_A, v_1) \neq P_A(v_A, v_2) = P_A(a_2, v_2) \iff$$

$$P_A(a_2, v_1) \neq P_A(a_2, v_2)$$

\[ Q.E.D. \]

Corollary 3.7. $cc(P_A) \geq 2^k$.

Proof. By Claim 3.6, different values of Bob induce a different payment for Alice whenever she bids the valuation $v_A$ such that $r_{v_A} = 2^{k+1} - 1$. Thus, $|\text{Im} P_A| \geq |V_B|$, where $|V_B| = \left(\begin{array}{c}k+1 \\ 2^k\end{array}\right)$. We deduce that every protocol that computes $P_A$ has at least $\left(\begin{array}{c}k+1 \\ 2^k\end{array}\right)$ leaves, so $cc(P_A)$ is bounded from below by $\log \left(\begin{array}{c}k+1 \\ 2^k\end{array}\right)$. Hence:

$$cc(P_A) \geq \log \left(\begin{array}{c}k+1 \\ 2^k\end{array}\right) \geq 2^k \cdot \log \frac{2^{k+1} - 1}{2^k} = 2^k \cdot \log 2 = 2^k$$

where the second inequality is due to the binomial bound $\left(\begin{array}{c}n \\ k\end{array}\right) \geq \left(\frac{n}{k}\right)^k$.

To conclude the proof of Theorem 3.2, combining Claim 3.4 and Corollary 3.7 yields that $cc_C(f) \geq 2^{k-1}$. In contrast, as discussed above, $cc(f) \leq k + 1 + \log |\mathcal{A}| \leq 3k$. \[ Q.E.D. \]
3.2 Proof II: Hardness via Interaction

We now show that if there are more than two players, then payment computation might be hard even if the number of possible payments is small. The idea is to construct a social choice function such that the payment is determined by the number of bit intersections of Bob’s and Charlie’s inputs (note that determining this number is harder than solving the disjointness problem). The challenge is to design such an $f$ with the additional property that $cc(f)$ is still small. We achieve that by constructing $f$ in which the chosen alternative depends only on $v_A$ and a constant number of bits of Bob and Charlie. That is, determining the output can be done “locally” but determining the payments is done “globally”.

**Theorem 3.8.** For every integer $k \geq 1$, there exists a single parameter social choice function $f_k$ over three players and $O(k)$ alternatives, where $cc(f_k) = \Theta(\log k)$, $cc(f_k^C) = \Omega(k)$ and $Imp_A = O(k)$, where $Imp_A$ is the image of the normalized payment function for Alice (by Proposition 2.1).

**Proof.** We describe the function $f_k$. For every integer $k \geq 1$, we define the set of alternatives as $A = \{1, \ldots, k\}$. There are three players, Alice, Bob, and Charlie, with single parameter domains. Alice’s private information is $r_A \in [0, k - 1]$ and her value for each alternative $a$ is $v_A(a) = a$ for every alternative. The private information of Bob and Charlie is $r_B, r_C \in \{0, 1\}^k$. Their values for all alternatives are identical and equal to the integer representations of their private information, so we use $v_B$ and $r_B$ interchangeably, and the same applies to $v_C$ and $r_C$. Denote with $v_B(j)$ and $v_C(j)$ the $j$th bits of $v_B$ and of $v_C$. We define $f_k : V_A \times V_B \times V_C \to A$ as follows. For every $v_A \in V_A$, $v_B \in V_B$, $v_C \in V_C$:

$$f_k(v_A, v_B, v_C) = \begin{cases} r_{oA} + 1 & v_B(r_{oA}) = v_C(r_{oA}) = 1 \\ r_{oA} & \text{otherwise} \end{cases}$$

**Lemma 3.9.** $f_k$ is monotone and hence implementable.

**Proof.** $f_k$ is clearly monotone with respect to Bob and Charlie. In order to prove monotonicity with respect to Alice, we fix $v_A, v_B', v_C' \in V_A \times V_B \times V_C$ such that $r_{oA} > r_{oA}'$. We want to show that $w_A(f_k(v_A, v_B', v_C')) \geq w_A(f_k(v_A, v_B, v_C'))$. If $|r'| = |r|$, by definition $w_A(f_k(v_A, v_B', v_C')) = w_A(f_k(v_A, v_B, v_C'))$ and we are done. Otherwise, we know that $|r_A'| > |r_{oA}|$, so $|r_{oA}'| \geq |r_{oA}| + 1$. Therefore:

$$w_A(f_k(v_A', v_B, v_C')) \geq w_A(f_k(v_A, v_B, v_C')) = w_A(f_k(v_A, v_B', v_C'))$$

By Proposition 2.1, the fact that Alice’s type space is an interval implies that the only normalized payment scheme that implements $f_k$ for Alice is:

$$P_A(v_A, v_B, v_C) = r_{oA} \cdot w_A(f_k(v_A, v_B, v_C)) = \int_0^{r_{oA}} w_A(f_k(z, v_B, v_C))dz$$

The following lemma is key in showing that $P_A$ is "small". For all $v_B \in V_B$ and $v_C \in V_C$, denote $v_B^j$ and $v_C^j$ as the prefixes of length $j$ of $v_B$ and of $v_C$.

**Lemma 3.10.** Fix an alternative $j \in A$, and an integer $t \in \{0, \ldots, j\}$. Then, for all types $(v_B, v_C) \in V_B \times V_C$ where the intersection of $v_B^j$ and $v_C^j$ is of size $t$ and alternative $j$ is reachable from $(v_B, v_C)$, Alice’s price for $j$ according to the payment scheme $P_A$ is one of the following:

$$(1) \quad j \cdot (j - 1) - \frac{(j - 1)(j - 2)}{2} - t + 1,

(2) \quad j^2 - \frac{j(j - 1)}{2} - t.$$
Lemma 3.12. Computing $P_A$ requires $O(k)$ bits of communication.

Proof. We reduce from disjointness with $k - 1$ bits. Let Bob’s type be the input of the first player in the disjointness problem with extra zero bit at the end, and let Charlie’s type be the input of the second player with an extra zero bit at the end. Let Alice’s type be $\omega_A = (k - 1) \cdot \omega_A$, i.e., $\omega_A = k - 1$. Since the $(k - 1)$th bits are by construction not intersecting, alternative $k - 1$ is chosen. If there are no intersecting bits in the $(k - 1)$-disjointness problem, then by equation (12), the payment is $(k - 1)^2 - \frac{(k - 1)(k - 2)}{2}$. If there is an intersecting bit, then the payment is strictly smaller. Thus, computing the payment is at least as hard as computing disjointness with $k - 1$ bits, and the lemma follows. □

Lemma 3.13. $2 \cdot cc_{IC}(f_k) \geq cc(P_A)$.

Proof. Let $P^M$ be the most efficient protocol that implements $f_k$, i.e., $cc_{IC}(f_k) = cc(P^M)$. Fix the valuations $\omega_A, \omega_B, \omega_C \in V_A \times V_B \times V_C$. For the payment of Alice, denote Alice’s payment scheme according to $P^M$ with $P^A_A$. Denote Alice’s valuation when $\omega_A = 0$ with $\omega_0$. The uniqueness of $P^A_A$ as a normalized payment scheme allows us to use a similar argument to the proof of Claim 3.4, and get that $P_A(\omega_A, \omega_B, \omega_C) = P^A_A(\omega_A, \omega_B, \omega_C) = P^A_A(\omega_0, \omega_B, \omega_C)$. Thus, $P_A$ can be computed using two calls to $P^M$.

