Multidimensional mechanism design: key results and research issues

Debasis Mishra*

Indian Statistical Institute, 7 S.J.S. Sansanwal Marg, New Delhi 110 016, India

Mechanism design is the study of designing procedures for interaction of strategic agents such that the designer's objective is achieved in an *equilibrium*. The main challenge in mechanism design is that strategic (i.e. utility maximizing) agents have private information not known to the designer. Multidimensional mechanism design deals with mechanism design when strategic agents have multidimensional information, i.e. private information consists of different components. This article briefly surveys the literature on mechanism design where monetary transfers are allowed and agents have multidimensional private information. It identifies some of the fundamental results in the literature and some interesting open research issues.

Keywords: Auctions, game theory, multidimensional mechanism design.

Introduction

IT is a well known fact that individuals tend to be strategic in interactions with other individuals. Strategic interactions are aimed at maximizing the utility of the individual. Game theory is the study of strategic interactions between individuals (agents).

Mechanism design is the *reverse engineering* aspect of game theory. Given a strategic environment with agents, game theory tries to predict the outcome of such an environment in some 'equilibrium' – stronger the notion of equilibrium better the prediction. On the other hand, mechanism design theory tries to design the environment in which strategic agents will interact. This designed environment specifies the actions that participating agents can take, and the outcome from each of these actions. Such a complete specification of the environment is termed a *mechanism*. The designer of a mechanism has particular objectives in mind, and the mechanism is designed such that when strategic agents interact in the mechanism, the *equilibrium outcome* coincides with the objective of the designer.

We give two examples to illustrate the idea further.

*e-mail: dmishra@gmail.com

1. Voting. Consider a scenario where a set of agents needs to choose a candidate from a set of candidates. Each agent has a preference (represented by a strict ordering) over the candidates. The designer wants to restrict attention to voting procedures where agents submit their preferences (need not be the true ones) and a candidate is selected – restricting attention to such voting procedures is without loss of generality, and follows from a seminal result in mechanism design called the *revelation principle*. Suppose the designer wants to design a mechanism (a voting procedure) which satisfies a mild condition called *unanimity* – if all the agents agree that a certain candidate is the best, then the mechanism must choose that candidate. Since this is a strategic environment, the designer wants to design a mechanism in which the 'equilibrium' outcome must satisfy unanimity. Suppose he imposes a strong notion of equilibrium – irrespective of the actions of other agents, each agent must submit his true preference. Such mechanisms are called *strategy-proof* mechanisms.

   Which voting procedures should the designer choose?

   Mechanism design theory answers such questions. A celebrated result in mechanism design, due to Gibbard[1] and Satterthwaite[2], asserts that the only strategy-proof voting procedure which satisfies the mild condition of unanimity is *dictatorship* if there are at least three candidates. Dictatorship voting procedure selects a *dictator* agent and always chooses his most preferred candidate.

2. Auction. Consider a scenario where a single indivisible object, say a house, is sold by a seller to a set of potential buyers. Each buyer has a valuation for the object, which is only known to him. The sale of a house typically requires payments. We assume that if a buyer has a value v and pays an amount p after winning the object, then his net utility is $v - p$ – this particular assumption on net utility is called the *quasi-linear* utility assumption.

   Now, there are many ways (mechanisms) to sell the object to the buyers. Of all the mechanisms available, let us restrict attention to those mechanisms in which each buyer reveals his value (may not be the true value), and the seller decides who to allocate the object and the amount of payment based on this information – again, restricting attention to such mechanisms is without loss of generality due to the well-known revelation principle. We would like the agents to reveal their *true value* irrespective of what other agents are revealing – this particular notion of equilibrium is called a *dominant strategy equilibrium*. Such mechanisms are called the *incentive compatible* mechanisms. One then asks: Is there an incentive compatible mechanism where the highest valued buyer gets the object?

In other words, can we design payments such that when we give the object to the highest valued buyer, the best strategy (in the sense of net utility maximization) of every buyer will be to reveal his true value. In a seminal work, Vickrey[3] showed that a second-price auction, where the highest valued buyer wins the object but pays an amount equal to the second highest value, achieves this objective.

We can ask another natural question: Of all the incentive compatible mechanisms, which mechanism maximizes the *expected* payments to the seller?

Note that since the seller is not aware of the values of the buyers, the question of maximizing payments at every possible value is asking for too much. If the seller knows the *distribution* of values of the bidders, then the question is to maximize the expected revenue from the mechanism. This is referred to as the *optimal auction design*. In his seminal work, Myerson[4] derived the optimal auction for single object auction.

