Discussion Papers in Economics

# Implementation in Multidimensional Domains with Ordinal Restrictions

**Debasis Mishra**

**Anup Pramanik**

**Souvik Roy**

May 2013

Discussion Paper 13-07

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

# Implementation in Multidimensional Domains with Ordinal Restrictions

Debasis Mishra, Anup Pramanik, and Souvik Roy *

May 28, 2013

**Abstract**

We consider implementation of a deterministic allocation rule using transfers in quasi-linear private values environments. We show that if the type space is a multidimensional domain satisfying some ordinal restrictions, then an allocation rule is implementable in such a domain if and only if it satisfies a familiar and simple condition called *2-cycle monotonicity*. Our ordinal restrictions cover type spaces which are non-convex, e.g., the single peaked domain and its generalizations. We apply our result to show that in the single peaked domain, a local version of 2-cycle monotonicity is necessary and sufficient for implementation and every locally incentive compatible mechanism is incentive compatible.

KEYWORDS. implementation, 2-cycle monotonicity, revenue equivalence, local incentive compatibility.

JEL CODES. D44, D47, D71, D82, D86.

*All authors are affiliated to Economics and Planning Unit, Indian Statistical Institute, New Delhi. Corresponding author: Debasis Mishra, Economics and Planning Unit, Indian Statistical Institute, 7, SJS Sansanwal Marg, New Delhi - 110016, India, Email: dmishra@isid.ac.in.

# 1 Introduction

An enduring theme in mechanism design is to investigate conditions that are necessary and sufficient for implementing an allocation rule. We investigate this question in private values and quasi-linear utility environments when the set of alternatives is finite and the allocation rule is deterministic (i.e., does not randomize). An allocation rule in such an environment is implementable if there exists a payment rule such that truth-telling is a dominant strategy for the agents in the resulting mechanism. Our main result is that in a large class of multidimensional domains (type spaces) that satisfy some ordinal restrictions, implementability is equivalent to a simple condition called *2-cycle monotonicity*. The 2-cycle monotonicity condition requires the following: given the types of other agents, if the alternative chosen by the allocation rule is $a$ when agent $i$ reports its type to be $t_i$ and the alternative chosen by the allocation rule is $b$ when agent $i$ reports its type to be $s_i$, then it must be that

$$t_i(a) - t_i(b) \geq s_i(a) - s_i(b),$$

where for any alternative $x$, $t_i(x)$ and $s_i(x)$ denote the values of alternative $x$ in types $t_i$ and $s_i$ respectively.

One of the earliest papers to pursue this question was Rochet (1987), who proved a very general result. He showed that a significantly stronger condition called cycle monotonicity is necessary and sufficient for implementability in any type space - see also Rockafellar (1970). When the set of alternatives is finite and the type space is convex, 2-cycle monotonicity implies cycle monotonicity (Bikhchandani et al., 2006; Saks and Yu, 2005; Ashlagi et al., 2010). Though convexity is a natural geometric property satisfied in many economic environments, it excludes many interesting type spaces. Moreover, how far this result extends to domains that do not satisfy convexity remain an intriguing question - we discuss this issue in detail in Section 2. A primary objective of this paper is to formulate restrictions on type spaces without the convexity assumption made in the literature and answer the question of implementability in such multidimensional type spaces. Indeed, our restrictions allow many interesting multidimensional non-convex type spaces. Prominent type spaces covered by our formulation are the single peaked domain [1] and its generalizations. In all these domains, we show that 2-cycle monotonicity is necessary and sufficient for implementability. To our knowledge, this paper is the first to identify such a large class of interesting non-convex domains where 2-cycle monotonicity characterizes implementability.

A characterization of implementability using 2-cycle monotonicity is useful because the

[1]Roughly, a single peaked type is defined using a strict and complete order on the set of alternatives. A type is single peaked if the values of alternatives decrease as we go to the *left* or *right* (where left and right are defined with respect to the given order) of the peak (the highest valued alternative).

cycle monotonicity condition, which can be used to characterize implementability in any domain, is a difficult condition to use and interpret. On the other hand, 2-cycle monotonicity is a simpler condition and the appropriate extension of the monotonicity condition used by Myerson (1981) to characterize implementability in the single object auction model. For this reason, 2-cycle monotonicity is often referred to as *weak monotonicity* (Bikhchandani et al., 2006; Saks and Yu, 2005) or *monotonicity* (Ashlagi et al., 2010).

A contribution of this paper is to apply the 2-cycle monotonicity characterization to derive sharper results in the single peaked domain. We consider a local version of the 2-cycle monotonicity condition, where we require the 2-cycle monotonicity condition to hold between a type and all types in its $\epsilon$-neighborhood. We show that such a local 2-cycle monotonicity condition is necessary and sufficient for implementation in the single peaked domain. Then, we explore the consequence of imposing local incentive compatibility for mechanisms in the single peaked domain. Local incentive compatibility requires that incentive constraints to hold between a type and all its types in the $\epsilon$-neighborhood. We show that in the single peaked domain, every locally incentive compatible mechanism is incentive compatible. The counterparts of these results are only known for convex type spaces (Archer and Kleinberg, 2008; Carroll, 2012).

We use a novel method to impose *ordinal* restriction on type spaces. Such a method of imposing ordinal restriction is usually followed in the mechanism design literature without transfers (a la strategic voting or social choice theory literature). To see how such restrictions can be imposed in a cardinal environment like ours, note that a type in our environment is a vector in $\mathbb{R}^{|A|}$, where $A$ is the set of alternatives. Now, let us restrict attention to *strict* types, where value of no two alternatives is the same. Such a type must induce a complete and strict ordering on $A$. We put restrictions by allowing only a subset of orderings that can be induced by any type. In particular, we require the permissible orderings to satisfy two properties, which we call *connectedness* and *lifting* - we discuss them in detail in Section 3. The set of all strict types in $\mathbb{R}^{|A|}_{++}$ that induce an ordering belonging to a set of permissible orderings define our ordinally admissible domain. Ordinally admissible domains can be generated, for instance, by considering all orderings over $A$ that are single peaked and then considering all types that induce these single peaked orderings - this will generate the single peaked domain. The complete domain ($\mathbb{R}^{|A|}_{++}$) is also an ordinally admissible domain. Many generalizations of the single peaked domain are also ordinally admissible.

We are aware of only one instance where such an idea of ordinal restriction was pursued in this literature. The *order-based* domain considered in Bikhchandani et al. (2006) is defined by considering a *weak partial* order on the set of alternatives and *every* type must induce this order. Firstly, an order-based domain is convex. Second, we consider *a set of strict and complete* orders and a type in our domain must induce one of the orders in this set. In that sense, our domain restriction is different from the order-based domains, but neither is stronger than the other.

We allow for indifferences by considering domains that are closures of some ordinally admissible domain and show that 2-cycle monotonicity is necessary and sufficient for implementability in these domains too. Further, in all these domains, revenue equivalence holds.

The rest of the paper is organized as follows. We start by giving a background on implementability and cycle monotonicity in Section 2. This allows us to cover the relevant literature in some detail in that section. We then motivate our research question by analyzing an example in Section 2.1. Our domain restriction and the main results are stated in Section 3. We give various examples where our domain restrictions hold in Section 3.1. Section 3.2 discusses the proof of our main result (Theorem 1), but all other proofs are in the Appendix. Section 3.3 discusses payments and revenue equivalence results. Section 4 discusses application of our results in the single peaked domain. In order to make formal connection with our model and definitions, we defer a detailed discussion on related literature till Section 5. We conclude by giving some future research directions in Section 6.

## 2 Implementation and Cycle Monotonicity

We consider a model with a single agent. As is well known in this literature, this is without loss of generality. All our results generalize easily to a model with multiple agents. The single agent is denoted by $i$. The set of alternatives for agent $i$ is denoted by $A_i$. In an $n$-agent model, $A_i$ denotes the possible allocations of agent $i$. [2] The type (private information) of agent $i$ is a vector $t_i \in \mathbb{R}^{|A_i|}$. If agent $i$ has type $t_i$, then $t_i(a)$ will denote the value of agent $i$ for alternative $a$. We assume private values and quasi-linear utility. This means that if alternative $a$ is chosen and agent $i$ with type $t_i$ makes a payment of $p_i$, then his net utility is given by $t_i(a) - p_i$.

Not all possible vectors in $\mathbb{R}^{|A_i|}$ can be a type of agent $i$. Let $D_i \subseteq \mathbb{R}^{|A_i|}$ be the type space of agent $i$ - these are the permissible types of agent $i$. The set $D_i$ will be referred to as a **domain** of agent $i$. An **allocation rule** is a mapping $f : D_i \to A_i$. A **payment rule** of agent $i$ is a mapping $p_i : D_i \to \mathbb{R}$. A **mechanism** consists of an allocation rule and a payment rule.

Definition 1 *An allocation rule $f$ is* **implementable** *if there exists a payment rule $p_i$ such*

[2] For instance, in a model with $n$ agents and $n$ objects, where each agent can be assigned exactly one object, $A_i$ will be the set of objects and *not* the set of *matchings*.

*that for every $s_i, t_i \in D_i$, we have*

$$s_i(f(s_i)) - p_i(s_i) \geq s_i(f(t_i)) - p_i(t_i).$$

*In this case, we will say that $p_i$ implements $f$ and $(f, p_i)$ is an* **incentive compatible** *mechanism.*

The primary objective of this paper is to give a simple necessary condition on the allocation rule that is also sufficient for implementability in a large class of interesting domains. For this, we revisit a classic condition that is already known to be necessary and sufficient for implementability in *any domain.*

Definition **2** *An allocation rule $f$ is $K$-***cycle monotone***, where $K \geq 2$ is a positive integer, if for every finite sequence of types $(t_i^1, t_i^2, \ldots, t_i^k)$, with $k \leq K$, we have*

$$\sum_{j=1}^{k} \left[ t_i^j(f(t_i^j)) - t_i^j(f(t_i^{j-1})) \right] \geq 0, \tag{1}$$

*where $t_i^0 \equiv t_i^k$. An allocation rule $f$ is* **cyclically monotone** *if it is $K$-cycle monotone for every positive integer $K \geq 2$.*

It is well known that implementability is equivalent to cycle monotonicity (Rochet, 1987; Rockafellar, 1970). This result is very general - it works on any domain $D_i$ and does not even require $A_i$ to be finite. [3] However, cycle monotonicity is a difficult condition to use and interpret since it requires verifying non-negativity of Inequality 1 for arbitrary length sequences of types. In a series of papers, it has been established that a significantly weaker condition than cycle monotonicity is sufficient for implementation in various interesting domains. Bikhchandani et al. (2006) showed that 2-cycle monotonicity is sufficient for implementability if $D_i$ is an *order-based* domain - this includes many interesting auction domains. Saks and Yu (2005) show that 2-cycle monotonicity is sufficient for implementation if $D_i$ is convex - this extends the result in Bikhchandani et al. (2006) because an order-based domain is convex. Ashlagi et al. (2010) extend this result to show that if the closure of $D_i$ convex, then 2-cycle monotonicity is sufficient for implementation.

