

Characterizing implementable allocation rules in multi-dimensional environments

Citation for published version (APA):

Berger, A., Müller, R. J., & Naeemi, S. H. (2014). *Characterizing implementable allocation rules in multi-dimensional environments*. Maastricht University, Graduate School of Business and Economics. GSBE Research Memoranda No. 021 <https://doi.org/10.26481/umagsb.2014021>

Document status and date:

Published: 01/01/2014

DOI:

[10.26481/umagsb.2014021](https://doi.org/10.26481/umagsb.2014021)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

André Berger, Rudolf Müller,
Seyed Hossein Naeemi

**Characterizing implementable
allocation rules in multi-
dimensional environments**

RM/14/021

GSBE

Maastricht University School of Business and Economics
Graduate School of Business and Economics

PO Box 616
NL- 6200 MD Maastricht
The Netherlands

Characterizing implementable allocation rules in multi-dimensional environments^{*}

André Berger, Rudolf Müller^{**}, and Seyed Hossein Naeemi ^{***}

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

Email:{a.berger,r.muller}@maastrichtuniversity.nl

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 continuous valuation function that maps an agent's type to his value for this outcome. Sets of types are assumed to be convex. Our main characterization theorem implies 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, they are implementable if and only if they are implementable on every one-dimensional subset of the type set.

Our results complement and extend significantly a characterization result by Saks and Yu (2005), as well as follow-up results thereof. Contrary to our model, this stream of literature identifies types with valuation vectors. In such models, convexity of the set of valuation vectors allows for a similar characterization as ours. Furthermore, implementability on one-dimensional subsets of valuation vectors is equivalent to monotonicity. We show by example that the latter does not hold anymore in our more general setting.

Our proofs demonstrate that the linear programming approach to mechanism design, pioneered in Gui et al (2004) and Vohra (2011), can be extended from models with linear valuation functions to arbitrary continuous valuation functions. This provides a deeper understanding of the role of monotonicity and local implementation. In particular, we provide a new, simple proof of the Saks and Yu theorem, and generalizations thereof.

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.

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

^{**} R. Müller and S.H. Naeemi acknowledge support by Hausdorff Center of Mathematics, Bonn, Germany.

^{***} S.H. Naeemi is supported by GSBE, the Graduate School of Business and Economics at Maastricht University.

Keywords: Mechanism design, multi-dimensional types, incentive compatible, monotonicity

JEL codes: D 82

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 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 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 :

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

Equivalent to using a revelation mechanism, the principle could set a price menu for A satisfying (1) and let the agent choose an action that maximizes his utility¹. Note that participation constraints, requiring that an agent gets at least as much utility from truthful reports as under non-participation, can be captured by (1) by adding some (artificial) type t_0 with $v(a, t_0)$ being equal to the value of the outside option for all $a \in A$, and setting $p(t_0) = 0$. We call the triple (T, A, v) an *environment*.

Such environments occur as building block in numerous applications in the theory of incentives. In particular they have been studied in the context of mechanism design. Though mechanism design deals typically with more than one agent, an environment as defined above can be used to grasp the perspective of each individual agent. For example, if we are interested in dominant strategy implementable allocation rules f , we get an environment for each select 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.

¹ The taxation principle which is readily derived from (1), forces equal prices for types that yield the same allocation.

Central questions in many applications 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. Having such insights at hand is, for example, instrumental to solve optimal mechanism design problems. Given a distribution on T , optimal mechanism design tries to find among all implementable f one that minimizes expected payments to agents (or, equivalently maximizes expected payments to the principal). A characterization of all implementable f allows for a compact encoding of all feasible solutions of this optimization problem, and an explicit representation of p allows to formulate the objective expected revenue as a function of an encoding of f . In the best case this yields an optimization problem with a closed form solution, with Myerson’s optimal single-item auction being the most prominent example (Myerson, 1981).

