Discussion Papers in Economics

Local Incentive Compatibility with Transfers.

Debasis Mishra, Anup Pramanik, and Souvik Roy

January 2016

Discussion Paper 16-01

Indian Statistical Institute, Delhi
Economics and Planning Unit
7, S. J. S. Sansanwal Marg, New Delhi 110016, India

Local Incentive Compatibility with Transfers [*]

Debasis Mishra, Anup Pramanik, and Souvik Roy [†]

January 20, 2016

Abstract

We consider locally incentive compatible mechanisms with deterministic allocation rules and transfers with quasilinear utility. We identify a rich class of type spaces, which includes the single peaked type space, where local incentive compatibility does not imply incentive compatibility. Our main result shows that in such type spaces, a mechanism is locally incentive compatible and *payment-only incentive compatible* if and only if it is incentive compatible. Payment-only incentive compatibility requires that a mechanism that generates the same allocation at two types must have the same payment at those two types. Our result works on a class of *ordinal type spaces*, which are generated by considering a set of ordinal preferences over alternatives and then considering all non-negative type vectors representing such preferences.

JEL Codes: D82, D44, D47

Keywords: local incentive compatibility; payment-only incentive compatibility; single peaked type space; ordinal type space

[*]We are grateful to Gabriel Carroll, Francois Maniquet, Benny Moldovanu, Rudolf Muller, Arunava Sen and seminar audience at Bonn University, CORE, Indian Statistical Institute (Delhi), Maastricht University for useful comments.

[†]Debasis Mishra - Indian Statistical Institute, New Delhi, Email: dmishra@isid.ac.in; Anup Pramanik - Institute of Social and Economic Research, Osaka University, Email: anup.isid@gmail.com; Souvik Roy - Indian Statistical Institute, Kolkata, souvik_roy2004@yahoo.com

1 Introduction

The fundamental principle in mechanism design is that an agent should have no incentive to report any type in the type space other than his true type. However, in many settings, verifying all possible incentive constraints of a mechanism may be too demanding. Further, a reasonable behavioral assumption is that an agent of a particular type only misreports types in a neighborhood of his true type which is strictly smaller than the entire type space. This paper is concerned with weakening incentive compatibility to *local* incentive compatibility in mechanism design problems with deterministic allocation rules and transfers with quasilinear utility.

In an important contribution, Carroll (2012) shows that in convex type spaces every locally incentive compatible mechanism is incentive compatible - his result also works with randomized allocation rules. While convexity may be a reasonable assumption that is satisfied by many type spaces, there are important type spaces that are non-convex. For instance, if types are such that the underlying ordinal preferences over alternatives satisfy single-peaked property, we get a non-convex type space. We give an example to show that the result in Carroll (2012) is no longer true in such non-convex type spaces.

However, we show that this break down is only due to violation of a particular type of (non-local) incentive constraint. We consider the notion of payment-only incentive compatibility. Suppose s and t are two types where the allocation decision of a mechanism is the same. Payment-only incentive compatibility requires that an agent of type s should have no incentive to report t to this mechanism. Since types s and t have identical allocation decisions, payment-only incentive compatibility can be equivalently stated as requiring that the payment decisions at s and t should also be the same. It is well known that payment-only incentive compatibility is necessary for incentive compatibility. However, it is not implied by local incentive compatibility.

We identify a large class of non-convex type spaces where local incentive compatibility along with payment-only incentive compatibility is equivalent to incentive compatibility. [1] We call such type spaces *top connected ordinal type spaces*. Ordinal type spaces are defined by taking a set of permissible strict linear orders over alternatives and considering the closure of all non-negative type vectors that represent these orders. Thus, restriction on the underlying set of strict linear orders translates to restrictions on type spaces. Top connectedness is a technical property on the permissible set of strict linear orders.

Two familiar ordinal domain restrictions satisfy our top connectedness property: (a) set of all single peaked orderings and (b) a particular set of single crossing orderings. In other

[1] We also require a mild range condition on the allocation rule.

words, in the type space constructed by taking the closure of all types representing the strict linear orders in these domains, local incentive compatibility and payment-only incentive compatibility will imply incentive compatibility.

While single peakedness and single crossing property are well motivated restrictions on preferences in ordinal environment, they are equally important restrictions in our cardinal environment. For instance, single peaked type restriction may make sense in certain scheduling problems and location problems. Consider a scheduling problem, where a supplier is deciding on delivery of products to various firms on days of the week. Each firm will have a *best* day of the week and as the delivery day goes away from the best day, the value of the firm for the products is likely to decrease because of delivery delay costs or inventory-holding costs. Note that we are imposing these restrictions on the underlying preferences over alternatives (delivery dates in this example) - preferences over transfers is still determined using quasi-linear utility functions. Also, our results apply to both public and private good allocation problems in quasilinear setting. [2]

An allocation rule is said to be implementable if there exists a payment function such that the resulting mechanism is incentive compatible. Well known results on revenue equivalence (Chung and Olszewski, 2007) in this setting implies that for every implementable rule, a payment rule that makes it incentive compatible is uniquely determined up to an additive constant. Hence, the question of verifying whether a mechanism is incentive compatible or not can be broken down into two parts: (a) verify whether the allocation rule is implementable or not and (b) check if the payment rule corresponds to the payment rule prescribed by the revenue equivalence formula. As a result, the local incentive compatibility question of a mechanism can be rephrased in terms of local implementability.

A well known result in Rockafellar (1970) and Rochet (1987) shows that implementability is equivalent to a condition called *cycle monotonicity*. With deterministic allocation rules, a weaker cycle monotonicity condition, called 2-cycle monotonicity, is known to be necessary and sufficient for implementability in convex type spaces (Saks and Yu, 2005; Ashlagi et al., 2010) and in single peaked type spaces (Mishra et al., 2014). Archer and Kleinberg (2008) define local implementability of an allocation rule as follows: an allocation rule is locally implementable if for every type in the type space, it is 2-cycle monotone in an arbitrary neighborhood around that type. In a result parallel to Carroll (2012), Archer and Kleinberg (2008) show that local implementability implies implementability in convex type spaces - see

[2]An example of a public good provision problem where agents have single peaked types is locating a public facility on a street. Again, it is not difficult to think that transfers are involved in these problems - in the form of tax or subsidy to participating agents.

also Archer and Kleinberg (2014). [3]

We show that in any polygonally connected type space, if local implementability implies implementability, then local incentive compatibility implies incentive compatibility. This implies that we cannot hope to extend our main result if we use the notion of local implementability. In particular, in the single peaked type space, a local implementable allocation rule may not be implementable.

Relation to Literature. In the auction design literature with transfers and quasilinear utility, a long standing research agenda has been to identify a minimal set of incentive constraints that will imply overall incentive compatibility - see discussions on relaxed problem in Chapter 7 of Fudenberg and Tirole (1991), Armstrong (2000), and Chapter 6 in Vohra (2011). While most of this literature was not very explicit about the local versus global incentive compatibility problem, Carroll (2012) was the first to provide a general definition and a result in a broad set of mechanism design settings. The analogous results for local implementability were shown in Archer and Kleinberg (2008) - see also Archer and Kleinberg (2014). Berger et al. (2010) show that the results in Archer and Kleinberg (2008) can be extended to certain connected type spaces under strong additional technical conditions. A recent paper by Fotakis and Zampetakis (2013) also show that the local implementability result in Archer and Kleinberg (2008) can be extended to some non-convex type spaces. The type spaces discussed in our paper is not related to theirs. Moreover, they discuss local implementability and our main focus is on type spaces where local incentive compatibility does not imply incentive compatibility.

The local to global incentive compatibility issue is also central in the principal-agent literature, but the central issue in both the literature is quite different. In the principal-agent literature, the incentive constraints are complicated because of the structure of the utility function and uncertainty over the outcome space - see a recent take on this in Kirkegaard (2014). On the other hand, the incentive constraints in our paper is complicated due to the presence of multidimensional type spaces.

The rest of the paper is organized as follows. We introduce the notion of local incentive compatibility formally in Section 2 followed by a motivating example in Section 2.1. Section 3 introduces the notion of payment-only incentive compatibility and states the main result with a detailed description of type spaces where the main result works. Section 4 discusses the notion of local implementation. We conclude in Section 5. All the omitted proofs are

[3]Unlike Carroll (2012), this result requires the allocation rule to be deterministic. For randomized allocation rules, Archer and Kleinberg (2008) consider a stronger version of local implementability, and show it to be sufficient for implementability in convex type spaces.

relegated to an Appendix at the end.

2 Local Incentive Compatibility

There is a single agent and a finite set of alternatives A. [4] The type of the agent is a vector in $\mathbb{R}^{|A|}$. The type space (possible set of types) of the agent is a set $T \subseteq \mathbb{R}^{|A|}$. If the agent has type $t \in T$, then his valuation for any alternative $a \in A$ is denoted by $t(a)$.

An **allocation rule** is a map $f : T \to A$. Note that we only consider deterministic allocation rules. A **payment rule** is a map $p : T \to \mathbb{R}$. A (direct) **mechanism** M consists of an allocation rule f and a payment rule p. If the agent has type t but reports s to the mechanism, then his net utility is given by

$$t(f(s)) - p(s),$$

where we assumed quasilinearity to evaluate utility from payments.

An agent of type t cannot **manipulate** to a type s if

$$t(f(t)) - p(t) \geq t(f(s)) - p(s).$$

Definition 1 *A mechanism (f, p) is* **incentive compatible** *if for all $t \in T$, t cannot manipulate to s for all $s \in T$.*

To define *local* incentive compatibility, we define the notion of a neighborhood. For every $\epsilon > 0$ and every $t \in T$, let $B^\epsilon(t) := \{s \in T : ||s - t|| < \epsilon\}$ be the open $|A|$-dimensional ball around t contained in T.