The theory of mechanism design covers such topics and answers such questions. We will be mainly concerned with the auction type of topics involving monetary transfers. These have tremendous practical applications: sale of spectrum waves using auctions and sale of advertisement slots on search engines like Google and Yahoo! using auctions have become the success story of multi-dimensional mechanism design.

A model of multidimensional mechanism design

In this survey, we will focus on a general model mechanism design that covers the single object auction as a special case. In particular, we will consider a model with monetary transfers and quasi-linear utility function of agents – thus, we will exclude the voting type of models. Further, we will assume that each agent knows his own valuation information completely, and his valuation information is not known to other agents and to the mechanism designer. This is known as the *private values model* – it is plausible to think of settings where a buyer is not completely aware of his own value of the object, but we do not discuss such models in this survey.

Formally, let $N := \{1,..., n\}$ be the set of agents in our model. There is a finite set of alternatives, denoted by $A := \{a, b, c,...\}$. In the sale of an indivisible object, the alternatives are the different possible outcomes, e.g. the object is not sold to anyone, the object is sold to agent 1, the object is sold to agent 2, etc. Each agent has a valuation for each alternative. In particular, each agent $i \in N$ has a valuation vector $v_i \in \mathbb{R}^{|A|}$, which is completely known to him. Though each agent knows his own valuation vector, he does not know the valuation vectors of other agents. Similarly, the mechanism designer does not know the valuation vector of any agent.

Though a particular valuation vector v_i belongs to $\mathbb{R}^{|A|}$, not every vector in $\mathbb{R}^{|A|}$ may be feasible. Consider the following examples.

- *Choosing a public project*. Suppose a city needs to decide to build one of the three public projects: (a) a museum, (b) a stadium and (c) a school. It is plausible to assume that agents (residents of the city) have non-negative value for such projects. Hence, a valuation vector of every agent will lie in $\mathbb{R}^3_+$.
- *Sale of an indivisible object*. Suppose there are two buyers to buy a single indivisible object. Then, the set of alternatives can be denoted as a_0 (nobody gets the object), a_1 (agent 1 gets the object) and a_2 (agent 2 gets the object). It is standard to assume that an agent gets a value from an alternative only if he gets the object. In that case, the valuation vector of agent 1 will have $v_i(a_0) = v_i(a_2) = 0$ and the valuation vector of agent 2 will have $v_i(a_0) = v_i(a_1) = 0$. Thus, one sees that the set of possible valuations of an agent is a strict subset of $\mathbb{R}^3_+$.

These examples emphasize the fact that not every vector in $\mathbb{R}^{|A|}$ may be a feasible valuation vector in all settings. Indeed, the example of sale of an indivisible object is a typical example of a *one-dimensional mechanism* design setting. In one-dimensional mechanism design problems, there is exactly one alternative in A whose valuation to the agent is not known to the mechanism designer – call such an alternative the *private alternative*. The valuation for all the alternatives except the private alternative is commonly known to all the agents and the designer. In the sale of an indivisible object example, the private alternative of agent i is the alternative where agent i gets the object. Consider another example. Suppose the designer is choosing between two alternatives: whether to provide a public good (say, a children's park) or not. If the public good is not provided, then all the agents get zero valuation, and if it is provided, then each agent gets a valuation which is privately known to him. Hence, the alternative that provides the public good is the private alternative for all the agents. This illustrates that our formulation captures settings with one-dimensional private information. In usual one-dimensional mechanism design problem, we care about just one number from each agent, which is the private information of the agent. A multidimensional mechanism design problem cannot be handled by revealing just one number, and our formulation allows that.

We denote the set of possible valuation vectors of agent i as $V_i \subseteq \mathbb{R}^{|A|}$. A profile of valuation vectors will be denoted by $v := (v_1,..., v_n)$. The set of all profiles of valuation vectors will be denoted by $V = V_1 \times \cdots \times V_n$. The mechanism designer knows $V_1,..., V_n$. We note here that since the private information of every agent is a vector in $\mathbb{R}^{|A|}$, we can incorporate 'multidimensional' private information.

An *allocation rule* is a mapping $f : V \to A$. In particular, an allocation rule chooses an alternative at every profile of valuations. Notice that we do not allow for randomization. More generally, an allocation rule can pick a probability distribution over elements in A at every profile of valuations. However, we restrict attention to *deterministic* allocation rules. An alternative way to interpret an allocation rule is as follows. Suppose we write a profile of valuations as a matrix, where the rows correspond to agents and columns correspond to alternatives. Then, a row vector is a valuation vector for the agent corresponding to that row. Each profile of valuations in V corresponds to one such matrix. An allocation rule chooses a *column vector* at every such matrix. One can think of many ways to choose a column vector of a matrix – a *constant allocation rule* chooses the same column at every profile of valuations, an *efficient allocation rule* chooses the column whose column sum of values is the highest, a *weighted efficient allocation rule* chooses the column whose weighted column sum of values is the highest, where weights are assigned to each agent (row).