Some answers to the three questions above are available without making any assumptions on the environment. As for (1), Rochet’s *cyclic monotonicity* condition provides a complete characterization of implementable allocation rules (Rochet, 1987). Gui et al (2004) provide a network interpretation of cyclic monotonicity by defining so-called *type* or *allocation graphs*. A type graph has a node for each $t \in T$ and directed edges in both directions between any two types with a particular length that reflects benefits of false type reports. Cyclic monotonicity tells that these graphs must not have any cycle of negative length. As for (2), Rochet’s proof is constructive in the sense that it provides a recipe how to compute p , which, in network terms, translates into computing shortest paths from some source node to any other node in terms of the edge length. As for (3), Heydenreich et al (2009) show that payments are unique up to a constant if and only if shortest path lengths are anti-symmetric, that is, for any two types s, t in the network the length of the shortest path from s to t equals minus the length of the shortest path from t to s . Kos and Messner (2012) generalize this insight by showing that generally payments form – up to normalization – a lattice in which shortest path lengths from and to a particular type form minimal and maximal points.

Having environments with additional structure allows for more compact, or local, characterizations of implementable allocation rules. For example, if T is a subset of \mathbb{R} , $A = [0, 1]$, and $v(a, t) = at$ – e.g. t could be the value of an indivisible good and a the probability of getting the good – then cyclical monotonicity holds if and only if f is weakly increasing. In terms of the type graph, this is the case if and only if all cycles consisting of 2 nodes are non-negative. The latter property is called (*weak*) *monotonicity* or *2-cycle-monotonicity*. The special role of 2-cycles has triggered quite some results on environments where 2-cycle monotonicity is sufficient for monotonicity (Archer and Kleinberg, 2014; Ashlagi et al, 2010; Bikhchandani et al, 2006; Mishra et al, 2013; Saks and Yu, 2005). These papers have in common that they use a different representation of an environment, which we shall call a *domain representation*, as opposed to a *parameter representation* as given by our model. A domain associates every type with a function, mapping

outcomes to values, as explained next. Both representations are closely linked, but since convex sets of types in parameter representations do not necessarily induce convex domains, characterization results on domains do not necessarily hold for parameter representations.

An environment (T, A, v) induces for each $t \in T$ a function $v(\cdot, t) : A \rightarrow \mathbb{R}$, or, equivalently, a valuation vector $x \in \mathbb{R}^A$, given by $x_a = v(a, t)$ for all $a \in A$ ². Given (T, A, v) , we will denote by ϕ the projection $\phi : T \rightarrow \mathbb{R}^A$ given by $\phi(t)_a = v(a, t)$ for all $a \in A$. We call $D = \phi(T)$ the *domain* corresponding to the environment (T, A, v) . Saks and Yu have shown that for finite A and convex domains D 2-cycle monotonicity implies cycle-monotonicity. Ashlagi et al (2010) strengthened this result in the following way. They consider randomized allocation rules that map valuation vectors to (sub-)probability distributions over outcomes, that is they consider functions $f : T \rightarrow Z(A)$ and $f : T \rightarrow \bar{Z}(A)$, where $Z(A) = \{z \in \mathbb{R}^A \mid z_a \geq 0, \sum_{a \in A} z_a = 1\}$ and $\bar{Z}(A) = \{z \in \mathbb{R}^A \mid z_a \geq 0, \sum_{a \in A} z_a \leq 1\}$, respectively. Agents are assumed to be risk neutral with the valuation for an allocation $z \in Z(A)$ being defined as $v(z, x) = \sum_{a \in A} z_a x_a$. Given a finite A , they call a domain a *monotonicity domain* if every monotone randomized allocation rule with finite range is implementable, and a *proper monotonicity domain* if every monotone allocation rule $f : T \rightarrow \bar{Z}(A)$ with finite range is implementable. Their main result is to show that a domain D is a proper monotonicity domain if and only if the closure of D is convex, and a monotonicity domain if and only if its projection on the hyperplane $\{x \in \mathbb{R}^A \mid \sum_{a \in A} x_a = 1\}$ is a proper monotonicity domain. The finite range assumption is thereby crucial. Archer and Kleinberg (2014) construct for any $k \geq 3$ a domain for which there exists an allocation rule f with infinite range that is k -cycle monotone but not $(k + 1)$ -cycle monotone.

