Essays on Auctions and Mechanism Design

Abdul Quadir

Thesis submitted to the Indian statistical Institute in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy

Essays on Auctions and Mechanism Design

Abdul Quadir

October, 2014

Thesis Supervisor : Professor Debasis Mishra

Thesis submitted to the Indian statistical Institute in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy

Acknowledgement

I consider myself fortunate enough to have Debasis Mishra as my thesis supervisor. I am indebted beyond words to him for his constant guidance and support. I will always appreciate the way he structured my thinking. He also showed me how to present the mathematical argument clearly. Whenever I approached him, he did not hesitate in lending his patient ear to me. He always encouraged me during my stressful times and showed the path to move forward. He introduced me mechanism design theory and auction theory which lead to writing this thesis.

I am also grateful to prof. Arunava Sen, who always encourage me to work more by his enthusiasm. It is my privilege that I have been introduced Game Theory and Social choice Theory by him. I am also grateful to all the faculty members of Economics and Planning Unit, especially Tridip Ray and Souvik Roy.

Among all of my undergraduate teachers, one of them deserves special mention, Somesh Mathur, who taught me mathematical economics during my undergraduate studies. He always motivated me. I am very grateful to him for encouragement and support.

I am very grateful to my seniors, classmates and juniors during my days in the hostel. I thank Mridu, Anup, Soumendu, Dushyant, Mihir, Sonal, Madhav, Priyanka, Anuradha, Bipul, Soham, Pawan, Kanika, Subroto, Sutirtha, Kumarjit and Aditya for stimulating discussion and enjoyable company.

During my stay in ISI hostel, I came in contact with some wonderful people and later we became friends, I thank all of them. Two of them deserve special mention, Shoumitro and Siddharth.

I will always be in debt to Rashid Khalil who convinced me to take economics as my honours subject in my undergraduate course, otherwise I would be doing something else. Later, Nazrul Bhai encouraged me to study economics further. In fact, he introduced me the great economics institutes in India for pursuing further degree. I am very grateful to him.

I am also grateful to my friends Imran Khan, Farrokh, Naved, Faiz, Ravi Pathak, Shuja, Rizwan, Asif and Ayub. They were always present in sharing happiness and sadness. I am also thankful to my younger brother Sabir. I have learned a lot of concept of mathematics from him and we use to discuss mathematics at home during our holidays.

Finally, I would like to thank my parents for reposing in me their unflinching faith. I must acknowledge that without their support and love, none of this would have been possible.

Fellowship from the Indian Statistical Institute is gratefully acknowledged.

To my Parents

Contents

1 Introduction **1**
- 1.1 Non-bossy Single Object Auctions . . . 2
- 1.2 Mechanism Design In Single Dimensional Type Spaces . . . 4
- 1.3 Single Object Auctions With Externalities: A Tractable Model . . . 5

2 Non-bossy Single Object Auctions **7**
- 2.1 Introduction . . . 7
  - 2.1.1 Relationship with Literature . . . 9
- 2.2 The Single Object Auction Model . . . 11
- 2.3 Implementation, Non-Bossiness, and Rationalizability . . . 12
  - 2.3.1 Rationalizability . . . 12
  - 2.3.2 Non-bossy Single Object Auctions . . . 15
  - 2.3.3 Simple Utility Maximization . . . 18
  - 2.3.4 Randomization and Bayesian Implementation via Border's Hierarchical Allocation Rules . . . 21
  - 2.3.5 Extension of Roberts' Theorem . . . 22
- 2.4 The Complete Characterization . . . 23
- 2.5 Discussions . . . 25

3 Mechanism Design in Single Dimensional Type Spaces **37**
- 3.1 Introduction . . . 37
  - 3.1.1 Relation of Our Result to the Literature . . . 38
- 3.2 The Connected Graph Model . . . 39
  - 3.2.1 The Complete Characterization . . . 40
  - 3.2.2 Payments and Revenue Equivalence . . . 44
- 3.3 The Matroid Model . . . 44
  - 3.3.1 The Complete Characterization For The Matroid Model . . . 45
  - 3.3.2 Applications . . . 47
- 3.4 Anonymity and Efficient Allocation Rule . . . 49
- 3.5 Conclusion . . . 55

4 Single Object Auctions With Externalities: A Tractable Model **61**
- 4.1 Introduction . . . 61
  - 4.1.1 Related Literature . . . 62
- 4.2 The Model . . . 63
- 4.3 Implementable Allocation Rules . . . 65
- 4.4 Revenue Maximization For The Known Ranking . . . 68
- 4.5 Private Rankings . . . 73

4.5.1 Implementability and Payment Rules 74
4.6 Conclusion 78

Chapter 1

Introduction

This thesis consists of three chapters that aim to characterize incentive compatible mechanisms in specific mechanism design settings. In these settings, the designer is allowed to use payments but the net utility of every agent is linear in payments. This particular assumption on net utility is called quasi-linearity. Each of the three chapters in the thesis identifies a class of mechanisms and characterizes them (in quasilinear private value environment) using dominant strategy incentive compatibility and some additional reasonable conditions.

In quasi-linear environment, a mechanism can be decomposed into an allocation rule and a payment rule for every agent. If a mechanism is dominant strategy incentive compatible, then we say that the corresponding allocation rule is implementable. The classic Vickrey-Clarke-Groves (Vickrey, 1961; Clarke, 1971; Groves, 1973) mechanisms implement the efficient allocation rule by using Groves payments. Under reasonable conditions, these are the only payment rules that implement the efficient allocation rule (Holmstrom, 1979). This feature generalizes to any implementable allocation rule, and is known as the revenue equivalence principle. As a consequence of the revenue equivalence, the characterization of the class of incentive compatible mechanism can be done in two steps: (a) characterize the implementable allocation rules (b) for each implementable allocation rule, identify one payment rule that implements it. We follow this prescription in all three chapters.