Definition 2 *A mechanism $M \equiv (f, p)$ is* **locally incentive compatible** *if for every $t \in T$ there exists an $\epsilon > 0$ such that for all $s \in B^\epsilon(t)$, t cannot manipulate to s and s cannot manipulate to t.*

Local incentive compatibility requires that for every type t, there is an open ball $B^\epsilon(t)$, where ϵ can depend on t, such that the (pair of) incentive constraints between t and every type in $B^\epsilon(t)$ hold. This notion of local incentive compatibility was introduced in Carroll (2012). [5] He showed that if a type space is convex, then local incentive compatibility implies incentive compatibility. We start off by giving an example of an important non-convex type space, where local incentive compatibility is no longer equivalent to incentive compatibility.

[4] This is without loss of generality since all the results generalize, albeit with extra notation, to a setting with multiple agents.

[5] Naturally, one can define other plausible notions of local incentive compatibility too. For discussions on how this particular definition compares to some other notions, see Carroll (2012).

2.1 A Motivating Example

Consider the following type space. Let $\succ$ be a strict linear order over the set of alternatives. A strict preference ordering P is **single peaked** with respect to $\succ$ if for every $a, b \in A$ with $b \succ a \succ P(1)$ or $P(1) \succ a \succ b$, where $P(1)$ is the highest ranked alternative in P, we have aPb. A type $t \in \mathbb{R}^{|A|}$ represents an ordering P if for all $a, b \in A$, $t(a) > t(b)$ if and only if aPb. Let $\mathcal{D}^{\succ}$ denote the set of all single peaked preference orderings and $T^{\succ}$ be the closure of all non-negative types representing types in $\mathcal{D}^{\succ}$.

As discussed earlier, the single peaked type space $T^{\succ}$ is a natural type space in many problems. We show that in the single peaked type space, local incentive compatibility does not imply incentive compatibility. [6]

Example **1**

Let $A = \{a, b, c\}$ and $\mathcal{D}^{\succ}$ be the following domain of orderings.

P^1	P^2	P^3	P^4
a	b	b	c
b	a	c	b
c	c	a	a

Note that $\mathcal{D}^{\succ}$ is single peaked with respect to $a \succ b \succ c$. We consider the following mechanism on $T^{\succ}$. We partition $T^{\succ}$ as $T^1 \cup T^2 \cup T^3 \cup T^4 \cup T^5 \cup T^6$, where each T^j, $j \in \{1, \ldots, 6\}$ is defined below in Table 1 along with the values of f and p in them.

Notice that the mechanism in Table 1 in not incentive compatible - types in T^1 and T^6 both get the alternative a, but types in T^1 pay 2 and types in T^6 pay zero. However, types in T^1 and T^6 are not *local* to each other, which allows us to show that this mechanism is locally incentive compatible.

Claim **1** *The mechanism (f, p) in Table 1 is locally incentive compatible but not incentive compatible.*

The proof of Claim 1 is in the Appendix. This example illustrates that there are important type spaces where local incentive compatibility *does not* imply incentive compatibility. In our

[6] Carroll (2012) provides a necessary condition on type spaces where every *randomized* locally incentive compatible mechanism is incentive compatible. This necessary condition is violated by the single peaked type space. Since we restrict attention to deterministic mechanisms, this necessary condition cannot be applied in our setting. Example 1 directly shows that local incentive compatibility does not imply incentive compatibility in single peaked type space.

type space	$f(\cdot)$	$p(\cdot)$
$T^1 := \{t : t(a) - t(b) \geq 2 \text{ and } t(a) > t(b) \geq t(c)\}$	a	2
$T^2 := \{t : t(a) - t(b) < 2 \text{ and } t(a) > t(b) \geq t(c)\}$	b	0
$T^3 := \{t : t(b) \geq \max(t(a), t(c))\}$	b	0
$T^4 := \{t : t(c) - t(b) \geq 1 \text{ and } t(c) > t(b) \geq t(a)\}$	c	1
$T^5 := \{t : t(c) - t(b) < 1 \text{ and } t(c) \geq t(b) > t(a)\}$	b	0
$T^6 := \{t : t(c) - t(b) < 1 \text{ and } t(c) > t(b) = t(a)\}$	a	0

Table 1: A locally incentive compatible mechanism

main result in the next section, we introduce a notion of non-local incentive compatibility, which along with local incentive compatibility implies incentive compatibility in such type spaces.

3 The Main Result

We state and prove our main result in this section. We start out by introducing a new notion of (non-local) incentive compatibility.

Definition 3 *A mechanism (f, p) is* **payment-only incentive compatible** *if for every pair of types s, t with $f(s) = f(t)$, we have*

$$t(f(t)) - p(t) \geq t(f(s)) - p(s).$$

It is easy to verify that payment-only incentive compatibility is equivalent to the following condition on payments.

Observation 1 *A mechanism (f, p) is payment-only incentive compatible if and only if for every pair of types s, t with $f(s) = f(t)$, we have $p(s) = p(t)$.*

Notice that an incentive compatible mechanism is payment-only incentive compatible. Obviously, payment-only incentive compatibility is not a local condition as it takes care of some non-local incentive constraints also. However, its implications from a design perspective is innocuous - it requires the domain of the payment function to be the set of alternatives instead of the type space. In other words, local incentive compatibility along with payment-only incentive compatibility still requires us to verify only local incentive constraints once the domain of the payment function is changed. Thus, the nature of the verification problem for the designer remains the same, though the verification of the mechanism is now easier.

3.1 Top Connected Domains

We now identify a class of type spaces where local incentive compatibility along with payment-only incentive compatibility implies incentive compatibility. We give two important non-convex type spaces that are covered by our general result.

To introduce our type spaces, we state below some definitions. Let $\mathcal{P}$ the set of all strict orderings over A. A **domain** is any subset $\mathcal{D} \subseteq \mathcal{P}$. A type $t \in \mathbb{R}^{|A|}$ is strict if $t(a) \neq t(b)$ for all $a, b \in A$. Every strict type induces an ordering in $\mathcal{P}$. We say a strict type t represents an ordering $P \in \mathcal{P}$ if for every $a, b \in A$, $t(a) > t(b)$ if and only if aPb. For any ordinal domain $\mathcal{D} \subseteq \mathcal{P}$, let $V(\mathcal{D})$ be the set of all strict types in $\mathbb{R}^{|A|}_{+}$ representing orderings in $\mathcal{D}$ - note that though we required our strict type to be non-negative, we can state and prove our results without this restriction. A type space T is an **ordinal type space** if there exists a domain $\mathcal{D}$ such that $T = cl(V(\mathcal{D}))$, where $cl(V(\mathcal{D}))$ is the closure of the set $V(\mathcal{D})$. Hence, an ordinal type space consists of the closure of union of a finite number of cones in $\mathbb{R}^{|A|}$, with each cone representing an ordering in the domain.

Given an ordering P and an alternative a, we denote by $r(P, a)$ the rank of a in P. A sequence of orderings $(P^1, \ldots, P^k)$ is a-improving for an alternative a if for every $j \in \{1, \ldots, k-1\}$, $r(P^j, a) \geq r(P^{j+1}, a)$.

We say two orderings P and P' are **adjacent** if there exists $a, b \in A$ such that $r(P, a) = r(P, b) + 1$, $r(P', a) = r(P, b)$, $r(P', b) = r(P, a)$, and for all $c \in A \setminus \{a, b\}$, $r(P, c) = r(P', c)$.

A sequence of orderings $(P^1, \ldots, P^k)$ is **connected** if for every $j \in \{1, \ldots, k-1\}$, P^j and P^{j+1} are adjacent.

Definition 4 *A domain $\mathcal{D}$ is* **top connected** *if for every $a \in A$, there exists a $P \in \mathcal{D}$ such that*

- **Richness.** $r(P, a) = 1$ *and*
- **Monotone Connectedness.** *for every $P' \in \mathcal{D}$, there exists an a-improving connected sequence $(P' \equiv P^1, \ldots, P^k \equiv P)$.*

We give an example to illustrate top-connectedness. Let $A = \{a, b, c, d\}$ and consider the set of preference orderings shown in Table 2. Notice that for every $x \in A$, there is a preference ordering P in Table 2, where $r(P, x) = 1$. Further, for any $P' \neq P$, we can construct a sequence of connected orderings such that the position of x is improving in the sequence. As an example, consider $d \in A$ and note that $r(P^8, d) = 1$. Now, consider P^5 and note that $r(P^5, d) = 3$. But consider the sequence (P^5, P^7, P^8) and notice that

P^1	P^2	P^3	P^4	P^5	P^6	P^7	P^8
a	b	b	b	c	c	c	d
b	a	c	c	b	b	d	c
c	c	a	d	d	a	b	b
d	d	d	a	a	d	a	a

Table 2: Top connected domain.

$r(P^5, d) > r(P^7, d) > r(P^8, d)$. Further P^8 and P^7 are adjacent and P^7 and P^5 are adjacent.

We discuss some specific domains that are top connected later. We require the following range condition for our main result.

Definition 5 *An allocation rule $f : T \to A$ satisfies* **strict type range (STR)** *condition if for every $a \in A$, there is a strict type $t \in T$ such that $f(t) = a$.*

Note that STR is stronger than ontoness since we require the alternative to be chosen at some *strict* type. Before we state our main result, we state a notion of local incentive compatibility that is commonly used in ordinal voting models without transfers (Carroll, 2012; Sato, 2013a,b), but it is relevant for our ordinal type spaces also.

Definition 6 *A mechanism (f, p) defined on an ordinal type space $cl(V(\mathcal{D}))$ is* **adjacent incentive compatible** *if for every $P, P' \in \mathcal{D}$ such that P and P' are adjacent, and for every $s, t \in cl(V(\{P, P'\}))$, t cannot manipulate to s.*

Adjacent incentive compatibility is a notion of local incentive compatibility that is specific to our ordinal type spaces. It requires that a type in a cone corresponding to a specific ordering cannot manipulate to a type in a cone corresponding to the same ordering or an adjacent ordering. As we show below in our main result, this is equivalent to local incentive compatibility in top connected type spaces.

