


Characterizing implementable allocation rules in multi-dimensional environments

André Berger¹ · Rudolf Müller¹  · Seyed Hossein Naeemi¹

Received: 10 June 2015 / Accepted: 10 November 2016 / Published online: 5 December 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract We study characterizations of implementable allocation rules when types are multi-dimensional, monetary transfers are allowed, and agents have quasi-linear preferences over outcomes and transfers. Every outcome is associated with a valuation function that maps an agent's type to his value for this outcome. The set of types are assumed to be convex. Our main characterization theorem shows that allocation rules are implementable if and only if they are implementable on any two-dimensional convex subset of the type set. For finite sets of outcomes and continuous valuation functions, they are implementable if and only if they are implementable on every one-dimensional subset of the type set. This extends a characterization result by Saks and Yu (Weak monotonicity suffices for truthfulness on convex domains, pp 286–293, 2005) from models with linear valuation functions to arbitrary continuous valuation functions, and provides a simple proof of their result. Modeling multi-dimensional mechanism design the way we propose it here is of relevance whenever types are given by few parameters, while the set of possible outcomes is large, and when values for outcomes are non-linear functions in types.

1 Introduction

We investigate the following basic setting of asymmetric information, which appears in various forms in the theory of incentives. There are a single agent and a principal. The

We would like to thank Juan Carlos Carbajal, Benny Moldovanu, Xianwen Shi, and Rakesh Vohra for fruitful discussions and helpful comments.

✉ Rudolf Müller
r.muller@maastrichtuniversity.nl

¹ Department of Quantitative Economics, Maastricht University, PO Box 616, 6200 MD Maastricht, Netherlands

agent holds private information t from some set T . We call t the *type* of the agent, and T the *type set*. Depending on the agent's type the principal wants to select an allocation, or take an action, a from some set A . We call the function $f : T \rightarrow A$ which determines this selection the *allocation rule*. We allow for monetary transfers given by a payment function $p : T \rightarrow \mathbb{R}$ which the principal uses in order to orchestrate incentives. The agent has cardinal preferences for allocations parameterized by his type, given by a *valuation function* $v : A \times T \rightarrow \mathbb{R}$, and quasi-linear utility for allocations and payments. We will call the triple (T, A, v) an *environment*. Such environments occur as building blocks in numerous applications in the theory of incentives. In particular they have been studied in the context of mechanism design.

We assume that the agent and the principal interact by a revelation mechanism in which the agent announces a type t , which may be different from his true type s , and the principal allocates $f(t)$ and makes transfer $p(t)$, yielding utility $v(f(t), s) + p(t)$ for the agent. We call f *implementable* if there exists a p that makes truthful reports of the agent a weakly dominant strategy, that is, for all s, t in T :

$$v(f(s), s) + p(s) \geq v(f(t), s) + p(t). \quad (1)$$

Central questions in the theory of incentives are (1) a characterization of all implementable allocation rules f , (2) ways to construct payments p , and (3) conditions on when p is unique up to a constant. The latter property is called *revenue equivalence* due to its applications in auctions.

In this paper we provide answers to these questions for the case when T is a convex subset of \mathbb{R}^d . Our first result shows that an allocation rule is implementable if and only if it is implementable on every 2-dimensional subset of T . Let us call a rule f *line-implementable* if it is implementable when restricted to any line-segment on T , and *locally implementable* if for every type t there exists some open neighborhood of t on which it is implementable. Our second result states for the same setting as above that a rule is implementable if and only if it is locally implementable and line implementable. Both results do not require any assumptions on the valuation functions v or the cardinality of A . We only need convexity of T and that the allocation rule in question, if line-implementable, satisfies revenue equivalence on lines. We then restrict to valuation functions that are continuous in types. For such settings, we strengthen the second result by showing that an allocation rule with a finite range is implementable if and only if it is line implementable. The last result generalizes a well-known theorem by [Saks and Yu \(2005\)](#), and yields as well a new, very easy proof of the original result.

All results are achieved by using directed graphs whose node sets consist of the set of types, and which have arcs between any two types s and t with length equal to value differences (more precisely, the length of arc (s, t) equals the value difference for the outcome at t if an agent is of type t versus of type s). Non-existence of cycles with negative lengths, called *cycle monotonicity* determines whether a rule is implementable, in which case shortest paths lengths yield incentive compatible payments. Also, it is a property on path lengths that determines whether a rule satisfies revenue equivalence. This approach goes back to [Rochet \(1987\)](#). It has been put into graph theoretic terms by [Gui et al. \(2004\)](#) and [Heydenreich et al. \(2009\)](#), and fully refined in the monograph of [Vohra \(2011\)](#). Non-existence of negative cycles of two nodes, called

2-cycle monotonicity is a direct generalization of monotonicity of allocation rules in single-item auctions. This triggered a series of results showing that for particular multi-dimensional environments 2-cycle monotonicity implies cycle-monotonicity. All these results require v to be linear in T , in which case 2-cycle monotonicity coincides with line-implementability. Our results show that replacing 2-cycle monotonicity by line-implementability provides the right means to extend these results beyond the case of linear valuations. By way of an example we show that line-implementability is a strictly stronger condition than 2-cycle monotonicity.

The definition of an environment used here assumes a single agent, while mechanism design deals typically with more than one agent. This is not limiting the applicability of our results as our definition can be used to grasp the perspective of each individual agent in a mechanism design context. For example, if we are interested in dominant strategy implementable allocation rules f , we get an environment for each selected agent and for each possible type report of all other agents. The allocation rule describes the influence of the selected agent's type reports, given the reports of other agents. Similar does the payment rule determine his payment for outcomes, given the report of the other agents. If we are interested in Bayesian Nash Implementation, the allocation rule maps types of a selected agent to a distribution of outcomes, induced by the distribution of truthful type reports of other agents.