However, Mishra and Roy (2013) show that there are interesting non convex domains where 2-cycle monotonicity is not sufficient for implementation. Further, they identify an interesting class of non-convex domains where 3-cycle monotonicity is sufficient for implementation but 2-cycle monotonicity is not sufficient.

[3] When the set of alternatives is finite, this result can be slightly strengthened to say that implementability is equivalent to $|A_i|$-cycle monotonicity (Mishra and Roy, 2013).

Interestingly, Ashlagi et al. (2010) establish a surprising result by allowing for *randomization*, i.e., an allocation rule picks a probability distribution over alternatives. They show that if every 2-cycle monotone randomized allocation rule is also cyclically monotone in a domain $D_i$ of dimension at least 2, then the closure of $D_i$ must be convex.

It is not clear how far this result is true if $f$ is allowed to be deterministic. Vohra (2011) contains a simple example of a non-convex domain with four alternatives where every *deterministic* allocation rule satisfying 2-cycle monotonicity is implementable. In his example, Vohra (2011) considers the sale of two objects $\alpha$ and $\beta$ to agents. The set of alternatives is the set of all subsets of $\{\alpha, \beta\}$. The restriction on values of agents is the following: $t_i(\{\alpha, \beta\}) = \max(t_i(\{\alpha\}), t_i(\{\beta\}))$ for each $i$. Hence, each agent desires at most one object, though he may be assigned both the objects. The type space here is non-convex. To see this, consider two types of agent $i$

$$t_i(\emptyset) = 0, t_i(\{\alpha\}) = 3, t_i(\{\beta\}) = 4, t_i(\{\alpha, \beta\}) = 4$$
$$s_i(\emptyset) = 0, s_i(\{\alpha\}) = 5, s_i(\{\beta\}) = 4, s_i(\{\alpha, \beta\}) = 5.$$

A convex combination of $(0.5, 0.5)$ of these two types generates values 4 for objects $\alpha$ and $\beta$ but a value of 4.5 for the bundle of objects $\{\alpha, \beta\}$. This violates the restriction on the type space.

Note that if we allow at most one object to be assigned to an agent, then the type space becomes convex, and we can apply earlier result to conclude that 2-cycle monotonicity is sufficient for implementation. However, by allowing the alternative $\{\alpha, \beta\}$, but still having a restriction that agents desire at most one object, we get to a non-convex type space. The result in Vohra (2011) shows that 2-cycle monotonicity is sufficient for implementation in such an example. It is not clear on how to extend the proof of this example if there are more than two objects.

## 2.1 A Motivating Example

Since the type space in the example in Vohra (2011) seems to be a slight modification of a convex type space, it is still unclear whether there are interesting non-convex type space where 2-cycle monotonicity is sufficient for implementation. The result in Ashlagi et al. (2010) shows that if every 2-cycle monotone randomized allocation rule is implementable in a multidimensional domain, then it must be convex. This shows that there is a significant gap in understanding implementability of deterministic allocation rules in non-convex multidimensional domains. We give below a motivating example to show that there are interesting non-convex domains where the current results are silent. Our results in the paper will apply to such domains.

Consider a general scheduling problem as follows. A number of firms procure products/parts from a supplier over a time horizon. In each time period, the supplier can only supply to one firm. Every firm $i$ has a time period $\tau_i^*$ where it gets the maximum value from getting its products supplied. The firms have *single peaked* preference over time, i.e., for firm $i$ for any time periods $\tau, \tau'$, if $\tau < \tau' < \tau_i^*$ or $\tau > \tau' > \tau_i^*$, then firm $i$ values supply of its products at time period $\tau'$ to time period $\tau$ (this may be due to inventory carrying cost and delivery delay costs).

The type space in this example is non-convex. To see this, suppose there are just three time periods $\{1, 2, 3\}$ and consider two single peaked types of agent (firm) $i$: $s_i := (6, 4, 3)$ (peak is period 1) and $t_i := (3, 4, 6)$ (peak is period 3). A convex combination $\frac{s_i+t_i}{2}$ produces the type $(4.5, 4, 4.5)$, which is no longer single peaked.

In such non-convex domains, we characterize implementability using 2-cycle monotonicity and apply this result to obtain other interesting results on incentive compatible mechanisms. Thus, there are interesting type spaces where earlier results are silent and our results provide sharp characterizations of implementability and incentive compatibility.

## 3 Domains with Ordinal Restrictions

We now formally define our domain. Every type $t_i$ induces a weak order on the set of alternatives. Let $\mathcal{P}$ be the set of all strict orderings of the set of alternatives $A_i$. A type $t_i$ is **strict** if $t_i(a) \neq t_i(b)$ for all $a, b \in A_i$. A strict type $t_i$ is consistent with a strict ordering $P \in \mathcal{P}$ if $t_i$ induces the strict ordering $P$. Let $\mathcal{D} \subseteq \mathcal{P}$ be some subset of strict orderings of the set of alternatives $A_i$. We denote by $T(\mathcal{D})$ the set of *all* strict types in $\mathbb{R}^{|A_i|}_{++}$ that are consistent with the strict orderings in $\mathcal{D}$, i.e.,

$$T(\mathcal{D}) := \{t_i \in \mathbb{R}^{|A_i|}_{++} : t_i \text{ is consistent with some } P \in \mathcal{D}\}.$$

The type spaces we consider in the paper are of the nature $T(\mathcal{D})$, where $\mathcal{D}$ is some subset of $\mathcal{P}$. [4] Our restriction on type space will be derived by imposing conditions on $\mathcal{D}$. We use the notation $P(k)$ to denote the $k$-th ranked alternative in ordering $P$. Let $\mathcal{D} \subseteq \mathcal{P}$.

1. The first condition is a notion of *connectedness*. Construct an undirected graph $G(\mathcal{D})$ as follows. The set of nodes in $G(\mathcal{D})$ is the set of alternatives $A_i$. For any $a, b \in A_i$, there is an edge $\{a, b\}$ in $G(\mathcal{D})$ if and only if there is an ordering $P \in \mathcal{D}$ such that $P(1) = a$ and $P(2) = b$, and another ordering $P' \in \mathcal{D}$ such that $P'(1) = b$ and $P'(2) = a$. So, there is an edge between a pair of alternatives $a$ and $b$ if there is a preference ordering

[4] We will extend our results to allow for indifferences and then the type space will be of the form $cl(T(\mathcal{D}))$, where $cl(\cdot)$ denotes the closure.

where $a$ is top and $b$ is second and another preference ordering where $b$ is top and $a$ is second. We will say a pair of alternatives $a, b \in A_i$ are **neighbors** if there is an edge $\{a, b\}$ in the graph $G(\mathcal{D})$. We say $\mathcal{D}$ is **connected** if $G(\mathcal{D})$ is connected, i.e., for every pair of alternatives $a, b \in A_i$ there is a path in $G(\mathcal{D})$ between $a$ and $b$. Note that we do *not* require that there is an edge between every pair of alternatives. We only require that they are connected by a path.

2. Then, we have two conditions on *lifting*.

   (a) We say $\mathcal{D}$ satisfies **top lifting** if for every pair of alternatives $a, b \in A_i$ and for every ordering $P \in \mathcal{D}$ such that $aPb$, there exists an ordering $P' \in \mathcal{D}$ such that (a) $P'(1) = a$ and (b) if $bPc$ for any alternative $c$ then $bP'c$.

   Table 1 illustrates the idea of top lifting. Suppose $A_i = \{x, y, z, x', y', a, b\}$ and $P$ (as shown in Table 1) is in $\mathcal{D}$. Consider $a, b \in A_i$. Note that in $P'$, $a$ is the top ranked alternative. The alternatives that were worse than $b$ in $P$ continue to be worse than $P'$. The ordering $P'$ thus allows the pair of alternatives $a$ and $b$ to satisfy top lifting at $P$.

| $P$ | $P'$ |
|---|---|
| $x$ | $[a]$ |
| $y$ | $x$ |
| $[a]$ | $z$ |
| $z$ | $[b]$ |
| $[b]$ | $y$ |
| $x'$ | $y'$ |
| $y'$ | $x'$ |

Table 1: Top Lifting Property

   (b) Another condition allows us to *lift* a neighbor. We say $\mathcal{D}$ satisfies **neighbor lifting** if for any pair of alternatives $a$ and $b$ such that $a$ and $b$ are neighbors and for any ordering $P$ where $P(1) = a$ and $P(k+1) = b$ for some integer $k \geq 1$, there exists an ordering $P'$ where $P'(k) = P(k+1) = b$ and $P'(k+1) = P(k)$ and for all $j \neq \{k, k+1\}$ we have $P'(j) = P(j)$.

   We illustrate the neighbor lifting property using an example in Table 2. Suppose $A_i = \{x, y, z, a, b, x', y'\}$. Consider $x$ and $z$ that are neighbors in some $\mathcal{D}$. Suppose $P \in \mathcal{D}$, where $P$ is as shown in Table 2. If $\mathcal{D}$ satisfies neighbor lifting, then all the orderings shown in Table 2 belongs to $\mathcal{D}$. Note that as we go from the left most ordering to right in Table 2, the alternative $z$ swaps position with the alternative just above it and the ranking of other alternatives remain the same - this is highlighted in Table 2 by putting the position of alternatives $x$ and $z$ in square brackets.

| $P$ | $P'$ | $\bar{P}$ | $\hat{P}$ |
|---|---|---|---|
| $[x]$ | $[x]$ | $[x]$ | $[z]$ |
| $y$ | $y$ | $[z]$ | $[x]$ |
| $a$ | $[z]$ | $y$ | $y$ |
| $[z]$ | $a$ | $a$ | $a$ |
| $b$ | $b$ | $b$ | $b$ |
| $x'$ | $x'$ | $x'$ | $x'$ |
| $y'$ | $y'$ | $y'$ | $y'$ |

Table 2: Neighbor Lifting Property

We say $\mathcal{D}$ satisfies **lifting** if it satisfies top lifting and neighbor lifting.

Definition 3 *A domain $D_i$ is* **ordinally admissible** *if there exists a connected $\mathcal{D} \subseteq \mathcal{P}$ satisfying lifting such that $D_i = T(\mathcal{D})$.*

Our main result is that in ordinally admissible domains, 2-cycle monotonicity is necessary and sufficient for implementation.

Theorem **1** *Suppose $D_i$ is an ordinally admissible domain and $f : D_i \to A_i$ is an onto allocation rule. Then, $f$ is implementable if and only if it is 2-cycle monotone.*

The proof of Theorem 1 is given in Section 3.2. Though we assume that every type in $D_i$ induces a strict ordering on $A_i$, we can allow for indifference. We denote by $cl(D_i)$ the closure of the set $D_i$.