We now finish the proof of Theorem 3.8. $cc(f_k) \geq \log(k)$, because $f_k$ can be easily computed by a protocol where Alice sends $[\omega_0]$ using $\log k$ bits, and Bob and Charlie send back their $[\omega_0]$. Also, there are $k$ alternatives, so by standard communication complexity arguments, $cc(f) \geq \log k$. Combining Lemmas 3.12 and 3.13, we get that $cc_{IC}(f_k) = O(k)$. By Corollary 3.11, $|ImP_A| = O(k^2)$, which completes the proof. □

3.3 Hardness of Welfare Maximizing Mechanisms

In Subsections 3.1 and 3.2, we gave two examples for social choice functions for which there is an exponential gap between computing the output and payment computation. In this section we show that there are natural social choice functions that exhibit this exponential gap.

The natural social choice function that we construct is simply a welfare maximizer in a multi-unit auction with $m$ items. We use this multi-unit auction instance to show another gap: between the approximation ratio of truthful and non-truthful algorithms. For this instance, non-truthful algorithms achieve the optimal welfare with $O(\log m)$ communication (approximation ratio of 1), whereas their truthful counterparts achieve the optimal welfare only if they use at least $\Omega(m)$ communication. It means, that all truthful algorithms with running time $o(m)$ have approximation ratio which is strictly less than 1.

In a multi-unit auction with $m = 2^k$ items, all items are identical and values of players are determined solely by the number of items they get: $v : \{0, \ldots, m\} \to \mathbb{R}_+$. The valuations are monotone ($l > j$ implies $v(l) \geq v(j)$) and normalized ($v(0) = 0$).\(^7\)

\(^7\) Note that we use the term “normalized” to describe two different notions. The first one is a property of mechanisms (bidding zero guarantees a payment of zero) and the second one is a property of multi-unit valuations ($v(0) = 0$).

Theorem 3.14. For every integer $k \geq 1$, there exists a multi-unit auction instance with $m = 2^k$ items such that a welfare maximizing allocation can be computed with $O(k) = O(\log m)$ bits, but every truthful mechanism that maximizes the welfare requires at least $\exp(k) = \Omega(m)$ communication. As a corollary, there exists a welfare-maximizing allocation rule $f$ such that $cc_{IC}(f) = \exp(cc(f))$.

The proof of Theorem 3.14 can be found in the full version of the paper.

4 Truthful in Expectation Mechanisms

Up until now we showed that the communication cost of ex-post implementations of social choice functions might be exponential comparing to output computation. However, we observe that in many domains the payments of the (deterministic) social choice function can be computed randomly so that the expected value of the payment equals the value of the (deterministic) ex-post payment scheme. If the players are risk neutral, this gives us a truthful-in-expectation implementation of the social choice function. For all the domains below we prove that $cc_{TIE}(f) \leq poly(n, cc(f))$.

For single parameter domains, where the private information $R_i$ of each player is either an interval $[0, b_i]$ or a finite set with non-negative values, the computation of payment is based on the observation that to compute the expected value of the integral in Myerson’s payment formula, it suffices to evaluate the value of the integral at a random valuation and know the type of the player $i$, $v_i$. One point of potential complication is that representing the type of player $i$ might be much more costly than computing $cc(f)$. We get around this problem by essentially providing a "similar" type to $v_i$, which is based on the communication protocol.

We then extend our results to some multi-parameter domains. We first consider scalable domains: domains where for each constant $\lambda \in [0, 1]$, if the type $v(\cdot)$ is in the domain, then so does $\lambda \cdot v(\cdot)$. We rely on a result of [5] who show an integral-based payments formula similar to Myerson’s for this domain. We rely on this formula in the sense that we compute a payment which is equal in expectation to it, similarly to the single parameter case. Finally, we show that scalable and convex domains are computationally equivalent. We begin by formally defining truthfulness in expectation.

Definition 4.1. Let $f : V_1 \times \cdots \times V_n \to A$ be a (deterministic) social choice function. A mechanism $M = (f, P)$ is truthful in expectation if for every player $i$, every $v_{-i} \in V_{-i}$ and every $v_i, v'_i \in V_i$: $\mathbb{E}[v_i(f(v_i, v_{-i})-P_i(v_i, v_{-i})) \geq \mathbb{E}[v_i(f(v'_i, v_{-i})-P_i(v'_i, v_{-i}))]$ (14) where the expectation is taken over the randomness of $P$.

We denote with $cc_{TIE}(f)$ the communication complexity of the optimal truthful-in-expectation implementation for $f$.

4.1 Single Parameter Domains

We are especially interested in truthful in expectation implementation for single parameter mechanisms, because they demonstrate an exponential gap between the communication complexities of deterministic truthfulness and truthfulness in expectation. The gap is established by observing that the functions used in Section 3 to derive lower bounds satisfy $cc_{IC}(f) = \exp(cc(f))$ and
\[ cc_{\text{ITE}}(f) \leq \text{poly}(cc(f)), \]

where the latter statement follows from Theorem 4.2.

Theorem 4.2. Let \( f : V = V_1 \times \cdots \times V_n \rightarrow \mathcal{A} \) be an implementable social choice function over a single parameter domain where for every player \( i \), \( R_i \) is an interval \([0, b_i]\) such that \( b_i \in \mathbb{R} \). Then:

\[ cc_{\text{ITE}}(f) \leq (n+1) \cdot cc(f) \]

As a corollary, if \( R_i \) is a finite set with non-negative values for every player \( i \), \( cc_{\text{ITE}}(f) \leq (n+1) \cdot cc(f) \).

For the proof of Theorem 4.2, we obtain an unbiased estimator for an integral using uniform sampling, similarly to [1] and [5]. Let \( U[a, b] \) be the continuous uniform distribution over the interval \([a, b]\).

Lemma 4.3. Let \( g : [a, b] \rightarrow \mathbb{R} \) be an integrable function. Define a random variable \( R = (b - a) \cdot g(z) \), where \( z \sim U[a, b] \). Then, \( R \) is an unbiased estimator of \( \int_a^b g(x) \) \( dx \).

Proof.

\[
\mathbb{E}_z[(b - a) \cdot g(z)] = (b - a) \cdot \mathbb{E}_z[g(z)] = (b - a) \cdot \int_{-\infty}^{\infty} g(z) \cdot f(z) \, dz = (b - a) \cdot \int_{a}^{b} g(z) \, \frac{1}{b-a} \, dz = \int_{a}^{b} g(z) \, dz
\]

where \( (15) \) is due to the law of the unconscious statistician, and \( (16) \) is achieved by plugging in the density function of the continuous uniform distribution. \( \square \)

Proof of Theorem 4.2. Denote a protocol of \( f \) with \( \pi \). We begin by running \( \pi(v_1, \ldots, v_n) \). Fix a player \( i \), \( V_i \) is an interval, so Proposition 2.1 yields that \( f \) is deterministically implemented by:

\[ P_i(v_i, v_{-i}) = r_{v_i} \cdot w_i(f(v_i, v_{-i})) - \int_0^{r_{v_i}} w_i(f(z \cdot w_i, v_{-i})) \, dz \] (17)

Hence, in order to obtain a payment which is truthful in expectation it suffices to compute a payment scheme whose expected value is \( (17) \). Each leaf in the protocol \( \pi \) is a combinatorial rectangle; \( L = L_1 \times \cdots \times L_n \). For each leaf \( L \), the players agree in advance on a profile which belongs in the leaf, i.e. \( (a_1^*, \ldots, a_n^*) \in L \). Denote the leaf that \( (a_1, \ldots, a_n) \) reaches with \( L^* \), and denote its agreed upon type for player \( i \) with \( v_i^* \).