Related work Characterizing settings where 2-cycle monotonicity is sufficient for cycle monotonicity has been a prominent topic in mechanism design (Archer and Kleinberg 2014; Ashlagi et al. 2010; Bikhchandani et al. 2006; Carbajal and Müller 2015; Mishra et al. 2013; Saks and Yu 2005). Common to this literature is a representation of an environment that differs from ours. We shall call it a *domain representation*, as opposed to our *parameter representation*. A domain representation associates every type with a function τ , mapping outcomes to values. A domain is a set of such functions. Both representations are closely linked since domains can be considered as type sets in parameter representations with $v(\tau, a) = \tau(a)$, implying that v is in fact a linear function in types. Vice versa, parameter representations induce a domain $\{v(\cdot, t) \mid t \in T\}$. However, convex sets of types in parameter representations do not necessarily induce convex domains. In fact, our results strictly extend work on domain representations whenever the induced domain representation is not convex. We provide an example in Sect. 4. Also, the type set T in a parameter representation might be finite-dimensional while A is infinite, in which case the previous literature is silent as well. For finite A , a parametrization by types may allow for a low-dimensional compact representation of private information, contrary to a high-dimensional corresponding domain. Think for example of additive valuations in multi-item auctions, where a type represents a value for each of the m items, allowing for types of dimension m , while the corresponding domain has dimension $2^m - 1$.

Within the framework of parameter representations, a different strand of mechanism design literature deals with explicit representations of payments in terms of path integrals of a particular vector field, yielding generalizations of the Mirrlees representation of indirect utility (Mirrlees 1971). For valuation functions that are differentiable in types Milgrom and Segal (2002) show that such representations follow from the envelope theorem. Krishna and Maenner (2001) prove a similar representation to hold for convex valuation functions. Mirrlees' representation can as well be used for character-

izing implementable allocation rules. For example, Jehiel et al. (1999) and Jehiel and Moldovanu (2001) derive characterizations for auction environments with linear valuation functions. They show that existence and path-independence of certain integrals provides necessary and sufficient conditions for implementation. This approach has been simplified by Archer and Kleinberg (2014). They show that for convex type set T and linear valuation functions monotonicity on line-segments and path-independence of path-integrals along the border of triangles are sufficient for implementability. Furthermore, they show that any rule that is locally implementable, is implementable on all of T . Berger et al. (2009) extend the results of Archer and Kleinberg (2014) to convex valuation functions. We show that neither linearity nor convexity of valuation functions is needed to yield such characterizations, if one replaces the Mirrless representation of indirect utility by distances along line-segments in directed graphs. All what is needed is revenue equivalence on lines and convexity of the type set. Thereby, we significantly extend the approach by Archer and Kleinberg (2014) and Berger et al. (2009).

A different generalization of the characterization literature based on Mirrlees representations has been proposed by Carbajal and Ely (2013). They show how for settings without revenue equivalence a weaker form of Mirrlees representation can be achieved by integration of correspondences. We elaborate on this work in Sect. 3. Carroll (2012) has investigated the role of local implementability as well. He shows that every locally incentive compatible mechanism is incentive compatible if the type space is convex. In related work, Mishra et al. (2015) give a different characterization for ordinal type spaces that includes *payment-only incentive compatibility* as an additional necessary and sufficient condition. While Carroll as well as Mishra et al. derive characterizations in terms of local properties of an allocation rule *and* a payment rule, our results, as those of Archer and Kleinberg (2014) and Berger et al. (2009), yield characterizations in terms of local properties of just the allocation rule. At the same time, the results by Carroll and Mishra et al. are more general as they cover ordinal as well as polyhedral type spaces.

Organization In Sect. 2, we define our setting and introduce necessary notation and previous results. We present our main characterization of implementability (Theorem 3) in Sect. 3.1. Then we provide extensions of the results of Archer and Kleinberg (2014) about local implementability (Theorem 4, Sect. 3.2) and of Saks and Yu (2005) for allocation rules with finite range (Theorem 5, Sect. 3.3). In Sect. 4 we apply our results to an example given in Vohra (2011).

2 Incentive compatibility, cycle monotonicity and 2-cycle monotonicity

In this section we provide precise definitions and recall the network approach for our basic model. We consider environments (T, A, v) , where T is a set of types, A is a set of allocations, and $v : A \times T \rightarrow \mathbb{R}$ is a valuation function. We assume quasi linear utilities, so the utility of an agent of type $t \in T$ for some outcome $a \in A$ and payment π is equal to $v(a, t) + \pi$.

Definition 1 A *direct mechanism* (f, p) , consisting of an allocation rule $f : T \rightarrow A$ and a payment function $p : T \rightarrow \mathbb{R}$ is called *incentive compatible* (IC) if for all $s, t \in T$:

$$v(f(s), s) + p(s) \geq v(f(t), s) + p(t). \tag{2}$$

An allocation rule f is called *implementable* if there exists a payment function p that makes the mechanism (f, p) IC.

It is straightforward to see that adding a constant to a payment rule p of an IC mechanism yields again an IC mechanism. If payment rules are unique up to such modifications, we say that *revenue equivalence* holds:

Definition 2 An implementable allocation rule f satisfies *revenue equivalence* if for any two incentive compatible mechanisms (f, p) and (f, p') there exists $c \in \mathbb{R}$ such that $p(t) = p'(t) + c$ for all $t \in T$. An environment (T, A, v) satisfies revenue equivalence, if all implementable allocation rules satisfy revenue equivalence.