Theorem 1 *Suppose $\mathcal{D}$ is a top connected domain and $T = cl(V(\mathcal{D}))$. Let $f : T \to A$ be a deterministic allocation rule satisfying STR and (f, p) be a mechanism defined on type space T. Then, the following statements are equivalent.*

1. *(f, p) is locally incentive compatible and payment-only incentive compatible.*

2. *(f, p) is adjacent incentive compatible and payment-only incentive compatible.*

3. *(f, p) is incentive compatible.*

Proof: Obviously, 3 $\Rightarrow$ 1. We only show 1 $\Rightarrow$ 2 and 2 $\Rightarrow$ 3. For 1 $\Rightarrow$ 2, consider the following elementary fact.

FACT **1** *Suppose $P, P' \in \mathcal{D}$ are adjacent. Then, $cl(V(\{P, P'\}))$ is a convex set.*

Proof: Since P and P' are adjacent, there exists $a, b \in A$ such that $r(P, a) = r(P, b) + 1$, $r(P', a) = r(P, b)$, $r(P', b) = r(P, a)$, and for all $c \in A \setminus \{a, b\}$, $r(P, c) = r(P', c)$. Now, choose $s, t \in cl(V(\{P, P'\}))$. By definition, s and t only differ in their ranking of a and b, which are consecutively ranked in P and P'. As a result, any convex combination of s and t will lie in $cl(V(\{P, P'\}))$. ■

Now, let (f, p) be a locally incentive compatible mechanism in $T = cl(V(\mathcal{D}))$, where $\mathcal{D}$ is top connected. Now, choose any $P, P' \in \mathcal{D}$ such that P, P' are adjacent. By Carroll (2012), we know that local incentive compatibility in a convex type space implies incentive compatibility. Hence, (f, p) restricted to $cl(V(\{P, P'\}))$ is incentive compatible by Fact 1. As a result, (f, p) is adjacent incentive compatible.

We now show 2 $\Rightarrow$ 3. Let (f, p) be an adjacent incentive compatible and payment-only incentive compatible mechanism. Since (f, p) is payment-only incentive compatible, we have $p(t) = p(t')$ for all $t, t' \in T$ with $f(t) = f(t')$. Hence, with a slight abuse of notation, we write p as a map $p : A \to \mathbb{R}$. Consider any type $s \in T$ and let $f(s) = a$. Choose an arbitrary $b \in A$. We need to show that

$$s(a) - s(b) \geq p(a) - p(b).$$

We do the proof in two steps.

STEP 1. In this step, we show that there exists an ordering $\bar{P} \in \mathcal{D}$ such that $r(\bar{P}, b) = 1$ and a type $\bar{t}$ representing $\bar{P}$ such that $f(\bar{t}) = b$.

By STR, there is a strict type $t \in T$ such that $f(t) = b$. Let $\tilde{P} \in \mathcal{D}$ be an ordering such that $t \in V(\{\tilde{P}\})$. By top-connectedness, there exists an ordering $\bar{P}$ such that $r(\bar{P}, b) = 1$ and a b-improving connected sequence $(\tilde{P} \equiv P^1, \ldots, P^k \equiv \bar{P})$.

Since P^1 and P^2 are adjacent, there is some pair of alternatives $x, y \in A$ such that $r(P^1, y) = r(P^1, x) + 1$, $r(P^2, x) = r(P^1, y)$, $r(P^2, y) = r(P^1, x)$, and $r(P^2, z) = r(P^1, z)$ for all $z \in A \setminus \{x, y\}$. Note that since the sequence is b-improving, $x \neq b$. Let $t^1 \equiv t$ and construct t^2 as follows by choosing $\epsilon > 0$ but sufficiently close to zero:

$$t^2(z) = \begin{cases} t^1(z) & \text{if } z \notin \{x, b\}, \\ t^1(z) + \epsilon & \text{if } z = b, \\ t^1(y) - \epsilon & \text{if } z = x. \end{cases}$$

Observe that t^2 represents P^2 - t^2 is constructed by lowering $t^1(x)$ to $t^1(y) - \epsilon$, increasing $t^1(b)$ by ϵ and keeping all values unchanged from t^1. Further, t^2 can be constructed since t^1 is a strict type.

Since P^1 and P^2 are adjacent, by adjacent incentive compatibility,

$$t^1(b) - p(b) \geq t^1(f(t^2)) - p(f(t^2))$$
$$t^2(f(t^2)) - p(f(t^2)) \geq t^2(b) - p(b).$$

Adding these incentive constraints, we get

$$t^2(f(t^2)) - t^1(f(t^2)) \geq t^2(b) - t^1(b) = \epsilon.$$

By definition of t^2, the above inequality can only be satisfied if $f(t^2) = b$. Hence, $f(t^2) = b$.

Now, we repeat this procedure inductively. For some $j \in \{1, \ldots, k-1\}$, suppose we have found a type t^j representing the ordering P^j such that $f(t^j) = b$. Since P^j and P^{j+1} are adjacent, there is some pair of alternatives $x, y \in A$ such that $r(P^j, y) = r(P^j, x) + 1$, $r(P^{j+1}, x) = r(P^j, y)$, $r(P^{j+1}, y) = r(P^j, x)$, and $r(P^{j+1}, z) = r(P^j, z)$ for all $z \in A \setminus \{x, y\}$. Note that since the sequence is b-improving, $x \neq b$. We construct t^{j+1} as follows by choosing $\epsilon > 0$ but sufficiently close to zero:

$$t^{j+1}(z) = \begin{cases} t^j(z) & \text{if } z \notin \{x, b\}, \\ t^j(z) + \epsilon & \text{if } z = b, \\ t^j(y) - \epsilon & \text{if } z = x. \end{cases}$$

Again, using an argument similar to above with adjacent incentive compatibility (by adding incentive constraints corresponding to t^j and t^{j+1}), we can show that $f(t^{j+1}) = b$. Using induction, we thus conclude that there is a type $\bar{t} \equiv t^k$ representing $\bar{P} \equiv P^k$ such that $f(\bar{t}) = b$. Note that $r(\bar{P}, b) = 1$.

Step 2. We complete the proof in this step. Let P be an ordering such that $s \in cl(V(\{P\}))$. Let s^1 be a strict type representing P such that s^1 is arbitrarily close to s. Let $f(s^1) = a_1$. Since s^1 is arbitrarily close to s and represents the same ordering as s, adjacent incentive compatibility implies that

$$s(a) - s(a_1) \geq p(a) - p(a_1). \tag{1}$$

By top connectedness, there is a b-improving connected sequence $(P \equiv P^1, \ldots, P^k \equiv \bar{P})$. Now, we construct a sequence of types $(s^1, \ldots, s^k)$ such that for all $j \in \{1, \ldots, k\}$, s^j represents P^j.

Since P^1 and P^2 are adjacent, there is some pair of alternatives $x, y \in A$ such that $r(P^1, y) = r(P^1, x) + 1$, $r(P^2, x) = r(P^1, y)$, $r(P^2, y) = r(P^1, x)$, and $r(P^2, z) = r(P^1, z)$ for all $z \in A \setminus \{x, y\}$. Note that since the sequence is b-improving, $x \neq b$. Now, construct s^2 as follows by choosing $\epsilon > 0$ but sufficiently close to zero:

$$s^2(z) = \begin{cases} s^1(z) & \text{if } z \neq x, \\ s^1(y) - \epsilon & \text{if } z = x. \end{cases}$$

Observe that s^2 represents P^2 and $s^1(z) \geq s^2(z)$ for all $z \in A$ with strict inequality holding for $z = x$ and equality holding for $z \neq x$. Since s^1 is a strict type, we can construct s^2 by lowering $s^1(x)$ to a value just below $s^1(y)$ and keeping all other values of alternatives the same.

Denote $f(s^2) \equiv a_2$. Since P^1 and P^2 are adjacent, adjacent incentive compatibility gives,

$$s^1(a_1) - s^1(a_2) \geq p(a_1) - p(a_2).$$

We now construct the sequence inductively using the above procedure - having defined s^j, we define s^{j+1} by exactly the same procedure as we did for constructing s^2 from s^1. Let $f(s^j) \equiv a_j$ for all $j \in \{1, \ldots, k\}$. As before incentive compatibility implies for every $j \in \{1, \ldots, k-1\}$,

$$s^j(a_j) - s^j(a_{j+1}) \geq p(a_j) - p(a_{j+1}).$$

Adding over all $j \in \{1, \ldots, k-1\}$ and telescoping the right hand side, we get

$$\sum_{j=1}^{k-1} \left[s^j(a_j) - s^j(a_{j+1})\right] = \left[s^1(a_1) - s^{k-1}(a_k)\right] + \sum_{j=1}^{k-2} \left[s^{j+1}(a_{j+1}) - s^j(a_{j+1})\right] \geq p(a_1) - p(a_k).$$

By construction, $s^j(a_{j+1}) \geq s^{j+1}(a_{j+1})$ for all $j \in \{1, \ldots, k-2\}$. Hence, the above inequality implies

$$s^1(a_1) - s^{k-1}(a_k) \geq p(a_1) - p(a_k). \tag{2}$$

Now, both s^k and $\bar{t}$ represent $\bar{P}$. Hence, adjacent incentive compatibility gives,

$$s^k(a_k) - s^k(b) \geq p(a_k) - p(b), \tag{3}$$

where we used the fact that $f(\bar{t}) = b$. Adding Inequalities 2 and 3, and using the fact that $s^k(a_k) \leq s^{k-1}(a_k)$, we get

$$s^1(a_1) - s^k(b) \geq p(a_1) - p(b). \tag{4}$$

Since $(P \equiv P^1, \ldots, P^k \equiv \bar{P})$ is a b-improving connected sequence, by construction $s^j(b) = s^{j+1}(b)$ for all $j \in \{1, \ldots, k-1\}$. As a result, we have $s^1(b) = s^k(b)$ and using Inequality 4, we get

$$s^1(a_1) - s^1(b) \geq p(a_1) - p(b). \tag{5}$$

Adding Inequalities 5 and 1, we get

$$s(a) + [s^1(a_1) - s(a_1)] - s^1(b) \geq p(a) - p(b). \tag{6}$$

This can be rewritten as

$$[s(a) - s(b)] + [s^1(a_1) - s(a_1)] - [s^1(b) - s(b)] \geq p(a) - p(b).$$

Since s^1 is arbitrarily close s, Inequality 6 reduces to

$$s(a) - s(b) \geq p(a) - p(b),$$

which is the required incentive compatibility condition. ■

3.2 Remarks on Theorem 1

We present some brief remarks on Theorem 1.