Lemma 4.4. \( P_i(v_i^*, v_{-i}) = P_i(v_i, v_{-i}) \).

Proof. By definition, \( (v_i, v_{-i}) \in L^* \) and \( v_i^* \in L_i^* \), so by the mixing property \( (v_i^*, v_{-i}) \in L^* \). Thus, \( w_i(f(v_i^*, v_{-i})) = w_i(f(v_i, v_{-i})) \), so from truthfulness we get that \( P(v_i^*, v_{-i}) = P(v_i, v_{-i}) \). \( \square \)

Hence, it suffices to compute a random variable with expectation \( P_i(v_i^*, v_{-i}) \). Note that \( v_i^* \) is known to all players, due to the execution of \( \pi \). Some player \( j \neq i \) sample \( z \sim U[0, r_{v_i}^*] \). Note that if we let player \( i \) sample \( z \) he could have potentially misreported the sample in order to increase the profit. All players in \( N \setminus \{i\} \) simulate \( (z \cdot w_i, v_{-i}) \) in order to obtain \( f(z \cdot w_i, v_{-i}) \). The output is:

\[ \hat{P} = r_{v_i} \cdot w_i(f(v_i^*, v_{-i})) = r_{v_i} \cdot w_i(f(z \cdot w_i, v_{-i})) \] (18)

By Lemma 4.3, \( \mathbb{E}[\hat{P}] = P_i(v_i^*, v_{-i}) \). Due to the two executions of \( f \), the players know all the components of \( \hat{P} \). Hence, we obtained a truthful in expectation payment for player \( i \) by making one extra call to \( \pi \). Thus, truthful in expectation implementation of \( f \) requires at most \((n+1) \cdot cc(f)\) bits.

For the corollary, let \( R_i \) be a finite domain with non-negative values. Fix a player \( i \) and a type \((v_1, \ldots, v_n)\). The players simulate \( \pi(v_1, \ldots, v_n) \). Let \([0, b_i]\) be an interval that contains all the elements in \( R_i \). We extend \( f \) to output for every \( x \neq R_i \) the same alternative as it assigns to the nearest \( R_i \) which is smaller than \( x \). For all \( x \) smaller than \( R_i \), the extension \( f_{\text{ext}} \) always outputs an arbitrary alternative \( a \) such that \( w_i(a) \) is minimal. This extension preserves the monotonicity of \( f \) and its new domain is an interval, so by Proposition 2.1, \( f_{\text{ext}} \) is deterministically implemented by:

\[ P_i(v_i, v_{-i}) = r_{v_i} \cdot w_i(f_{\text{ext}}(v_i, v_{-i})) = \int_0^{r_{v_i}} w_i(f_{\text{ext}}(z \cdot w_i, v_{-i})) \, dz \]

which is smaller than \( x \). Thus, truthful in expectation implementation of \( f \) requires \((n+1) \cdot cc(f)\) bits.

□

4.2 Scalable Domains

Roughly speaking, scalable domains are multi-parameter domains that can be “stretched”. They are useful because of two main properties. The first is that a scalable domain can be projected to a single parameter domain, so upper bounds of payment computation in single parameter settings extend to them. The latter is that they are essentially equivalent (up to translation) to convex domains, so we use them as a means to derive upper bounds for them. Formally, scalable domains are:

Definition 4.5. A domain of a player \( V_i \) is scalable if for every \( v_i \in V_i \) and every \( \lambda \in [0,1] \), \( \lambda \cdot v_i \in V_i \).

By definition, a scalable domains necessarily contains a zero valuation, \( v_i \equiv 0 \). Thus, for scalable domains, we say that a mechanism is normalized if \( P_i(0, v_{-i}) = 0 \) for all \( v_{-i} \). As observed by [5], a corollary of Rochet [15] is:

Proposition 4.6 ([5,15]). Let \( f : V_1 \times \cdots \times V_n \rightarrow \mathcal{A} \) be an implementable social choice function with scalable domains. Then, the mechanism \((f, P)\) is truthful and normalized if and only if for every player \( i \):

\[ P_i(v_i, v_{-i}) = v_i(f(v)) - \int_0^1 v_i(f(t \cdot v_i, v_{-i})) \, dt \] (19)

Theorem 4.7. Let \( f : V_1 \times \cdots \times V_n \rightarrow \mathcal{A} \) be a social choice function with scalable domains. Then, \( cc_{\text{ITE}}(f) \leq (n+1) \cdot cc(f) \).

Proof. Let \( \pi \) be a communication protocol for \( f \). The players simulate \( \pi(v_1, \ldots, v_n) \). By Proposition 4.6, the payment in (19) deterministically implements \( f \), so for every player we wish to compute a random variable whose expected value is equal to it. For every leaf in \( \pi \), the players agree in advance on a profile which belongs in the leaf. Denote the leaf that \( (v_1, \ldots, v_n) \) reaches with \( L^* \), and its agreed type with \( v_i^* \). Lemma 4.4 allows us to focus on computing a random variable with expectation \( P_i(v_i^*, v_{-i}) \). We obtain such a
random variable when a player \( j \neq i \) samples \( t \sim U[0, 1] \). The players simulate \( \pi(t; v_i, v_{-i}) \) and output:

\[
\hat{P}_i = v_i^*(f(v_i^*, v_{-i})) - v_i^*(f(t; v_i^*, v_{-i}))
\]

By Lemma 4.3, \( \mathbb{E}[\hat{P}_i] = P_i(v_i^*, v_{-i}) \). Due to the two executions of \( \pi \), the players know all the components of \( \hat{P} \). By repeating for all players, we get a truthful-in-expectation implementation of \( f \) with \( (n + 1) \cdot cc(f) \) bits.

\[\square\]

4.3 Convex Domains

Convex domains are useful for mechanism design since they are weak monotonicity domains [17], i.e., domains where social choice function that satisfy weak monotonicity are necessarily implementable.

**Theorem 4.8.** Let \( f: V = V_1 \times \ldots \times V_n \to A \) be a social choice function with convex domains. Then, \( cc_{\text{IE}}(f) \leq (n + 1) \cdot cc(f) \).

We prove properties of convex domains by reducing them to scalable ones: we show that for every function, translating its domain by a constant has no effect on it, and that all convex domains translate to scalable domains. The complete proof can be found in the full version of the paper.

5 AN ALGORITHM FOR SINGLE PARAMETER SETTINGs

We now return to considering deterministic ex-post implementations. Notice that the exponential lower bounds of Section 3 were proven using single parameter social choice functions. We now provide an algorithm for all such functions. The upper bound on the communication complexity of the algorithm has a linear in \(|A|\) factor. The communication complexity of the algorithm is optimal in the sense that the dependence on \(|A|\) is necessary, as demonstrated by the examples in Theorems 3.2 and 3.8.

**Theorem 5.1.** For all single parameter environments, \( cc_{\text{IC}}(f) = O(n \cdot cc(f) |A|) \). As a corollary, for binary single parameter domains, \( cc_{\text{IC}}(f) = O(n \cdot cc^2(f)) \).