Rochet (1987) identified a property called *cycle monotonicity* that characterizes implementable allocation rules. It has later been related to node potentials in *type graphs* by Gui et al. (2004). Here, and further on, a graph consists of a set of *nodes* and a set of (directed) *arcs* between pairs of nodes.

Given an allocation rule f , the set of nodes of the type graph T_f is equal to T . Every pair of types $s, t \in T$ is connected by arcs from s to t and from t to s . We define arc lengths $l_u(s, t)$ for arcs of T_f as follows (and call them *u-length* between types $s, t \in T$):

$$l_u(s, t) = v(f(t), t) - v(f(t), s).$$

A path from node s to node t in T_f , or (s, t) -path for short, is defined as $P = (s = s_0, s_1, \dots, s_k = t)$ such that $s_i \in T$ for $i = 0, \dots, k$. The *u-length* of P is defined as

$$\text{length}_u(P) = \sum_{i=0}^{k-1} l_u(s_i, s_{i+1}).$$

A cycle is a path with $s = t$. For any t , we regard the path from t to t without any arcs as a (t, t) -path and define its length to be 0. Let $P(s, t)$ be the set of all (s, t) -paths. The *u-distance* from s to t is defined as

$$\text{dist}_u(s, t) = \inf_{P \in P(s, t)} \text{length}_u(P).$$

A *node potential* π with respect to *u-length* is a function $\pi : T \rightarrow \mathbb{R}$ such that for all $s, t \in T$ we have

$$\pi(t) \leq \pi(s) + l_u(s, t). \tag{3}$$

By the definition of *u-length*, implementability of an allocation rule f is equivalent with the existence of node potentials with respect to *u-length*. Thereby, for all t , potential $\pi(t)$ equals the net utility $v(f(t), t) + p(t)$ with respect to some incentive compatible payment rule p . Furthermore, revenue equivalence coincides with uniqueness of node potentials with respect to *u-lengths* up to a constant.

It is straightforward that if T_f has a node potential it cannot have a negative cycle. The opposite holds as well, as in the absence of negative cycles we can fix a type s and take distances from s to any type t to yield the node potential $\pi(t) := \text{dist}_u(s, t)$. This motivates Rochet's definition of cycle monotonicity and yields his characterization of implementability.

Definition 3 An allocation rule $f : T \rightarrow A$ is called *cycle monotone*, if for all cycles C , $\text{length}_u(C) \geq 0$. f is called *2-cycle monotone*,¹ if for all $s, t \in T$ it holds that:

$$l_u(s, t) + l_u(t, s) \geq 0.$$

Theorem 1 (Rochet (1987)) *An allocation rule $f : T \rightarrow A$ is implementable if and only if it is cycle monotone.*

For later reference we state Rochet's theorem in terms of distances and combine it with a relation between distances and payment differences that is straightforward to prove.

Corollary 1 *An allocation rule $f : T \rightarrow A$ is implementable if and only if for any $s, t \in T$:*

$$\text{dist}_u(s, t) + \text{dist}_u(t, s) \geq 0. \quad (4)$$

In this case, every payment p satisfies:

$$- \text{dist}_u(t, s) \leq v(f(t), t) + p(t) - v(f(s), s) - p(s) \leq \text{dist}_u(s, t). \quad (5)$$

Finally, a characterization of revenue equivalence due to Heydenreich et al. (2009) is a direct consequence of what has been said so far:

Theorem 2 (Heydenreich et al. (2009)) *Let f be an allocation rule that is implementable. Then f satisfies revenue equivalence if and only if for any $s, t \in T_f$:*

$$\text{dist}_u(s, t) + \text{dist}_u(t, s) = 0. \quad (6)$$

Combining Theorem 2 and Corollary 1 yields the following.

Corollary 2 *An allocation rule $f : T \rightarrow A$ is implementable and satisfies revenue equivalence if and only if for any $s, t \in T$:*

$$\text{dist}_u(s, t) + \text{dist}_u(t, s) = 0 \quad (7)$$

In this case, every payment p satisfies:

$$\text{dist}_u(s, t) = v(f(t), t) + p(t) - v(f(s), s) - p(s). \quad (8)$$

¹ In the literature, the terms *weakly monotone*, or *just monotone* is often used instead of 2-cycle monotone. For readability purposes we prefer to use the longer name 2-cycle monotone.

3 Characterizing implementability on convex type sets

In this section we consider environments (T, A, v) , where T is a convex subset of \mathbb{R}^d ($d \geq 1$), A is an arbitrary set, and $v : A \times T \rightarrow \mathbb{R}$ is a valuation function. We start by introducing the notion of line-implementability: when restricting the input of the allocation rule to any line segment in T , it is implementable. We show how line-implementability can be used to characterize implementability. Next we show that every allocation rule that is locally implementable is globally implementable. Both results need the additional assumption that line-implementable allocation rules satisfy revenue equivalence on these line segments. We explain at the end of this section, how this revenue equivalence assumption can be omitted and thereby relate our results to Carbajal and Ely (2013).

Finally, we turn to valuation functions that are continuous in types and prove that any allocation rule that is line-implementable and has finite range is globally implementable. In such settings, the revenue equivalence assumption can be made without loss of generality (Chung and Olszewski 2007; Heydenreich et al. 2009).

3.1 Line-implementability

We denote by $L_{s,t}$ the line segment between s and t in T :

$$L_{s,t} = \{s + \lambda(t - s) : \lambda \in [0, 1]\}.$$

Definition 4 Let T be convex. An allocation rule $f : T \rightarrow A$ is called *line implementable* if for any $s, t \in T$, the restriction of f to the line segment $L_{s,t}$ is implementable.