The characterization of implementable allocation rules will depend, among other things, on the type space considered in the problem. The type space that we consider in all these three chapters is one dimensional (a connected subset of $\mathbb{R}_{++}$). This makes the characterization harder since the type space is restricted. If the type space was unrestricted, then Roberts (1979) has shown that under mild additional condition only affine maximizer allocation rules are implementable. Affine maximizers are linear generalization of efficient allocation rule. In one dimensional type spaces that we consider, we identify a significantly larger class of implementable allocation rules and characterize them.

The first chapter considers the standard single object auction model in private values setting. It identifies the class of allocation rules that is called strongly rationalizable allocation rules. These are the only rules that are implementable and satisfy a condition known as non-bossiness (Satterthwaite and Sonnenschein, 1981). Non-bossiness requires that if an agent changes his type such that his own allocation does not change, then the allocation of other agents must not change. Under additional technical condition, this characterization can be sharpened. In particular, we identify a class of allocation rules called simple utility maximizers and show that each strongly rationalizable allocation rule is equivalent to a simple utility maximizer under a technical condition. The advantage of this characterization is that simple utility maximizer is easier to use and interpret than the strongly rationalizable allocation rule. They are also a natural extension of Roberts' affine maximizer rules.

The second chapter considers an abstract model where the set of alternatives exhibits certain discrete structural properties. In particular, we assume that alternatives are *bases* of a *matroid*. Each agent owns an element of the ground set of the matroid. The advantage of this model is that it covers many practical models as special cases. We show that this model covers the single object auction problem, multi-unit auction model with unit demand, the heterogeneous good auction with dichotomous preferences and a connected graph model. Our main result in this chapter is the complete characterization of implementable allocation rules for the general matroid model. In particular, we identify a class of implementable allocation rules called the generalized utility maximizers and show them to be the only implementable allocation rules. We also discuss implications of anonymity and non-bossiness in some of these models.

The third chapter deals with the single object auction with externalities. It models externalities in a specific way. Each agent's type is his valuation for the object. Further, it also identifies his valuation when another agent gets the object by assuming a specific nature of externality. In this problem, we identify a property of the allocation rule that we call the interval property and show that it is necessary and sufficient for implementability. The interval property can be thought as a generalization of the monotonicity condition identified in Myerson (1981) to our model. Using this characterization, we find the revenue maximizing auction for this model. We also discuss a multi-dimensional model of externalities in this chapter and discuss the challenges one faces while extending our result to this setting.

We discuss briefly each chapter below.

1.1 Non-bossy Single Object Auctions

In this chapter, we study single object auctions in the private values model and restrict attention to deterministic single object auctions. We try to identify a set of implementable allocation rules. In fact, in theory as well as in practice, we know only a very few implementable allocation rules. For instance *efficient* allocation rule in the single object auction private values model is implementable by the Vickrey auction and *constraint efficient* with reserve price is implementable by Myerson's revenue maximizing auction in the independent private values model etc.

Therefore, it is important to understand how these allocation rules distinguish themselves from the remaining implementable allocation rules. A primary motivation of this chapter is to carry out a systematic analysis of this question axiomatically.

Every auction that we observe generally involves in some form of maximization and are deterministic dominant strategy incentive compatible. If ties in these maximizations are broken carefully, then the allocations in these auctions also satisfy an appealing property-*non-bossiness*. Non-bossiness is the following requirement. Suppose agent i is not winning the object at a particular valuation profile (v_i, v_{-i}) and we go to another valuation profile (v'_i, v_{-i}) where the valuation of agent i changes, such that agent i still does not win the object. Then, the agent who is winning the object at valuation profile (v_i, v_{-i}) continues to win the object at (v'_i, v_{-i}). In other words, if an agent cannot change his own outcome, then he cannot change the outcome of any other agent.

We give a complete characterization of implementable and non-bossy allocation rules. For this characterization, we introduce a novel idea of rationalizability in the single object allocation model and, use it to define a class of implementable allocation rules that we call the *strongly rationalizable* allocation rules. Our result says that an allocation rule is implementable and non-bossy if and only if it is strongly rationalizable allocation rule.

We further sharpen our characterization by imposing a mild condition of *continuity* on allocation rules. We define a notion of a *simple utility* function for every agent, that is a non-decreasing function that maps all the possible valuation of an agent to a real number. Using these simple utility function, we introduce a new class of implementable allocation rules, which we call *simple utility maximizer*. A simple utility maximizer is an allocation rule that chooses a simple utility function for every agent. Then, at every valuation profile (a) it does not allocate the object if every agent has negative simple utility function and (b) if at least one agent has positive simple utility, then it allocates the object to the agent who has the highest simple utility. Our result shows that if an allocation rule satisfies a mild continuity condition, then it is implementable and non-bossy if and only if it is a simple utility maximizer allocation rule (with a suitable tie-breaking rule).

Simple utility maximizer includes all the commonly used allocation rules in single object auctions like efficient allocation rule, efficient allocation rule with reserve price and the optimal auction rule in Myerson (1981). Hence, our characterization provides an axiomatic foundation to a very rich class of implementable allocation rules. Since we have characterized the implementable and non-bossy allocation rules, using revenue equivalence principle we can pin down all the payments that implement these allocation rules. Thus, we get a complete characterization of mechanisms that use non-bossy allocation rule.

We also extend our idea of simple utility function to a more general utility function that maps all the possible type space of all agent to a real number. We define a larger class of implementable allocation rule than simple utility maximizer, that we call *generalized utility maximizer*. We show that implementability is equivalent to generalized utility maximizer for single object auction model. Generalized utility maximizers are more complex allocation rules. This shows how a natural condition like non-bossiness helps to have a more simple allocation rules.