While the domain model is easy to work with, it is for many applications neither natural nor practical. Often T might be finite-dimensional, while A may be infinite, in which case the previous literature is silent. Even for finite A , a parametrization by types may allow for a low-dimensional compact representation of private information, contrary to valuation vectors. 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 $2^m - 1$. In this example the projection ϕ would still be linear, which helps to lift results from the domain model. However, in general, valuation functions may be non-linear functions in types, in which case the projection $\phi(T)$ of a convex T usually yields a non-convex domain representation. Non-linear valuation functions appear in particular in behavioral models of utility, like for example in Köszegi and Rabin's theory of preferences that depend on endogenous reference points (Köszegi and Rabin, 2006), or Gul's model of disappointment aversion (Gul, 1991). Carbajal and Ely (2013) discuss principal agent problems with non-contractible actions that fall into this category as well.

Another example of a non-convex domain is given by Vohra (2011, Example 4, p. 59). In this example there are two goods a and b , and the agent can be

² We assume here that $|A| < \infty$ for ease of presentation, see Section 2 for more details.

allocated either good a , good b , or both (that is $A = \{a, b, ab\}$). The agent's type is a non-negative vector $t \in \mathbb{R}^3$, where t_a and t_b are the valuations that the agent has for receiving either good a or good b , and $t_{ab} = \max\{t_a, t_b\}$, i.e. the agent has no additional value when receiving both goods. Vohra shows that every deterministic monotone allocation rule on this type space is implementable. However, neither this type space nor its projection on the hyperplane $\{x \in \mathbb{R}^3 \mid t_a + t_b + t_{ab} = 1\}$ are convex, and therefore, by the result of Ashlagi et al (2010), there are monotone allocation rules with finite range that map to $Z(A)$ which are not implementable. This leads us to the question whether there are stronger, but still practical conditions that can be used instead of 2-cycle monotonicity to characterize implementability. In the remainder of the paper we give a comprehensive answer to this question under the following, very mild assumptions on the environment. We will assume that T is a convex subset of \mathbb{R}^d for some d . Varying, but very mild assumptions on v are required, as for example that $v(a, \cdot)$ is continuous on T . For such settings, we show that local implementability implies implementability, and that, for finite A and continuous valuation functions, implementability on lines is sufficient for implementability. The example by Vohra can be translated into our setting of environments with a convex type space, and we can use our results to show that for allocations that map into a particular subset of $Z(A)$ monotonicity is still sufficient for implementability.

Additional structure allows as well to provide 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 this follows from the envelope theorem. Krishna and Maenner (2001) prove a similar representation to hold for convex valuation functions. Mirrlees' representation can be used for characterization purposes as well. For example, Jehiel et al (1999) and Jehiel and Moldovanu (2001) apply this to auction environments with linear valuation functions. They show that existence and path-independence of these integrals provides necessary and sufficient conditions for implementation. This approach has been simplified by Archer and Kleinberg (2014) who show that for convex type set T and linear valuation functions path-integrals on line-segments and path-independence along the border of triangles is sufficient for implementation. Furthermore, they show that any rule that is implementable in some small enough neighborhood around each type t , is implementable on all of T . Based on this they provide an alternative proof for Saks and Yu's theorem, and at the same time generalize it to environments with convex T and linear valuation functions. Berger et al (2009) generalize this result to convex valuation functions and monotone allocation rules which satisfy the additional property *decomposition monotonicity*. In the following we significantly extend the approach by Archer and Kleinberg (2014) and Berger et al (2009). To do so, we disentangle the characterization question from the Mirrlees representation. Contrary to Carbajal and Ely (2013), who show how for settings without revenue equivalence a weaker form of Mirrlees representation can be achieved

by integration of correspondences, we choose a network approach that is purely based on measuring distances along line segments in type graphs. When allocation rules satisfy revenue equivalence, such distances provide unique payments up to a constant.