Recall that in single parameter settings, the valuations set of each player is composed of a public function \( w_i: A \to \mathbb{R} \) and a type space which contains scalar private information \( R_i \). Each valuation \( w_i() \) is equal to \( r_i \cdot w_i() \) for some \( r_i \in R_i \). For brevity, throughout the paper we slightly abuse notation by writing \( w_i \) both for a valuation and for the scalar private information associated with it, \( r_i \).

For every player \( i \), denote the range of \( w_i \) with \( Im w_i \). We will show that \( cc_{\text{IC}}(f) \leq O(n \cdot cc^2(f) \cdot \max_i |Im w_i|) \). It implies the theorem, because \( \max_i |Im w_i| \leq |A| \). Since binary single parameter functions satisfy that \( Im w_i = \{0, 1\} \) for every player \( i \), it is immediate that they satisfy \( cc_{\text{IC}}(f) = O(n \cdot cc^2(f)) \).

The proof is based on the observation that the payment fully depends on \( f(v_i, v_{-i}) \) and that by multiple binary searches, the players know \( w_i(f(v_i, v_{-i})) \) for all \( v_i \in V_i \). The binary searches are for the sake of finding the “threshold” values of each alternative. The theorem has no assumptions at all neither the domain

\[\text{not the function, but it comes at a price: the proof involves a lot of technicalities in order to include all single parameter domains, also very degenerate ones. The proof can be found in the full version of the paper.}\]

**Tightness.** We explain why the factors \(|A|\) and \( cc(f) \) cannot be omitted, i.e. that it cannot be the case that for all functions, or even for all single parameter functions satisfy that \( cc_{\text{IC}}(f) \leq poly(n, cc(f)) \) or that \( cc_{\text{IC}}(f) \leq poly(n, |A|) \). For non-degenerate functions \( cc(f) \geq n \), so we consider the \( n \) factor to be less significant.

The social choice functions in Section 3 serve as counterexamples to \( cc_{\text{IC}}(f) \leq poly(n, cc(f)) \), since all of them satisfy that \( cc_{\text{IC}}(f) = \exp(cc(f)) \) with a constant number of players. Similarly, we can easily provide a function with two alternatives and two players where the communication complexity of its implementation is arbitrarily large: let \( f_k: V_1 \times V_2 \to \{a_0, a_1\} \). The valuations of the players are their private information, and they do not depend on the alternative chosen: \( R_1 = V_1 = R_2 = V_2 = \{0, 1, \ldots, 2^k - 1\} \). \( f_k(a_1, a_2) = a_1 \) if and only if the bit representations of \( a_1, a_2 \) are disjoint. Clearly, \( f_k \) is harder than the function \( DISJ_k \) and it is well known that \( cc(DISJ_k) = \Omega(k) \). \( f_k \) is implementable with no payments, because the valuations of both players do not depend on the outcome. Hence, \( cc_{\text{IC}}(f_k) = cc(f_k) = \Omega(k) \), whereas \( n = 2 \) and \(|A| = 2\).

6 PAYMENT COMPUTATION IN MULTI-PARAMETER SETTINGs

So far, we considered deterministic algorithms only for single-parameter domains. In this section, we venture into the more challenging multi-parameter setting. We begin by proving an efficient algorithm for functions that satisfy uniqueness of payments. There is a vast literature on the topic of characterizing domains and functions where implementability guarantees uniqueness of payments (for example, \([10, 14]\)). Notice that “uniqueness of payments” is often called “revenue equivalence”.

We conclude by showing that an efficient algorithm for functions that satisfy uniqueness of payments yields efficient algorithms for implementable functions with scalable and convex domains (Claims 6.9 and 6.10).

The intuition to the proof is as follows. Instead of providing a deterministic protocol that proves this bound as usual, we provide a non-deterministic protocol that shows it. A deterministic mechanism follows by relying on the fact that the connection between deterministic and non-deterministic communication complexity of promise problems is polynomial (as we show in Section A).

Thus, the problem boils down to providing a succinct witness that determines the price of the alternative chosen. We make the observation that a payment can be determined by a conjunction of \( O(\mid |A|^2 \mid) \) inequalities. For illustration, fix \( v_{-i} \). Let \( \sigma^a \) be such that \( f(\sigma^a, v_{-i}) = a \) and \( \sigma^b \) such that \( f(\sigma^b, v_{-i}) = b \). Then, it obviously holds that \( \sigma^a(a) - \sigma^a(b) \geq c^a(b) - c^a(a) \) and \( c^b(b) - c^b(a) \geq c^b(a) - c^b(b) \). So \( \sigma^a(a) - \sigma^b(b) \geq \sigma^b(a) - \sigma^b(b) \). We show that if we choose, for each such alternatives \( a \) and \( b \), two types \( \sigma^a \) and \( \sigma^b \) such that the inequality is “tight” as possible, then the payment of an alternative can be determined. These \( O(|A|^2) \) types will serve as our non-deterministic witness, which completes the proof except
that the description of the types might be huge. We rely on the communication protocol of \( f \) to provide a succinct description of them. We begin with some formalities.

**Definition 6.1. (Uniqueness of Payments)** A social choice function \( f: V = V_1 \times \cdots \times V_n \rightarrow \mathcal{A} \) satisfies uniqueness of payments if for every pair of truthful mechanisms \((f, P)\) and \((f, P')\), it holds that there exist \( n \) functions \( h_1, \ldots, h_n \) where \( h_i: V_i \rightarrow \mathbb{R} \), such that for every player \( i \) and every \((v_1, \ldots, v_n) \in V:\)

\[
P_i(v_1, \ldots, v_n) = P'_i(v_1, \ldots, v_n) + h_i(v_{-i})
\]

(20)

In Section 2, we define normalized mechanisms for single parameter settings. We generalize the definition to multi-parameter domains. 

**Definition 6.2. (Multi-Parameter Normalization)** For every player \( i \), let \( a_i^0 \in V_i \) be its zero type. We say that a mechanism \( M = (f, P) \) is normalized if for every player \( i \) and every \( v_i \in V_i \), \( P_i(v_i^0, v_{-i}) = 0 \).

For every \( v_i \), define \( f(v_i, v_{-i}) = a_i^0 \) as the zero alternative with respect to \( v_{-i} \). It is easy to see that if \( f \) satisfies uniqueness of payments, there exists a unique normalized mechanism which implements it.

**Theorem 6.3.** Let \( f: V_1 \times \cdots \times V_n \rightarrow \mathcal{A} \) be an implementable social choice function that satisfies uniqueness of payments. Then, \( cc_{IC}(f) \leq \text{poly}(n, |\mathcal{A}|, cc(f)) \).

**Proof of Theorem 6.3.** Fix an implementable function \( f \) that satisfies uniqueness of payments, and denote the normalized mechanism for it as \( M \). We show an upper bound for \( cc_{IC}(f) \) by presenting a communication protocol for \( M \). Computing the output requires \( cc(f) \) bits, so clearly the tricky part is computing the payments. By the taxation principle, the payment of every player is a function of \( v_{-i} \) and of the alternative chosen. Hence, we define a promise function for the price of an alternative \( a \in \mathcal{A} \) given \( v_{-i} \), \( \text{price}^a_i : V_{-i} \rightarrow \mathbb{R} \), with the promise that \( a \) is reachable from \( v_{-i} \). We denote the image of \( \text{price}^a_i \) with \( Im(\text{price}^a_i) \). Note that all the elements in \( Im(\text{price}^a_i) \) are real.