1.2 Mechanism Design In Single Dimensional Type Spaces

In this chapter, we study a mechanism design problem over a connected graph in quasi-linear and private value environment. This problem on connected graph captures many practical situations like mobile network or procurement auction. The planner wants to use a subset of edges.

We assume that an edge is privately held by a single agent, who incurs some cost when his edge is used. The costs are private information of agents. The mechanism designer wants to design a mechanism. Our objective is to characterize the set of all dominant strategy incentive compatible mechanisms for this problem. We identify a rich set of implementable allocation rules like the last result for the single object auction model in the previous chapter. We call them generalized utility maximizers. We also show that there exist payment rules such that generalized utility maximizers and the corresponding payment rules are dominant strategy incentive compatible.

A generalized utility maximizer assigns a map, which we call generalized utility function (GUF), to every agent. A GUF assigns a real number, which we call generalized utility, to every profile of types. A generalized maximizer chooses a subset of edges at every type profile such that (a) all the nodes are connected and (b) the sum of generalized utilities of agents is maximized.

In fact, we extend the characterization of this model to a model where the planner is choosing a *base* of a matroid and the matroid is defined by the set of agents as ground set and the family of subsets of agents being *independent set*. This model allows us to capture the connected graph model and many well-studied models in the literature, including multi-unit auction model, heterogeneous good auction model with dichotomous preferences. Thus, we characterize the dominant strategy incentive compatible mechanisms for a family of problems where the type of an agent is a single number (i.e, type is one dimensional) and the set of alternatives exhibits certain structure.

We consider two more plausible conditions, a suitable version of *anonymity* and *non-bossiness* on top of incentive compatibility and sharpen our characterization. We show that only dominant strategy incentive compatible mechanisms are the Groves mechanism, i.e, uses efficient allocation rules. We have established this result for the connected graph model. But it can be easily extended to the matroid model. Thus, it extends and unifies the similar results that have been proved in specific models.

1.3 Single Object Auctions With Externalities: A Tractable Model

In this chapter, we study a single object auction model with externalities. We know that in many situations, individuals enjoy some utility even if they do not own a particular object. In auctioning of many objects like patents and paintings, although a bidder does not win the object, he may still enjoy some utility.

We restrict attention to deterministic single object auction. We identify a necessary and sufficient condition for implementable allocation rules in our model. Using revenue equivalence, we characterize all the dominant strategy incentive compatible mechanisms. Then, we design a revenue maximizing auction (optimal auction) for our model. The unique feature of this model is that every agent makes some payment irrespective of whether he is getting the object or not.

The innovative feature of this paper lies in the way we model externalities. Imagine a situation where there are certain features of every agent known to everyone that allows one to infer how he will use the object. For instance, in case of patents, a company's past use of patents may reflect how he will use any patent in the market. Such features directly influence the utility other agents will have from him owning the object. We model this aspect by assuming that each agent has a a strict ranking over the set of all agents (including himself) and the seller, where he keeps himself at the top and the seller at the bottom of the ranking. A commonly known real number is assigned to each position, with the top position getting 1, the bottom position getting zero, and each intermediate position getting a number strictly between 0 and 1 with the numbers decreasing with position. The utility for an agent when an agent gets the object or the seller keeps the object is a product of his own valuation and the real number associated with the position of the winning agent in his ranking. For instance, if agent i ranks agent j at the third position, then the utility of agent i when agent j wins the object is $\alpha_3 v_i$, where v_i is the valuation of agent i for the object and $\alpha_3 \in (0, 1)$ is the third position-specific number.

We identify a condition on the allocation rule, which we call the *interval property*, that is necessary and sufficient for implementability of the allocation rule. The interval property requires the following requirement: considering an arbitrary agent i and fixing the valuation of other agents, if agent j wins the object at v_i and agent k wins the object at v_i', where $v_i' > v_i$, then agent k is higher than agent j in the ranking of agent i.

This interval property allows us to pin down one payment rule that implements an implementable allocation rule. Using revenue equivalence, we can then pin down the entire class of payment rules that can implement an implementable allocation rule. These ideas are then used to derive a revenue maximizing auction for this model using Myersonian techniques.

Chapter 2

Non-bossy Single Object Auctions

2.1 Introduction

We study single object auctions in the private values model. We restrict attention to deterministic single object auctions, i.e., auctions where the probability of allocating the object to any agent is either zero or one.[1] An allocation rule for single object auction is implementable if we can find payments such that truth-telling is a dominant strategy for every agent. A central result in mechanism design is that the efficient allocation rule in the single object auction private values model is implementable using the Vickrey auction (Vickrey, 1961; Clarke, 1971; Groves, 1973). On the other hand, a revenue maximizing auction in the independent private values model maximizes the *virtual valuations* of the agents (Myerson, 1981). English auction with a reserve price is popular in practice (seen on EBay and other Internet sites) and in theory, for instance, in designing approximately optimal auctions (Hartline and Roughgarden, 2009; Dhangwatnotai et al., 2010). Such an auction implements a *constrained* efficient allocation rule with a reserve price - it does not allocate the object if the valuation of each bidder is less than the reserve price, but when it allocates the object it does so to the highest bidder.

While the set of implementable allocation rules is quite rich, we encounter only these particular simple class of implementable allocation rules in theory and practice. Hence, it is important to understand how these allocation rules distinguish themselves from the remaining implementable allocation rules. A primary motivation of this paper is to carry out a systematic analysis of this question axiomatically.