Carroll (2012) has investigated the role of local implementability as well, and shows in which cases local subsets of incentive constraints (1) are sufficient for implementability. While he derives characterizations in terms of local properties of an allocation rule *and* a payment rule, ours, 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, his results are more general as they cover ordinal as well as polyhedral type spaces.

Organization & Results. First, in Section 2, we define our setting and introduce necessary notation. In that section we also describe the above described relationship between domain and parameter representations in more detail. We present our main characterization of implementability (Theorem 4) in Section 3.1. Then we provide extensions of the results of Archer and Kleinberg (2014) about local implementability (Theorem 5, Section 3.2) and of Saks and Yu (2005) for finite outcome space (Theorem 6, Section 3.3). In Section 4 we apply our results to an example given in Vohra (2011).

2 Incentive Compatibility, Cyclic Monotonicity and 2-Cycle Monotonicity

In this section we provide precise definitions and recall the network approach for our basic model. Next we establish the relation between parameter representations and domain representations. In the first part, we need to make no assumptions on the environment, the second part applies to finite A .

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 $s, t \in T$:

$$v(f(s), s) + p(s) \geq v(f(t), s) + p(t). \quad (2)$$

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 *cyclical 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

$$\pi(t) \leq \pi(s) + l_u(s, t).$$

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*. 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 $p(t) := \text{dist}_u(s, t)$. This motivates Rochet's definition of cyclical monotonicity and yields his characterization of implementability.

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

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

³ 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.

Theorem 1 (Rochet (1987)). *An allocation rule $f : T \rightarrow A$ is implementable if and only if it is cyclical 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 $s, t \in T$:*

$$\text{dist}_u(s, t) + \text{dist}_u(t, s) \geq 0. \quad (3)$$

In this case, every payment p satisfies:

$$-\text{dist}_u(t, s) \leq v(f(t), t) + p(t) - v(f(s), s) - p(s) \leq \text{dist}_u(s, t). \quad (4)$$

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 $s, t \in T_f$:*

$$\text{dist}_u(s, t) + \text{dist}_u(t, s) = 0. \quad (5)$$

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 $s, t \in T$:*

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

In this case, every payment p satisfies:

$$\text{dist}_u(s, t) = v(f(t), t) + p(t) - v(f(s), s) - p(s). \quad (7)$$

As we have mentioned in the introduction, our intention is to derive for environments similar characterizations of implementable allocation rules as for domain representations, to the extent this is possible. In order to connect more easily to the previous literature on domain representations, we assume for the remainder of this section that A is finite, though part of the results hold as well for infinite A .

We define a projection $\phi : T \rightarrow \mathbb{R}^A$ and a domain $D \subseteq \mathbb{R}^A$ as follows. For a type $t \in T$ we define $\phi(t)_a = v(a, t)$ for all $a \in A$ and let $D = \phi(T)$. In the following we assume that ϕ is one-to-one. This assumption means that there are no two types in T which have the same valuation for each outcome. This is a reasonable assumption since two such types would be indistinguishable from the viewpoint of the principal⁴. Let $Z(A) = \{z \in \mathbb{R}^A | z_a \geq 0, \sum_{a \in A} z_a = 1\}$

⁴ At the cost of some technical care results can be extended to the case where ϕ is not one-to-one. The issue to be dealt with is that allocation rules on T may not have a unique counterpart on $\phi(T)$.

and $\bar{Z}(A) = \{z \in \mathbb{R}^A \mid z_a \geq 0, \sum_{a \in A} z_a \leq 1\}$. Note that valuations on A can be straightforwardly extended to $\bar{Z}(A)$ by setting $v(z, t) = \sum_{a \in A} z_a v(a, t)$. Again, in order to connect more easily to previous literature, we let allocation rules map from T into $Z(A)$ or $\bar{Z}(A)$. An allocation rule is said to have *finite range* if $f(T)$ is finite.