A *payment rule* for agent $i \in N$ is a mapping $p_i : V \to \mathbb{R}$. We will interpret the payment as the amount each agent pays to the mechanism designer. Due to quasi-linear utility function, if an alternative $a \in A$ is chosen and agent i with valuation vector v_i makes a payment of t_i, then his net utility is $v_i(a) - t_i$. Note that the payment of an agent can be positive, negative or zero. Additional requirements can be imposed depending on the context. For instance, in many situations payments should add up to be non-negative as the mechanism designer has no money to pay the bidders.

Implementation of allocation rules

As discussed earlier, a mechanism designer has certain objectives in mind and wants to design a mechanism where the equilibrium strategies of the agents (who want to maximize their net utility) achieve this objective. We will now discuss the basic elements of a mechanism. A mechanism in general may be quite complicated. Abstractly, a mechanism may involve arbitrary messages between the designer and the agents, where messages are restricted to lie in some abstract space. But eventually, the outcome of a mechanism is the selection of an alternative and a payment for each agent.

We restrict attention to a particular class of mechanisms here, called the *direct mechanisms*. The only message that is allowed in a direct mechanism is the revelation of the valuation vector (need not be the true one) by each agent. As a consequence a (direct) mechanism is a tuple (f, p), where $p \equiv (p_1,..., p_n)$ are the payment rules for the agents and f is an allocation rule. We will now require a notion of equilibrium. The equilibrium we use is a very strong one – it requires that no matter what other agents reveal, the best strategy of each agent is to reveal his true valuation vector. This is the notion of a dominant strategy equilibrium. Fix an agent $i \in N$, and suppose other agents reveal valuation vectors v_{-i} (this $(-i)$ notation will be used to denote information of agents other than agent i). If agent i has true valuation vector v_i, then his net utility for being truthful in a mechanism (f, p) is $v_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i})$, whereas if he deviates to another revelation of v'_i, his net utility becomes $v_i(f(v'_i, v_{-i})) - p_i(v'_i, v_{-i})$. Our equilibrium requires that irrespective of the revelation of other agent v_{-i}, truth telling must maximize net utility for agent i.

Definition 1. *An allocation rule f is implementable if there exists payment rules $p_1,..., p_n$ such that for every agent $i \in N$, for every $v_{-i} \in V_{-i}$, and for every $v_i, v'_i \in V_i$,*

$$v_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i}) \geq v_i(f(v'_i), v_{-i}) - p_i(v'_i, v_{-i}).$$

In this case, we say $(p_1,..., p_n)$ implement f, and the mechanism (f, p) is incentive compatible.

There are weaker notions of equilibrium, like Bayes–Nash and ex-post Nash, which we will skip here – see ref. 5 for details. The focus on direct mechanisms is without loss of generality since the equilibrium outcome of any mechanism can be replicated by an incentive-compatible directed mechanism. This is known as the revelation principle – any standard textbook in microeconomics covers this[6].

The central question of this survey is the following. What allocation rules are implementable?

Efficiency is implementable

One of the most celebrated allocation rules is the *efficient allocation rule*.

Definition 2. *An allocation rule $f : V \to A$ is an efficient allocation rule if at every $v \in V$,*

$$f(v) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a).$$

Efficiency maximizes the sum of valuations of agents over all alternatives. In the case of single-object auctions, it awards the object to the agent with the highest value. Vickrey[3], Clarke[7] and Groves[8] showed that efficiency is implementable by a payment rule, now called the Groves payment rule. A payment rule p_i of agent $i \in N$ is a Groves payment rule for an efficient allocation rule f^e if for every $v \in V$,

$$p_i(v) = h_i(v_{-i}) - \sum_{j \neq i} v_j(f^e(v)),$$

where $h_i : V_{-i} \to \mathbb{R}$ is an arbitrary function.

Theorem 1. (*Vickrey[3], Clarke[7] and Groves[8]*). *The Groves payment rules implement an efficient allocation rule.*

The proof is straightforward and boils down to checking that the incentive constraints for implementability hold. There are many choices for the h_i function in the definition of Groves payment rule. One particular choice has many interesting properties in specific contexts. For every $i \in N$ and $v_{-i} \in V_{-i}$, let $h_i(v_{-i}) = \max_{a \in A} \sum_{j \neq i} v_j(a)$. Such a choice of h_i defines the *pivotal payment rule*, and the corresponding mechanism is called the Vickrey–Clarke–Groves (VCG) mechanism. A particular instance of the VCG mechanism is the second-price auction or the Vickrey auction in the case of single-object auction. VCG mechanisms have excellent theoretical properties in the case of *combinatorial auctions*[9].