Obviously, every implementable allocation rule is line-implementable. Furthermore every line-implementable allocation rule is 2-cycle monotone. It is well-known that for linear valuation functions 2-cycle monotonicity and line-implementability are equivalent. The following example shows that for convex, but non-linear valuation functions, 2-cycle monotonicity is not sufficient for line-implementability, even if they are piecewise linear.

Example 1 Suppose $T = [0, 1]$ and $A = \{a, b, c\}$, and the valuation function is given by

$$v(a, t) = \begin{cases} 0 & t \leq \frac{2}{3} \\ 3t - 2 & t > \frac{2}{3}, \end{cases}$$

$v(b, t) = 3t$ and

$$v(c, t) = \begin{cases} 2 - 3t & t \leq \frac{1}{3} \\ 3t & t > \frac{1}{3}. \end{cases}$$

Consider the following allocation rule:

$$f(t) = \begin{cases} a & 0 \leq t < \frac{1}{3} \\ b & \frac{1}{3} \leq t \leq \frac{2}{3} \\ c & \frac{2}{3} < t \leq 1. \end{cases}$$

We verify 2-cycle monotonicity by calculating u -length for the following three cases.

(i) $0 \leq s \leq \frac{1}{3}$ and $\frac{1}{3} < t \leq \frac{2}{3}$

$$l_u(s, t) + l_u(t, s) = 3(t - s) \geq 0,$$

(ii) $0 \leq s \leq \frac{1}{3}$ and $\frac{2}{3} < t \leq 1$

$$l_u(s, t) + l_u(t, s) = 3s \geq 0,$$

(iii) $\frac{1}{3} < s \leq \frac{2}{3}$ and $\frac{2}{3} < t \leq 1$

$$l_u(s, t) + l_u(t, s) = 0.$$

However, there is a cycle with negative length:

$$l_u(0, 1) + l_u\left(1, \frac{1}{3}\right) + l_u\left(\frac{1}{3}, 0\right) = 1 - 2 + 0 = -1,$$

which means f is not implementable.

Archer and Kleinberg (2014) prove that for convex type spaces and linear valuation functions, 2-cycle monotonicity of an allocation rule together with path-independence on triangles of particular integrals defined by f is equivalent with implementability. Example 1 shows that this equivalence cannot hold for arbitrary valuations. Still, we can show that the same principle, as well as its consequences, applies if we replace 2-cycle monotonicity by line-implementability. Thereby, we do not even need integrals, but can fully rely on distances in the type graph. To do so, we need to define distances on lines.

Definition 5 Let T be convex. For any $s, t \in T$, the L_u -distance from s to t is defined as

$$\text{dist}_u^L(s, t) = \inf_{P \in P^L(s,t)} \text{length}_u(P),$$

where $P^L(s, t)$ is the set of all (s, t) -paths contained in $L_{s,t}$. For any $s \in T$, we define $\text{dist}_u^L(s, s) = 0$.

Definition 6 Let T be convex and $f : T \rightarrow A$ be implementable. We say f satisfies *revenue equivalence on lines*, if $f|_L$ satisfies revenue equivalence for all line segments $L = L_{s,t}, s, t \in T$.

Using these definitions we get the following theorem.

Theorem 3 Let $T \subseteq \mathbb{R}^d$ be convex and $f : T \rightarrow A$ an allocation rule. The following are equivalent:

1. f is implementable and satisfies revenue equivalence on lines.
2. f is line implementable and for any $s_1, s_2, s_3 \in T$:

$$\sum_{i=1}^3 \text{dist}_u^L(s_i, s_{i+1}) = 0, \tag{9}$$

where $s_4 = s_1$ and distances in (9) are taken with respect to $L_{s_i, s_{i+1}}$.

Proof (\implies) Since f is implementable by some payment function p , it is also implementable on each line segment using the same p . By Corollary 2 and (8), applied to any $L = L_{s,t}, s, t \in T$, revenue equivalence on lines implies

$$\text{dist}_u^L(s, t) = v(f(t), t) + p(t) - v(f(s), s) - p(s).$$

By summing up along a triangle given by types s_1, s_2 and s_3 we get

$$\sum_{i=1}^3 \text{dist}_u^L(s_i, s_{i+1}) = 0.$$

(\impliedby) Fix $x \in T$. For every $w \in T$ define the payment as:

$$p(w) = \text{dist}_u^L(x, w) - v(f(w), w),$$

where $L = L_{x,w}$. Now for every $s, t \in T$ we have:

$$\begin{aligned} p(t) - p(s) &= \text{dist}_u^L(x, t) - v(f(t), t) - \text{dist}_u^L(x, s) + v(f(s), s) \\ &\leq \text{dist}_u^L(x, t) + \text{dist}_u^L(s, x) - v(f(t), t) + v(f(s), s) \\ &= -\text{dist}_u^L(t, s) - v(f(t), t) + v(f(s), s) \\ &\leq \text{dist}_u^L(s, t) - v(f(t), t) + v(f(s), s) \\ &\leq l_u(s, t) - v(f(t), t) + v(f(s), s) \\ &= v(f(s), s) - v(f(t), s), \end{aligned}$$

where the first and the second inequality follows from Corollary 1, the second equality from (9), and the third inequality from the definition of dist_u .

If we take $s_3 = s_2$ we have:

$$\text{dist}_u^L(s_1, s_2) + \text{dist}_u^L(s_2, s_1) = 0.$$

Since s_1 and s_2 are arbitrary we can conclude according to Theorem 2 that f satisfies revenue equivalence on lines. \square

A few remarks about the conditions in the above theorem are at place.