For any allocation rule $f : T \rightarrow \bar{Z}(A)$ we can identify a corresponding allocation rule $g : D \rightarrow \bar{Z}(A)$ by letting $g(x) := f(\phi^{-1}(x))$ for $x \in D$.

Lemma 1. *Let (T, A, v) be an environment with finite A . Let $f : T \rightarrow \bar{Z}(A)$ be an allocation rule and $g : \phi(T) \rightarrow \bar{Z}(A)$ be the corresponding allocation rule on the domain induced by (T, A, v) . Then f is cyclical monotone (2-cycle monotone) if and only if g is cyclical monotone (2-cycle monotone).*

Proof. We show that the u -length between any pair of types is the same for f and for g . For this purpose let $s, t \in T$ and let $z^s = f(s)$ and $z^t = f(t)$. Then

$$\begin{aligned} l_u(s, t) &= v(f(t), t) - v(f(t), s) = \sum_{a \in A} z_a^t v(a, t) - \sum_{a \in A} z_a^t v(a, s) \\ &= \sum_{a \in A} z_a^t \phi(t)_a - \sum_{a \in A} z_a^t \phi(s)_a = v(g(\phi(t)), \phi(t)) - v(g(\phi(t)), \phi(s)) = l_u(\phi(s), \phi(t)). \end{aligned}$$

□

Lemma 1 allows to apply the following results by Ashlagi et al (2010).

Theorem 3 (Ashlagi et al (2010)). *Let $D \subseteq \mathbb{R}^A$ and $|A| \geq 2$. Then every 2-cycle monotone allocation rule $f : D \rightarrow \bar{Z}(A)$ with finite range is implementable if and only if the closure of D is convex.*

Moreover, every 2-cycle monotone allocation rule $f : D \rightarrow Z(A)$ with finite range is implementable if and only if the projection of D onto the hyperplane $\{x \in \mathbb{R}^A \mid \sum_{a \in A} x_a = 1\}$ is convex.

From the above theorem and Lemma 1 we immediately conclude the following corollary for general environments.

Corollary 3. *Let (T, A, v) be an environment where A is finite, $|A| \geq 2$, and let $D = \phi(T)$ be defined as above. Moreover, assume that ϕ is one-to-one.*

Then every 2-cycle monotone allocation rule $f : T \rightarrow \bar{Z}(A)$ with finite range is implementable if and only if the closure of D is convex.

Moreover, every 2-cycle monotone allocation rule $f : T \rightarrow Z(A)$ with finite range is implementable if and only if the projection of D onto the hyperplane $\{x \in \mathbb{R}^A \mid \sum_{a \in A} x_a = 1\}$ is convex.

Turning back to environments (T, A, v) and allocation rules $f : T \rightarrow A$, Corollary 3 has the following consequences. If the closure of $\phi(T)$ is convex, any 2-cycle monotone $f : T \rightarrow A$ with finite range is implementable. To see this, apply Corollary 3 with A being $f(T)$, noting that $|f(T)| = 1$ is trivial. If the closure of $\phi(T)$ is not convex and $|A| \geq 2$, there exists a 2-cycle monotone

$f : T \rightarrow \overline{Z}(A)$ with finite range that is not implementable. If the closure of the projection of $\phi(T)$ (as described in 3) is not convex, there even exists a monotone $f : T \rightarrow Z(A)$ with finite range that is not implementable. Even for convex $T \subset \mathbb{R}^d$ neither of the two conditions on $\phi(T)$ might hold if the projection ϕ is not a linear mapping. This raises the question of a suitable replacement of the monotonicity condition, which we will answer in the remainder of the paper.