Common features of all these auctions are that the allocation rules are deterministic, dominant strategy implementable, and involve maximization of some form. If ties in these maximizations are broken carefully, then the allocation rules mentioned above satisfy another appealing property - *non-bossiness*. Non-bossiness is the following requirement. Suppose agent i is not winning the object at a particular valuation profile (v_i, v_{-i}) and we go to another valuation profile (v'_i, v_{-i}), where the valuation of only agent i changes, such that

[1] A shorter version of this chapter has been published.

agent i still does not win the object. Then, the agent who was winning the object at the valuation profile (v_i, v_{-i}) continues to win the object at (v'_i, v_{-i}). In other words, if an agent cannot change his own outcome, then it cannot change the outcome of any other agent. [2]

We provide a complete characterization of implementable and non-bossy allocation rules. For this characterization, we introduce a novel notion of rationalizability in the single object allocation model, and use it to define a class of allocation rules that we call the *strongly rationalizable* allocation rules. Our characterization says that an allocation rule is implementable and non-bossy if and only if it is a strongly rationalizable allocation rule.

Under a mild continuity condition, we sharpen our characterization. We define the notion of a *simple utility function*, which is any non-decreasing function that maps the set of possible valuations of an agent to the set of real numbers. A simple utility maximizer is an allocation rule that chooses a simple utility function for every agent. Then, at every valuation profile (a) it does not allocate the object if every agent has negative simple utility and (b) if at least one agent has positive simple utility, then it allocates the object to an agent with the highest simple utility. We show that if an allocation rule satisfies a mild continuity condition, then it is implementable and *non-bossy* if and only if it is a simple utility maximizer allocation rule (supplemented with an appropriate tie-breaking rule).

All the commonly used allocation rules in single object auctions (e.g., efficient allocation rule, efficient allocation rule with a reserve price, the optimal auction allocation rule in Myerson (1981)) are simple utility maximizer allocation rules. Hence, our results provide an axiomatic foundation for a rich class of commonly used allocation rules. Although we characterize implementable and non-bossy allocation rules, using revenue equivalence (Myerson, 1981), we can pin down the payments that will implement these allocation rules. Thus, we get a complete characterization of "mechanisms" that use non-bossy allocation rules.

Our characterizations have a common feature - implementability and non-bossiness is equivalent to some form of maximization by the seller at every valuation profile. These results relate to two fundamental results in mechanism design and auction theory. A benchmark result in private value mechanism design in quasi-linear environments is the Roberts' affine maximizer theorem (Roberts, 1979). It considers general multidimensional type spaces with finite set of alternatives. A type of an agent in such models is a vector in $\mathbb{R}^{|A|}$, where A is the set of alternatives. Roberts (1979) showed that if there are at least three alternatives and the type space is *unrestricted* (i.e., $\mathbb{R}^{|A|}$), then every onto implementable allocation rule is an affine maximizer. It can be shown that every affine maximizer is implementable. [3] An

[2] The use of non-bossiness axiom in social choice theory with private good allocations, specially matching problems, is extensive - it was first used by Satterthwaite and Sonnenschein (1981), and subsequently in matching problems (Svensson, 1999; Papai, 2000; Ehlers, 2002; Hatfield, 2009) and cost sharing problem (Mutuswami, 2005).

[3] Carbajal et al. (2013) show that if there are at least three alternatives and the type space of every agent is unrestricted, then an onto allocation rule is implementable if and only if it is a *lexicographic* affine maximizer. Lexicographic affine maximizers contain a particular class of affine maximizers where ties are broken carefully.

affine maximizer can be thought to be a linear simple utility function. The single object auction model has a restricted type space. As a result, Roberts' result does not apply. Our characterizations can be thought as extension of Roberts' affine maximizer result to the single object auction model.

Further, in a seminal result, Border (1991) showed that the interim allocation probability obtained by every Bayesian and randomized allocation rule can be obtained by taking convex combination of certain dominant strategy implementable allocation rules that he called *hierarchical allocation rules* - see also Manelli and Vincent (2010); Deb and Pai (2013). As we discuss later, a hierarchical allocation rule can be written as a convex combination of simple utility maximizer allocation rules that we identify (which are deterministic, dominant strategy implementable, and non-bossy allocation rules). Hence, the set of dominant strategy implementable and non-bossy deterministic allocation rules occupy a pivotal role in the set of all randomized and Bayesian implementable allocation rules.

Finally, we extend our idea of simple utility maximizer allocation rule to define an even larger class of allocation rules that we call *generalized utility maximizer* allocation rules. We show that implementability is equivalent to these allocation rules. While this result is also in the spirit of Roberts' affine maximizer theorem, the proof is a simple consequence of Myerson's monotonicity characterization of implementable allocation rule, which we discuss below. Generalized utility maximizers are more complex allocation rules than simple utility maximizers. This shows how a natural condition like non-bossiness helps us to separate complex auction rules from simple and commonly used auction rules.

2.1.1 Relationship with Literature

Myerson (1981) shows that implementability is equivalent to a monotonicity property of the allocation rules. [4] The monotonicity property is equivalent to requiring that for every agent i and for every valuation profile of other agents, there is a cutoff valuation of agent i below which he does not get the object and above which he gets the object. [5]

The relationship between our results and the monotonicity characterization can be best illustrated by reference to parallel results in the strategic voting literature. Muller and Satterthwaite (1977) show that Maskin monotonicity, the counterpart of monotonicity in the strategic voting models, is necessary for dominant strategy implementation, and if the domain is unrestricted then it is also sufficient. However, the seminal results of Gibbard (1973) and Satterthwaite (1975) show that dictatorship is the only dominant strategy implementable voting rule satisfying unanimity.

[4] See also extensions of this characterization to the multidimensional private values models in Bikhchandani et al. (2006); Saks and Yu (2005); Ashlagi et al. (2010); Cuff et al. (2012); Mishra and Roy (2013).