Remark 1 Revenue equivalence on lines is a fairly mild assumption. For example, it holds when A is countable and valuation functions are equi-continuous (Chung and Olszewski 2007; Heydenreich et al. 2009), and for arbitrary A when valuation functions are differentiable functions of types (Milgrom and Segal 2002), or convex functions of types (Krishna and Maenner 2001). Berger et al. (2009) contains a direct proof of the last fact using type graphs.

Remark 2 Line-implementability of an allocation rule has to be verified on a case by case basis. However, in some situations more structure on the environment can make this task easier. One property for an environment that ensures line-implementability for any 2-cycle monotone allocation rule is the *increasing differences property* (Müller et al. 2007). An environment satisfies this property if and only if for all $s, t \in T$ and $x \in L_{s,t}$, and $a, b \in A$, we have that $v(a, t) - v(b, t) \geq v(a, x) - v(b, x)$ implies that $v(a, x) - v(b, x) \geq v(a, s) - v(b, s)$. Note that this definition is independent of the allocation rule f and therefore gives an easy way of identifying environments in which line-implementability can be replaced by 2-cycle monotonicity in Theorem 3.

Remark 3 For certain settings, the distances on lines in the above theorem can be explicitly computed using line integrals over corresponding vector fields, in particular when the valuation functions are linear (Archer and Kleinberg 2014), convex (Berger et al. 2009; Krishna and Maenner 2001) or differentiable functions of types (Berger et al. 2009; Milgrom and Segal 2002).

Remark 4 Carbajal and Ely (2013) show that for particular environments one can also get a characterization in the flavor of Theorem 3 without requiring revenue equivalence on lines. The trick is to have sufficient structure in order to be able to replace distances on lines $L_{s,t}$ by some function $\delta(s, t)$ which satisfies $\delta(s, t) \leq l_u(s, t)$ and $\delta(s, t) = -\delta(t, s)$ for all s, t in T . Carbajal and Ely show that integrals on the line segment between s and t with respect to an integrable correspondence defined by the allocation rule f provide us with such δ , if one imposes sufficient structure on the environment to guarantee the existence of the integrals. They also show for their environments that the existence of these integrals is implied by implementability. The functions δ given by these integrals satisfy in particular

$$-\text{dist}_u(t, s) \leq \delta(s, t) \leq \text{dist}_u(s, t),$$

which implies that $\delta(s, t) = \text{dist}_u(s, t)$ if and only if the allocation rule satisfies revenue equivalence.

We close this section by a corollary of Theorem 3, which extends a result by Vohra (2011, Theorem 4.2.11), who has proven it for randomized allocation rules over finitely many outcomes.

Corollary 3 *Let $T \subseteq \mathbb{R}^d$ be convex and (T, A, v) be an environment such that every line implementable allocation rule satisfies revenue equivalence on lines. Then an allocation rule $f : T \rightarrow A$ is implementable if and only if it is implementable on every two-dimensional convex subset of T .*

3.2 Local implementability

Archer and Kleinberg (2014) were the first ones who characterized implementability based on local implementability. Their proof requires valuation functions to be linear. Motivated by their results we introduce in this section the notion of local implementability and extend their results to general valuation functions. The characterization holds for any outcome space and any valuation function, except that we will need revenue equivalence on lines.

Definition 7 An allocation rule $f : T \rightarrow A$ is called *locally implementable* if for every $t \in T$ there exists an open neighborhood $U(t)$ around t such that $f|_{T \cap U(t)}$ is implementable.

Obviously, implementability guarantees local implementability. To prove the other direction we need the following lemma.

Lemma 1 *Let $T \subseteq \mathbb{R}^d$ be convex. If the allocation rule f is line implementable and satisfies revenue equivalence on lines, then for any $s, t \in T$ and $x \in L_{s,t}$ between s and t :*

$$\text{dist}_u^L(s, t) = \text{dist}_u^L(s, x) + \text{dist}_u^L(x, t). \tag{10}$$

Proof Fix $s, t \in T$ and x between s and t . Since f is implementable and satisfies revenue equivalence on $L_{s,t}$, $L_{s,x}$ and $L_{x,t}$, according to Theorem 2

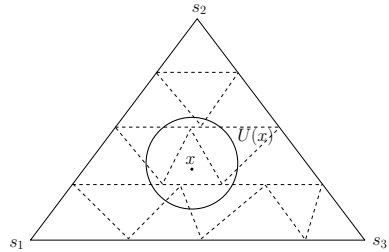
$$\begin{aligned} \text{dist}_u^L(s, t) &= -\text{dist}_u^L(t, s) \\ \text{dist}_u^L(s, x) &= -\text{dist}_u^L(x, s) \\ \text{dist}_u^L(x, t) &= -\text{dist}_u^L(t, x) \end{aligned}$$

and according to Theorem 1

$$\begin{aligned} \text{dist}_u^L(s, t) + \text{dist}_u^L(t, x) + \text{dist}_u^L(x, s) &\geq 0 \\ \text{dist}_u^L(t, s) + \text{dist}_u^L(s, x) + \text{dist}_u^L(x, t) &\geq 0. \end{aligned}$$

Altogether this yields 10. □

Fig. 1 Subdividing $\blacktriangle_{s_1,s_2,s_3}$ into sufficiently small triangles



In the following we denote for $s_1, s_2, s_3 \in T$, all three distinct, by $\blacktriangle_{s_1,s_2,s_3}$ the convex hull of s_1, s_2, s_3 and by Δ_{s_1,s_2,s_3} the path describing the boundary of $\blacktriangle_{s_1,s_2,s_3}$, i.e $L_{s_1,s_2} \cup L_{s_2,s_3} \cup L_{s_3,s_1}$, with direction $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_1$.