3 Simplifying Rochet's characterization

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 restrict ourselves to allocation rules that satisfy revenue equivalence. We explain at the end of this section, how this assumption can be omitted and thereby relate our results to Carbajal and Ely (2013). Throughout, we assume that T is a convex subset of \mathbb{R}^d for some $d \geq 0$. The main result is that every allocation rule that is locally implementable is globally implementable. This allows us to prove that any allocation rule that is implementable on every line segment and has finite range is globally implementable. We start by introducing the notion of line-implementability and showing how it can be used to characterize implementable rules. Then we present our local implementability result. We conclude with a treatment of finite outcome spaces.

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 $s, t \in T$ it is implementable on $L_{s,t}$.*

Obviously, every implementable allocation rule is line-implementable. Furthermore every line-implementable allocation rule is monotone. It is well-known that for linear valuation functions 2-cycle monotonicity and line-implementability are equivalent. This follows in particular from Corollary 3. The same Corollary tells us that this equivalence does not hold in general for non-linear valuation functions, as the projection of the type space into the domain representation might not be convex. The following example of convex, almost everywhere linear valuation functions provides an illustration where even a deterministic allocation rule can be constructed that is 2-cycle monotone but not implementable.

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:

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

We verify 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:

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

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 and Corollary 3 tell 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 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 4. 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 $s_1, s_2, s_3 \in T$:

$$\sum_{i=1}^3 \text{dist}_u^L(s_i, s_{i+1}) = 0, \quad (8)$$

where $s_4 = s_1$ and distances in (8) 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 (7), 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 (8), 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 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 verifying in which environments line-implementability can be replaced by monotonicity in Theorem 4.

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 convex (Archer and Kleinberg, 2014; 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 4 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

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

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 4, which extends a result by Vohra (2011, Theorem 4.2.11), who has proven it for randomized allocation rules over finitely many outcomes.

Corollary 4. *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 monotonicity. 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 2. *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 :*

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

Proof. Fix $s, t \in T$ and x between s and t . Since f is implementable on $L_{s,t}$, according to Corollary 1 and Theorem 2 there exist payments p such that:

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

Using the same observation for $\text{dist}_u^L(s, x)$ and $\text{dist}_u^L(x, t)$ and a simple calculation completes the proof. \square

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 5. *Let $T \subseteq \mathbb{R}^d$ be convex and assume that every 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 4 and Lemma 2 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:

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

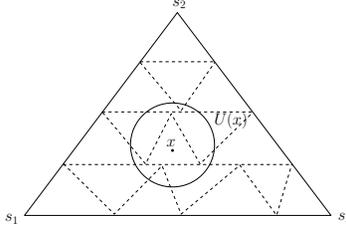


Fig. 1. Subdividing $\triangle_{s_1, s_2, s_3}$ into sufficiently small triangles

where $x_4 = x_1$.

Recall that by the Lebesgue Number Lemma for any open covering \mathcal{A} 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 A containing it⁵. This implies that there is a $\delta > 0$ such that every subset of $\triangle_{s_1, s_2, s_3}$ of diameter less than δ is contained in at least one of the neighborhoods in which f is implementable. In particular, if we subdivide $\triangle_{s_1, s_2, s_3}$ into M triangles $\triangle_{s_1^1, s_2^1, s_3^1}, \triangle_{s_1^2, s_2^2, s_3^2}, \dots, \triangle_{s_1^M, s_2^M, s_3^M}$ (see Figure1), 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

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

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 2, 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 4, f is implementable. \square

3.3 Finite Outcome Space

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 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 $a \in A$.

⁵ For more information refer to Munkres (2000) or other classic books on topology.

Lemma 3. 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

$$D_a := cl(f^{-1}(a)).$$

If $f : T \rightarrow A$ is monotone and $\bigcap_{a \in f(T)} D_a \neq \emptyset$, then f is implementable.⁶

Proof. Let $\{s_1, \dots, s_k\} \subseteq T$ for some $k \geq 3$ and $t \in \bigcap_{a \in A} D_a$. 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 monotonicity. By continuity of v in t we get:

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

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. \square

Now we simplify Theorem 4 in case of f with a finite range, which yields a generalization of the result by Saks and Yu (2005) for domain models, 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 6. Let (T, A, v) be an environment such $T \subseteq \mathbb{R}^d$ 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 5 it is sufficient to show that f is locally implementable.