Richness of Type Space. Our type space is the closure of the set of all non-negative types representing a top-connected domain $\mathcal{D}$. This ensures some amount of richness in the type space. In Carroll (2012), richness is achieved by assuming convexity of the type space.

At the same time, the proof of Theorem 1 goes through even if we do not include *all* the types representing $\mathcal{D}$. Suppose there is an upper bound β on the maximum value on any alternative. Then, T can be defined as

$$T := \{t \in cl(V(\mathcal{D})) : \max_{a \in A} t(a) \leq \beta\}.$$

Theorem 1 holds if the type space is modified to be T.

How Useful are Ordinal Type Spaces? Top connectedness is a general condition satisfied by many domains. We explicitly describe some interesting domains in the next section - this includes the single peaked type space discussed earlier. The conventional way of imposing type space restrictions in mechanism design (in quasilinear environment) is to impose geometric restrictions - for instance, connectedness or convexity. In that sense, our ordinal type space restriction is a novelty. However, we do not see any reason to believe why either restriction is more compelling than others. Given that preferences over transfers are separable using quasilinearity, it sounds plausible that agents first think of ordinal restriction on the preferences over alternatives and then consider cardinal types that respect this ordinal restriction.

Imposing such restrictions on preferences is standard in mechanism design literature without money (Barbera, 2010). Introducing transfers in some of those problems is a natural extension. For instance, consider the celebrated single peaked domain discussed in Moulin (1980) and Sprumont (1991). Application of this problem includes various public and private good allocation problems - e.g., locating a public facility on a street, allocating pollution limits to firms in a country, and various scheduling problems (as we discussed earlier). Our type space considers the same problems but with transfers and quasilinear utility. Recent papers which discuss mechanism design in such type spaces include Mishra et al. (2014); Carbajal and Muller (2015).

The Range Condition. We now present an example to show that the STR condition is required for Theorem 1 to hold. The following example gives a mechanism in the single peaked type space that violates STR and incentive compatibility, but satisfies local incentive compatibility and payment-only incentive compatibility.

Example **2**

Let $A = \{a, b, c\}$ and $\mathcal{D}$ be the following domain.

P^1	P^2	P^3	P^4
a	b	b	c
b	a	c	b
c	c	a	a

We consider the following mechanism on $cl(V(\mathcal{D}))$.

$$f(t) = \begin{cases} a & \text{if } t(a) - t(b) > 1 \text{ and } t(a) > t(b) \geq t(c) \\ b & \text{if } t(a) - t(b) \leq 1,\ t(a) \geq t(b) \geq t(c) \text{ and } \neg(t(a) - 1 = t(b) = t(c)) \\ c & \text{if } t(a) - 1 = t(b) = t(c) \\ b & \text{otherwise} \end{cases}$$

$$p(t) = \begin{cases} 1 & \text{if } f(t) = a \\ 0 & \text{if } f(t) = b \\ 0 & \text{if } f(t) = c \end{cases}$$

The mechanism partitions the type space into four parts. Types representing P^1 are partitioned into three parts and the remaining types (representing P^2, P^3, P^4) constitute the fourth part. A type s representing P^4 prefers c to b, but it is allocated b by the mechanism (at zero payment). It can manipulate by reporting a type to get the alternative c (at zero payment). Hence, the mechanism (f, p) is not incentive compatible. However, the incentive constraints violated here are not local. Further, the mechanism satisfies payment-only incentive compatibility, but the allocation rule f does not satisfy STR (outcome c is not attained by any type in the interior of the cone representing the orderings). These observations are formally proved in the following claim, whose proof can be found in the Appendix.

Claim **2** *Let (f, p) be the above mechanism.*

1. *(f, p) is locally incentive compatible and payment-only incentive compatible but not incentive compatible.*

2. *The allocation rule f does not satisfy STR.*

3.3 Domains Satisfying Top Connectedness

In this section, we show two important domains where Theorem 1 holds: (a) the domain consisting of all single peaked preferences (b) a single crossing domain. To show this, we provide a sufficient condition for a domain to be top connected. We then show that this sufficient condition holds in these domains.

To introduce our sufficient condition, we begin with some definitions. These definitions have been borrowed from the local incentive compatibility literature in voting environment (Sato, 2013b). Since we study ordinal type spaces, they turn out to be relevant for us also. For any preference ordering P and any alternative $a \in A$, let $r(P, a)$ denote the rank of a in P. We say P' is a (a, b)-swap of P if $r(P, x) = r(P', x)$ for all $x \notin \{a, b\}$, $r(P, a) = r(P, b) - 1$, $r(P', a) = r(P', b) + 1$. In other words, if a and b are consecutively ranked in P with a ranked above b, the preference ordering obtained by just swapping the positions of a and b is called an (a, b)-swap of P. Note that if P' is an (a, b)-swap of P for some $a, b \in A$, then P and P' are *adjacent*.

The following definition is borrowed from Sato (2013b).

Definition **7** *A distinct sequence of orderings* $(P^1, \ldots, P^k)$ *is* **without restoration** *if for every* $j \in \{1, \ldots, k-1\}$*,* P^j *and* P^{j+1} *are adjacent and there exists no distinct* $j, j' \in \{0, 1, \ldots, k\}$ *and* $x, y \in A$ *such that* P^{j+1} *is a* (x, y)*-swap of* P^j *and* $P^{j'+1}$ *is a* (y, x)*-swap of* $P^{j'}$*.*

A domain $\mathcal{D}$ *is* **connected without restoration** *if for every* $P, P' \in \mathcal{D}$*, there exists a sequence of distinct orderings* $(P = P^0, P^1, \ldots, P^k, P^{k+1} = P')$ *without restoration.*

The without restoration property requires that no pair of alternatives is swapped more than once along a sequence of adjacent orderings.

Now, consider the following richness condition on the domain, which also appeared in the definition of top connectedness.

Definition **8** *A domain* $\mathcal{D}$ *is* **rich** *if for every* $a \in A$*, there exists* $P \in \mathcal{D}$ *such that* $r(P, a) = 1$*.*

Richness and connectedness without restoration are sufficient for top connectedness. Since many domains are known to satisfy connectedness without restoration, and richness is easy to verify, this gives us an easy method to check top connectedness.

Lemma **1** *If* $\mathcal{D}$ *is rich and connected without restoration, then it is top connected.*

Proof: Choose any $a \in A$. By richness, there is a $P \in \mathcal{D}$ such that $r(P, a) = 1$. Now, choose any $P' \in \mathcal{D}$. Since $\mathcal{D}$ is connected without restoration, there is a sequence of preferences $(P' = P^0, P^1, \ldots, P^k, P^{k+1} = P)$ without restoration such that for all $j \in \{1, \ldots, k\}$, $P^j \in \mathcal{D}$. Since $r(P, a) = 1$ and the sequence of preferences is without restoration, we have $r(P^j, a) \geq r(P^{j+1}, a)$ for all $j \in \{0, 1, \ldots, k\}$. Hence, $\mathcal{D}$ is top connected. ■

An immediate consequence of Lemma 1 is that the single peaked domain (with respect to a given strict linear order) is top connected.

Proposition **1** *The single peaked domain is top connected.*

Proof: Sato (2013b) shows that the single peaked domain is connected without restoration. Clearly, it is a rich domain. Hence, Lemma 1 proves the claim. ■

This shows that Theorem 1 applies to the important single peaked type space. It can be shown that many extensions of the single peaked domain, like the single peaked domain on a tree (Demange, 1982) and some multidimensional versions of the single peaked domain (Reffgen, 2015), are also rich and connected without restoration. Hence, our main result applies to such domains too.

We now identify another top connected type space. This type space corresponds to the following domain of orderings.

Definition 9 *A set of preferences $\mathcal{D}$ is a* **single crossing** *domain if there exists a strict linear order $\succ$ on the set of alternatives and a strict linear order $\lhd$ on the set of preferences $\mathcal{D}$ such that for all $a, b \in A$ and for all $P, P' \in \mathcal{D}$,*

- *$a \succ b$, $P \lhd P'$, and aPb implies $aP'b$*
- *$a \succ b$, $P \lhd P'$, and $bP'a$ implies bPa.*

Single crossing domains are a well studied domain in voting and political economy (Saporiti, 2009). They can also be a plausible domain restriction in models with transfers. For instance, suppose A is the set of products in the market and $\succ$ reflects the ranking of products in terms of reputation. A preference of a consumer may or may not be sensitive to the reputation of the product. Single crossing ensures that the set of possible preferences of a consumer for the products can be *ordered* on how sensitive they are to reputation of the products. This is captured by the ordering $\lhd$.

For any ordering P over A and any ordering $\succ$ over A, let $X(P, \succ) := \{(a, b) : a \succ b, aPb\}$. Clearly, a set of preferences $\mathcal{D}$ is a **single crossing** domain if and only if there exists a strict linear order $\succ$ on the set of alternatives and a strict linear order $\lhd$ on the set of preferences $\mathcal{D}$ such that for any $P, P' \in \mathcal{D}$ with $P \lhd P'$, we have $X(P, \succ) \subsetneq X(P', \succ)$ (notice the strict inclusion). We will denote a single crossing domain as $\mathcal{D}^{\succ,\lhd}$.