Now we are prepared to prove the main theorem of this section.

Theorem 4 *Let $T \subseteq \mathbb{R}^d$ be convex and assume that every line implementable allocation rule satisfies revenue equivalence on lines. Then an allocation rule f is implementable if and only if it is locally implementable and line implementable.*

Proof (\Rightarrow) Implementability of f on T implies implementability on subsets of T . Therefore f is locally implementable and line implementable.

(\Leftarrow) The proof is similar to the proof for linear valuations given in Archer and Kleinberg (2014), however as we need to apply our more general results Theorem 3 and Lemma 1 we include it. Let f be locally implementable and line implementable. Let $s_1, s_2, s_3 \in T$, all three distinct. Since $\blacktriangle_{s_1,s_2,s_3}$ is closed and bounded it is compact. Since f is locally implementable, for any point x in $\blacktriangle_{s_1,s_2,s_3}$ there is an open neighborhood $U(x)$ such that for any $x_1, x_2, x_3 \in U(x) \cap T$, all three distinct:

$$\sum_{i=1}^3 \text{dist}_u^L(x_i, x_{i+1}) = 0,$$

where $x_4 = x_1$.

Recall that by the Lebesgue Number Lemma for any open covering Λ of a compact metric space X there is a $\delta > 0$ such that for each subset of X having diameter less than δ , there exists an element Λ containing it.² This implies that there is a $\delta > 0$ such that every subset of $\blacktriangle_{s_1,s_2,s_3}$ of diameter less than δ is contained in at least one of the neighborhoods in which f is implementable. In particular, if we subdivide $\blacktriangle_{s_1,s_2,s_3}$ into M triangles $\blacktriangle_{s_1^1,s_2^1,s_3^1}, \blacktriangle_{s_1^2,s_2^2,s_3^2}, \dots, \blacktriangle_{s_1^M,s_2^M,s_3^M}$ (see Fig. 1), each of which having diameter less than δ , and orient the borders $\Delta_{s_1^j,s_2^j,s_3^j}$ consistently with Δ_{s_1,s_2,s_3} , we get

$$0 = \sum_{j=1}^M \sum_{i=1}^3 \text{dist}_u^L(s_i^j, s_{i+1}^j).$$

² For more information refer to Munkres (2000) or other classic books on topology.

In this formula, the distances along Δ_{s_1, s_2, s_3} appear exactly once. All distances of sides of $\Delta_{s_1^j, s_2^j, s_3^j}$ which are not contained in Δ_{s_1, s_2, s_3} appear exactly once in each direction and cancel each other out because revenue equivalence holds on lines. Applying Lemma 1, we have

$$\sum_{i=1}^3 \text{dist}_u^L(s_i, s_{i+1}) = \sum_{j=1}^M \sum_{i=1}^3 \text{dist}_u^L(s_i^j, s_{i+1}^j) = 0,$$

where $s_4 = s_1$ and $s_4^j = s_1^j$. Now according to Theorem 3, f is implementable. \square

3.3 Finite outcome space and continuous valuations

We prove in this section a generalization of the Theorem of Saks and Yu. We make use of a lemma that is of interest by its own as it describes a fairly general setting for which 2-cycle monotonicity is sufficient for implementability. Ashlagi et al. (2010) have proven a similar lemma for linear valuations and finite set of outcomes. We show that the result holds in a much more general case. To make it work, we have to make the assumption that valuation functions $v(a, \cdot)$ are continuous in t for all $a \in A$.

Lemma 2 *Let $T \subseteq \mathbb{R}^d$ and $v : A \times T \rightarrow \mathbb{R}$ be continuous in t for all $a \in A$. For $a \in A$ let*

$$D_a := \text{cl}(f^{-1}(a)) \cap T.$$

If $f : T \rightarrow A$ is 2-cycle monotone and $\bigcap_{a \in f(T)} D_a \neq \emptyset$, then f is implementable.³

Proof Let $\{s_1, \dots, s_k\} \subseteq T$ for some $k \geq 3$ and $t \in \bigcap_{a \in f(T)} D_a$. Fix $1 \leq i \leq k$. Since $t \in D_{f(s_{i+1})}$, there is a sequence $(t_j)_{j \in \mathbb{N}}$, such that $f(t_j) = f(s_{i+1})$ for every $j \in \mathbb{N}$ and $\lim_{j \rightarrow \infty} t_j = t$ where indices are taken modulo k . Note that for every $j \in \mathbb{N}$

$$\begin{aligned} l_u(s_i, s_{i+1}) &= v(f(s_{i+1}), s_{i+1}) - v(f(s_{i+1}), s_i) \\ &= v(f(s_i), s_i) - v(f(s_{i+1}), s_i) + v(f(s_{i+1}), s_{i+1}) - v(f(s_i), s_i) \\ &= v(f(s_i), s_i) - v(f(t_j), s_i) + v(f(s_{i+1}), s_{i+1}) - v(f(s_i), s_i) \\ &\geq v(f(s_i), t_j) - v(f(t_j), t_j) + v(f(s_{i+1}), s_{i+1}) - v(f(s_i), s_i) \\ &= v(f(s_i), t_j) - v(f(s_{i+1}), t_j) + v(f(s_{i+1}), s_{i+1}) - v(f(s_i), s_i), \end{aligned}$$

where the inequality follows from 2-cycle monotonicity. By continuity of v in t we get:

$$l_u(s_i, s_{i+1}) \geq v(f(s_i), t) - v(f(s_{i+1}), t) + v(f(s_{i+1}), s_{i+1}) - v(f(s_i), s_i).$$

³ $\text{cl}(X)$ denotes the topological closure of a set $X \subseteq \mathbb{R}^d$.