[5] The results in Myerson (1981) are more general. In particular, he considers implementation in Bayes-Nash equilibrium and allows for randomization. But the expected revenue maximizing allocation rule he identifies is a deterministic and dominant strategy implementable allocation rule.

In the quasi-linear private values models, Roberts' theorem can be thought of as the counterpart of the Gibbard-Satterthwaite theorem (Gibbard, 1973; Satterthwaite, 1975). After the result of Gibbard (1973) and Satterthwaite (1975), a vast literature in social choice theory has pursued the characterization of implementable allocation rules in restricted "voting" domains, e.g., the median voting rule and its generalizations characterize implementable allocation rules in single-peaked domains (Moulin, 1980; Barbera et al., 1993). Indeed, these characterizations of implementable allocation rules are all in the spirit of Roberts' theorem - they describe the precise *parameters* that are required to design an implementable allocation rule. In this spirit, our results give explicit characterization of implementable allocation rules for the single object auction model.

There have been extensions of Roberts' theorem to certain environments. For instance, Mishra and Sen (2012) show that Roberts, theorem holds in certain bounded byt full dimensional type spaces under an additional condition of *neutrality* along with implementability. Their neutrality condition is vacuous in the single object auction model. Moreover, the type space in the single object auction model is not full dimensional. Carbajal et al. (2013) extend Roberts' theorem to certain restricted type spaces which satisfy some technical conditions. Though it covers many interesting models, including those with infinite set of alternatives, the single object auction model does not satisfy their technical conditions. Marchant and Mishra (2012) extend Roberts' theorem to the case of two alternatives. Since the number of alternatives in the single object auction model is more than two, their results do not hold in our model.

Jehiel et al. (2008) show that a version of the Roberts' theorem holds even in the interdependent values model (they require implementation in ex-post equilibrium). They also require the complete domain assumption like Roberts (1979), and remark that their result does not hold in restricted one-dimensional settings like the single object auction.

Two related work in computer science literature deserve special mention. Lavi et al. (2003) focus on a particular restricted domain, which they call order-based domains (this includes some auction domains). Under various additional restrictions on the allocation rule (which includes an independence condition), they show that every implementable allocation rule must be an "almost" affine maximizer - roughly, almost affine maximizers are affine maximizers for large enough values of types of agents.

Next, Archer and Tardos (2002) consider the single object auction model and show that if the *object is always allocated* then the only implementable allocation rules satisfying non-bossiness and three more additional conditions are *min function* allocation rules. [6] Min function allocation rules are simple utility maximizer allocation rules, but with some additional limiting and continuity properties. Though our characterization of simple utility

[6] Archer and Tardos (2002) consider a more general environment than ours in which a planner needs to select a path in a graph, where each edge represents an agent. Informally, their three additional conditions are various range and tie-breaking conditions, and called *edge autonomy, path autonomy,* and *sensitivity*. The non-bossy condition is called *independence* by them.

maximizer is related to their result, it has several important differences. First, their result requires that we *always* sell the good. This rules out any allocation rule with a reserve price, such as Myerson's revenue maximizing allocation rule. Further, our proof shows that allowing the object to be not sold adds several non-trivial complications in deriving our results. Second, they seem to require different types of range and tie-breaking conditions than our continuity requirement. On the other hand, our characterization of simple utility maximizer makes it explicit the way ties need to be broken. Finally, they have no analogue of our other characterizations.

There have been many simplifications of the original proof of Roberts (Jehiel et al., 2008; Lavi, 2007; Dobzinski and Nisan, 2009; Vohra, 2011; Mishra and Sen, 2012). But none of these proofs show how Roberts' theorem can be extended to a restricted domain like the single object auction model. Unlike most of the literature, our goal is not to characterize "affine maximizers" - indeed, all our characterizations capture a larger class of implementable allocation rules than affine maximizers.

An alternate approach is to characterize the set of dominant strategy mechanisms directly by imposing conditions on mechanisms rather than just on allocation rules. A contribution along this line is Ashlagi and Serizawa (2011). They show that any mechanism which always allocates the object, satisfies individual rationality, non-negativity of payments, *anonymity in net utility*, and dominant strategy incentive compatiblity must be the Vickrey auction. This result is further strengthened by Mukherjee (2014), who shows that any strategy-proof and anonymous (in net utility) mechanism which always allocates the object must use the efficient allocation rule. Further, Sakai (2013) characterizes the Vickrey auction with a reserve price using various axioms on the *mechanism* (this includes an axiom on the allocation rule which requires a weak version of efficiency). By placing minimal axioms on *allocation rules*, we are able to characterize a broader class of dominant strategy incentive compatible mechanisms (using revenue equivalence) than these papers.

2.2 The Single Object Auction Model

A seller is selling an indivisible object to n potential agents (buyers). The set of agents is denoted by $N := \{1, \ldots, n\}$. The private value of agent i for the object is denoted by $v_i \in \mathbb{R}_{++}$. The set of all possible private values of agent i is $V_i \subseteq \mathbb{R}_{++}$ - note that we do not allow zero valuations. We will use the usual notations v_{-i} and V_{-i} to denote a profile of valuations without agent i and the set of all profiles of valuations without agent i respectively. Let $V := V_1 \times V_2 \times \ldots \times V_n$.