**Claim 6.4.** For every player \( i \) and every alternative \( a \in \mathcal{A} \), it holds that \( N(\text{price}^a_i) \leq O(|\mathcal{A}|^2 \cdot cc(f)) \), where \( N(\text{price}^a_i) \) is the non-deterministic communication complexity of the promise problem \( \text{price}^a_i \). 

The proof of Claim 6.4 can be found in Section 6.1. Combining the polynomial relation between deterministic and non-deterministic communication complexity (Section A) with Claim 6.4, we get that:

\[
cc_{IC}(f) \leq cc(f) + \sum_{i=1}^n cc(\text{price}^a_i)
\]

(Note computing \( f(v=a) \))

(21)

\[
\leq cc(f) + \sum_{i=1}^n \text{poly}(n, N(\text{price}^a_i)) \quad (\text{by Theorem A.2})
\]

\[
\leq \text{poly}(n, |\mathcal{A}|, cc(f)) \quad (\text{by Claim 6.4})
\]

Observe that \( f(v) = a \) implies that for every player \( i \), \( a \) is reachable from \( v_{-i} \), so \( v_{-i} \) satisfies the promise of \( \text{price}^a_i \). \( \square \)

### 6.1 Proof of Main Claim

**Proof of Claim 6.4.** Fix a player \( i \) and an alternative \( a \). We will prove the claim by presenting a proof system for \( \text{price}^a_i \). Since \( \text{price}^a_i \) is a promise function, a valid proof system for it satisfies that if \( v_{-i} \) breaks the promise, i.e., \( a \) is not reachable from \( v_{-i} \), then the players reject all witnesses for it.\(^{10}\)

First, we make some definitions and prove useful properties. Following the notation of \([6]\), let \( \delta_{ab} : V_i \rightarrow \mathbb{R} \cup \{\infty\} \) be a function that maps \( v_{-i} \) to \( V_i \) to \( \in \{(a, b) \mid f(a, v_{-i}) = a \} \) for every pair of alternatives \( a, b \in \mathcal{A} \). Because of truthfulness, \( \in \{(a, b) \mid f(v, v_{-i}) = a \} \) is an upper bound of the difference between the payment of \( a \) and the payment of \( b \) in every payment scheme that implements \( f \). We begin with the following technical lemma:

**Lemma 6.5.** Let \( f: V_1 \times \cdots \times V_n \rightarrow \mathcal{A} \) be a social choice function. Then, for every player \( i \) and every pair of alternatives \( a, b \in \mathcal{A} \), \( cc(f) \geq \log |Im\delta_{ab}| \). In particular, \( Im\delta_{ab} \) is finite.\(^{12}\)

**Proof of Lemma 6.5.** First, \( cc(f) \geq \log |Im\delta_{ab}| \) implies that \( Im\delta_{ab} \) is finite, because we assume that \( cc(f) \) is finite (see Remark 2.2). In order to prove \( cc(f) \geq \log |Im\delta_{ab}| \), we take an arbitrary finite subset \( T \) of \( Im\delta_{ab} \). Denote the \( t \) elements with \( \delta_1 < \cdots < \delta_t \).

The proof is by the fooling set method. For every \( j \in \{t\} \), we take a type \( v_{-i}^j \) such that \( \delta_{ab}(v_{-i}^j) = \delta_j \). We pair it with a type \( v_j^i \) in \( V_i \) in the following way. If \( \delta_j = \infty \), we take \( v_j^i \) to be an arbitrary type in \( V_i \). Otherwise, we take \( v_j^i \) such that \( f(v_j^i, v_{-i}^j) = a \) and \( \delta_j \leq f(v_j^i, v_{-i}^j) - f(v_j^i, v_{-i}^j) < \delta_{j+1} \).

We need to show that every \( v_{-i}^j \) has a matching \( v_j^i \) that satisfies those requirements. If \( \delta_j = \infty \), it is trivial that an arbitrary type in \( V_i \) exists. If \( \delta_j < \infty \), a matching \( v_j^i \) for \( v_{-i}^j \) necessarily exists because \( \in \{(a, b) \mid f(a, v_{-i}^j) = a \} \) is non empty and \( \delta_j \) is its real lower bound. For \( j < t \), such a type necessarily exists, since \( \delta_j \) is by definition the greatest lower bound of \( \in \{(a, b) \mid f(v, v_{-i}^j) = a \} \). The fact that \( \delta_{j+1} \) is not the infimum even though it is larger than \( \delta_j \) implies that it is not a lower bound of the subset, so there exists a type \( v_{-i}^{j+1} \) such that \( \delta_j \leq f(v_{-i}^{j+1}) - f(v_{-i}^{j+1}) < \delta_{j+1} \).

Let \( S = \{(v_j^i, v_{-i}^j) \mid 1 \leq j \leq t \} \). We will show that any two inputs in \( S \) cannot belong in the same leaf in any communication protocol.

---

\(^9\)The reason for it is as follows. The price of a reachable alternative \( a \) given \( v_{-i} \) cannot be \(-\infty \), because then a player would be incentivized to misreport his type whenever it is \( v_0 \), by reporting instead a type that reaches \( a \). Similarly, the price cannot be \( \infty \), because in this case the player would deviate from truthfulness whenever his real type is the type that reaches \( a \).

\(^{10}\)See Definition A.1.

\(^{11}\)If the set defined by \( \delta_{ab} \) is not bounded from below, \( \delta_{ab}(v_{-i}) \) outputs \(-\infty \).

\(^{12}\)We remind that \( Im(\cdot) \) stands for the range of a function.
for $k$. Let $(v_k^1, v_k^2), (v'_k, v'_k)$ be two inputs in $S$ where $k < l$. If $\delta_l = \infty$, clearly $f(v_k^1, v_k^2) \neq v$, whereas $f(v'_k, v'_k) = v$. Then, they clearly do not belong in the same leaf. If $\delta_{ab}(v_k^1) < \infty$, we know that $f(v_k^1, v_k^2) = f(v'_k, v'_k)$ = $a$. Note that $\delta_k \leq \delta_k(a) = \delta_k(b) < \delta_{k+1}$.


does not belong in the same leaf because they violate the mixing property. Hence, the number of leaves is at least $t$, so $cc(f) \geq \log t$. If every finite subset $T$ of the set $Im\delta_{ab}$, its size at most $2^{cc(f)}$, then the size of the set itself is at most $2^{2cc(f)}$. Therefore, $2^{cc(f)} \geq |Im\delta_{ab}| \implies cc(f) \geq \log |Im\delta_{ab}|$. □

**Proposition 6.6.** Let $f$ be a function that satisfies uniqueness of payments and suppose that $Im\delta_{ab}$ is countable for all $a, b \in A$. Then, for every $v_{-i}$ and for every reachable alternative from $i$'s, there exists a set $R \subseteq V_i$ of types $v_1, \ldots, v_t = O(|Arc|)$, such that if $v'_i \in V_i$ satisfies $f(v, v_{-i})$ = $f(v', v'_{-i})$ for all $v \in R$, then price$_i(v_{-i})$ = price$_i(v'_{-i})$. Moreover, $R$ contains the zero type $v_0$ and a type $v^*$ such that $f(v^*, v_{-i}) = a$.