If we sum up all inequalities, we have:

$$\sum_{i=1}^k l_u(s_i, s_{i+1}) \geq \sum_{i=1}^k v(f(s_i), t) - v(f(s_{i+1}), t) + v(f(s_{i+1}), s_{i+1}) - v(f(s_i), s_i) = 0.$$

Invoking Theorem 1 completes the proof. □

Now we simplify Theorem 3 in case of allocation rules f with a finite range, which yields a generalization of the result by Saks and Yu (2005) and by Archer and Kleinberg for environments with linear valuation functions. The theorem simplifies identifying the implementability of an allocation rule f to verifying whether f is implementable on any one dimensional subset of T .

Theorem 5 *Let (T, A, v) be an environment such that $T \subseteq \mathbb{R}^d$ is convex and $v(a, \cdot) : T \rightarrow \mathbb{R}$ is continuous in t for all $a \in A$. An allocation rule $f : T \rightarrow A$ with finite range is implementable if and only if it is line implementable.*

Proof (\Leftarrow) As the range of f is finite and v continuous, it follows from Heydenreich et al. (2009) that f satisfies revenue equivalence on lines. According to Theorem 4 it is sufficient to show that f is locally implementable.

Fix $t \in T$. For all $a \in f(T)$ let $\varepsilon_a(t) := \inf_{x \in D_a} \|x - t\|_2$.⁴ Then,

$$t \in D_a \Leftrightarrow \varepsilon_a(t) = 0.$$

We show the existence of a neighborhood $U(t)$ around t such that $t \in D_a$ for all $a \in f(U(t))$. Set $A(t) := \{a \in f(T) : \varepsilon_a(t) = 0\}$. As $t \in D_{f(t)}$, we have that $A(t) \neq \emptyset$ and $t \in \bigcap_{a \in A(t)} D_a$. If $A(t) = f(T)$ we let $U(t) = \mathbb{R}^d$. Otherwise let

$$\varepsilon = \min\{\varepsilon_a(t) : a \in f(T) \setminus A(t)\}.$$

Note that $\varepsilon > 0$. Define $U(t) = \{x \in \mathbb{R}^d : \|x - t\|_2 < \varepsilon\}$.

Since line implementability implies 2-cycle monotonicity, we can invoke Lemma 2 to prove that f is implementable on $U(t)$. In other words, f is locally implementable.

(\Rightarrow) is obvious. □

Note that in the above theorem we cannot replace line-implementability by the weaker condition 2-cycle monotonicity, despite the fact that 2-cycle monotonicity is all we need to apply Lemma 2. This follows from Example 1.

4 Example

In this section we illustrate by example how our results can be used to identify a large class of allocation rules on an environment with a convex type set but non-convex domain for which 2-cycle monotonicity is sufficient for implementability.

⁴ See Lemma 2 for the definition of D_a .

The example is based on [Vohra \(2011, Example 4, p. 48\)](#). Vohra provides in this example a non-convex domain, in which each deterministic and 2-cycle monotone allocation rule is implementable. We extend this result by providing a class of randomized allocation rules with the same property. We model his setting as an environment with convex type set and convex valuation functions. Applying [Theorem 5](#) yields that line-implementability is sufficient for implementability. Proving the increasing difference property of the valuation functions gives us line-implementability from 2-cycle monotonicity (see [Remark 2](#)).

We consider a set of outcomes $A = \{a, b, ab\}$ and the set of all lotteries over outcomes in A , that is $Z(A) = \{(p_a, p_b, p_{ab}) : p_a + p_b + p_{ab} = 1, p_a, p_b, p_{ab} \geq 0\}$. The type set is $T = [0, 1]^2$ and the valuations for a type $(t_1, t_2) \in T$ are given by

$$\begin{aligned} v(a, (t_1, t_2)) &= t_1 \\ v(b, (t_1, t_2)) &= t_2 \\ v(ab, (t_1, t_2)) &= \max\{t_1, t_2\}. \end{aligned}$$

These are linearly extended to outcomes in $Z(A)$. The domain arising from this environment (as a subset of \mathbb{R}^3) is not convex. Moreover, its projection onto the hyperplane $\{x \in \mathbb{R}^3 \mid t_a + t_b + t_{ab} = 1\}$ is also not convex. Therefore, by a result of ([Ashlagi et al. 2010](#)), there are randomized allocation rules on this domain which are 2-cycle monotone but not implementable.

However, as [Proposition 1](#) below shows, there is a large class of randomized allocation rules for which 2-cycle monotonicity implies implementability.

Proposition 1 *Let T, A and $v : T \times Z(A) \rightarrow \mathbb{R}$ be as above. Let $f : T \rightarrow Z(A)$ be a 2-cycle monotone allocation rule with finite range such that for any $(p_a, p_b, p_{ab}), (r_a, r_b, r_{ab}) \in f(T)$ we have that*

$$(p_a - r_a) \cdot (p_b - r_b) < 0.$$

Then f is implementable.