The set of alternatives is denoted by $A := \{e^0, e^1, \ldots, e^n\}$, where each e^i is a vector in $\mathbb{R}^n$. In particular, e^0 is the zero vector in $\mathbb{R}^n$ and e^i is the unit vector in $\mathbb{R}^n$ with i-th component 1 and all other components zero. The j-th component of the vector e^i will be denoted by e^i_j. The alternative e^0 is the alternative where the seller keeps the object and for every $i \in N$, e^i is the alternative where agent i gets the object. Notice that our model focuses on deterministic alternatives. Every agent $i \in N$ gets zero value from any alternative where he does not get the object. An allocation rule is a mapping $f : V \to A$. For every $v \in V$ and for every $i \in N$, the notation $f_i(v) \in \{0, 1\}$ will denote if agent i gets the object ($f_i(v) = 1$) or not ($f_i(v) = 0$) at valuation profile v in allocation rule f.

Payments are allowed and agents have quasi-linear utility functions over payments. A payment rule of agent $i \in N$ is a mapping $p_i : V \to \mathbb{R}$.

Definition 1 *An allocation rule f is* **implementable** *(in dominant strategies) if there exist payment rules $(p_1, \ldots, p_n)$ such that for every agent $i \in N$ and for every $v_{-i} \in V_{-i}$*

$$v_i f_i(v_i, v_{-i}) - p_i(v_i, v_{-i}) \geq v_i f_i(v'_i, v_{-i}) - p_i(v'_i, v_{-i}) \qquad \forall\ v_i, v'_i \in V_i.$$

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

Notice that we focus on deterministic dominant strategy implementation.

Myerson (1981) showed that the following notion of monotonicity is equivalent to implementability - see also Laffont and Maskin (1980) for a similar characterization.

Definition 2 *An allocation rule f is* **monotone** *if for every $i \in N$, for every $v_{-i} \in V_{-i}$, and for every $v_i, v'_i \in V_i$ with $v_i < v'_i$ and $f_i(v_i, v_{-i}) = 1$, we have $f_i(v'_i, v_{-i}) = 1$.*

Myerson (1981) shows that an allocation rule is implementable if and only if it is monotone - this result does not require any restriction on the space of valuations (see Vohra (2011), for instance). Throughout the paper, our results will be driven by the monotonicity condition.

2.3 Implementation, Non-Bossiness, and Rationalizability

We now provide the main results of this chapter. We will define the notion of a *non-bossy* allocation rule. Then, we will provide a complete characterization of non-bossy and implementable allocation rules. Finally, we will add a mild continuity-like condition to sharpen this characterization even further.

The backbone of this result is a notion of rationalizability in our model, and this reveals an elegant structure of implementable and non-bossy allocation rules. We introduce this idea of rationalizability in the single object auctions next.

2.3.1 Rationalizability

To define rationalizability in our context, we view the mechanism designer as a decision maker who is making choices using his allocation rule. Notice that at every profile of valuations, by choosing an alternative, the mechanism designer assigns values to each agent - zero to all agents who do not get the object but positive value to the agent who gets the object. Denote by $\mathbf{1}_{v_i}$ the vector of valuations in $\mathbb{R}^n_+$, where all the components except agent i has zero and the component corresponding to agent i has v_i. Further, denote by $\mathbf{1}_0$ the n-dimensional zero vector. For convenience, we will write $\mathbf{1}_0$ as $\mathbf{1}_{v_0}$ at any valuation profile.

Using this notation, at a valuation profile $(v_1, \ldots, v_n)$, a mechanism designer's choice of an alternative in A can lead to the selection of one of the following $(n+1)$ vectors in $\mathbb{R}^n_+$ to be chosen - $\mathbf{1}_{v_0}, \mathbf{1}_{v_1}, \ldots, \mathbf{1}_{v_n}$. We will refer to these vectors as *utility vectors*. Any allocation rule f can alternatively thought of choosing utility vectors at every valuation profile. The domain of valuations V_i of agent i gives rise to a set of feasible utility vectors where only agent i gets positive value. In particular define for every $i \in N$, $D_i := \{\mathbf{1}_{v_i} : v_i \in V_i\}$. Further, let $D_0 := \{\mathbf{1}_{v_0}\}$ and $V_0 = \{0\}$. Denote by $D := D_0 \cup D_1 \cup D_2 \cup \ldots \cup D_n$ the set of all utility vectors consistent with the domain of profile of valuations V.

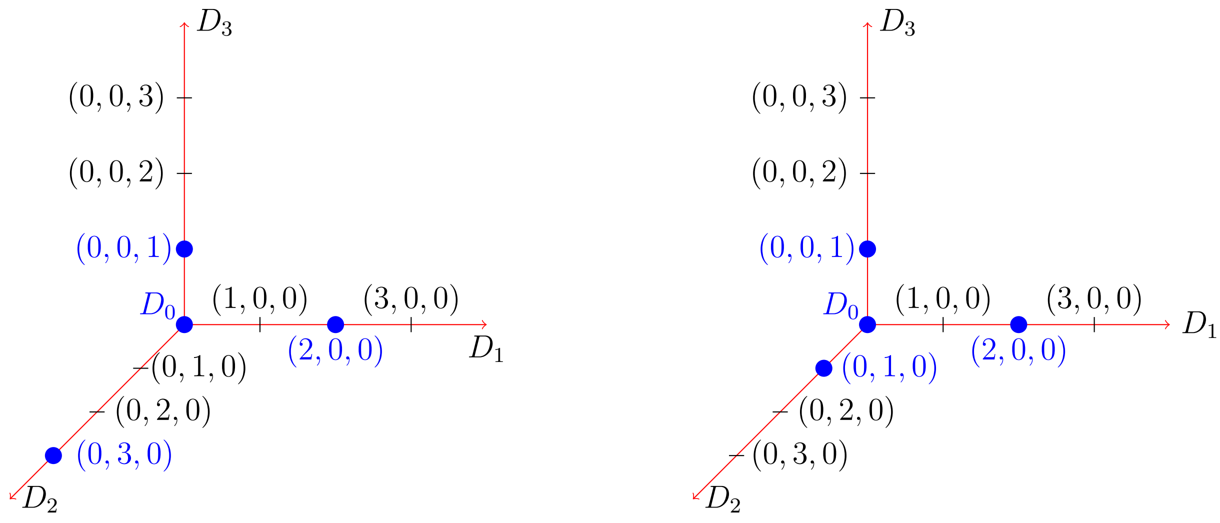
To define the notion of a rational allocation rule, we will use orderings (reflexive, complete, and transitive binary relation) on the set of utility vectors D. For any ordering $\succeq$ on D, let $\succ$ be the asymmetric component of $\succeq$ and $\sim$ be the symmetric component of $\succeq$. A strict linear ordering is an anit-symmetric ordering with no symmetric component. An ordering $\succeq$ on D is monotone if for every $i \in N$, for every $v_i, v'_i \in V_i$ with $v_i > v'_i$, we have $\mathbf{1}_{v_i} \succ \mathbf{1}_{v'_i}$. Our notion of rational allocation requires that at every profile of valuations it must choose a maximal element among the utility vectors at that valuation profile, where the maximal element is defined using a monotone ordering on D.