Fix a function $f$ that satisfies those assumptions. By the taxation principle, we interpret a truthful mechanism as a process where a menu $M_{a,b}$ is presented to every player $i$ with her prices for all alternatives, and then an alternative which maximizes the utilities of all players is chosen. We assume that if an alternative is not reachable from $v_{-i}$, its price in the menu is $\infty$ and otherwise by definition $M_{a,b}(v) = price(v_{-i})$. Let $M \in \mathcal{M}$ be a menu. Given a menu $M$ and a subset of alternatives $A \subseteq \mathcal{M}$, we denote with $M^A$ the "restricted" version of $M$, with prices only for the alternatives in $A$. We say that a menu $M$ is truthful for $v_{-i}$ if player $i$ never increases his utility by misreporting his type given the prices in the menu $M$.

**Lemma 6.8.** Fix a type $v_{-i}$ and a pair of reachable alternatives $a, b \in A$ from $v_{-i}$. Fix $\delta_i \in Im\delta_{ab}$ where $j < |Im\delta_{ab}|$. Then, $\delta_{ab}(v_{-i}) \leq \delta_j$ if and only if there exists a type $v \in V_i$ such that:

1. $f(v, v_{-i}) = a$
2. $v(a) - v(b) < \delta_{j+1}$

Therefore, the existence of a type that satisfies conditions 1 and 2 implies that every truthful menu $M$ for $v_{-i}$ satisfies that $M(a) - M(b) \leq \delta_j$.

Proof. Assume $\delta_{ab}(v_{-i}) \leq \delta_j$, $\delta_j < \delta_{j+1}$, so $\delta_{ab}(v_{-i}) < \delta_{j+1}$. Recall that by definition, $\delta_{ab}(v_{-i}) = \inf\{v(a) - v(b) \mid f(v, v_{-i}) = a\}$. The fact that $\delta_{ab}(v_{-i})$ is the greatest lower bound and not $\delta_{j+1}$ implies that it is not a lower bound at all, and thus there exists a type $v \in V_i$ such that $f(v, v_{-i}) = a$ and $v(a) - v(b) < \delta_{j+1}$.

For the other direction, assume that there exists $v \in V_i$ such that $f(v, v_{-i}) = a$ and $v(a) - v(b) < \delta_{j+1}$ implies that $\delta_{ab}(v_{-i}) = \inf\{v(a) - v(b) \mid f(v, v_{-i}) = a\}$. Then, $\delta_{ab}(v_{-i}) < \delta_{j+1}$. Combining it with the fact that $\delta_{ab}(v_{-i}) < \delta_{j+1}$, it follows that $\delta_{ab}(v_{-i}) \leq \delta_{j+1}$.

Combining the two inequalities gives that every truthful menu $M$ for $v_{-i}$ satisfies that $M(a) - M(b) \leq \delta_j$.

**Proof of Proposition 6.6.** Fix a type $v_{-i}$ and an alternative $a'$ which is reachable from it. Denote its set of reachable alternatives with $A$, its zero alternative with $a_0$ and its normalized menu with $M_{a_0}$. We construct a subset $R \subseteq V_i$ as follows. First, we add the zero type $v_0$ to $R$. Then, for every reachable alternative $a \in A$, we add to $R$ a type $v_a$ such that $f(v_a, v_{-i}) = a$. For every ordered pair $(a, b) \in A$, where $\delta_{ab}(v_{-i})$ is not the largest element in $Im\delta_{ab}$, we add to $R$ the following type: denote with $J_{ab}$ the index of $\delta_{ab}(v_{-i})$ in $Im\delta_{ab}$. By Lemma 6.8, there exists a type $v_a$ such that $f(v_a, v_{-i}) = a$ and $v_a(a) - v_a(b) < \delta_{j+1}$. We add it to $R$. Notice that $|R| = O(|Arc|^2)$.

Let $v'_{-i} \in V_i$ be a type such that $f(v', v_{-i}) = f(v_a, v'_{-i})$ for all $a \in R$. We want to prove that price$_i(v'_{-i}) = price_i(v_{-i})$. Denote the set of reachable alternatives of $v'_{-i}$ as $A'$. By construction, all alternatives in $A$ are reachable from $v'_{-i}$ as well, so $A \subseteq A'$. Also, $f(v_0, v'_0) = f(v_0, v'_0) = a_0$, i.e. $v_0$ and $v'_0$ have the same

Now, let $M^A \neq M_{a_0}^A$ be a normalized menu vector that satisfies $M(a) - M(b) \leq \delta_{ab}(v_{-i})$ for all $a, b \in A$. It means that $f(v, v_{-i}) = a$ implies that:

$$v(a) - v(b) \geq \inf\{v(a) - v(b) \mid f(v, v_{-i}) = a\} = \delta_{ab}(v_{-i}) \geq M(a) - M(b)$$

which means that $v(a) - M(a) \geq v(b) - M(b)$. In words, the menu $M^A$ induces truthful behaviour for player $i$ given $v_{-i}$. We extend $M^A$ with the same prices as $M_{a_0}^A$, for all $a \in A$ and with arbitrary prices for all $a \notin A$ and obtain a non-restricted menu, $M$. By assumption, $M$ is normalized, so we get that it is truthful and normalized, but differs from $M_{a_0}$, which is a contradiction to the uniqueness of payments assumption. □

Denote the elements in $Im\delta_{ab}$ with $\{\delta_1 < \ldots < \delta_j < \ldots\}$. The fact that $Im\delta_{ab}$ is countable allows us to enumerate its elements.
zero alternative, so a normalized menu for $\alpha_{-i}$ necessarily satisfies $M(a_0) = 0$. Fix an ordered pair $(a, b) \in A$, and denote $\delta_{ab}(v_{-i})$ with $\delta_{ab}$. If $j_{ab} < |\text{Im}_m|$, by construction there exists $v_{ab} \in R$ such that $a = f(v_{ab}, v_{-i}) = f(v_{ab}, \alpha_{-i})$ and $v_{ab}(a) = v_{ab}(b) < \delta_{ab}$. Recall that $A \subseteq A'$, so $a$ and $b$ are reachable from $\alpha_{-i}$ as well. Hence, we can use Lemma 6.8 for $\alpha_{-i}$ and get that $\delta_{ab}(v_{-i}) \leq |\text{Im}_m|$ and that every truthful menu $M$ for $\alpha_{-i}$ satisfies that $M(a) - M(b) \leq \delta_{ab}$. If $j_{ab} = |\text{Im}_m|$, $\text{Im}_m$ is necessarily finite. By Lemma 6.7, the fact that alternatives $a$ and $b$ are reachable from $\alpha_{-i}$ implies that every truthful menu for $\alpha_{-i}$ satisfies:

$$M(a) - M(b) \leq \delta_{ab}(v_{-i}) \leq \max_{v_{-i} \in V_{-i}} \delta_{ab}(v_{-i}) = \delta_{ab}$$