A result in unrestricted domain of valuations

There are allocation rules other than efficient allocation rules which can be implemented. For instance, consider the following class of allocation rules which generalize the efficient allocation rule.

Definition 3. *An allocation rule f is an affine maximizer allocation rule if there exists non-negative weights $\lambda_1,\ldots, \lambda_n$, not all equal to zero, and a mapping $\kappa : A \to \mathbb{R}$ such that at every $v \in V$,*

$$f(v) \in \arg\max_{a \in A} \left[\sum_{i \in N} \lambda_i v_i(a) + \kappa(a) \right].$$

Such an affine maximizer allocation rule is unresponsive to irrelevant agents (UIA) if for every $i \in N$ such that $\lambda_i = 0$ and for every $(v_i, v_{-i}), (v'_i, v_{-i}) \in V$, we have $f(v_i, v_{-i}) = f(v'_i, v_{-i})$.

The UIA condition requires that the allocation should not change if an agent whose weight is zero changes his valuation vector. A payment rule p_i of agent $i \in N$ is a *generalized Groves payment rule* for an affine maximizer allocation rule f^a satisfying UIA if for every $v \in V$, $p_i(v) = 0$ if $\lambda_i = 0$ and

$$p_i(v) = h_i(v_{-i}) - \frac{1}{\lambda_i} \left[\sum_{j \neq i} \lambda_j v_j(f^a(v)) + \kappa(f^a(v)) \right],$$

if $\lambda_i > 0$, where $h_i : V_{-i} \to \mathbb{R}$ is an arbitrary function.

Theorem 2. (*Roberts[10]*). *Every affine maximizer satisfying UIA can be implemented by the generalized Groves payment rules.*

Remarkably, under some restrictions, every implementable allocation rule is an affine maximizer. We say an allocation rule f is *onto* if for every $a \in A$, there exists a $v \in V$ such that $f(v) = a$. Roberts[10] proved the following remarkable result.

Theorem 3. (*Roberts[10]*). *Suppose $V_i = \mathbb{R}^{|A|}$ for every $i \in N$ and $|A| \geq 3$. If f is an implementable and onto allocation rule, then it is an affine maximizer allocation rule.*

If an allocation rule is implementable in a domain, then its restriction to a smaller domain is also implementable in that domain – it is a good exercise to prove this result formally. So, in smaller domains affine maximizers (satisfying UIA) are implementable, but there may be non-affine maximizers that can be implemented also. Roberts' theorem breaks down if some of the assumptions made in its statement are weakened. For instance, if there are two alternatives then there are more allocation rules than affine maximizers that can be implemented. If V_i is a strict subset of $\mathbb{R}^{|A|}$, which is the case in many practical settings like combinatorial auctions, public project selection, etc. then many allocation rules can be implemented besides the affine maximizer allocation rules. An open question is to find the counterpart of Roberts' theorem in some of these restricted domains of valuation. One of the problems about extending Roberts' theorem to other settings is that its proof is extremely involved. Researchers have tried to simplify Roberts' original proof[5,11–13]. Some progress has been done to extend Roberts' theorem to restricted domains[13,14]. Mishra and Sen[13] show that if V_i is an open multidimensional interval for every $i \in N$, then if f is implementable and satisfies *neutrality*, it is an affine maximizer with $\kappa(a) = 0$ for all $a \in A$. Neutrality roughly requires that if we consider two valuation profiles v, v' such that v' is obtained by permuting the columns of two alternatives at v, then the outcomes chosen at v' must be obtained by applying the same permutation to the outcome chosen at v. It says that the allocation rule should not discriminate between alternatives based on their *names*. Carbajal *et al.*[14] show that Roberts' theorem holds in a technical domain. Besides these two papers, there is little known about extending Roberts' theorem to restricted domains of valuations.

Implicit monotonicity characterizations

Roberts' theorem is very precise in describing the set of implementable allocation rules. However, such explicit characterization is difficult in restricted domains of valuation. For this reason, researchers have turned their attention to implicit characterizations of implementability. The results in the literature are stated in a one-agent model. In particular, we fix an agent $i \in N$ and the profile of valuations of other agents at v_{-i}, and look at the image of an allocation rule f as agent i changes his valuation over V_i. We denote this image of f as $F : V_i \to A$ and assume that a payment rule $P_i : V_i \to \mathbb{R}$, which depends only on the valuation vector of agent i (of course, putting together such P_is for all v_{-i} will give us a payment rule $p_i : V \to \mathbb{R}$). This saves us a lot of notations.