An example with three agents will clarify some of the concepts.

Example 1

Let $N = \{1, 2, 3\}$. So, the set of alternatives is $A = \{e^0, e^1, e^2, e^3\}$. Let $V_1 = V_2 = V_3 = \{1, 2, 3\}$. In that case, the utility vectors are vectors in $\mathbb{R}^3_+$. In particular, D_0 contains the origin, $D_1 = \{(1,0,0), (2,0,0), (3,0,0)\}$, $D_2 = \{(0,1,0), (0,2,0), (0,3,0)\}$, and $D_3 = \{(0,0,1), (0,0,2), (0,0,3)\}$. Figures 2.1(a) and 2.1(b) show D_0, D_1, D_2, D_3 with two valuation profiles (shown in dark circles in each figure). A valuation profile corresponds to four points in $D \equiv (D_0 \cup D_1 \cup D_2 \cup D_3)$. The valuation profile (v_1, v_2, v_3) corresponding to Figure 2.1(a) is $(2, 3, 1)$ (the corresponding utility vectors are shown in dark blue dots in the figure) and that corresponding to Figure 2.1(b) is $(2, 1, 1)$.

Now, consider the following ordering $\succeq$ defined on D: $(0,0,3) \succ (0,3,0) \succ (0,2,0) \succ (3,0,0) \succ (0,0,0) \sim (0,0,2) \succ (0,1,0) \sim (2,0,0) \succ (1,0,0) \succ (0,0,1)$. Note that $\succeq$ is monotone. Consider an allocation rule f, which chooses the $\succeq$-maximal utility vector at every valuation profile. For instance, consider the utility vectors corresponding to valuation profile $(2, 3, 1)$ (shown in Figure 2.1(a)). The $\succeq$-maximal utility vector at this valuation profile is $(0, 3, 0)$ and hence, f allocates the object to agent 2. Similarly, consider the utility vectors corresponding to valuation profile $(2, 1, 1)$ (shown in Figure 2.1(b)). The $\succeq$-maximal utility vector at this valuation profile is $(0, 0, 0)$ and hence, f does not allocate the object to any agent. We call such allocation rules *rationalizable* allocation rules.



(a) (b)

Figure 2.1: Illustration of rationalizable allocation rule

We now formally define a rationalizable allocation rule. For every allocation rule f, let $G^f : V \to D$ be a *social welfare function* induced by f, i.e., for all $v \in V$, $G^f(v) = \mathbf{1}_{v_j}$ if $f(v) = e^j$ for any $j \in \{0, 1, \dots, n\}$.

Definition 3 *An allocation rule f is* **rationalizable** *if there exists a monotone ordering $\succeq$ on D such that for all $v \in V$, $G^f(v) \succeq \mathbf{1}_{v_j}$ for all $j \in \{0, 1, \dots, n\}$. In this case, we say $\succeq$ rationalizes f.*

An allocation rule f is **strongly rationalizable** *if there exists a monotone strict linear ordering $\succ$ on D such that for all $v \in V$, $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$ for all $j \in \{0, 1, \dots, n\} \setminus \{i\}$, where $G^f(v) = \mathbf{1}_{v_i}$. In this case, we say $\succ$ strongly rationalizes f.*

We will investigate the relationship between (strongly) rationalizable allocation rules and implementable allocation rules. The following lemma establishes that a rational allocation rule is implementable.

Lemma 1 *Every rationalizable allocation rule is implementable.*

Proof: Consider a rationalizable allocation rule f and let $\succeq$ be the corresponding ordering on D. Fix an agent i and valuation profile v_{-i}. Consider two valuations of agent i: v_i and v'_i with $v_i < v'_i$ with $f(v_i, v_{-i}) = e^i$. By definition of $\succeq$, $\mathbf{1}_{v_i} \succeq \mathbf{1}_{v_j}$ for all $j \in (N \cup \{0\}) \setminus \{i\}$. Since $\succeq$ is monotone, $\mathbf{1}_{v'_i} \succ \mathbf{1}_{v_i}$. By transitivity, $\mathbf{1}_{v'_i} \succ \mathbf{1}_{v_j}$ for all $j \in (N \cup \{0\}) \setminus \{i\}$. Then, by the definition of $\succeq$, $f(v'_i, v_{-i}) = e^i$. Hence, f is monotone, which further implies that it is implementable (Myerson, 1981). ■

The converse of Lemma 1 is not true. The following example establishes that.