A single crossing domain $\mathcal{D}^{\succ,\lhd} := \{P^1, \ldots, P^l\}$ with $P^1 \lhd \ldots \lhd P^l$ is a **successive single crossing domain** if for every $j \in \{1, \ldots, l-1\}$, $|X(P^j, \succ)| + 1 = |X(P^{j+1}, \succ)|$. Successive single crossing domains were introduced in Carroll (2012).

Proposition 2 *A rich successive single crossing domain is top connected.*

Proof: We will show that a rich successive single crossing domain is connected without restoration, and by Lemma 1, we will be done. Let $\mathcal{D}^{\succ,\lhd} := \{P^1, \ldots, P^l\}$ be a rich successive single crossing domain with $P^1 \lhd P^2 \lhd \ldots \lhd P^l$. Pick $P^j, P^k \in \mathcal{D}^{\succ,\lhd}$ with $j < k$. The sequence of preferences $(P^j, P^{j+1}, \ldots, P^k)$ satisfies the fact that for any $j' \in \{j, j+1, \ldots, k-1\}$, $P^{j'}$ and $P^{j'+1}$ are adjacent - this follows from the definition of successive single crossing domain. Now, assume for contradiction, there is some pair of alternatives $x, y \in A$ such that they are swapped more than once in this sequence. But the single crossing property requires that if $x \succ y$, once $xP^{k'}y$ for some $P^{k'}$ in the sequence, it must remain $xP^{l'}y$ for all $l' > k'$. Hence, getting swapped more than once will violate the single crossing property. This means that every successive single crossing domain is connected without restoration. ■

This shows that Theorem 1 applies to a rich successive single crossing type space. Every successive single crossing domain need not be rich. However, there is a rich successive single crossing domain. In Appendix B, we describe an algorithm to construct one rich successive single crossing domain.

3.4 Discussions on Non-Ordinal Type Spaces

We have not been able to identify sufficiently interesting non-ordinal type spaces where Theorem 1 holds. We give below an example to illustrate the difficulty in such type spaces.

Suppose $A = \{a, b\}$ and consider the type space shown in Figure 1 - the type space is shown in dashed and dotted areas and it does not include the boundaries of the region shown.

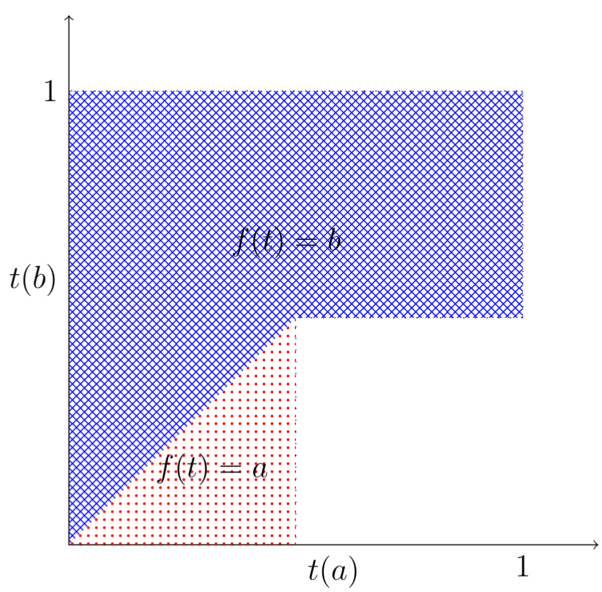

Figure 1: An example with two alternatives

The allocation rule f chooses a and b in the regions shown in Figure 1. Take the payment rule $p(t) = 0$ for all t. Hence, the mechanism (f, p) is payment-only incentive compatible. It is now easy to verify that (f, p) is locally incentive compatible (note boundaries of the region are not included).

But it is not incentive compatible since a type in the region where $f(t) = b$ but $t(a) > t(b)$ can manipulate to a type s where $f(s) = a$.

Though this type space is not convex, it is full-dimensional and connected. Still, we do not get the richness required for local incentive compatibility and payment incentive compatibility to imply incentive compatibility.

Appendix C contains two specific non-ordinal type spaces. In one of them, we show that our main result (Theorem 1) can be extended. When we generalize that type space, we see that the main result does not extend any more. We defer elaborate discussions on these type spaces to Appendix C because they are very specific type spaces, and may not be of general interest.

4 Local Implementation

In quasilinear environment, instead of analyzing a mechanism (f, p), one can consider implementability of the allocation rule f.

Definition 10 *An allocation rule f is* **implementable** *if there exists a payment rule p such that (f, p) is incentive compatible.*

Because of well know revenue equivalence results, for every implementable allocation rule, the corresponding payment rule can be identified up to an additive constant in connected type spaces (Chung and Olszewski, 2007). As a result, if we want to verify if a mechanism is incentive compatible or not, it is enough to verify implementability of the allocation rule. The payment rule can be verified to match with the payment rule prescribed by the revenue equivalence formula.

If the objective of local incentive compatibility is to make the process of verification of incentive compatibility easier, this can also be accomplished by verifying the implementability of the allocation rule. We investigate this question in this section and come up with two interesting conclusions. First, we formally establish a connection between these two parallel strands of local verifications. Second, our results establish that an analogue of our main result (Theorem 1) is not possible if we consider the notion of local implementation.

We now formally define the notion of local implementation. Definition 10 has an existential qualifier. We can remove this by using the well known characterization of implementability due to Rochet (1987). First, for any allocation rule f, we define $\ell^f(s, t) := t(f(t)) - t(f(s))$ for every pair of types s, t in the type space.

Definition 11 *An allocation rule f is* **cyclically monotone** *if for every finite sequence of types $(t^1, \ldots, t^k)$, we have*

$$\sum_{j=1}^{k} \ell^f(t^j, t^{j+1}) \geq 0,$$

where $(k+1) \equiv 1$.

Fact 2 *An allocation rule is implementable if and only if it is cyclically monotone.*

Fact 2 is true in every type space (Rockafellar, 1970; Rochet, 1987).

Definition 12 *An allocation rule f is* **strongly locally implementable** *if for every $t \in T$, there exists an $\epsilon > 0$ such that f restricted to $B^\epsilon(t)$ is implementable.*

Equivalently, f is strongly locally implementable if for every $t \in T$, there exists an $\epsilon > 0$ such that f restricted to $B^\epsilon(t)$ is cyclically monotone.

This notion of strong local implementation was introduced in Archer and Kleinberg (2008). However, it is not the counterpart of local incentive compatibility since local incentive compatibility requires that for every t in the type space, there is an $\epsilon > 0$ such that for every $s \in B^\epsilon(t)$, incentive constraints between s and t must hold - it is silent about incentive constraints among other types in $B^\epsilon(t)$. Local incentive compatibility can be generalized in the context of implementation using a weaker form of cycle monotonicity.

Definition 13 *An allocation rule f is* **locally implementable** *if for every $t \in T$, there exists an $\epsilon > 0$ such that for all $s \in B^\epsilon(t)$, we have*

$$\ell^f(s, t) + \ell^f(t, s) \geq 0.$$

This notion of local implementation is also discussed in Archer and Kleinberg (2008), who called it weak local implementation and showed that this may not imply implementability in convex type spaces if the allocation rule is a randomized allocation rule. However, if the allocation rule is deterministic, a locally implementable allocation rule is implementable in a convex type space (Archer and Kleinberg, 2008).

Hence, in convex type spaces with deterministic allocation rules, (a) a locally implementable allocation rule is implementable (Archer and Kleinberg, 2008) and (b) a locally incentive compatible mechanism is incentive compatible (Carroll, 2012). Our objective here is to show that local implementation implying implementation is a stronger result than local incentive compatibility implying incentive compatibility in a large class of domains.

DEFINITION **14** *A type space T is a* **locally implementable (LIM)** *type space if every locally implementable allocation rule $f : T \to A$ is implementable.*

A type space T is a **locally incentive compatible (LIC)** *type space if every locally incentive compatible mechanism (f, p) defined on T is incentive compatible.*

We identify sufficient conditions on type spaces where every LIM type space is a LIC type space. We do this for the following type spaces.

DEFINITION **15** *A k-tuple of points $(s_1, \ldots, s_k)$ will be called a* **polygonal connection** *from s to t in T if $s_1 = s, s_k = t$ and for every $j \in \{1, \ldots, k-1\}$, the line segment $\{\lambda s_j + (1-\lambda)s_{j+1} : \lambda \in [0,1]\}$ is in T.*

The type space T is **polygonally connected** *if for every pair of points $s, t \in T$, there is a polygonal connection from s to t.*

Every open connected set is polygonally connected. This leads to the main result of this section.

THEOREM **2** *Suppose type space $T \subseteq \mathbb{R}^{|A|}$ is polygonally connected. If T is an LIM type space, then it is an LIC type space.*

Proof: Consider a mechanism $M \equiv (f, p)$ in a polygonally connected type space T, which is locally incentive compatible. Choose a type $t \in T$, $\epsilon > 0$, and $s \in B^{\epsilon}(t)$. By local incentive compatibility,

$$t(f(t)) - p(t) \geq t(f(s)) - p(s)$$
$$s(f(s)) - p(s) \geq s(f(t)) - p(t).$$

Adding these incentive constraints, gives us

$$\ell^f(s,t) + \ell^f(t,s) = \big[t(f(t)) - t(f(s))\big] + \big[s(f(s)) - s(f(t))\big] \geq 0.$$

This shows that f is locally implementable. Since T is a LIM domain, f is implementable. This implies that there exists a payment rule $q : T \to \mathbb{R}$ such that (f, q) is incentive compatible.