Theorem **2** *Let $D_i$ be an ordinally admissible domain and $f : cl(D_i) \to A_i$ is an onto allocation rule. Then, $f$ is implementable if and only if it is 2-cycle monotone.*

The proof of Theorem 2 is given in the Appendix and it uses Theorem 1.

## 3.1 Examples

Before presenting the proof of Theorem 1, we give various examples of $\mathcal{D}$ that satisfy the connectedness and the lifting property. The first example is a convex domain, and hence, the earlier results already imply Theorem 1 in this domain. The remaining examples are of non-convex multidimensional domains, and hence, the earlier results are silent on such examples.

1. Complete domain. In the complete domain, $\mathcal{D}$ is the set of all possible orderings over $A_i$. As a result, $G(\mathcal{D})$ is a complete graph - there is an edge between every pair of alternatives. So, connectedness holds. Further, lifting is satisfied since all possible orderings are in $\mathcal{D}$. Note that the complete domain covers the multi-object auction model with unit demand. To see this, let $A_i$ be the set of heterogeneous objects and agent $i$ can be assigned exactly one object from $A_i$. Then, the complete domain assumption requires that any vector of positive valuations can be assigned to the objects.

2. Single peaked domain. The single-peaked domain is a well studied domain in social choice theory (Moulin, 1980). The set of admissible orderings $\mathcal{D}$ in the single peaked domain is defined as follows. There is an exogenously given ordering $\succ$ on the set of alternatives $A_i$. We say an ordering $P$ is a single-peaked ordering if for any pair of alternatives $a, b \in A_i$, $P(1) \succ a$ and $a \succ b$ implies $aPb$ and $b \succ a$ and $a \succ P(1)$ implies $aPb$. In words, as we go away from the *peak* of an ordering, the agent must prefer the alternatives less.

   The set of admissible orderings $\mathcal{D}$ in the single peaked domain induces a graph $G(\mathcal{D})$ in which every node has degree either one or two. Alternatives $a$ and $b$ are *neighbors* in the single peaked domain if there are no alternatives between $a$ and $b$ in the ordering $\succ$. Clearly, if an alternative $a \in A_i$ is a peak in an ordering $P \in \mathcal{D}$, then only one of the neighbors of $a$ can be second ranked in $P$. Hence, there is an edge between $a$ and each of its neighbors. The top ranked and worst ranked alternatives according to $\succ$ has exactly one neighbor and they have only degree one in $G(\mathcal{D})$. It is easy to see that $G(\mathcal{D})$ is a *line graph*, and hence, $\mathcal{D}$ is connected. Figure 1 shows $G(\mathcal{D})$ for $A_i = \{a, b, c, x, y\}$ and $a \succ b \succ c \succ x \succ y$.

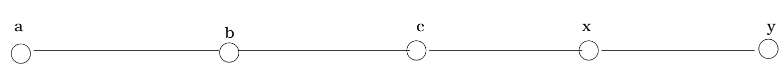


Figure 1: Graph $G(\mathcal{D})$ for the single peaked domain $\mathcal{D}$.

We next argue that $\mathcal{D}$ satisfies lifting. To see this, consider any pair of alternatives $a, b \in A_i$. Suppose $a \succ b$ (a similar argument works if $b \succ a$). Let $L(a) := \{c : c \succ a\}$, $R(b) := \{c : b \succ c\}$, and $B(a, b) := \{c : c \succ b\} \cap \{c : a \succ c\}$. Now, we can always find a single peaked preference ordering in which (1) $a$ is the peak, (2) alternatives in $B(a, b)$ lie below $a$ but above $b$, (3) alternatives in $R(b)$ lie below $b$, and (4) alternatives in $L(a)$ are below $a$ but can be put in any position. Hence, from any preference ordering $P \in \mathcal{D}$ where $aPb$ we can construct an ordering $P'$ such that top lifting is satisfied. Figure 2 shows a graphical illustration of how a type $s_i$ (in solid blue lines) can be converted to a type $s'_i$ to satisfy top lifting for a pair of alternatives $a, b$. Note that if $P$ is the ordering induced by $s_i$ we have $aPb$. Now, consider the type $s'_i$ (in dashed red lines in Figure 2) and suppose it induces the ordering $P'$. Now, note that in $P'$, $a$ is the top alternative and the alternatives that were below $b$ in $P$ continue to be below $b$ in $P'$. Hence, top lifting is satisfied.

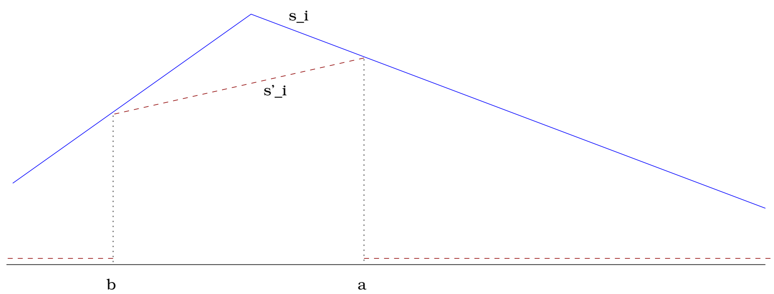


Figure 2: Illustration of top lifting in the single peaked domain.

The same observation also enables the domain to satisfy neighbor lifting. To see this, suppose $a$ and $b$ are two neighbors with $a \succ b$ and there is an ordering $P$ where $aPb$ and $P(1) = a$. By definition $b$ is the top ranked alternative among alternatives in $R(a)$. Clearly, increasing the rank of $b$ will either maintain $a$ to be the peak or make $b$ the peak, and in both cases, it will not disturb the single peakedness.

3. Semi single peaked domain. In semi single peaked domain, the ordering of alternatives to one of the sides may not decrease but on the other side, it must decrease as in the single peaked domain. Formally, there is an exogenously given ordering $\succ$ on the set of alternatives $A_i$. We say an ordering $P$ is a single-peaked to left if for any pair of alternatives $a, b \in A_i$, $b \succ a$ and $a \succ P(1)$ implies $aPb$. Similarly, an ordering $P$ is single-peaked to right if for any pair of alternatives $a, b \in A_i$, $P(1) \succ a$ and $a \succ b$ implies $aPb$.

   The set of admissible orderings $\mathcal{D}$ is a semi single peaked domain if it consists of either all left single peaked orderings or all right single peaked orderings. Consider a semi single peaked domain $\mathcal{D}$ and assume that it consists of all right single peaked orderings. Then, every alternative $a$ has an edge with alternative $b$ in $G(\mathcal{D})$ if $B(a, b) = \emptyset$, where $B(a, b)$ is the set of alternatives between $a$ and $b$ according to $\succ$. Note that $a$ can be top ranked in an ordering and any alternative $b \in L(a)$ can be second ranked. However, if $b$ is top ranked, $a$ can be second ranked only if $B(a, b) = \emptyset$. For this reason, $G(\mathcal{D})$ is a line graph, which is connected. [5]

We can verify that the semi single peaked domain satisfies top lifting. To see this, consider an ordering $P \in \mathcal{D}$, where $aPb$. If $b \in L(a)$, we can construct an ordering where $a$ is top ranked and $b$ is second ranked by lowering all the alternatives (except $a$) below $b$ but maintaining single peaked to the right of $a$. If $b \in R(a)$, then alternatives to the left of $a$ can be lowered sufficiently to make $a$ the peak and it will automatically maintain single peakedness to the right of $a$.

Finally, we verify that the semi single peaked domain satisfies neighbor lifting. Since the neighbors of an alternative are the same in the semi single peaked domain and the single peaked domain, the argument for this is similar to the argument which showed that the single peaked domain satisfies neighbor lifting.

4. Single peaked domain with characteristics. This is a generalization of the single peaked domain. We are now exogenously given a set of orderings $\mathcal{S}$ over the set of alternatives. The domain $\mathcal{D}$ consists of all orderings that are single peaked with respect to some $\succ \in \mathcal{S}$. If $\mathcal{S}$ is a singleton, this is precisely the single peaked domain. Suppose the set of alternatives are objects. An element of $\mathcal{S}$ can be interpreted as a "characteristic" of the objects. Depending on the characteristic used by an agent to rank the objects, his preference must be single peaked with respect to that characteristic.

   Consider an example with $A_i = \{a, b, c, x, y\}$ and let $\mathcal{S} = \{\succ_1, \succ_2\}$, where $a \succ_1 b \succ_1 c \succ_1 x \succ_1 y$ and $y \succ_2 a \succ_2 b \succ_2 x \succ_2 c$. Figure 3 shows the graph $G(\mathcal{D})$ for this domain. The edges are derived from the single peaked restrictions on each characteristics.

   In general, the graph $G(\mathcal{D})$ is connected since $\mathcal{D}$ contains the single peaked domain, which is connected. Also, $\mathcal{D}$ satisfies lifting property. To see this, consider any $a, b \in A$ and suppose there is a preference ordering $P$ where $aPb$. Since $P$ is single peaked with respect to some $\succ \in \mathcal{S}$, we can apply the arguments for the single peaked domain to show that there is some ordering $P'$ that is single peaked with respect to $\succ$ such that top lifting holds for $a$ and $b$ at $P$. Similarly, neighbor lifting lifting is satisfied by

[5] The single peaked and the single peaked domains induce the same graph. This shows that two different domains can induce the same graph.

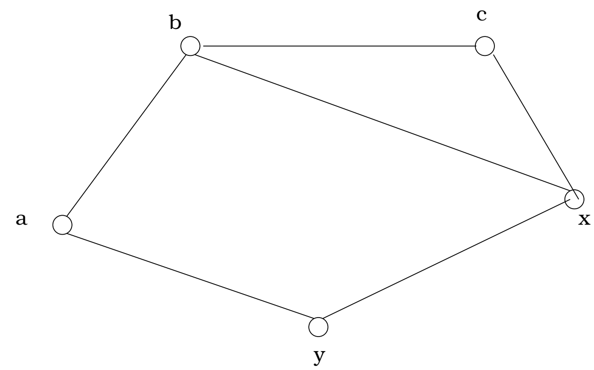


Figure 3: Graph $G(\mathcal{D})$ for the single peaked domain with two characteristics.

mimicking the argument of the single peaked domain.