As before, an allocation rule $F : V_i \to A$ is implementable if there exists a payment rule P_i such that $v_i(F(v_i)) - P_i(v_i) \geq v_i(F(v_i')) - P_i(v_i')$ for all $v_i, v_i' \in V_i$. Now, we will like to identify simple conditions on F which are necessary and sufficient for it to be implementable. Consider the following necessary condition.

Definition 4. *An allocation rule F is K-cycle monotone, where $K \geq 2$ is a positive integer, if for every finite sequence of valuation vectors $(v_i^1, v_i^2, ..., v_i^k)$ with $k \leq K$, we have*

$$\sum_{j=1}^{k} [v_i^j (F(v_i^j)) - v_i^j (f(v_i^{j-1}))] \geq 0, \qquad (1)$$

where $v_i^0 \equiv v_i^k$. An allocation rule F is cyclically monotone if it is K-cycle monotone for all positive integers $K \geq 2$.

Note that if an allocation rule f is $(K + 1)$-cycle monotone, then it is also K-cycle monotone. The necessity of cycle monotonicity can be shown by adding the incentive constraints associated with any sequence of valuation vectors – payment terms cancel out. In a seminal work, Rochet[15] showed the following.

Theorem 4. (*Rochet*[15]). *An allocation rule is implementable if and only if it is cyclically monotone.*

This result is related to the characterization of subgradients of a convex function due to Rockafellar[16] – the allocation rule serves as a subgradient of the convex net utility function of agent i. The explicit graph theoretic interpretation is due to Gui *et al.*[17], where they associate a *graph* with every domain of valuation vectors V_i, every set of alternatives A, and every allocation rule $F : V_i \to A$. This type graph contains the set of valuation vectors as the set of nodes, and is a complete graph (i.e. a directed edge exists from every node to every other node). The length of the edge from node v_i to v_i' is

$$\ell(v_i, v_i') := v_i(F(v_i)) - v_i(F(v_i')).$$

Then, it is easy to notice that inequality (1) is requiring the length of the cycle $(v_i^1, ..., v_i^k, v_i^1)$ to be non-negative.

Rochet's characterization holds for various extensions of our model – for instance, with allocation rules which randomize, when the set of alternatives is not finite, etc. (see refs 5, 18–20). Though mathematically elegant, this characterization of implementability involves verifying the length of cycles involving arbitrary number of nodes. When the set of alternatives is finite, as is assumed here, one only needs to verify cycles involving no more than $|\mathcal{A}|$ nodes. The following result is a due to Mishra and Roy[21].

Theorem 5. (*Mishra and Roy*[21]). *An allocation rule f is implementable if and only if it is $|A|$-cycle monotone.*

Even with this result, the cycle monotonicity characterization looks opaque. In a series of papers, starting from Bikhchandani *et al.*[22], followed by Saks and Yu[23] and Ashlagi *et al.*[24], researchers have shown that if the closure of a domain of valuation vectors is convex, then two-cycle monotonicity is sufficient for implementation. Since two-cycle monotonicity is an easy condition to interpret, the relation of implementability and cycle monotonicity is clearer in such convex domains.

Theorem 6. (*Bikhchandani et al.*[22], *Saks and Yu*[23] *and Ashlagi et al.*[24]). *Suppose closure of $V_i \subseteq \mathbb{R}^{|A|}$ is convex. Then, $F : V_i \to A$ is implementable if and only if it is two-cycle monotone.*

Two-cycle monotonicity reduces to a simple condition in the case of single-object auction (a convex domain). It says that if an agent is winning the object at a valuation, then he should continue to get the object at a larger valuation. This monotonicity condition was first discovered to be equivalent to implementability in the single-object auction setting by Myerson[4]. Two-cycle monotonicity generalizes this condition to arbitrary convex domains of valuation.

While many interesting domains in the literature are convex, many are not. One of the open questions in the literature is to find counterparts of the above theorem in non-convex domains. A recent result in Mishra and Roy[21] finds such an extension. They study *dichotomous domains*. In dichotomous domains, an agent has two pieces of private information: (1) a set of alternatives he finds permissible, and (2) a valuation in $\mathbb{R}_{++}$ for the permissible alternatives. Note that for non-permissible alternatives, the valuation is always zero. Such dichotomous domains are non-convex, and the earlier result does not apply. However, for a large class of dichotomous domains, termed *rich dichotomous domains*, Mishra and Roy[21] showed that three-cycle monotonicity is necessary and sufficient for implementation.