Now, choose any pair of points $s, t \in T$. By polygonal connectedness, there exists a sequence of points $(s = s_1, \ldots, s_k = t)$ in T such that for each $j \in \{1, \ldots, k-1\}$, the line segment joining s_j and s_{j+1} lie in T. Now, consider any $j \in \{1, \ldots, k-1\}$ and the line segment L_j joining s_j and s_{j+1}. Now, (f,p) restricted to L_j is also locally incentive compatible in L_j. Since L_j is convex, by Carroll (2012), (f,p) restricted to L_j is incentive compatible. Also, (f,q) restricted to L_j is incentive compatible.

Now, we apply revenue equivalence.

Definition 16 *An implementable allocation rule $f : T \to A$ satisfies* **revenue equivalence** *if for every $p : T \to \mathbb{R}$ and $q : T \to \mathbb{R}$ such that (f,p) and (f,q) are incentive compatible, there exists a constant α such that*

$$p(t) = q(t) + \alpha \qquad \forall\ t \in T.$$

Since L_j is convex, revenue equivalence holds (Rockafellar, 1970). [7] As a result, we have

$$p(s_{j+1}) - p(s_j) = q(s_{j+1}) - q(s_j).$$

Summing the above equations for all $j \in \{1, \ldots, k-1\}$, and telescoping, we get

$$p(s_k) - p(s_1) = q(s_k) - q(s_1).$$

Using $s = s_1$ and $t = s_k$, we get $p(t) - p(s) = q(t) - q(s)$.

Thus, we have shown that for any pair of types $s, t \in T$, $p(t) - p(s) = q(t) - q(s) \leq t(f(t)) - t(f(s))$, where the last inequality came from incentive compatibility of (f,q). But this implies that (f,p) is also incentive compatible. ■

We do not know if the converse of Theorem 2 holds. But Theorem 2 also holds (with identical proof) if we allow for randomization. Further, it also holds if we strengthen the notion of local incentive compatibility and local implementability in the lines of Definition 12.

An immediate corollary of Theorem 2 is that there are top connected type spaces that are not LIM type space.

Corollary 1 *The single peaked type space is not an LIM type space.*

Proof: If the single peaked type space is an LIM type space, by Theorem 2, it is an LIC type space. By Claim 1, Example 1 contains a locally incentive compatible mechanism that is not incentive compatible. This is a contradiction. ■

One wonders if imposing the range condition STR helps. Theorem 2 can be restated with the STR condition, i.e., if every locally implementable allocation rule satisfying STR is

[7] See also, Krishna and Maenner (2001); Milgrom and Segal (2002); Chung and Olszewski (2007); Heydenreich et al. (2009).

implementable in a type space, then every locally incentive compatible mechanism (f,p) with f satisfying STR is incentive compatible in that type space. The same proof of Theorem 2 goes through. Since the allocation rule in Example 1 satisfies STR, we again conclude that in the single peaked type space there are locally implementable allocation rules satisfying STR that are not implementable.

5 Conclusion

In an important class of non-convex type spaces, we show that if we restrict attention to deterministic mechanisms, then local incentive compatibility along with payment-only incentive compatibility implies incentive compatibility. We also show a relationship between local implementation and local incentive compatibility. A natural future research direction is to study the implication of randomization and Bayesian incentive compatibility in various non-convex type spaces.

Appendix A: Omitted Proofs

Proof of Claim 1

Proof: As argued, earlier (f,p) is not incentive compatible. We say that a type t can manipulate to a type s if misreporting s leads to higher net utility in the mechanism (f,p). We consider all possible cases of misreport and verify the possible manipulations in this mechanism.

Case 1: Suppose the true type of the agent is $t \in T^1$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) - p(t) = t(a) - 2$. Obviously, misreporting to another type in T^1 does not change the net utility. If he reports $s \in T^2 \cup T^3 \cup T^5$, then the $f(s) = b$ and $p(s) = 0$. Hence, the net utility of the agent is $t(b) - 0 = t(b)$. By definition, $t(a) - 2 \geq t(b)$, and hence, t cannot manipulate to $s \in T^2 \cup T^3 \cup T^5$.

If t manipulates to a type in T^4, then he gets a net utility of $t(c) - 1$. But $t(a) - 2 \geq t(b) \geq t(c)$ implies that t cannot manipulate to a type in T^4 also. This shows that t cannot manipulate to a type in $T^1 \cup T^2 \cup T^3 \cup T^4 \cup T^5$.

However, if t reports a type in T^6, then he gets a net utility of $t(a)$, whereas truthtelling gives him a net utility of $t(a) - 2$.

Case 2. Suppose the true type of the agent is $t \in T^2$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(f)) - p(t) = t(b)$. Clearly, misreporting to another type in T^2 does not change the net utility. Since $t(a) > t(b) \geq t(c)$ and payments are all non-negative, misreporting to a type that gives either b or c as outcome will not be beneficial. Hence, t cannot manipulate to a type in $T^3 \cup T^4 \cup T^5$. Also, by misreporting to a type in T^1 gives the agent a net utility of $t(a)-2$. By definition, $t(a)-2 < t(b)$. Hence, misreporting to a type in T^1 is also not beneficial. Thus, t cannot manipulate to a type in $T^1 \cup T^2 \cup T^3 \cup T^4 \cup T^5$.

However, misreporting to a type in T^6 gives the agent a net utility of $t(a)$, which is strictly higher than $t(b)$.

Case 3. Suppose $t \in T^3$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) - p(t) = t(b)$. Since $t(b) \geq \max(t(a), t(c))$ and payments in the mechanism are non-negative, t cannot manipulate to any other type.

Case 4. Suppose $t \in T^4$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) - p(t) = t(c) - 1$. By definition, $t(c) - 1 \geq t(b) \geq t(a)$. Since payments in the mechanism are always non-negative, t cannot manipulate to any other type.

Case 5. Suppose $t \in T^5$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) - p(t) = t(b)$. Since $t(c) \geq t(b) > t(a)$ and payments in the mechanism are non-negative, no manipulation to a type s such that $f(s) \in \{a, b\}$ is not beneficial. On the other hand, if t manipulates to a type s with $f(s) = c$, then $s \in T^4$, and his net utility is $t(c) - 1$. But, by definition, $t(c) - 1 < t(b)$. Hence, t cannot manipulate to any other type.

Case 6. Suppose $t \in T^6$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) - p(t) = t(a)$. Since $t(c) \geq t(b) = t(a)$ and payments in the mechanism are non-negative, no manipulation to a type s such that $f(s) \in \{a, b\}$ is possible. On the other hand, if t manipulates to a type s with $f(s) = c$, then $s \in T^4$, and his net utility is $t(c) - 1$. But, by definition, $t(c) - 1 < t(a)$. Hence, t cannot manipulate to any other type.

This shows that the only manipulation possible in the mechanism is when the true type $t \in (T^1 \cup T^2)$, the agent can manipulate to a type in T^6. But note that every type in $s \in T^6$ satisfies $s(c) > s(a) = s(b)$. But $t(a) > t(b) \geq t(c)$. Hence, there exists an $\epsilon > 0$ such that every $t' \in B^{\epsilon}(t)$ satisfies $t' \notin T^6$. Similarly, there exists $\epsilon > 0$ such that every $t' \in B^{\epsilon}(s)$ satisfies $t' \notin (T^1 \cup T^2)$. This shows that (f, p) is locally incentive compatible. ∎

Proof of Claim 2

Proof: Proof of 1. We consider the following cases to complete the proof.

Case 1: $t \in T_a$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) - p(t) = t(a) - 1$. Obviously, misreporting to another type in T_a does not change the net utility.

If he reports $s \in T_b$, then the $f(s) = b$ and $p(s) = 0$. Hence, the net utility of the agent is $t(b) - 0 = t(b)$. Since for all $t \in T_a$, $t(a) - 1 > t(b)$, t cannot manipulate to $s \in T_b$.

If he reports $s \in T_c$, then the $f(s) = c$ and $p(s) = 0$. Hence, the net utility of the agent is $t(c)$. Since for all $t \in T_a$, $t(a) - 1 > t(c)$, t cannot manipulate to $s \in T_c$.

This shows that t cannot manipulate to a type in $T_a \cup T_b \cup T_c$.

Case 2: $t \in T_c$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) = t(c)$. Obviously, misreporting to another type in T_c does not change the net utility.

If he reports $s \in T_a$, then the $f(s) = a$ and $p(s) = 1$. Hence, the net utility of the agent is $t(a) - 1$. Since for all $t \in T_c$, $t(c) = t(a) - 1$, t cannot manipulate to $s \in T_a$.

If he reports $s \in T_b$, then the $f(s) = b$ and $p(s) = 0$. Hence, the net utility of the agent is $t(b)$. Since for all $t \in T_c$, $t(c) = t(b)$, t cannot manipulate to $s \in T_b$.

This shows that t cannot manipulate to a type in $T_a \cup T_b \cup T_c$.

Case 3: $t \in T_b$. Truthful reporting in the mechanism gives the agent a net utility of $t(f(t)) - p(t) = t(b)$. Obviously, misreporting to another type in T_b does not change the net utility.

If he reports $s \in T_a$, then the $f(s) = a$ and $p(s) = 1$. Hence, the net utility of the agent is $t(a) - 1$. Since for all $t \in T_b$, $t(b) \geq t(a) - 1$, t cannot manipulate to $s \in T_a$.

However, t can manipulate to $s \in T_c$ only if $t(c) > t(b) \geq t(a)$. This shows that (f, p) is not incentive compatible. However if $t(c) > t(b) \geq t(a)$, there exists an $\epsilon > 0$ such that every $s \in B^\epsilon(t)$ satisfies $s(c) > s(b) \geq s(a)$. Therefore (f, p) is locally incentive compatible. Since (f, p) is also payment-only incentive compatible, we are done.