5. Single peaked domain on a tree. This is another generalization of the single peaked domain. Here, we are given a graph $G$ which is a tree (connected and without any cycles). The preferences are single peaked along paths of this tree $G$. It is not difficult to see that the graph $G(\mathcal{D})$ is exactly the graph $G$ itself. For the example with $A_i = \{a, b, c, x, y\}$, Figure 4 shows a possible tree. The graph $G(\mathcal{D})$ for this domain is also the same graph. To see that $\mathcal{D}$ satisfies top lifting, consider $P \in \mathcal{D}$ and $a, b \in A_i$

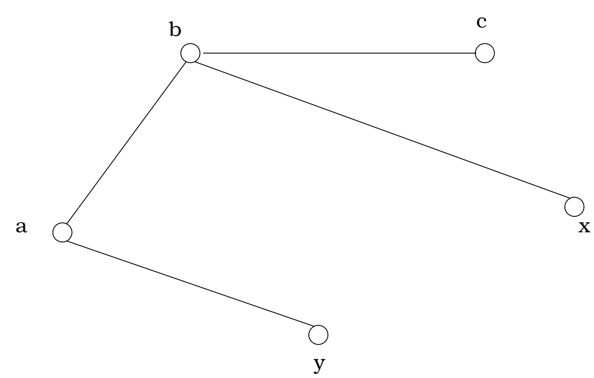


Figure 4: Graph $G(\mathcal{D})$ for the single peaked domain on a tree.

such that $aPb$. There is a unique path $\Pi$ in $G$ involving $a$ and $b$. By definition, we can construct an ordering $P'$ where all alternatives that do not lie on $\Pi$ lie below the alternatives in $\Pi$ by maintaining single peakedness on paths. For instance, in Figure 4, if the unique path between $c$ and $x$ is $(c, b, x)$ and we can construct an ordering where $x$ is the top, followed by $b, c, a, y$, and this will satisfy single peakedness on the tree. Now, we can mimic the arguments of the single peaked domain on alternatives in $\Pi$ to show that top lifting is satisfied. Similarly, if $P(1) = a$ and $a$ and $b$ are neighbors then again we can use this argument to construct $P'$ and then mimic the arguments of the single peaked domain on alternatives in $\Pi$ to show that neighbor lifting is satisfied.

### 3.2 Proof of Theorem 1

The proof of Theorem 1 will be done using a series of Lemmas. These lemmas will reveal the underlying structure of the domain. First, by Rochet (1987), if $f : D_i \to A_i$ is implementable, then it is 2-cycle monotone. Next, again by Rochet (1987), if $f$ is cyclically monotone, then it is implementable. So, we will show that if $f$ is 2-cycle monotone, then it is cyclically monotone. In the remainder of the section, we assume that $f$ is 2-cycle monotone.

Suppose $D_i = T(\mathcal{D})$ for some connected $\mathcal{D}$ satisfying lifting. For every $a \in A_i$, define $D_i(a)$ as follows.

$$D_i(a) := \{t_i \in D_i : f(t_i) = a\}.$$

Since $f$ is onto, $D_i(a)$ is non-empty. Next, for every $s_i, t_i \in D_i$, define $\ell(s_i, t_i)$ as follows.

$$\ell(s_i, t_i) := t_i(f(t_i)) - t_i(f(s_i)).$$

Notice that 2-cycle monotonicity is equivalent to requiring that for every $s_i, t_i \in D_i$, we have $\ell(s_i, t_i) + \ell(t_i, s_i) \geq 0$. Now, for every $a, b \in A_i$, define $d(a, b)$ as follows.

$$d(a, b) := \inf_{t_i \in D_i(b)} \big[t_i(b) - t_i(a)\big].$$

We state below a well known fact - see, for instance, Lemma 6 in Bikhchandani et al. (2006).

Lemma 1 *For every $a, b \in A_i$, $d(a, b) + d(b, a) \geq 0$.*

*Proof*: Suppose $d(a, b) + d(b, a) = -\epsilon < 0$ for some $a, b \in A_i$. This means, there is a $s_i \in D_i(b)$ and $t_i \in D_i(a)$ such that $[s_i(b) - s_i(a)] + [t_i(a) - t_i(b)] < 0$. But this means that $\ell(s_i, t_i) + \ell(t_i, s_i) < 0$, a contradiction to 2-cycle monotonicity. ■

The first step of the proof of Theorem 1 is the following lemma.

Lemma 2 *If $a, b$ are neighbors, then $d(a, b) + d(b, a) = 0$.*

*Proof*: Consider $a, b \in A_i$ such that $a$ and $b$ are neighbors. By Lemma 1, $d(a, b) + d(b, a) \geq 0$. Assume for contradiction $d(a, b) + d(b, a) = \epsilon > 0$. Then, either $d(a, b) > 0$ or $d(b, a) > 0$. Suppose $d(a, b) > 0$ - a similar proof works if $d(b, a) > 0$. Then, there is a type $s_i \in D_i(b)$ such that $d(a, b) \leq s_i(b) - s_i(a) < d(a, b) + \epsilon_1$, for any $\epsilon_1 > 0$ arbitrarily close to zero. Hence, $s_i(b) > s_i(a)$. Since $a$ and $b$ are neighbors and $\mathcal{D}$ is connected, there exists a $P \in \mathcal{D}$ such that $b$ is top ranked and $a$ is second ranked. We can construct a type $u_i \in D_i$ that induces $P$ and

$$u_i(x) = \begin{cases} s_i(x) & \text{if } x = a \\ s_i(x) - \delta & \text{if } x = b \\ < \min(s_i(x) - \delta, s_i(a)) & \text{if } x \notin \{a, b\}, \end{cases}$$

where $\delta \in (\epsilon_1, s_i(b) - s_i(a))$. Notice that since $s_i(b) > s_i(a)$, we have $u_i(b) > u_i(a)$ for sufficiently small $\delta > \epsilon_1$.

We will now argue that $f(u_i) = a$. First, if $f(u_i) = x \notin \{a, b\}$, we have $u_i(x) - u_i(b) < (s_i(x) - \delta) - (s_i(b) - \delta) = s_i(x) - s_i(b)$, which violates 2-cycle monotonicity. Second, if $f(u_i) = b$, we have $u_i(b) - u_i(a) = s_i(b) - \delta - s_i(a) \leq d(a, b) - (\delta - \epsilon_1) < d(a, b)$, which violates the definition of $d(a, b)$. Hence, $f(u_i) = a$.

But this implies that $d(b, a) \leq u_i(a) - u_i(b) = s_i(a) - s_i(b) + \delta \leq -d(a, b) + \delta$. Hence, $d(b, a) + d(a, b) \leq \delta$. Since $\delta, \epsilon_1$ can be chosen arbitrarily close to zero, this contradicts the fact that $d(a, b) + d(b, a) = \epsilon > 0$. ■

The next lemma establishes a crucial property.

Lemma 3 *For every pair of alternatives $a, c \in A_i$ such that $a$ and $c$ are not neighbors and any path $\Pi(a, c)$ between $a$ and $c$ in $G(\mathcal{D})$, there exists an alternative $b$ in this path such that $d(a, b) + d(b, c) \leq d(a, c)$.*

*Proof*: Fix $a, c \in A_i$ and a path $\Pi(a, c)$ between $a$ and $c$ in $G(\mathcal{D})$. Choose an $\epsilon > 0$ arbitrarily close to zero and a $t_i \in D_i(c)$ such that $d(a, c) \leq t_i(c) - t_i(a) < d(a, c) + \epsilon$. We consider two cases.

Case 1. $t_i(c) > t_i(a)$. Choose an alternative $b$ in $\Pi(a, c)$ between $a$ and $c$ in $G(\mathcal{D})$ such that $b$ is a neighbor of $c$. Let the ordering induced by $t_i$ be $P$. By top lifting, there exists an ordering $P' \in \mathcal{D}$ such that (a) $P'(1) = c$ and (b) if $aPc'$ for any alternative $c'$ then $aP'c'$. Hence, we can construct a type $t'_i \in D_i$ that induce $P'$ and $t'_i(c) = t_i(c)$, $t'_i(a) = t_i(a) - \epsilon'$, and $t'_i(x) < t_i(x)$ for all $x \notin \{a, c\}$, where $\epsilon' > 0$ but arbitrarily close to zero. Since $t'_i(c) = t_i(c)$ and $t'_i(x) < t_i(x)$ for all $x \neq c$, we have $[t'_i(x) - t'_i(c)] + [t_i(c) - t_i(x)] < 0$ for all $x \neq c$, and hence, by 2-cycle monotonicity, $f(t'_i) = c$. Further, $d(a, c) \leq t'_i(c) - t'_i(a) < d(a, c) + \epsilon$.

Let $\delta \in (t'_i(c) - t'_i(b) - d(b, c), t'_i(c) - t'_i(b) - d(b, c) + \epsilon'')$, for some $\epsilon'' > 0$ but arbitrarily close to zero. Since $f(t') = c$, we have $t'_i(c) - t'_i(b) \geq d(b, c)$. Hence, $\delta > 0$ but arbitrarily close to $t'_i(c) - t'_i(b) - d(b, c)$, which in turn is arbitrarily close to $t_i(c) - t_i(b) - d(b, c)$. Now, we construct a new type $s_i$ as follows.

$$s_i(x) = \begin{cases} t'_i(x) & \text{if } x = c \\ t'_i(x) + \delta & \text{if } x = b \\ t'_i(x) - \epsilon'' & \text{if } x \in A \setminus \{b, c\} \end{cases}$$

Note that $s_i$ is constructed by increasing the value of alternative $b$ from $t'_i(b)$ to $t'_i(b)+\delta$ and decreasing the value of other alternatives by arbitrarily small amount. Further, $c$ is top at $t'_i$ and $b$ is the neighbor of $c$ in $\Pi(c,a)$. Hence, by successive application of neighbor lifting, $s_i \in D_i$.

We argue that $f(s_i) = b$. First, suppose $f(s_i) = x \notin \{b,c\}$. Then, $s_i(x) - s_i(c) < t'_i(x) - t'_i(c)$, and this contradicts 2-cycle monotonicity. Next, suppose $f(s_i) = c$. Then, $d(b,c) \leq s_i(c) - s_i(b) = t'_i(c) - t'_i(b) - \delta < d(b,c)$, a contradiction. Hence, $f(s_i) = b$.

Now, $d(a,b) \leq s_i(b) - s_i(a) = [t'_i(b) - t'_i(a) + \delta] + \epsilon''$. Since $\delta < [t'_i(c) - t'_i(b)] - d(b,c) + \epsilon''$, we have $d(a,b) < [t'_i(c) - t'_i(a)] - d(b,c) + 2\epsilon'' \leq d(a,c) + \epsilon - d(b,c) + 2\epsilon''$. This implies that $d(a,b) + d(b,c) < d(a,c) + \epsilon + 2\epsilon''$. Since $\epsilon$ and $\epsilon''$ can be chosen arbitrarily close to zero, we conclude that $d(a,b) + d(b,c) \leq d(a,c)$.

Case 2. $t_i(c) < t_i(a)$. This case is similar to Case 1 except that the roles of $a$ and $c$ are reversed from Case 1. ■

Now, consider the following lemma.