Denote with $M_{\alpha_{-i}}^A$ and with $M_{\alpha_{-i}}^A$ the menus presented by the mechanism $M$ for $\alpha_{-i}$ and for $\alpha_{-i}'$, restricted to the alternatives in $A$. By Lemma 6.7, $M_{\alpha_{-i}}^A$ is the only menu that satisfies $M(a_0) = 0$ and $M(a) - M(b) \leq \delta_{ab}(v_{-i}) \leq \delta_{ab}$ for every $a, b \in A$. As we have just shown, a truthful and normalized menu $M$ for $\alpha_{-i}$ must satisfy those conditions as well. Therefore, $M_{\alpha_{-i}}^A \equiv M_{\alpha_{-i}}^A$, because otherwise we get a contradiction to the uniqueness of $M_{\alpha_{-i}}^A$. By definition, $a' \in A$, so $M_{\alpha_{-i}}^A(a') = M_{\alpha_{-i}}^A(a')$, and thus $price_{\alpha_{-i}}^A(v_{-i}) = price_{\alpha_{-i}}^A(v_{-i}')$, which completes the proof. □

Observe the following naïve proof system. Denote the most efficient protocol of $f$ with $\Pi_f$. Fix a type $\alpha_{-i}$. Notice that by Lemma 6.5, $\text{Im}_m$ is finite for all $a, b \in A$ and $f$ satisfies uniqueness of payments, so there exists a set $R \subseteq V_i$ of $O(|A|^2)$ types that satisfies the conditions of Proposition 6.6. The protocol is as follows: the prover sends all the types in $R$, and the players simulate $\Pi_f(r, v_{-i})$ for every $r \in R$. By Proposition 6.6, this process allows the players to extract $price_{\alpha_{-i}}^f(v_{-i})$. However, this naïve protocol might be too costly, because if the size of the domain $V_i$ is large, pointing to a single index in it might require too many bits.

We overcome this problem as follows. Instead of sending the types in $R$ themselves, the prover sends for each $r \in R$ the leaf in $\Pi_f$ that $(r, v_{-i})$ reaches. By that, we take advantage of the fact that the protocol $\Pi_f$ is public and has at most $2^{cc(f)}$ leaves. We denote the leaf (combinatorial rectangle) of each $r \in R$ with $L' = L'_{1} \times \cdots \times L'_{n}$, and the set of leaves sent by the prover with $L$. We also define a set of types in $V_{-i}$ that are congruent with $L$, $\text{cands}(L) = \{v_{-i} \in V_{-i} \mid \forall L \in L, v_{-i} \in L_{-i}\}$. Before outputting a price, the players verify that:

1. For every player $j \in N \setminus \{i\}$, and for every $L \in L$, $v_j \in L_j$. In other words, the leaves sent by the prover are congruent with the players’ types.
2. $\text{cands}(L) \neq \emptyset$.
3. There exists a leaf $L_0$ such that $(v_0, v_{-i}) \in L_0$. The alternative associated with this leaf is the zero alternative.
4. There exists a leaf $L \in L$ labelled with alternative $a'$.
5. For every $v_{-i}, v_{-i}' \in \text{cands}(L)$, $price_{\alpha_{-i}}^f(v_{-i}) = price_{\alpha_{-i}}^f(v_{-i}')$.

Denote this price as $price_{\alpha_{-i}}^f(L)$. Note that verifying all conditions does not require any communication between the players. If the verification fails, they reject. Otherwise, they output $price_{\alpha_{-i}}^f(L)$.

Correctness. First, we will show that for every $v_{-i}$ and every reachable $a'$, if the prover sends the set $R$ specified in Proposition 6.6, the players output $price_{\alpha_{-i}}^f(v_{-i})$. By construction, the set of leaves $L$ sent by the prover satisfies that $v_{-i} \in L_{-i}$ for all $L \in L$ because $\Pi_f(r, v_{-i})$ reaches the leaf $L_{-i}$, so conditions 1 and 2 hold. Proposition 6.6 guarantees that conditions 3 and 4 hold as well.

For condition 5, fix a type $\alpha_{-i}' \in \text{cands}(L)$. We will show that for all $r \in R, (r, v_{-i})$ and $(r, v_{-i}')$ belong in the same leaf of $\Pi_f$, and thus $f(r, v_{-i}) = f(r, v_{-i}')$. Using Proposition 6.6, it implies that $price_{\alpha_{-i}}^f(v_{-i}) = price_{\alpha_{-i}}^f(v_{-i}')$ as needed.

To this end, fix a type $r \in R$, and denote the leaf sent for it in $\Pi_f$ with $L$. By construction, $(r, v_{-i})$ reaches $L'$ and $v_{-i}' \in L_{-i}'$ because $v_{-i}' \in \text{cands}(L)$. $L'$ is a combinatorial rectangle, so using its mixing property, we get that $(r, v_{-i}')$ reaches $L'$ as well.

We still need to prove that if $a'$ is not reachable from $v_{-i}$, i.e., $v_{-i}$ violates the promise, then the players reject all witnesses for it. We also need to prove that if $price_{\alpha_{-i}}^f(v_{-i}) = p$, there is no witness that convinces the players that the price is different.

If $a'$ is not reachable from $v_{-i}$, no witness for $v_{-i}$ satisfies conditions 1 and 4 simultaneously, so the players always reject types in $V_{-i}$ that violate the promise. Fix a type $v_{-i}$ such that $price_{\alpha_{-i}}^f(v_{-i}) = p$, and denote its set of reachable alternatives with $A_f$. $f$ satisfies uniqueness of payments, so $M_{\alpha_{-i}}^A$ is the only restricted menu which is truthful and normalized for $A$. By definition, $M_{\alpha_{-i}}^A(a') = p$. If the players return a payment other than $p$, it means that there exist leaves in the protocol $\Pi_f$ that point at outcomes of $f$ that disqualify $p$ from being the price of $a'$ for $v_{-i}$ because of violations of truthfulness or normalization. If $p$ is disqualified, the menu $M_{\alpha_{-i}}^A$ is invalidated as well. However, $M_{\alpha_{-i}}^A$ is truthful and normalized, so it cannot be invalidated and we get a contradiction.

Communication. The total communication of the protocol is $O(|A|^2 \cdot cc(f))$ bits, because $R$ is of size $O(|A|^2)$ and for every type in $R$ the prover sends an index of a leaf in $\Pi_f$, using $cc(f)$ bits.

$\square$

6.2 Scalable and Convex Domains

Scalable domains satisfy unique payments (Proposition 4.6) and convex domains are basically equivalent to scalable domains, so we get the following claims for free.

Claim 6.9. Let $f : V_1 \times \cdots \times V_n \rightarrow A$ be an implementable social choice function with scalable domains. Then, $cc_{IC}(f) \leq poly(n, cc(f), |A|)$. □

Proof. By Proposition 4.6, $f$ satisfies uniqueness of payments. Hence, by Theorem 6.3, $cc_{IC}(f) \leq poly(n, |A|, cc(f))$. □

Claim 6.10. Let $f : V_1 \times \cdots \times V_n \rightarrow A$ be a social choice function with convex domains. Then, $cc_{IC}(f) \leq poly(n, cc(f), |A|)$. □

The proof is based on the same ideas as the proof of Theorem 4.8, and can be found in the full version.

ACKNOWLEDGMENTS

Work supported by BSF grant 2016192 and ISF grant 2185/19. Also, we would like to thank Uriel Feige for his valuable suggestions.
A DETERMINISTIC AND NONDETERMINISTIC COMMUNICATION COMPLEXITY

The proof of the unique payments algorithm (Theorem 6.3) relies on the polynomial relations between deterministic and non-deterministic communication complexity of promise problems. We hereby prove this assertion. We stress that despite the fact that we focus on promise problems, all the results of this section can be easily extended to non-promise problems. Usually, given a deterministic function \( f : X = X_1 \times \cdots \times X_n \rightarrow O \) and a protocol \( \pi \) we say that \( \pi \) computes \( f \) if \( f(x) = \pi(x) \) for all \( x \in X \). For promise problems, we allow \( \pi \) to err sometimes.