Proof of 2. Since there does not exist a strict type t such that $f(t) = c$, f violates the STR condition. ■

Appendix B: An Algorithm to Construct a Rich Successive Single Crossing Domain

Here, we provide an algorithm to explicitly construct a rich successive single crossing domain. For our construction, we assume that the underlying ordering over A to be $\succ$. Let P^1 be the preference ordering which is the reverse ordering of $\succ$. We now describe how we construct a sequence of orderings that constitute our domain. The construction is inductive. Suppose we have found a preference ordering P^j. Then, we adopt the following procedure to construct P^{j+1} or stop. We say that a pair of alternatives $a, b \in A$ are inconsistent between P^j and $\succ$ if $a \succ b$ and $bP^j a$. If a and b are not inconsistent between P^j and $\succ$, then they are consistent between P^j and $\succ$.

- If P^j is the same ordering as $\succ$, we stop. Our domain is $\{P^1, \ldots, P^j\}$.
- If P^j is different from $\succ$, then we define

  $A(P^j, \succ) := \{\{a, b\} : |r(P^j, a) - r(P^j, b)| = 1$ and a, b are inconsistent between P^j and $\succ\}$.

- Choose $\{x, y\} \in A(P^j, \succ)$ such that for every $\{a, b\} \in A(P^j, \succ)$, we have

  $$\min(r(P^j, x), r(P^j, y)) \leq \min(r(P^j, a), r(P^j, b)).$$

  Let P^{j+1} be the ordering obtained as by swapping x and y and maintaining the positions of other alternatives as in P^j, i.e., $r(P^{j+1}, x) = r(P^j, y), r(P^{j+1}, y) = r(P^j, x)$ and $r(P^{j+1}, z) = r(P^j, z)$ for all $z \in A \setminus \{x, y\}$. The procedure is then repeated.

Since P^1 is initialized, this sequence of ordering is well defined. Moreover, the sequence will terminate since there will come a stage with preference ordering P^j where $A(P^j, \succ)$ will be empty, and thus, P^j will be the same ordering as $\succ$. Let $(P^1, \ldots, P^l)$ be the sequence of ordering constructed by this algorithm. We define $\lhd$ as the relation $P^1 \lhd P^1 \lhd \ldots \lhd P^l$. We show the following.

CLAIM **3** *The domain $\mathcal{D}^{\succ, \lhd} := \{P^1, \ldots, P^l\}$ is a rich successive single crossing domain.*

Proof: By construction, for every pair of alternatives $a, b \in A$ with $a \succ b$, if there is a preference ordering $P^j \in \mathcal{D}^{\succ, \lhd}$ with $aP^j b$, then $aP^k b$ for all $k > j$. This ensures that $\mathcal{D}^{\succ, \lhd}$ is a single crossing domain. Further, for any $j \in \{1, \ldots, k-1\}$, P^j and P^{j+1} are adjacent. This ensures that $\mathcal{D}^{\succ, \lhd}$ is a successive single crossing domain.

We show that $\mathcal{D}^{\succ,\lhd}$ is rich. Pick any alternative $a \in A$. We will show that there exists some $P^j \in \{P^1, \ldots, P^k\}$ such that $r(P^j, a) = 1$. Let $r(P^1, a) = \ell$. We show this using induction on ℓ. If $\ell = 1$, then we are done. Suppose $\ell > 1$ and the claim is true for all $k' < \ell$. Note that the claim is true if $\ell = |A|$ - this is because the terminating preference ordering P^k of the algorithm is $\succ$. Hence, $\ell < |A|$.

Let b be an alternative such that $r(P^1, b) = \ell - 1$. By the induction hypothesis, there is a preference ordering $P^h \in \{P^1, \ldots, P^k\}$ such that $r(P^h, b) = 1$. Without loss of generality, let P^h be such that for all $h' > h$, we have $r(P^{h'}, b) \neq 1$. We argue that $r(P^h, a) = 2$.

Assume for contradiction that $r(P^h, a) \neq 2$. Then, $r(P^h, a) > 2$. Let $r(P^h, c) = 2$, where $c \neq a$. Since $r(P^{h+1}, b) \neq 1$ and P^h and P^{h+1} are adjacent, it must be that $r(P^{h+1}, c) = 1$ and $r(P^{h+1}, b) = 2$. Since c and b are not consistent between P^h and $\succ$, it must be that bP^1c. This implies that bP^1aP^1c (since a and b are consecutively ranked in P^1). But note that c and a are consistent between P^h and $\succ$. By the definition of our procedure, there must exist $h'' < h$ such that a and b are consistent between $P^{h''}$ and $\succ$ - this follows from the fact that higher ranked inconsistent pairs are made consistent earlier in our procedure. By the single crossing property, a and b must be consistent between P^h and $\succ$. This is a contradiction since $a \succ b$ and $r(P^h, b) = 1$.

Hence, $r(P^h, a) = 2$. Since P^h and P^{h+1} are adjacent and $r(P^{h+1}, b) \neq 1$, it must be that $r(P^{h+1}, a) = 1$. This shows that $\mathcal{D}^{\succ,\lhd}$ is rich. ■

P^1	P^2	P^3	P^4	P^5	P^6	P^7	P^8	P^9	P^{10}	P^{11}
e	d	d	c	c	c	b	b	b	b	a
d	e	c	d	d	b	c	c	c	a	b
c	c	e	e	b	d	d	d	a	c	c
b	b	b	b	e	e	e	a	d	d	d
a	a	a	a	a	a	a	e	e	e	e

Table 3: A rich successive single crossing domain

We give an example in Table 3 of a rich successive single crossing domain constructed from our procedure. This domain is single crossing with respect to the ordering $\succ$ over alternatives where $a \succ b \succ c \succ d \succ e$ and the preferences are ordered according to $\lhd$ as $P^1 \lhd P^2 \lhd P^3 \lhd P^4 \lhd P^5 \lhd P^6 \lhd P^7 \lhd P^8 \lhd P^9 \lhd P^{10} \lhd P^{11}$.

The rich successive single crossing domain produced by our algorithm has $\frac{n(n-1)}{2} + 1$ preference orderings as compared to 2^{n-1} preference orderings in the single peaked domain. [8]

[8] A proof of this fact is available upon request.

Hence, these domains are not the same.

Appendix C: Examples of Non-ordinal Type Spaces

We now discuss the implication of payment-only incentive compatibility in general type spaces using some particular examples. The discussions below suggests that the equivalence between local incentive compatibility along with payment-only incentive compatibility and incentive compatibility is quite nuanced.

Consider an example with $A = \{a, b, c\}$ and a type space where type of an agent assigns a non-negative value to one of the alternatives and assigns zero to all the other alternatives. Hence, the type space T is defined as follows:

$$T = \{t \in \mathbb{R}^3_+ : \text{ for some } x \in \{a, b, c\}, t(x) \geq 0, t(y) = 0 \ \forall \ y \neq x\}.$$

Notice that T consists of the non-negative parts of three axes in $\mathbb{R}^3$. We will denote these three parts of T as T_a, T_b, T_c and note that $T = T_a \cup T_b \cup T_c$ and intersection of any two of them contains only the origin.

Suppose (f, p) is a locally incentive compatible mechanism. For every $x \in A$, convexity of T_x implies (by Carroll (2012)) (f, p) restricted to each T_x is incentive compatible. But this alone is not sufficient to guarantee incentive compatibility. Consider the following mechanism in Table 4. It is easy to verify that this mechanism is locally incentive compatible - one easy way to see this is that the mechanism is incentive compatible on T_a, T_b, T_c and it is incentive compatible on a small neighborhood around the origin. However, it is not incentive compatible since a type t with $t(b) > 0.5$ gets a utility of $t(b) - 0.5$ from truthtelling but can manipulate to a type s with $s(a) \in [0.25, 0.5]$ to get a utility of $t(b)$. This manipulation is possible since the mechanism is not payment-only incentive compatible - if $s(a) \in [0.25, 0.5]$ we have $f(s) = b$ and $p(s) = 0$ but when $t(b) > 0.5$ we have $f(t) = b$ and $p(t) = 0.5$.

Type space	f	p
$t(a) > 0.5$, $t(b) = t(c) = 0$	$f(t) = a$	$p(t) = 0.5$
$t(a) \in [0.25, 0.5]$, $t(b) = t(c) = 0$	$f(t) = b$	$p(t) = 0$
$t(a) \in [0, 0.25)$, $t(b) = t(c) = 0$	$f(t) = c$	$p(t) = 0$
$t(b) > 0.5$, $t(a) = t(c) = 0$	$f(t) = b$	$p(t) = 0.5$
$t(b) \in [0, 0.5]$, $t(a) = t(c) = 0$	$f(t) = c$	$p(t) = 0$
$t(c) \geq 0$, $t(a) = t(b) = 0$	$f(t) = c$	$p(t) = 0$

Table 4: A locally incentive compatible mechanism

Payment-only incentive compatibility alone cannot get rid of this problem. To see this, consider the modification of this mechanism in Table 5. Again, it is straightforward to verify that the mechanism in Table 5 is locally incentive compatible and payment-only incentive compatible. However, an agent with type t, where $t(b) > 0$ and $t(a) = t(c) = 0$, can still manipulate to a type s, where $s(a) \in [0.25, 0.5], s(b) = s(c) = 0$.

Type space	f	p
$t(a) > 0.5$, $t(b) = t(c) = 0$	$f(t) = a$	$p(t) = 0.5$
$t(a) \in [0.25, 0.5]$, $t(b) = t(c) = 0$	$f(t) = b$	$p(t) = 0$
$t(a) \in [0, 0.25)$, $t(b) = t(c) = 0$	$f(t) = c$	$p(t) = 0$
$t(b) \geq 0$, $t(a) = t(c) = 0$	$f(t) = c$	$p(t) = 0$
$t(c) \geq 0$, $t(a) = t(b) = 0$	$f(t) = c$	$p(t) = 0$

Table 5: A locally incentive compatible and payment-only incentive compatible mechanism

We now formally define this type space.