Lemma 4 *For any pair of alternatives $a_1, a_k \in A_i$, let $(a_1, a_2, \ldots, a_k)$ be a sequence of alternatives on any path $\Pi(a_1, a_k)$ between $a_1$ and $a_k$ in $G(\mathcal{D})$. Then, the following are true.*

$$d(a_1,a_2) + d(a_2,a_3) + \ldots + d(a_{k-1},a_k) \leq d(a_1,a_k)$$
$$d(a_k,a_{k-1}) + d(a_{k-1},a_{k-2}) + \ldots + d(a_2,a_1) \leq d(a_k,a_1).$$

*Proof*: Consider any pair of alternatives $a_1, a_k \in A_i$ and let $(a_1, a_2, \ldots, a_k)$ be a sequence of alternatives on any path $\Pi(a_1, a_k)$ between $a_1$ and $a_k$ in $G(\mathcal{D})$. We do the proof using induction on $k$. If $k = 2$, then the claim is vacuously true. Suppose the claim is true for all $k < K$. If $k = K$, then by Lemma 3, there is an alternative $a_r \in \{a_2, \ldots, a_{K-1}\}$ such that $d(a_1,a_r) + d(a_r,a_K) \leq d(a_1,a_K)$. The paths $(a_1, \ldots, a_r)$ and $(a_r, \ldots, a_K)$ each contain less than $K$ nodes. By our induction hypothesis, $d(a_1,a_2) + \ldots + d(a_{r-1},a_r) \leq d(a_1,a_r)$ and $d(a_r,a_{r+1}) + \ldots + d(a_{K-1},a_K) \leq d(a_r,a_K)$. Hence, $d(a_1,a_2) + \ldots + d(a_{K-1},a_K) \leq d(a_1,a_K)$.

A similar argument shows that $d(a_k,a_{k-1}) + d(a_{k-1},a_{k-2}) + \ldots + d(a_2,a_1) \leq d(a_k,a_1)$ ■

The following lemma is well known - see, for instance, Heydenreich et al. (2009).

Lemma 5 *Suppose for every sequence of alternatives $(a_1, \ldots, a_k)$, we have*

$$\sum_{j=1}^{k} d(a_j, a_{j+1}) \geq 0,$$

*where $a_{k+1} \equiv a_1$. Then, $f$ is cyclically monotone.*

*Proof*: Consider any sequence of types $(t_i^1, \ldots, t_i^k)$ such that $f(t_i^j) = a_j$ for all $j \in \{1, \ldots, k\}$. Then, $[t_i^2(a_2) - t_i^2(a_1)] + \ldots + [t_i^k(a_k) - t_i^k(a_{k-1})] + [t_i^1(a_1) - t_i^1(a_k)] \geq d(a_1, a_2) + \ldots + d(a_{k-1}, a_k) + d(a_k, a_1) \geq 0$, where $d(a, a) = 0$ for any $a \in A_i$ by convention. Hence, $f$ is cyclically monotone. ■

LEMMA **6** *Suppose $(a_1, \ldots, a_k)$ is a path in $G(\mathcal{D})$. Then,*

$$\sum_{j=1}^{k} d(a_j, a_{j+1}) \geq 0,$$

*where $a_{k+1} \equiv a_1$.*

*Proof*: Let $(a_1, \ldots, a_k)$ be a path in $G(\mathcal{D})$. By Lemma 4, $d(a_k, a_1) \geq d(a_k, a_{k-1}) + \ldots + d(a_2, a_1)$. Hence,

$$\begin{aligned} d(a_1, a_2) + d(a_2, a_3) + \ldots + d(a_{k-1}, a_k) + d(a_k, a_1) &\geq \sum_{j=1}^{k-1} \big[d(a_j, a_{j+1}) + d(a_{j+1}, a_j)\big] \\ &= 0, \end{aligned}$$

where the last equality follows from the fact that $a_j$ and $a_{j+1}$ are neighbors for all $j \in \{1, \ldots, k-1\}$ and Lemma 2. ■

At this point, it will be useful to consider another graph $G^f$. [6] The set of nodes in $G^f$ is the set of alternatives $A_i$. It is a complete directed graph. Hence, for every pair of alternatives $a, b \in A_i$, there is an edge from $a$ to $b$ and an edge from $b$ to $a$. The path from an alternative $a$ to another alternative $b$ in $G^f$ is a directed path. Note that for every path $(a_1, a_2, \ldots, a_k)$ in $G^f$ from $a_1$ to $a_k$, the corresponding undirected path may or may not exist in $G(\mathcal{D})$. For any pair of alternatives $a_1, a_k \in A_i$, denote by $dist^f(a_1, a_k)$ the shortest path length from $a_1$ to $a_k$ in $G^f$.

LEMMA **7** *For any pair of alternatives $a, b \in A_i$, there exists a path $(a_1, a_2, \ldots, a_k)$ in $G(\mathcal{D})$, where $a \equiv a_1$ and $b \equiv a_k$, such that*

$$\sum_{j=1}^{k-1} d(a_j, a_{j+1}) = dist^f(a, b).$$

*Proof*: Fix $a, b \in A_i$ and choose a shortest path from $a$ to $b$ in $G^f$. Let this path be $(a'_1, \ldots, a'_h)$, where $a'_1 \equiv a$ and $a'_h \equiv b$. Now, take any edge $(x, y)$ in this path. If $x$ and $y$ are

[6] In Heydenreich et al. (2009), this graph is called the *allocation graph*.

not neighbors in $G(\mathcal{D})$, then by Lemma 5, we can pick any path $(x, c_1, \ldots, c_r, y)$ in $G(\mathcal{D})$ from $x$ to $y$, and $d(x,y) \geq d(x,c_1)+d(c_1,c_2)+\ldots+d(c_{r-1},c_r)+d(c_r,y)$. For every $j \in \{1,\ldots,h-1\}$, denote such a path from $a'_j$ to $a'_{j+1}$ in $G(\mathcal{D})$ as $\Pi(a'_j, a'_{j+1})$. Combining the paths $\Pi(a'_j, a'_{j+1})$ for all $j \in \{1,\ldots,k-1\}$, we get a path from $a$ to $b$ in $G(\mathcal{D})$, which we denote by $(a_1,\ldots,a_k)$ with $a \equiv a_1$ and $b \equiv a_k$, and some cycles in $G(\mathcal{D})$. By Lemma 6, these cycles have non-negative length (according to weights defined in $G^f$). Hence, $dist^f(a,b) \geq \sum_{j=1}^{k-1} d(a_j, a_{j+1})$. By definition, $dist^f(a,b) \leq \sum_{j=1}^{k-1} d(a_j,a_{j+1})$. Hence, $dist^f(a,b) = \sum_{j=1}^{k-1} d(a_j,a_{j+1})$. ■

This leads to the final lemma in the proof of Theorem 1.

LEMMA **8** *Every cycle of $G^f$ has non-negative length.*

*Proof*: Consider a cycle $(a_1,\ldots,a_k,a_1)$ in $G^f$. By Lemma 7, there is some path $(a_1,b_1,\ldots,b_r,a_k)$ in $G(\mathcal{D})$ such that $d(a_1,b_1)+d(b_1,b_2)+\ldots+d(b_{r-1},b_r)+d(b_r,a_k) = dist^f(a_1,a_k) \leq d(a_1,a_2)+\ldots+d(a_{k-1},a_k)$. This shows that

$$d(a_1,a_2)+\ldots+d(a_{k-1},a_k) \geq d(a_1,b_1)+d(b_1,b_2)+\ldots+d(b_{r-1},b_r)+d(b_r,a_k).$$

Now, consider the path $(a_k,b_r,\ldots,b_1,a_1)$ from $a_k$ to $a_1$. By Lemma 4,

$$d(a_k,a_1) \geq d(a_k,b_r)+d(b_r,b_{r-1})+\ldots+d(b_2,b_1)+d(b_1,a_1).$$

Adding the previous two inequalities, we get

$$\begin{aligned}\sum_{j=1}^{k} d(a_j,a_{j+1}) \geq {} & [d(a_1,b_1)+d(b_1,a_1)]+[d(b_1,b_2)+d(b_2,b_1)]+\ldots \\ & +[d(b_{r-1},b_r)+d(b_r,b_{r-1})]+[d(a_k,b_r)+d(b_r,a_k)] \\ = {} & 0,\end{aligned}$$

where $a_k \equiv a_1$ and the last equality follows from Lemma 2 and the fact that consecutive alternatives on the path $(a_1,b_1,\ldots,b_r,a_k)$ are neighbors. ■

Lemmas 8 and 5 establish that $f$ is cyclically monotone, and hence, implementable. This completes the proof of Theorem 1.

REMARK. The proof of Theorem 1 shows that we only use the lifting properties in Lemma 3. Hence, Theorem 1 is true in any domain induced from a connected $\mathcal{D}$ and satisfying Lemma 3. Further, lifting property can be weakened and we can still satisfy Lemma 3. Such modifications turn out to be cumbersome and do not add any significant domain where Theorem

1 holds. Hence, we do not report them here.

Remark. In many contexts, it is natural to assume that there is an alternative whose value is always zero (for instance, in auction problems, the alternative of not getting any object gives zero value to the agent). Though we do not explicitly allow this in our model, our proof can be modified straightforwardly to accommodate the fact that there is an alternative which is worst ranked and has value zero at every type.

### 3.3 Payments and Revenue Equivalence

It is well know that if $f$ is implementable, then the following payment rule implements $f$. Fix a type $s_i \in D_i$ and set $p_i(t_i) = 0$ for all $t_i$ with $f(t_i) = f(s_i)$. For all $t_i \in D_i$ such that $f(t_i) \neq f(s_i)$, set $p_i(t_i)$ equal to $dist^f(f(s_i), f(t_i))$. If $f$ is cyclically monotone, then, $p_i$ implements $f$ - see for instance, Vohra (2011) and Kos and Messner (2013).

The characterization of the set of *all* payment rules that implement an allocation rule is done using the revenue equivalence principle.

Definition 4 *An allocation rule $f$ satisfies* **revenue equivalence** *if for all payment rules $p_i, q_i$ that implement $f$, there exists a constant $\alpha_i \in \mathbb{R}$ such that for all $t_i \in D_i$*

$$p_i(t_i) = q_i(t_i) + \alpha_i.$$

In ordinally admissible domain, every onto implementable allocation rule satisfies revenue equivalence.

Theorem 3 *Suppose $D_i$ is an ordinally admissible domain. Then every onto implementable allocation rule $f : D_i \to A_i$ satisfies revenue equivalence.*

The proof of Theorem 3 is in the Appendix. We remark that Chung and Olszewski (2007) and Heydenreich et al. (2009) have shown that if $D_i$ is a connected subset of a topological space, then every implementable allocation rule satisfies revenue equivalence in such a domain. However, since $\mathcal{D}$ consists of strict orderings, $D_i$ is not connected and hence, our result is not a direct corollary of their results.