There are many interesting domains which are non-convex. For instance, consider the single-peaked domain studied extensively in voting theory[25]. In single-peaked domain, there is a pre-specified ordering over the set of alternatives, and the valuations of agents must respect this ordering in the sense that as we go away from the top alternative of the agent, the values must decrease. It is easy to show that this is a non-convex domain. It is an open question to characterize implementability using K-cycle monotonicity in such domains.

There are other notions of implicit characterizations studied in the literature. For instance, Carroll[26] showed that a particular form of *local incentive compatibility* condition is necessary and sufficient for implementability in convex domains of valuation.

Revenue equivalence

We will stay in the single agent framework of the last section. As we saw, the incentive compatibility constraints have a graph theoretic interpretation – we saw that cycle monotonicity is equivalent to requiring no negative cycles in the associated *type graph*. We can define shortest path lengths between any pair of valuation vectors in the type graph. For any $v_i, v'_i \in V_i$, let $\text{dist}(v_i, v'_i)$ be the length of the shortest path from v_i to v'_i, where a path is any sequence of distinct edges connecting v_i to v'_i and the length of a path is the sum of lengths of edges in the path. Gui *et al.*[17] showed that if F is implementable, then the following payment rule implements F. Fix a $v_i \in V_i$ and let $P_i(v_i) = 0$ and for every $v'_i \in V_i$ and $v'_i \neq v_i$, let $P_i(v'_i) = \text{dist}(v_i, v'_i)$. It is easily verified that if P_i is a payment rule that implements F, and if we let for all $v'_i \in V_i$, $P'_i(v'_i) = P_i(v'_i) + \alpha$ for some $\alpha \in \mathbb{R}$, then P'_i also implements F. A well-known result in multidimensional mechanism design states that by choosing appropriate α, we can generate *all* payment rules that implement F. This is known as the revenue equivalence result.

Definition 5. *An implementable allocation rule F satisfies revenue equivalence if for every pair of payment rules P_i, P'_i that implement F, there is some $\alpha \in \mathbb{R}$ such that for all $v'_i \in V_i$ we have*

$$P'_i(v'_i) = P_i(v_i) + \alpha.$$

Revenue equivalence is an extremely powerful property since it characterizes the payments that implement an allocation rule. It was first discovered by Myerson[4] in the single-object auction setting. Later extensions in the multi-dimensional settings were found in literature[27–30].

Theorem 7. (Heydenreich *et al.*[30]) *An allocation rule $F : V_i \to A$ satisfies revenue equivalence if and only if for every $v_i, v'_i \in V_i$,* $\text{dist}(v_i, v'_i) + \text{dist}(v'_i, v_i) = 0$.

This theorem, like Rochet's theorem, holds in settings with infinite number of alternatives and randomization over alternatives. It does not require any restriction on V_i. But one can use this theorem to identify domains where every implementable allocation rule satisfies revenue equivalence.

Theorem 8. (*Chung and Olszewski*[29] *and Heydenreich et al.*[30]). *Suppose V_i* is *connected. Then, every implementable allocation rule $F : V_i \to A$ satisfies revenue equivalence.*

In the n-agent environment, where $f : V \to A$ and payment rule $p_i : V \to \mathbb{R}$ takes the general form, the revenue equivalence property will look slightly different. It will say that an allocation rule f satisfies revenue equivalence if for every $i \in N$ and every $v_{-i} \in V_{-i}$, for every p_i, p'_i that implement f, we have $p'_i(v'_i) = p_i(v'_i) + h_i(v_{-i})$ for all $v'_i \in V_i$, where $h_i: V_{-i} \to \mathbb{R}$ is an arbitrary function.

Revenue equivalence allows one to write an explicit formula for payments of an implementable allocation rule, and as a result, several optimization problems involving payments of agents become an easy exercise – for instance optimal auctions in Myerson[4] and other problems in Nisan *et al.*[31].

Other extensions

So far we have focused attention on two types of results in multidimensional mechanism design: (a) explicit characterizations like Roberts' theorem and (b) implicit characterizations involving cycle monotonicity. We now discuss some extensions and open questions of these results.

1. *Randomization*. The allocation rules we considered so far are deterministic, i.e. they choose an alternative with probability one. If we allow for randomization, then a randomized allocation rule will choose a probability distribution over the alternatives at every profile of valuation vectors. It is known that Roberts' theorem and the two-cycle monotonicity characterizations do not extend when the allocation rule randomizes. Berger *et al.*[20], Carbajal and Ely[32], Rahman[18], Archer and Kleinberg[33] and Jehiel *et al.*[34] consider randomization and extend these results. Their results show the difficulty encountered with randomization. However, randomization is not a big problem for

extending the revenue equivalence result. Krishna and Maenner[28] and Vohra[5] showed that if the domain of possible valuation vectors is convex for every agent, then revenue equivalence holds even with randomization.