Proof According to [Theorem 5](#) it is sufficient to show that any 2-cycle monotone $f : T \rightarrow Z(A)$ with finite range that satisfies the above condition is line-implementable. In order to show line-implementability, it is sufficient to show that the environment (T, A, v) satisfies the increasing differences property (see [Remark 2](#)). In our setting, this property holds, if for all $s, t \in T, x \in L_{s,t}$ and $z_p = (p_a, p_b, p_{ab}), z_r = (r_a, r_b, r_{ab}) \in Z(A)$, we have that

$$v(z_r, t) - v(z_p, t) \geq v(z_r, x) - v(z_p, x)$$

implies that

$$v(z_r, x) - v(z_p, x) \geq v(z_r, s) - v(z_p, s).$$

For explanatory purpose we show that the increasing difference property holds for the case when $s = (\alpha, 1)$ and $t = (1, \beta)$, where $\alpha, \beta \in [0, 1]$. The proof for the other cases is similar.

We consider the function $v(z_r, \cdot) - v(z_p, \cdot)$ on the line segment $L_{s,t}$, given by the parametrization $g : [0, 1] \rightarrow T, g(\lambda) = s + \lambda(t - s)$. This function is piecewise linear with one breakpoint.

Moreover,

$$v(z_r, g(0)) - v(z_p, g(0)) = (1 - \alpha)(p_a - r_a),$$

and

$$v(z_r, g(1)) - v(z_p, g(1)) = (1 - \beta)(p_b - r_b).$$

Therefore the condition on the outcomes in the theorem ensures that $v(z_r, g(\lambda)) - v(z_p, g(\lambda))$ as a function of λ is strictly monotone, and from this the increasing difference property follows immediately. \square

5 Conclusions

In this paper we provide characterization of implementable allocation rules for general environments with convex type spaces. Our main theorem implies that, for any environment where revenue equivalence on lines holds, allocation rules are implementable if and only if they are implementable on any two-dimensional convex subset of the type set. For finite sets of outcomes and continuous valuations, they are implementable if and only if they are implementable on every one-dimensional subset of the type set. The latter provides a natural generalization of a theorem by [Saks and Yu \(2005\)](#). Our proofs extend the linear programming approach to mechanism design ([Gui et al. 2004](#); [Vohra 2011](#)) from models with linear valuation functions to arbitrary continuous valuation functions. This provides a deeper understanding of the role of 2-cycle monotonicity and local implementation. It remains a challenging task to develop further techniques that enable us to verify line-implementability of allocation rules. If the increasing differences property holds, it is sufficient to verify 2-cycle monotonicity (see Sect. 4). This property does not hold in general for convex valuation function, even if 2-cycle monotonicity is satisfied (see Example 1).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

Archer A, Kleinberg R (2014) Truthful germs are contagious: a local-to-global characterization of truthfulness. *Games Econ Behav* 38:340–366

- Ashlagi I, Braverman M, Hassidim A, Monderer D (2010) Monotonicity and implementability. *Econometrica* 78(5):1749–1772
- Berger A, Müller R, Naeemi SH (2009) Characterizing incentive compatibility for convex valuations. In: Mavronicolas M, Papadopoulou VG (eds) *Algorithmic Game Theory*, Lecture Note in Computer Science, vol 5814. Springer, Heidelberg, pp 24–35
- Bikhchandani S, Chatterjee S, Lavi R, Mu'alem A, Nisan N, Sen A (2006) Weak monotonicity characterizes deterministic dominant strategy implementation. *Econometrica* 74(4):1109–1132
- Carbajal J, Ely J (2013) Mechanism design without revenue equivalence. *J Econ Theory* 148(1):104–133
- Carbajal J, Müller R (2015) Implementability under monotonic transformations in differences. *J Econ Theory* 116:114–131
- Carroll G (2012) When are local incentive constraints sufficient? *Econometrica* 80(2):661–686
- Chung KS, Olszewski W (2007) A non-differentiable approach to revenue equivalence. *Theor Econ* 2:469–487
- Gui H, Müller R, Vohra R (2004) Dominant strategy mechanisms with multidimensional types. Discussion Paper 1392, The Center for Mathematical Studies in Economics & Management Sciences, Northwestern University, Evanston
- Heydenreich B, Müller R, Uetz M, Vohra R (2009) Characterization of revenue equivalence. *Econometrica* 77:307–316
- Jehiel P, Moldovanu B (2001) Efficient design with interdependent valuations. *Econometrica* 69(5):1237–1259
- Jehiel P, Moldovanu B, Stacchetti E (1999) Multidimensional mechanism design for auctions with externalities. *J Econ Theory* 85(2):258–293
- Krishna V, Maenner E (2001) Convex potentials with an application to mechanism design. *Econometrica* 69(4):1113–1119
- Milgrom P, Segal I (2002) Envelope theorems for arbitrary choice sets. *Econometrica* 70(2):583–601
- Mirrlees J (1971) An exploration in the theory of optimal income taxation. *Rev Econ Stud* 38:175–208
- Mishra D, Pramanik A, Roy S (2013) Multidimensional mechanism design in single peaked type spaces. Working paper, Indian Statistical Institute
- Mishra D, Pramanik A, Roy S (2015) Local incentive compatibility with transfers. Working paper, Indian Statistical Institute
- Müller R, Perea A, Wolf S (2007) Weak monotonicity and Bayes-Nash incentive compatibility. *Games Econ Behav* 61(2):344–358
- Munkres JR (2000) *Topology*, 2nd edn. Prentice Hall, Englewood Cliffs
- Rochet JC (1987) A necessary and sufficient condition for rationalizability in a quasi-linear context. *J Math Econ* 16(2):191–200
- Saks M, Yu L (2005) Weak monotonicity suffices for truthfulness on convex domains. In: Proc. 6th ACM conference on Electronic Commerce, pp 286–293
- Vohra R (2011) Mechanism design, A linear programming approach, *Econometric Society Monographs*, vol 47. Cambridge University Press, Cambridge