Our domain allows us to be precise on the nature of the shortest paths between any pair of nodes in $G^f$. Suppose $f$ is implementable. Now, for any pair of alternatives $a, b \in A_i$, consider any path $(a_1, \ldots, a_k)$ in $G(\mathcal{D})$, where $a_1 \equiv a$ and $b \equiv a_k$. Then, $dist^f(a, b) =$

$\sum_{j=1}^{k-1} d(a_j, a_{j+1})$. This follows from the fact that

$$\begin{aligned} 0 &= dist^f(a,b) + dist^f(b,a) \\ &\leq \sum_{j=1}^{k-1} d(a_j, a_{j+1}) + \sum_{j=k-1}^{1} d(a_{j+1}, a_j) \\ &= \sum_{j=1}^{k-1} \left[ d(a_j, a_{j+1}) + d(a_{j+1}, a_j) \right] \\ &= 0, \end{aligned}$$

where the first equality follows from Theorem 3 and the last equality from Lemma 2. Since in many examples, we know the structure of $G(\mathcal{D})$, this allows us to know the payments in these domains explicitly.

# 4 Applications to Single Peaked Domain

We show some applications of our main result in the single peaked domain. To remind, a single peaked domain is defined as follows. There is a strict ordering $\succ$ on the set of alternatives. For any type $t_i$ (inducing a strict ordering), we say $a \in A_i$ is a **peak** at $t_i$ if $t_i(a) > t_i(b)$ for all $b \in A_i$. The peak at type $t_i$ will be denoted as $\tau(t_i)$. A type $t_i$ is single peaked (with respect to $\succ$) if (a) $\tau(t_i) \succ a$ and $a \succ b$, implies $t_i(a) > t_i(b)$ and (b) $a \succ \tau(t_i)$ and $b \succ a$ implies $t_i(a) > t_i(b)$. Let $D_i^{\succ}$ be the set of all single peaked types in $\mathbb{R}_{++}$. As was shown earlier, $D_i^{\succ}$ is an ordinally admissible domain and our results can be directly applied on this domain. We show some applications of our results on this domain. We will use the following notation for the single peaked domain. For any pair of alternatives $a, b \in A_i$ with $a \succ b$, denote by $B(a,b)$ the set of all alternatives that are between $a$ and $b$ according to $\succ$, i.e., $B(a,b) := \{c : a \succ c, c \succ b\}$. As before, for any pair of alternatives $a, b \in A_i$ with $a \succ b$, $a$ and $b$ are called **neighbors** if $B(a,b) = \emptyset$.

## 4.1 Local 2-Cycle Monotonicity

In this section, we show that a *local* version of the 2-cycle monotonicity condition is equivalent to implementability in single peaked domain. A similar notion of local 2-cycle monotonicity was defined in Archer and Kleinberg (2008), who used it to show that if the domain is convex, then local 2-cycle monotonicity is equivalent to implementability. [7]

[7]The result in Archer and Kleinberg (2008) is more general since they consider the case where $A_i$ need not be finite, but in that case they assume an additional technical condition on $f$.

DEFINITION 5 *For any $\epsilon > 0$, an allocation rule $f : D_i \to A_i$ is 2-cycle monotone in $\epsilon$ neighborhoods if for every $t_i \in D_i$, and for every $s_i$ in the open ball around $t_i$ of radius $\epsilon$,*

$$\left[t_i(f(t_i)) - t_i(f(s_i))\right] + \left[s_i(f(s_i)) - s_i(f(t_i))\right] \geq 0.$$

*An allocation rule $f : D_i \to A_i$ is* **locally 2-cycle monotone** *if it 2-cycle monotone in $\epsilon$ neighborhoods for some $\epsilon > 0$.*

We show that this result is true in the single peaked domain.

THEOREM 4 *Let $D_i^{\succ}$ be the single peaked domain. An onto allocation rule $f : D_i^{\succ} \to A_i$ is implementable if and only if it is locally 2-cycle monotone.*

The proof of Theorem 4 is in the Appendix. It shows that 2-cycle monotonicity is ensured if it is satisfied in small neighborhoods around every type. Then, the result can be established by applying Theorem 1.

## 4.2 Local Incentive Compatibility

We now explore the implications of local incentive compatibility in the single peaked domain.

DEFINITION 6 *For any $\epsilon > 0$, a mechanism $(f, p)$ is incentive compatible in $\epsilon$ neighborhoods if for every $t_i \in D_i$, and for every $s_i$ in the open ball around $t_i$ of radius $\epsilon$,*

$$t_i(f(t_i)) - p_i(t_i) \geq t_i(f(s_i)) - p_i(s_i),$$

$$s_i(f(s_i)) - p_i(s_i) \geq s_i(f(t_i)) - p_i(t_i).$$

*A mechanism $(f, p)$ is* **locally incentive compatible** *if it is incentive compatible in $\epsilon$ neighborhoods for some $\epsilon > 0$.*

Notice that local incentive compatibility is a local version of incentive compatibility for *mechanisms*. Carroll (2012) shows that if the domain $D_i$ is convex, local incentive compatibility implies incentive compatibility. We show that in the single peaked domain, local incentive compatibility implies incentive compatibility. [8]

THEOREM 5 *Let $D_i^{\succ}$ be the single peaked domain and $f : D_i^{\succ} \to A_i$ be an allocation rule and $p_i : D_i^{\succ} \to A_i$ be a payment rule. If $(f, p_i)$ is locally incentive compatible then it is incentive compatible.*

[8] Carroll (2012) also considers local incentive compatibility in standard ordinal voting models *without transfers*, where he considers single peaked domains. His notion of local incentive compatibility in ordinal models is quite different (see also Sato (2013)).

The proof of Theorem 5 is in the Appendix. It uses Theorem 4 and other structural properties of ordinally admissible domains that we had shown in Section 3.2. The result in Carroll (2012) requires convex domain assumption and his proof is direct. However, we make use of local *implementability* characterization to derive our result. Thus, we make a connection between local implementability and local incentive compatibility.

## 5 Relation to the Literature

We discuss specific literature and its relation to our results. In the one dimensional model of single object auctions, Myerson (1981) characterizes implementable allocation rules using a monotonicity condition, which is equivalent to 2-cycle monotonicity (Myerson allows for randomization and considers Bayesian implementation). The cycle monotonicity characterization in Rochet (1987) can be thought of as an extension of Myerson's characterization to multidimensional models. The recent literature on multidimensional mechanism design started with the paper of Jehiel et al. (1999) who observed that besides 2-cycle monotonicity, an integral condition is required to ensure Bayesian implementability in multidimensional environments with randomization. However, if the set of alternatives is finite, the allocation rule is deterministic and the type space is convex, only 2-cycle monotonicity is sufficient (Bikhchandani et al., 2006; Saks and Yu, 2005; Ashlagi et al., 2010; Gui et al., 2004; Cuff et al., 2012). [9] Our results are extensions of these results to non-convex type spaces. Mishra and Roy (2013) also consider a non-convex domain, which they call rich dichotomous domain, and show that 3-cycle monotonicity is sufficient for implementability in their domain but 2-cycle monotonicity is not sufficient.

A parallel literature in multidimensional mechanism design pursues domains where revenue equivalence result in Myerson (1981) holds. Contributions to this are Krishna and Maenner (2001); Milgrom and Segal (2002); Chung and Olszewski (2007); Heydenreich et al. (2009); Carbajal (2010); Kos and Messner (2013). We use a characterization in Heydenreich et al. (2009) to prove revenue equivalence in our domains.

Most of the domain restrictions in multidimensional mechanism design is geometric (using assumptions like convexity or connectedness in topological spaces). Our ordinally admissible domain formulation is influenced by a vast literature in strategic social choice theory where transfers are not allowed. For instance, the connectedness and lifting properties

[9] There are many papers which characterize different extensions of implementability in convex domains using 2-cycle monotonicity and additional technical conditions - for Bayes-Nash implementation, see Jehiel et al. (1999) and Muller et al. (2007); for randomized implementation, see Archer and Kleinberg (2008); for implementation with general value functions, see Berger et al. (2010) and Carbajal and Ely (2013); for extension of cycle monotonicity to general environments, see Rahman (2011).

we discuss have close resemblance to similar properties being used to identify *dictatorial domains* (Aswal et al., 2003), *median domains* (Chatterji et al., 2013; Nehring and Puppe, 2007), *tops-only domains* (Chatterji and Sen, 2011; Weymark, 2008) in social choice theory. We find it interesting to observe that such conditions could be used in multidimensional mechanism design models with transfers to derive sufficient conditions for implementability. Since most of our non-convex domains are single peaked domains or their generalizations, we will like to point out that strategic social choice theory, starting with Moulin (1980) and Sprumont (1991), have a long tradition of studying these domains without monetary transfers. However, allowing for transfers in many of these domains is practical in many of these models. Hence, our results extend this literature to the case of transfers. At the same time, our characterizations using 2-cycle monotonicity are only implicit characterizations, unlike the characterizations in the strategic social choice theory, which are more explicit in describing the form of the implementable allocation rules. The counterpart to such explicit characterizations in the multidimensional mechanism design with transfers literature is Roberts' theorem (Roberts, 1979), who showed that *affine maximizers* are the only implementable allocation rules in the complete domain. We leave such characterizations in single peaked domains for future research.

We applied our results to derive specific results in the single peaked domain. A small literature in computer science has applied cycle monotonicity to derive computational results - see for instance Lavi and Swamy (2009); Babaioff et al. (2013). In contrast, our application is about understanding the connection between local incentive constraints and incentive compatibility.

# 6 Conclusion

We have provided necessary and sufficient conditions for implementability in domains involving some ordinal restrictions. Our domains include many interesting non-convex domains such as the single peaked domain and its generalizations. However, many interesting (non-convex) domains are still not covered by our result. Investigation of such domains is left as a direction for future research. Some other relaxations of the current model can also be investigated. For instance, the consequence of allowing infinite set of alternatives, considering randomized allocation rules, and Bayesian incentive compatibility in such models is not known yet. These questions are left for future. Finally, the difficult problem of finding expected revenue maximizing mechanisms in such multidimensional domains still remains an open problem.

# Appendix: Omitted Proofs

Proof of Theorem 2.

*Proof*: By Rochet (1987); Rockafellar (1970), implementability implies cycle monotonicity. Suppose $D_i$ is an ordinally admissible domain and $f : cl(D_i) \to A_i$ is 2-cycle monotone. Let $\bar{f} : D_i \to A_i$ be the restriction of $f$ to $D_i$. Since $f$ is 2-cycle monotone, $\bar{f}$ is 2-cycle monotone. By Theorem 1, $\bar{f}$ is implementable, and hence cyclically monotone. Assume for contradiction that $f$ is not cyclically monotone. Then, by Lemma 5, there exists a sequence of alternatives $(a_1, \ldots, a_k)$ such that $\sum_{j=1}^{k} d(a_j, a_{j+1}) < 0$, where $a_{k+1} \equiv a_1$.