Definition 17 *A type space $\bar{T} \subseteq \mathbb{R}^{|A|}$ is a* **unique dichotomous** *type space if*

$$\bar{T} := \{t \in \mathbb{R}^{|A|}_{+} : \text{ for some } x \in A, t(x) \geq 0, t(y) = 0 \;\forall\; y \neq x\}.$$

Every type in a unique dichotomous type space has a unique alternative for which it gets non-negative value and gets zero value on other alternatives. We need the following range condition on the allocation rules.

Definition 18 *An allocation rule $f : \bar{T} \to A$ satisfies* **strict range condition (SRC)** *if for every $a \in A$, there exists $t \in \bar{T}$ with $t(a) \geq 0, t(b) = 0$ for all $b \neq a$ such that $f(t) = a$.*

Theorem 1 extends to the unique dichotomous type space using SRC.

Theorem 3 *Suppose $f : \bar{T} \to A$ is an allocation rule satisfying SRC. Let (f, p) be a mechanism defined over $\bar{T}$. Then, the mechanism (f, p) is locally incentive compatible and payment-only incentive compatible if and only if it is incentive compatible.*

Proof: Let (f, p) be a locally incentive compatible and payment-only incentive compatible mechanism, where f satisfies SRC. As before, for any $x \in A$, let T_x denote all the types in $\bar{T}$ such that $t(x) \geq 0, t(y) = 0$ for all $y \neq x$. Consider a type t such that $f(t) = a$ and $t \in T_b$. Local incentive compatibility along with convexity of T_b implies that (f, p) restricted to T_b is incentive compatible (Carroll, 2012). Payment-only incentive compatibility allows us to write p as a map $p : A \to \mathbb{R}$. Choose any $c \neq a$. For incentive compatibility of (f, p), we need to show that $t(a) - p(a) \geq t(c) - p(c)$. If $c = b$, SRC implies that there is a type in T_b where f chooses b. Since (f, p) restricted to T_b is incentive compatible, we get the desired inequality. Else, $c \notin \{a, b\}$. Then, $t \in T_b$ implies that $t(c) = 0$. Now, the origin $\mathbf{0} \in T_x$ for all $x \in A$. Let $f(\mathbf{0}) = z$. Since (f, p) restricted to each T_x is incentive compatible, we can write

$$t(a) - p(a) \geq t(z) - p(z)$$
$$0 - p(z) \geq 0 - p(c).$$

Adding these two inequalities, we get $t(a) - p(a) \geq t(z) - p(c) \geq t(c) - p(c)$, where the last inequality followed from the fact $t(z) \geq 0 = t(c)$. ■

The result in Theorem 3 seems difficult to extend to other connected type spaces. For instance, consider an example with three alternatives $A = \{a, b, c\}$. We say a type $t \in \mathbb{R}^3_+$ is **dichotomous** if there exists a non-negative number $\alpha \geq 0$ and a subset of alternatives $S \subseteq A$ such that $t(a) = \alpha$ for all $a \in S$ and $t(a) = 0$ for all $a \notin S$. Let T^d be the set of all dichotomous types. The dichotomous type space is a generalization of the unique dichotomous type space. They were studied in Mishra and Roy (2013), who characterized the *implementable* rules in this type space.

The dichotomous type space T^d consists of rays from the origin. Local incentive compatibility only ensures that along each ray incentive compatibility holds and around the origin incentive compatibility holds. This is not sufficient to guarantee overall incentive compatibility even with payment-only incentive compatibility and range conditions. To see this, consider the example in Table 6. The mechanism (f, p) given in the example in Table 6 is locally incentive compatible - it can be easily verified from the fact that it is incentive compatible along each of the rays and incentive compatible around a neighborhood of the origin. It is also payment-only incentive compatible. However, it is not incentive compatible. For instance, consider a type t with $t(a) = t(b) = 0.55, t(c) = 0$. From Table 6, we see that $f(t) = c$ and $p(t) = 0$, and hence, truthtelling generates a utility of zero. However, by reporting a type s with $s(b) = 0.55, s(a) = s(c) = 0$, we have $f(s) = b, p(s) = 0.5$, which generates a utility of 0.05 for the agent with type t. Hence, agent with type t manipulates to s.

No reasonable range condition can fix these kind of problems in this type space. Essentially, every ray corresponds to a subset of alternatives whose value is non-negative. Incentive compatibility along the rays only ensures that there is a cutoff value below which the agents gets an alternative outside this subset of alternative (giving zero value) and above which

Type space	f	p
$t(a) > 0.6, t(b) = t(c) = 0$	$f(t) = a$	$p(t) = 0.6$
$t(a) \in (0, 0.6], t(b) = t(c) = 0$	$f(t) = c$	$p(t) = 0$
$t(b) > 0.5, t(a) = t(c) = 0$	$f(t) = b$	$p(t) = 0.5$
$t(b) \in (0, 0.5], t(a) = t(c) = 0$	$f(t) = c$	$p(t) = 0$
$t(a) = t(b) > 0.6, t(c) = 0$	$f(t) = a$	$p(t) = 0.6$
$t(a) = t(b) \in (0, 0.6], t(c) = 0$	$f(t) = c$	$p(t) = 0$
$t(c) > 0$	$f(t) = c$	$p(t) = 0$
$t(a) = t(b) = t(c) = 0$	$f(t) = c$	$p(t) = 0$

Table 6: A locally incentive compatible and payment-only incentive compatible mechanism

he gets an alternative inside this subset of alternative (giving positive value). But this is not enough for incentive compatibility in this type space. Mishra and Roy (2013) show that incentive compatibility also implies that the cutoffs along these rays have to be carefully chosen and the alternatives below and above these cutoffs have to be chosen in a particular manner.

References

Archer, A. and R. Kleinberg (2008): "Truthful Germs are Contagious: A Local to Global Characterization of Truthfulness," in *In Proceedings of the 9th ACM conference on Electronic commerce (EC-08)*, Springer (Lecture Notes in Computer Science).

——— (2014): "Truthful Germs are Contagious: A Local to Global Characterization of Truthfulness," *Forthcoming, Games and Economic Behavior*.

Armstrong, M. (2000): "Optimal multi-object auctions," *The Review of Economic Studies*, 67, 455–481.

Ashlagi, I., M. Braverman, A. Hassidim, and D. Monderer (2010): "Monotonicity and Implementability," *Econometrica*, 78, 1749–1772.

Barbera, S. (2010): "Strategy-proof Social Choice," in *Handbook of Social Choice and Welfare*, ed. by K. J. Arrow, A. K. Sen, and K. Suzumura, North-Holland, 441–449.

Berger, A., R. Müller, and S. H. Naeemi (2010): "Path-Montonicity and Incentive Compatibility," Working Paper, Maastricht University.

Carbajal, J. C. and R. Muller (2015): "Implementability under Monotonic Transformations in Differences," *Journal of Economic Theory*, 160, 114–131.

Carroll, G. (2012): "When are Local Incentive Constraints Sufficient?" *Econometrica*, 80, 661–686.

Chung, K.-S. and W. Olszewski (2007): "A Non-Differentiable Approach to Revenue Equivalence," *Theoretical Economics*, 2, 1–19.

Demange, G. (1982): "Single-peaked Orders on a Tree," *Mathematical Social Sciences*, 3, 389–396.

Fotakis, D. and E. Zampetakis (2013): "Truthfulness Flooded Domains and the Power of Verification for Mechanism Design," in *Proceedings of the Workshop on Internet and Network Economics (WINE)*, 202–215.

Fudenberg, D. and J. Tirole (1991): *Game Theory*, MIT Press, Cambridge.

Heydenreich, B., R. Muller, M. Uetz, and R. V. Vohra (2009): "Characterization of Revenue Equivalence," *Econometrica*, 77, 307–316.

Kirkegaard, R. (2014): "A Unifying Approach to Incentive Compatibility in Moral Hazard Problems," Working Paper, University of Guelph.

Krishna, V. and E. Maenner (2001): "Convex Potentials with an Application to Mechanism Design," *Econometrica*, 69, 1113–1119.

Milgrom, P. and I. Segal (2002): "Envelope Theorems for Arbitrary Choice Sets," *Econometrica*, 70, 583–601.

Mishra, D., A. Pramanik, and S. Roy (2014): "Multidimensional Mechanism Design in Single Peaked Type Spaces," *Journal of Economic Theory*, 153, 103–116.

Mishra, D. and S. Roy (2013): "Implementation in Multidimensional Dichotomous Domains," *Theoretical Economics*, 8, 431–466.

Moulin, H. (1980): "On Strategyproofness and Single-peakedness," *Public Choice*, 35, 437–455.

Reffgen, A. (2015): "Strategy-proof social choice on multiple and multi-dimensional single-peaked domains," *Journal of Economic Theory*, 157, 349–383.

Rochet, J. C. (1987): "A Necessary and Sufficient Condition for Rationalizability in a Quasi-linear Context," *Journal of Mathematical Economics*, 16, 191–200.

Rockafellar, R. T. (1970): *Convex Analysis*, Princeton University Press.

Saks, M. E. and L. Yu (2005): "Weak Monotonicity Suffices for Truthfulness on Convex Domains," in *Proceedings of 7^{th} ACM Conference on Electronic Commerce*, ACM Press, 286–293.

Saporiti, A. (2009): "Strategy-proofness and single-crossing," *Theoretical Economics*, 4, 127 – 163.

Sato, S. (2013a): "Strategy-proofness and the reluctance to make large lies: the case of weak orders," *Social Choice and Welfare*, 40, 479–494.

——— (2013b): "A sufficient condition for the equivalence of strategy-proofness and non-manipulability by preferences adjacent to the sincere one," *Journal of Economic Theory*, 148, 259–278.

Sprumont, Y. (1991): "The Division Problem with Single-Peaked Preferences: A Characterization of the Uniform Allocation Rule," *Econometrica*, 59, 509–519.

Vohra, R. V. (2011): *Mechanism Design: A Linear Programming Approach*, Cambridge University Press.