**Definition A.1.** (Promise Problems) Let \( f : X = X_1 \times \cdots \times X_n \rightarrow O \cup \{\ast\} \) be a function. We call \( x \) such that \( f(x) \in O \) promise inputs, and denote them with \( P \subseteq X \). We say that a deterministic protocol \( \pi \) computes \( f \) if \( \pi(x) = f(x) \) for all \( x \in P \). We say that a non-deterministic protocol computes \( f \) if it presents a valid witness for \( x \in P \), whilst all witnesses are rejected if \( f \) violates the promise. We denote with \( N(f) \) the non-deterministic communication complexity of \( f \).

Recall that a non-deterministic protocol is equivalent to a cover of inputs. Thus, we require that the cover associated with a non-deterministic protocol for \( f \) satisfies that every \( o \)-monochromatic rectangle in it does not contain \( \ast \)-inputs, i.e. inputs which violate the promise. Our goal is to prove that:

**Theorem A.2.** Let \( f : X_1 \times \cdots \times X_n \rightarrow O \cup \{\ast\} \) be a promise function. Then, there exists a deterministic protocol that computes \( f \) using \( \text{poly}(n, N(f)) \) bits.

To this end, we define for each \( o \in O \) a verifier function which is a promise problem as well: \( V_o^f : X \rightarrow \{0, 1, \ast\} \). \( V_o^f \) has the same promise inputs as the original \( f \). It outputs 1 if \( f(x) = o \), 0 if \( f(x) \in O \) but differs from \( o \) and \( \ast \) if \( x \notin P \). We define the most costly verifier with \( V_f \), i.e. \( V_f = \arg \max_{o \in O} \text{cc}(V_o^f) \). We prove the theorem by combining the two following propositions:

**Proposition A.3.** Let \( f : X_1 \times \cdots \times X_n \rightarrow O \cup \{\ast\} \) be a promise problem. Then, \( \text{cc}(f) \leq O(n^2 (\log^2 |O| + \log |O| \cdot \text{cc}(V_f) + \text{cc}^2(V_f))) \).

The proof of Proposition A.3 is by a reduction to unique-disjointness, followed by executing the unique-disjointness algorithm proposed by [7, 11]. The proof is relegated to the full version.

**Proposition A.4.** Let \( f : X_1 \times \cdots \times X_n \rightarrow \{0, 1, \ast\} \) be a boolean promise function. Then, there exists a deterministic protocol that computes \( f \) using \( O(n^2 \cdot N^9(f) \cdot N^4(f)) \). Thus, \( \text{cc}(f) \leq \text{poly}(n, N(f)) \).

We prove the proposition by showing that the proof in the two party model in [12, Theorem 2.11] can be extended to a multi-player promise setting. Dolev and Feder [8] provide a similar result, but they do not address the promise scenario (details appear in the full version of the paper).

**Proof of Theorem A.2.** By Proposition A.3, we get that:

\[
\text{cc}(f) \leq O(n^2 (\log^2 |O| + \log |O| \cdot \text{cc}(V_f) + \text{cc}^2(V_f)))
\]

\( N(f) \geq \log |O| \), so we only need to upper bound \( \text{cc}(V_f) \). By Proposition A.4, \( \text{cc}(V_f) \leq \text{poly}(n, N(V_f)) \). By definition, \( N(V_f) \leq N(f) \) because a non-deterministic protocol for \( f \) can be used for the non-deterministic computation of every \( V_o^f \) since \( f \) and \( V_o^f \) share the same promise \( P \subseteq X \). Therefore:

\[
\text{cc}(f) \leq O(n^2 (\log^2 |O| + \log |O| \cdot \text{cc}(V_f) + \text{cc}^2(V_f)))
\]

\[
\leq \text{poly}(n, N(V_f), \text{cc}(V_f))
\]

\[
\leq \text{poly}(n, N(f), N(V_f))
\]

(22)

where (22) holds due to Proposition A.4. □

**REFERENCES**

[1] Archer, A., Papadimitriou, C., Talwar, K., and Tardos, É. An approximate truthful mechanism for combinatorial auctions with single parameter agents. Internet Mathematics 1, 2 (2004), 129–150.

[2] Archer, A., and Tardos, É. Truthful mechanisms for one-parameter agents. In Proceedings 42nd IEEE Symposium on Foundations of Computer Science (2001), IEEE, pp. 482–491.

[3] Ashish, S., Khandeparkar, H., Saxena, R. R., and Weinberg, S. M. Separating the communication complexity of truthful and non-truthful combinatorial auctions. In Proceedings of the 32nd Annual ACM SIGACT Symposium on Theory of Computing (2020), pp. 1073–1085.

[4] Babaioff, M., Blumrosen, L., Naor, M., and Schapira, M. Informational overhead of incentive compatibility. In Proceedings of the 9th ACM conference on Electronic commerce (2008), pp. 88–97.

[5] Babaioff, M., Kleinberg, R. D., and Slivkins, A. Truthful mechanisms with implicit payment computation. Journal of the ACM (JACM) 62, 2 (2015), 1–37.

[6] Bikhchandani, S., Chatterji, S., Lavi, R., Mu’alem, A., Nisan, N., and Sun, A. Weak monotonicity characterizes deterministic dominant-strategy implementable mechanisms. Econometrica 74, 4 (2006), 1109–1132.

[7] Dobzinski, S. Computational efficiency requires simple taxation. In 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS) (2016), IEEE, pp. 209–218.

[8] Dolev, D., and Feder, T. Determinism vs. nondeterminism in multiparty communication complexity. SIAM Journal on Computing 21, 5 (1992), 889–895.

[9] Faigle, U., and K lays. I. The communication cost of selfishness. Journal of Economic Theory 144, 5 (2009), 1895–1920.

[10] Heydenreich, M., Müller, R., Uetz, M., and Vohra, R. V. Characterization of revenue equivalence. Econometrica 77, 1 (2009), 307–316.

[11] Kärcher, M., Newman, I., Saks, M., and Wigderson, A. Non-deterministic communication complexity with few witnesses. Journal of Computer and System Sciences 49, 2 (1994), 247–257.

[12] Kushilevitz, E., and Nisan, N. Communication Complexity. Cambridge University Press, 1996.

[13] Myerson, R. B. Optimal auction design. Mathematics of operations research 6, 1 (1981), 58–73.

[14] Nisan, N. Introduction to Mechanism Design (for Computer Scientists). Cambridge University Press, 2007, p. 209–242.

[15] Rochet, J.-C. A necessary and sufficient condition for rationalizability in a quasi-linear context. Journal of Mathematical Economics 16, 2 (1987), 191–200.

[16] Rubinstein, A., Saxena, R. R., Thomas, C., Weinberg, S. M., and Zhao, J. Exponential communication separations between notions of selfishness. In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (2021).

[17] Saks, M., and Yu, L. Weak monotonicity suffices for truthfulness on convex domains. In Proceedings of the 6th ACM Conference on Electronic Commerce (New York, NY, USA, 2005), EC ’05, Association for Computing Machinery, p. 286–293.