Fix $t \in T$. For all $a \in f(T)$ let $\varepsilon_a(t) := \inf_{x \in D_a} \|x - t\|_2$ ⁷. Then,

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

⁶ $cl(X)$ denotes the topological closure of a set $X \subseteq \mathbb{R}^d$.

⁷ See Lemma 3 for the definition of D_a

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 monotonicity, we can invoke Lemma 3 to prove that f is implementable on $U(t)$. In other words, f is locally implementable.

(\Rightarrow) is obvious. □

Theorem 6 holds as well for allocation rules $f : T \rightarrow \overline{Z}(A)$ as we may replace A by $\overline{Z}(A)$ at first place, given that we made no assumptions on A and that the extension of a continuous valuation from A to $\overline{Z}(A)$ remains continuous. Note that we cannot replace line-implementability by the weaker condition monotonicity, despite the fact that monotonicity is all we need to apply Lemma 3. This follows from Example 1 and Corollary 3.

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 non-convex domain for which monotonicity is sufficient for implementability.

The example is based on Vohra (2011, Example 4, p. 59). Vohra provides in this example a non-convex domain, in which each deterministic and 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 a *convex* type space and convex valuation functions and then apply Theorem 6.

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 space is $T = [0, 1]^2$ and the valuations for a type $(t_1, t_2) \in T$ are given by

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

and 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 the result of Ashlagi (see Theorem 3), there are randomized allocation rules on this domain which are monotone but not implementable-

However, as Proposition 1 below shows, there is a large class of randomized allocation rules for which 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 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 6 it is sufficient to show that any 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

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

The proof involves an extensive case analysis, where the cases depend on the relative location of s and t . For explanatory purpose we show that the increasing difference property holds for the case when $s = (\alpha, 1)$ and $t = (1, \beta)$, where $\alpha, \beta \in [0, 1]$.

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 we have $v(z_r, g(\lambda)) - v(z_p, g(\lambda))$ as a function of λ is strictly monotone, and from this the increasing difference property follows immediately. □

5 Conclusions

In this paper, we have characterized implementable allocation rules in multi-dimensional environments. Our main theorem implies that, for any environment where revenue on lines holds and where the set of types is convex, 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, they are implementable if and only if they are implementable on every one-dimensional subset of the type set. For the latter, revenue on lines holds whenever valuations are continuous. 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 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 monotonicity (see Remark 2). But already for convex, but non-linear valuation functions it is not.

Bibliography

- Archer A, Kleinberg R (2014) Truthful germs are contagious: a local to global characterization of truthfulness. *Games and Economic Behavior*
- 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, 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. *Journal of Economic Theory* 148(1):104–133
- 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. *Theoretical Economics* 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, IL
- Gul F (1991) A theory of disappointment aversion. *Econometrica* 59(3):667–686
- 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. *Journal of Economic Theory* 85(2):258–293
- Kos N, Messner M (2012) Extremal incentive compatible transfers. *Journal of Economic Theory* 148(1):134–164
- Köszegi B, Rabin M (2006) A model of reference-dependent preferences. *Quarterly Journal of Economics* 121(4):1133–1165
- 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. *Review of Economic Studies* 38:175–208
- Mishra D, Pramanik A, Roy S (2013) Multidimensional mechanism design in single peaked type spaces. Working paper, Indian Statistical Institute
- Müller R, Perea A, Wolf S (2007) Weak monotonicity and Bayes-Nash incentive compatibility. *Games and Economic Behavior* 61(2):344–358
- Munkres JR (2000) *Topology*, 2nd edn. Prentice Hall

- Myerson R (1981) Optimal auction design. *Mathematics of Operations Research* 6(1):58–73
- Rochet JC (1987) A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics* 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