Now, consider any $j \in \{1, \ldots, k\}$, let $t_i \in cl(D_i)$ be such that $f(t_i) = a_{j+1}$ and $d(a_j, a_{j+1}) < t_i(a_{j+1}) - t_i(a_j) < d(a_j, a_{j+1}) + \epsilon$ for some $\epsilon > 0$ and arbitrarily close to zero. By definition, there must exist an ordering $P \in \mathcal{D}$ such that $t_i$ is the limit point of a sequence of types in $D_i$ each inducing the ordering $P$. Notice that for any pair of alternatives $a, b \in A_i$, if $aPb$ then $t_i(a) \geq t_i(b)$.

Suppose $a_{j+1}Pa_j$ - a similar proof works if $a_jPa_{j+1}$. By top lifting, there is an ordering $P' \in \mathcal{D}$ such that $P'(1) = a_{j+1}$ and for all $a \in A_i$ if $aPa_j$ then $aP'a_j$ and if $a_jPa$ then $a_jP'a$. Now, choose $\epsilon' > 0$ but arbitrarily close to zero and construct a new type $t_i^{j+1}$ satisfying the following requirement:

$$t_i^{j+1}(x) = \begin{cases} t_i(x) & \text{if } x = a_{j+1} \\ t_i(x) - \epsilon & \text{if } x = a_j \\ \in (0, t_i(x)) & \text{if } x \notin \{a_j, a_{j+1}\}, \end{cases}$$

Note that by top lifting, such a $t_i^{j+1}$ can be constructed such that it lies in $D_i$ and induces $P' \in \mathcal{D}$. Further, since $t_i^{j+1}(a_{j+1}) - t_i(a_{j+1}) = 0 > t_i^{j+1}(a) - t_i(a)$ for all $a \neq a_{j+1}$, 2-cycle monotonicity implies that $f(t_i^{j+1}) = \bar{f}(t_i^{j+1}) = a_{j+1}$.

Finally, $t_i^{j+1}(a_{j+1}) - t_i^{j+1}(a_j) = t_i(a_{j+1}) - t_i(a_j) + \epsilon'$. Hence, $d(a_j, a_{j+1}) < t_i^{j+1}(a_{j+1}) - t_i^{j+1}(a_j) < d(a_j, a_{j+1}) + \epsilon' + \epsilon$. Adding over all $j \in \{1, \ldots, k\}$ and denoting $a_{k+1} \equiv a_1$, we get

$$\sum_{j=1}^{k} \left[t_i^{j+1}(a_{j+1}) - t_i^{j+1}(a_j)\right] < \sum_{j=1}^{k} d(a_j, a_{j+1}) + k(\epsilon' + \epsilon).$$

Since $\epsilon$ and $\epsilon'$ can be made arbitrarily close to zero,

$$\sum_{j=1}^{k} \left[t_i^{j+1}(a_{j+1}) - t_i^{j+1}(a_j)\right] \leq \sum_{j=1}^{k} d(a_j, a_{j+1}) < 0,$$

where the last inequality follows from our assumption. However, the ordering induced by

$t_i^{j+1}$ is a strict ordering, and hence, $t_i^{j+1} \in D_i$. By cycle monotonicity of $\bar{f}$, we know that

$$\sum_{j=1}^{k} \left[t_i^{j+1}(a_{j+1}) - t_i^{j+1}(a_j)\right] \geq 0.$$

This is a contradiction. ■

Proof of Theorem 3.

*Proof*: Heydenreich et al. (2009) showed that an implementable allocation rule $f$ satisfies revenue equivalence if and only if $dist^f(a,b) + dist^f(b,a) = 0$ for all $a, b \in A_i$. We show that this property is satisfied in our ordinally admissible domains. To see this, fix a pair of alternatives, $a, b \in A_i$. Since $f$ is cyclically monotone, $dist^f(a,b) + dist^f(b,a) \geq 0$ - the union of a shortest path from $a$ to $b$ and a shortest path from $b$ to $a$ gives rise to cycles, which have non-negative length due to cycle monotonicity. We show that $dist^f(a,b) + dist^f(b,a) \leq 0$, and this will prove the theorem. By Lemma 7, there is some path $(a_1, \ldots, a_k)$, with $a \equiv a_1$ and $b \equiv a_k$ between $a$ and $b$ in $G(\mathcal{D})$ such that $dist^f(a,b) = \sum_{j=1}^{k-1} d(a_j, a_{j+1})$. But

$$dist^f(a,b) + \sum_{j=k-1}^{1} d(a_{j+1}, a_j) = \sum_{j=1}^{k-1} \left[d(a_j, a_{j+1}) + d(a_{j+1}, a_j)\right] = 0,$$

where the equality follows from Lemma 2. By definition, $dist^f(b,a) \leq \sum_{j=k-1}^{1} d(a_{j+1}, a_j)$. Hence, $dist^f(a,b) + dist^f(b,a) \leq 0$. ■

Proof of Theorem 4.

*Proof*: Let $f : D_i^{\succ} \to A_i$ be a locally 2-cycle monotone allocation rule. We will show that $f$ is 2-cycle monotone, and by Theorem 1, $f$ is implementable. Suppose $\epsilon > 0$ such that for every $\bar{t}_i \in D_i^{\succ}$ and every $\bar{s}_i$ in the open ball around $\bar{t}_i$ of radius $\epsilon$, the 2-cycle monotonicity condition between $\bar{s}_i$ and $\bar{t}_i$ holds.

Consider any two types $s_i, t_i$ and let $f(s_i) = a, f(t_i) = b$. If $a = b$, then $[t_i(f(t_i)) - t_i(f(s_i))] + [s_i(f(s_i)) - s_i(f(t_i))] = 0$, and we are done. So, assume that $a \neq b$. In particular, suppose $a \succ b$. Assume for contradiction $[t_i(b) - t_i(a)] + [s_i(a) - s_i(b)] < 0$. Suppose $s_i(a) < s_i(b)$. Let the strict ordering induced by $s_i$ be $P$. Notice that the set of all types that induce the ordering $P$ is a convex set - we denote this set as $D_i(P)$. The restriction of $f$ onto $D_i(P)$ is locally monotone and since $D_i(P)$ is convex, by Archer and Kleinberg (2008), $f$ restricted to $D_i(P)$ is 2-cycle monotone. As a result, we can construct the following type $s_i' \in D_i(P)$ and apply 2-cycle monotonicity between $s_i$ and $s_i'$. Choose $\epsilon' > 0$ but arbitrarily close to zero.

$$s'_i(x) = \begin{cases} s_i(x) & \text{if } x = a \\ s_i(x) - \epsilon' & \text{if } x = b \\ \in (s_i(b) - \epsilon', s_i(b)) & \text{if } s_i(x) > s_i(b) \\ \in (s_i(a), s_i(a) + \epsilon') & \text{if } s_i(a) < s_i(x) < s_i(b) \\ \in (0, \epsilon') & \text{otherwise.} \end{cases}$$

Figure 5 shows a rough sketch of how $s'_i$ can be constructed from $s_i$. The type $s_i$ is shown with solid blue lines and the type $s'_i$ is shown with dashed red lines in Figure 5.

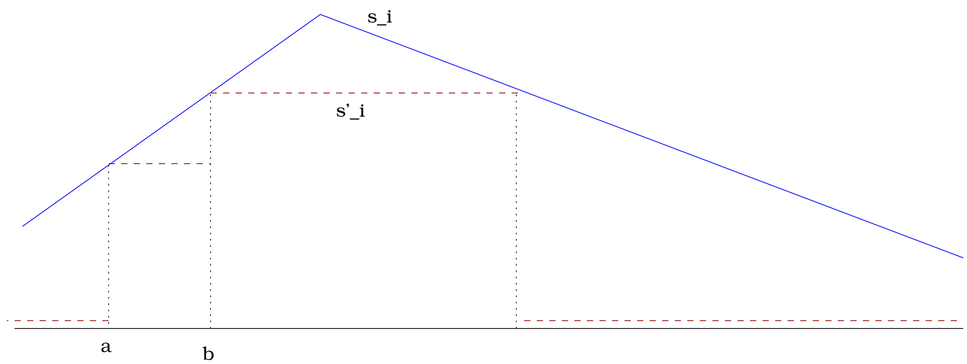


Figure 5: Illustration of $s_i$ and $s'_i$.

Since $s'_i(x) < s_i(x)$ for all $x \neq a$ and $s'_i(a) = s_i(a)$, by 2-cycle monotonicity $f(s'_i) = a$. Notice that the value of $s'_i(a) - s'_i(b)$ is arbitrarily close to $s_i(a) - s_i(b)$. The type $s'_i$ need not satisfy $\tau(s'_i) = b$. But we can construct a type $s''_i$ by perturbing $s'_i$ around its $\epsilon$ neighborhood and lowering the value of all alternatives which are above $b$ by a small amount and the value of all other alternatives except $a$, by even smaller amount such that $\tau(s''_i) = b$. This is possible because for all $x$ with $s'_i(x) > s'_i(b)$, we have $s'_i(x) \in (s_i(b) - \epsilon', s_i(b))$. Hence, we can apply local 2-cycle monotonicity between $s'_i$ and $s''_i$ to conclude that $f(s''_i) = a$. Let the ordering induced by $s''_i$ be $P'$. We can again lower the values of all the alternatives which lie in $B(a,b)$ to a value in $(s''_i(a), s''_i(a) + \epsilon')$ and which lie outside $B(a,b)$ (but not $a$ or $b$) to a value in $(0, \epsilon')$, while still inducing the ordering $P'$. Let this type be $\hat{s}_i$. Applying 2-cycle monotonicity between $s''_i$ and $\hat{s}_i$, we conclude that $f(\hat{s}_i) = a$. Further, $\hat{s}_i(a) - \hat{s}_i(b)$ is arbitrarily close to $s_i(a) - s_i(b)$ and $\tau(\hat{s}_i) = b$.

We now consider two cases.

Case 1. $t_i(b) > t_i(a)$. Then, as we constructed $\hat{s}_i$, we can construct a new type $\hat{t}_i$ inducing the ordering $P'$ with $f(\hat{t}_i) = b$, $\tau(\hat{t}_i) = b$, and $\hat{t}_i(b) - \hat{t}_i(a)$ arbitrarily close to $t_i(b) - t_i(a)$. Since $\hat{t}_i$ and $\hat{s}_i$ induce the same ordering $P'$, we can apply 2-cycle monotonicity between them to conclude $[\hat{s}_i(a) - \hat{s}_i(b)] + [\hat{t}_i(b) - \hat{t}_i(a)] \geq 0$. But $[\hat{s}_i(a) - \hat{s}_i(b)] + [\hat{t}_i(b) - \hat{t}_i(a)]$ is arbitrarily close to $[s_i(a) - s_i(b)] + [t_i(b) - t_i(a)]$, which is negative. This is a contradiction.