2. *Bayes–Nash implementation*. The dominant strategy incentive compatibility is a strict notion of equilibrium. One possibility is to relax it to require Bayes–Nash equilibrium. Bayes–Nash equilibrium will require that given other agents are revealing valuations truthfully, an agent maximizes his expected net utility by revealing his true valuation. Notice that the expectation requires prior information about valuations of agents. Relaxing the notion of equilibrium enlarges the set of implementable allocation rules. An extension of Roberts' theorem with Bayes–Nash equilibrium remains an open question. Muller *et al*.[35] showed that under a technical condition, an allocation rule is Bayes–Nash implementable if and only if it is two-cycle monotone. The analogue of revenue equivalence result continues to hold with Bayes–Nash implementation[30].
3. *Interdependence*. The model we consider is a private values model, where each agent is perfectly informed about his own valuation. However, in many situations, an agent may not be know his own valuation perfectly. This situation is modelled using interdependent valuation framework. Each agent receives a private signal, and the valuation(s) of an agent depends on the signals of all the agents. Even in the single-object auction framework, the second price auction is no longer incentive-compatible. With appropriate restrictions on how the valuation of an agent depends on the signals of other agents, efficiency can be implemented by a Groves-like payment rule[36]. With multidimensional signals, Jehiel *et al*.[37] showed that it is generically impossible to implement anything except constant allocation rules in sufficiently rich domains – see Bikhchandani[38] for more on this.
4. *Optimal auction design*. One of the primary applications of the results we discussed in this survey is finding an expected revenue maximizing (optimal) mechanism in various multidimensional settings. Indeed, Myerson[4] used these results to derive the optimal mechanism for the single-object auction case. An open question remains to derive the optimal auction for selling multiple objects – a multidimensional setting. There has been limited progress to this question in some multidimensional settings – see Vohra[5] for a survey. Some recent advances to this question can be found in Pai and Vohra[39,40].
5. *Computational constraints*. A growing literature in computer science is interested in multidimensional mechanism design[31]. One primary reason for this is that most mechanisms discussed in the mechanism design literature (e.g. the VCG mechanism) are computationally intractable in many settings. For instance, computing the VCG payments in the combinatorial auctions problem (selling multiple objects) belongs to the hardest computational class. The literature tries to find incentive compatible mechanisms which are computationally tractable, and yet have approximately good desirable properties. For instance, though the VCG mechanism is computationally hard in the combinatorial auctions problem, Dobzinski and Nisan[41] derived a non-affine maximizer allocation rule which is computationally easy. They showed that it approximates efficiency well in some settings.

1. Gibbard, A., Manipulation of voting schemes: a general result. *Econometrica*, 1973, **41**, 587–602.
2. Satterthwaite, M., Strategy-proofness and arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. *J. Econ. Theory*, 1975, **10**, 187–217.
3. Vickrey, W., Counterspeculation, auctions, and competitive sealed tenders. *J. Finance*, 1961, **16**, 8–37.
4. Myerson, R. B., Optimal auction design. *Math. Oper. Res.*, 1981, **6**, 58–73.
5. Vohra, R. V., *Mechanism Design: A Linear Programming Approach*, Cambridge University Press, New York, USA, 2011.
6. Mas-Collel, A., Whinston, M. and Green, J., *Microeconomic Theory*, Oxford University Press, Oxford, 1995.
7. Clarke, E., Multipart pricing of public goods. *Public Choice*, 1971, **11**, 17–33.
8. Groves, T., Incentives in teams. *Econometrica*, 1973, **41**, 617–631.
9. Cramton, P., Shoham, Y. and Steinberg, R., *Combinatorial Auctions*, MIT Press, 2006.
10. Roberts, K., Aggregation and revelation of preferences. In *The Characterization of Implementable Choice Rules* (ed. Laffont, J.-J.), North Holland Publishing, 1979, pp. 321–348.
11. Lavi, R., Mualem, A. and Nisan, N., Two simplified proofs of Roberts' theorem. *Soc. Choice Welfare*, 2009, **32**, 407–423.
12. Dobzinski, S. and Nisan, N., A modular approach to Roberts' theorem. In Proceedings of the 2nd International Symposium on Algorithmic Game Theory, Springer (Lecture Notes in Computer Science), 2009.
13. Mishra, D. and Sen, A., Roberts' theorem with neutrality: a social welfare ordering approach. *Games Econ. Behav.*, 2012, **75**, 283–298.
14. Carbajal, J. C., McLennan, A. and Tourky, R., Truthful implementation and aggregation in restricted domains. *J. Econ. Theor*. (in press).
15. Rochet, J. C., A necessary and sufficient condition for rationalizability in a quasi-linear context. *J. Math. Econ.*, 1987, **16**, 191–200.
16. Rockafellar, R. T., *Convex Analysis*, Princeton University Press, Princeton, NJ, USA, 1970.
17. Gui, H., Muller, R. and Vohra, R., Characterizing dominant strategy mechanisms with multidimensional types. Working Paper, Northwestern University, 2004.
18. Rahman, D., Detecting profitable deviations, Working Paper, University of Minnesota, Minneapolis, USA, 2011.
19. Kos, N. and Messner, M., Extremal incentive compatible transfers. *J. Econ. Theor*. (in press).
20. Berger, A., Muller, R. and Naeemi, S. H., Path-monotonicity and incentive compatibility. Working Paper, Maastricht University, 2010.
21. Mishra, D. and Roy, S., Implementation in multidimensional dichotomous domains. *Theor. Econ.*, 2012 (in press).

22. Bikhchandani, S., Chatterji, S., Lavi, R., Mualem, A., Nisan, N. and Sen, A., Weak monotonicity characterizes deterministic dominant strategy implementation. *Econometrica*, 2006, **74**, 1109–1132.
23. Saks, M. E. and Yu, L., Weak monotonicity suffices for truthfulness on convex domains. In Proceedings of 7th ACM Conference on Electronic Commerce, ACM Press, New York, USA, 2005, pp. 286–293.
24. Ashlagi, I., Braverman, M., Hassidim, A. and Monderer, D., Monotonicity and implementability. *Econometrica*, 2010, **78**, supplementary material.
25. Moulin, H., On strategyproofness and single-peakedness. *Public Choice*, 1980, **35**, 437–455.
26. Carroll, G., When are local incentive constraints sufficient? *Econometrica*, 2011.
27. Milgrom, P. and Segal, I., Envelope theorems for arbitrary choice sets. *Econometrica*, 2002, **70**, 583–601.
28. Krishna, V. and Maenner, E., Convex potentials with an application to mechanism design. *Econometrica*, 2001, **69**, 1113–1119.
29. Chung, K.-S. and Olszewski, W., A non-differentiable approach to revenue equivalence. *Theor. Econ.*, 2007, **2**, 1–19.
30. Heydenreich, B., Muller, R., Uetz, M. and Vohra, R. V., Characterization of revenue equivalence. *Econometrica*, 2009, **77**, 307–316.
31. Nisan, N., Roughgarden, T., Tardos, E. and Vazirani, V., *Algorithmic Game Theory*, Cambridge University Press, 2007.
32. Carbajal, J. C. and Ely, J., Mechanism design without revenue equivalence. Working Paper, Northwestern University and University of Queensland, 2011.
33. Archer, A. and Kleinberg, R., Truthful germs are contagious: a local to global characterization of truthfulness. In Proceedings of the 9th ACM Conference on Electronic Commerce, Springer (Lecture Notes in Computer Science), 2008.
34. Jehiel, P., Moldovanu, B. and Stacchetti, E., Multidimensional mechanism design for auctions with externalities. *J. Econ. Theory*, 1999, **85**, 258–293.
35. Muller, R., Perea, A. and Wolf, S., Weak monotonicity and Bayes–Nash incentive compatibility. *Games Econ. Behav.*, 2007, **61**, 344–358.
36. Krishna, V., *Auction Theory*, Academic Press, San Diego, CA, USA, 2002.
37. Jehiel, P., ter Vehn, M. M., Moldovanu, B. and Zame, W. R., The limits of ex post implementation. *Econometrica*, 2006, **74**, 585–610.
38. Bikhchandani, S., Ex post implementation in environments with private goods. *Theor. Econ.*, 2006, **1**, 369–393.
39. Pai, M. and Vohra, R., Optimal auctions with financially constrained bidders, Working Paper, Kellogg School of Management, 2008.
40. Pai, M. and Vohra, R., Optimal dynamic auctions, Working Paper, Kellogg School of Management, 2008.
41. Dobzinski, S. and Nisan, N., Multi-unit auctions: beyond Roberts. In Proceedings of 12th ACM Conference on Electronic Commerce, ACM Press, New York, USA, 2011, pp. 233–242.

ACKNOWLEDGEMENT. I am grateful to Y. Narahari (IISc, Bangalore) for his suggestions and comments.