Case 2. $t_i(a) > t_i(b)$. Then, as we constructed $\hat{s}_i$, we can construct a new type $\hat{t}_i$ such that $f(\hat{t}_i) = b$, $\tau(\hat{t}_i) = a$, and $\hat{t}_i(b) - \hat{t}_i(a)$ arbitrarily close to $t_i(b) - t_i(a)$. Next, we transform $\hat{t}_i$ to $\bar{t}_i$ as follows. From $\hat{t}_i$, we decrease the value of all alternatives except $a$ and $b$ by arbitrarily small amount and we decrease the value of alternative $a$ to a level arbitrarily close to $\hat{t}_i(b)$ but still making it the peak. This new type is $\bar{t}_i$. Note that we can do this transformation such that $\bar{t}_i$ and $\hat{t}_i$ induce the same ordering. Hence, by 2-cycle monotonicity, $f(\bar{t}_i) = b$. Further, $\tau(\bar{t}_i) = a$, and $\bar{t}_i(b) - \bar{t}_i(a) < 0$ but arbitrarily close to zero. Then, we construct a new type $t_i^*$ by perturbing $\bar{t}_i$ around its $\epsilon$ neighborhood to induce the ordering induced by $\hat{s}_i$. Further, we can do so by maintaining $t_i^*(b) = \bar{t}_i(b)$ and $t_i^*(x) < \bar{t}_i(x)$ for all $x \neq b$. By local 2-cycle monotonicity, $f(t_i^*) = f(\bar{t}_i) = b$. Since $t_i^*$ and $\hat{s}_i$ induce the same ordering, we can apply 2-cycle monotonicity between them to conclude $[\hat{s}_i(a) - \hat{s}_i(b)] + [t_i^*(b) - t_i^*(a)] \geq 0$. But $t_i^*(b) - t_i^*(a)$ is arbitrarily close to zero and $\hat{s}_i(a) - \hat{s}_i(b) < 0$. This is a contradiction. ■

Proof of Theorem 5.

*Proof*: Since $(f, p_i)$ is locally incentive compatible, adding the incentive constraints of any pair of types in an $\epsilon$ neighborhood ensures that $f$ is locally 2-cycle monotone. By Theorem 4, $f$ is cycle monotone. We use this fact and do the proof in three steps.

Step 1. First, we show that for any type $t_i \in D_i^{\succ}$, agent $i$ cannot manipulate to a type $s_i \in D_i^{\succ}$ such that $f(s_i) = f(t_i)$. Assume for contradiction that $s_i, t_i \in D_i^{\succ}$ such that $f(s_i) = f(t_i) = a$ and $t_i(a) - p_i(t_i) < t_i(a) - p_i(s_i)$ or $p_i(s_i) < p_i(t_i)$.

Now, we go from $s_i$ to another type $\bar{s}_i$, where $\tau(\bar{s}_i) = a$. This can be done in a sequence of steps. In each step, value of alternative $a$ is increased slightly, but value of other alternatives are decreased. In particular, value of those alternatives whose value is larger than $s_i(a)$ are decreased at a faster rate than those whose value is smaller than $s_i(a)$. These rates can be chosen such that the value of alternatives except $a$ can be made arbitrarily close to zero. If these sequence of types are within $\epsilon$ neighborhood of each other, local incentive compatibility holds between them. Further, by 2-cycle monotonicity $f$ chooses $a$ in each type of this sequence. By local incentive compatibility, the payment at each type in this sequence must be the same. Hence, $f(\bar{s}_i) = a$ and $p_i(\bar{s}_i) = p_i(s_i)$.

In a similar fashion, we can construct a type $\bar{t}_i$ with $f(\bar{t}_i) = a$, $\tau(\bar{t}_i) = a$, $p_i(\bar{t}_i) = p_i(t_i)$, and $\bar{t}_i(x)$ is arbitrarily close to zero if $x \neq a$. If $\bar{t}_i(a) = \bar{s}_i(a)$, then $\bar{s}_i$ is in the $\epsilon$ neighborhood of $\bar{t}_i$. Hence, local incentive compatibility implies that $p_i(\bar{t}_i) = p_i(\bar{s}_i)$. This is a contradiction. Else, suppose $\bar{t}_i(a) < \bar{s}_i(a)$ - a similar proof works if $\bar{t}_i(a) > \bar{s}_i(a)$. Now, we increase the value of $\bar{t}_i(a)$ in small steps to reach $\bar{s}_i(a)$ value, and denote this new type as $\tilde{s}_i$. By virtue of 2-cycle monotonicity, for each type in this sequence, $f$ chooses $a$. Further, local incentive compatibility ensures that the payment for each type in this sequence is the same. Hence, $f(\tilde{s}_i) = a$ and $p_i(\tilde{s}_i) = p_i(\bar{s}_i) = p_i(s_i)$. But $\tilde{s}_i$ is in the $\epsilon$ neighborhood of $\bar{t}_i$. Hence, local incentive compatibility implies $p_i(\tilde{s}_i) = p_i(\bar{t}_i)$. This implies that $p_i(s_i) = p_i(t_i)$, a contradiction.

Step 2. Next, we show that for any type $t_i \in D_i^{\succ}$, agent $i$ cannot manipulate to a type $s_i \in D_i^{\succ}$ such that $f(s_i)$ and $f(t_i)$ are neighbors. Assume for contradiction that $s_i, t_i \in D_i^{\succ}$ such that $f(t_i) = a$, $f(s_i) = b$, $a$ and $b$ are neighbors, and $t_i(a) - p_i(t_i) < t_i(b) - p_i(s_i)$. By Step 1, we can abuse notation to write $p_i$ as a map $p_i : A \to \mathbb{R}$. Hence, $p_i(a) - p_i(b) > t_i(a) - t_i(b) \geq d(b, a)$. Hence, there is a type $\bar{t}_i$ such that $f(\bar{t}_i) = a$ and $\bar{t}_i(a) - \bar{t}_i(b)$ is arbitrarily close to $d(b, a)$. Moreover, by 2-cycle monotonicity, such a type $\bar{t}_i$ can be chosen such that the value of all alternatives besides $a$ and $b$ are arbitrarily close to zero. By Lemma 2, if $f$ is 2-cycle monotone, $d(a, b) = -d(b, a)$. Hence, there is a type $\tilde{t}_i$ arbitrarily close to $\bar{t}_i$ such that $f(\tilde{t}_i) = b$ and $\tilde{t}_i(b) - \tilde{t}_i(a)$ is arbitrarily close to $d(a, b)$. Then, we can apply local incentive compatibility between $\bar{t}_i$ and $\tilde{t}_i$ to conclude that $\bar{t}_i(a) - p_i(a) \geq \bar{t}_i(b) - p_i(b)$. This implies that $\bar{t}_i(a) - \bar{t}_i(b) \geq p_i(a) - p_i(b)$. Since $\bar{t}_i(a) - \bar{t}_i(b)$ is arbitrarily close to $d(b, a)$, this contradicts the fact that $p_i(a) - p_i(b) > d(b, a)$.

Step 3. Now, to show $(f, p_i)$ is incentive compatible, it suffices to show that for every $a, b \in A$,

$$p_i(b) - p_i(a) \leq d(a, b).$$

Choose $a, b \in A$, and assume without loss of generality that $a \succ b$. Consider a sequence of alternatives $(a_0 \equiv a, a_1, \ldots, a_k, a_{k+1} \equiv b)$ such that for all $j \in \{0, 1, \ldots, k\}$, we have $a_j$ and $a_{j+1}$ are neighbors and $a_j \succ a_{j+1}$. By Step 2, we get $p_i(a_{j+1}) - p_i(a_j) \leq d(a_j, a_{j+1})$ for all $j \in \{0, 1, \ldots, k\}$. Adding these over all $j \in \{0, 1, \ldots, k\}$, we get $p_i(b) - p_i(a) \leq \sum_{j=0}^{k} d(a_j, a_{j+1}) \leq d(a, b)$, where the last inequality is due to 2-cycle monotonicity and Lemma 4. ■

## 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).

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

Aswal, N., S. Chatterji, and A. Sen (2003): "Dictatorial domains," *Economic Theory*, 22, 45–62.

Babaioff, M., R. Kleinberg, and A. Slivkins (2013): "Multi-parameter Mechanisms with Implicit Payment Computation," in *Proceedings of the ACM Conference on Electronic Commerce*, ACM Press, Forthcoming.

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

Bikhchandani, S., S. Chatterji, R. Lavi, A. Mualem, N. Nisan, and A. Sen (2006): "Weak Monotonicity Characterizes Deterministic Dominant Strategy Implementation," *Econometrica*, 74, 1109–1132.

Carbajal, J. C. (2010): "On the Uniqueness of Groves Mechanisms and the Payoff Equivalence Principle," *Games and Economic Behavior*, 68, 763–772.

Carbajal, J. C. and J. Ely (2013): "Mechanism Design without Revenue Equivalence," *Journal of Economic Theory*, 148, 104–133.

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

Chatterji, S., R. Sanver, and A. Sen (2013): "On domains that admit well-behaved strategy-proof social choice functions," *Journal of Economic Theory*, 148, 1050–1073.

Chatterji, S. and A. Sen (2011): "Tops-only domains," *Economic Theory*, 46, 255–282.

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

Cuff, K., S. Hong, J. A. Schwartz, Q. Wen, and J. Weymark (2012): "Dominant Strategy Implementation with a Convex Product Space of Valuations," *Social Choice and Welfare*, 39, 567–597.

Gui, H., R. Muller, and R. Vohra (2004): "Characterizing Dominant Strategy Mechanisms with Multidimensional Types," Working Paper, Northwestern University.

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

Jehiel, P., B. Moldovanu, and E. Stacchetti (1999): "Multidimensional Mechanism Design for Auctions with Externalities," *Journal Economic Theory*, 85, 258–293.

Kos, N. and M. Messner (2013): "Extremal Incentive Compatible Transfers," *Journal of Economic Theory*, 148, 134–164.

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

Lavi, R. and C. Swamy (2009): "Truthful Mechanism Design for Multidimensional Scheduling via Cycle Monotonicity," *Games and Economic Behavior*, 67, 99–124.

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

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.

Muller, R., A. Perea, and S. Wolf (2007): "Weak Monotonicity and Bayes-Nash Incentive Compatibility," *Games and Economics Behavior*, 61, 344–358.

Myerson, R. B. (1981): "Optimal Auction Design," *Mathematics of Operations Research*, 6, 58–73.

Nehring, K. and C. Puppe (2007): "The structure of strategy-proof social choiceÑPart I: General characterization and possibility results on median spaces," *Journal of Economic Theory*, 135, 269–305.

Rahman, D. (2011): "Detecting Profitable Deviations," Working Paper, University of Minnesota.

Roberts, K. (1979): *The Characterization of Implementable Choice Rules*, North Holland Publishing, chap. Aggregation and Revelation of Preferences, 321–348, editor: J-J. Laffont.

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.

Sato, S. (2013): "A Sufficient Condition for the Equivalence of Strategy-proofness and Nonmanipulability 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.

Weymark, J. A. (2008): "Strategy-Proofness and the Tops-Only Property," *Journal of Public Economic Theory*, 10, 7–26.