Example **2**

Suppose there are two agents: $N = \{1, 2\}$. Suppose $V_1 = V_2 = \mathbb{R}_{++}$. Consider an allocation rule f defined as follows. At any valuation profile (v_1, v_2), if $\max(v_1 - 2v_2, v_2 - v_1) < 0$, then $f(v_1, v_2) = e^0$. Else, if $v_1 - 2v_2 < v_2 - v_1$, then $f(v_1, v_2) = e^2$ and if $v_1 - 2v_2 \geq v_2 - v_1$, then $f(v_1, v_2) = e^1$. It is easy to verify that f is monotone, and hence, implementable.

We argue that f is not a rationalizable allocation rule. Assume for contradiction that f is a rationalizable allocation rule and $\succeq$ is the corresponding monotone ordering. Consider the profile of valuation (v_1, v_2), where $v_1 = 1$ and $v_2 = 2$. For $\epsilon > 0$ but arbitrarily close to zero, $f(v_1, v_2 - \epsilon) = e^2$. Hence, $\mathbf{1}_{v_2 - \epsilon} \succeq \mathbf{1}_{v_0}$. By monotonicity, $\mathbf{1}_{v_2} \succ \mathbf{1}_{v_0}$. Now, consider the profile of valuations (v'_1, v_2), where $v'_1 = 2 + \epsilon$ and $v_2 = 2$. Note that $f(v'_1, v_2) = e^0$. Hence, $\mathbf{1}_{v_0} \succeq \mathbf{1}_{v_2}$. This is a contradiction.

A feature of this example is that at valuation profile (v_1, v_2), the allocation rule was choosing e^2. But when valuation of agent 1 changed to v'_1, it chose e^0 at valuation profile (v'_1, v_2). Hence, agent 1 could change the outcome without changing his own outcome. As we show next, such allocation rules are incompatible with rationalizability.

2.3.2 Non-bossy Single Object Auctions

In this section, we will show that the set of implementable and *non-bossy* allocation rules are characterized by strongly rationalizable allocation rules.

Definition **4** *An allocation rule f is* **non-bossy** *if for every $i \in N$, for every $v_{-i} \in V_{-i}$ and for every $v_i, v'_i \in V_i$ with $f_i(v_i, v_{-i}) = f_i(v'_i, v_{-i})$, we have $f(v_i, v_{-i}) = f(v'_i, v_{-i})$.*

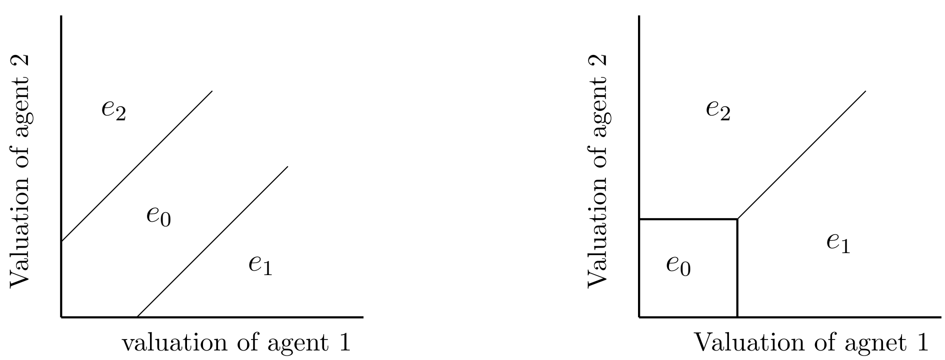
Non-bossiness requires that if an agent does not change his own allocation (i.e., whether he is getting the object or not) by changing his valuation, then he should not be able to change the allocation of anyone. It was first proposed by Satterthwaite and Sonnenschein (1981). As discussed in the introduction, it is a plausible condition to impose in private good allocation problems and has been extensively used in the strategic social choice theory literature.

The notion of non-bossiness that we use is a non-standard version. The standard version of non-bossiness in our setting will translate to a condition on *mechanism* not on allocation rule. We call this version of non-bossiness as *utility non-bossiness*. In particular, an incentive compatible mechanism (f, p) satisfies **utility non-bossiness** if for every $i \in N$, for every v_{-i}, and for every $v_i, v'_i \in V_i$, such that $v_i f_i(v_i, v_{-i}) - p_i(v_i, v_{-i}) = v'_i f_i(v'_i, v_{-i}) - p_i(v'_i, v_{-i})$, we have $v_j f_j(v_i, v_{-i}) - p_j(v_i, v_{-i}) = v_j f_j(v'_i, v_{-i}) - p_j(v'_i, v_{-i})$ for all $j \in N$. In words, if an agent changes his valuation such that his net utility does not change, then the net utility of every agent must remain unchanged.

We do not impose such utility non-bossiness because this is a condition on *mechanisms*, and we are interested in a condition on *allocation rules*. Further, utility non-bossiness is not satisfied by many canonical mechanisms. For instance, the second-price Vickrey auction is not utility non-bossy. To see this, consider an example with two agents with valuations 10 and 7 respectively. Note that the allocation rule in a second-price Vickrey auction is an efficient allocation rule. The net utilities of agents 1 and 2 in the second-price Vickrey auction are 3 and 0 respectively. Now, consider the valuation profile $(10, 8)$. At this valuation profile, agent 2 continues to get zero net utility in the second price Vickrey auction, but the net utility of agent 1 is reduced to 2. This shows that the second-price Vickrey auction is not utility non-bossy. On the other hand, the efficient allocation rule with appropriate tie breaking is a non-bossy allocation rule.

In a recent paper, Thomson (2014) discusses the various versions of non-bossiness and he named our version of non-bossiness **subspace non-bossiness**. He also illustrated the difference between subspace non-bossiness and standard definition of non-bossiness.

We give an example of a bossy and a non-bossy allocation rule in Figure 2.2(a) and Figure 2.2(b) respectively. These figures indicate a scenario with two agents. The possible outcomes of the allocation rules at different valuation profiles are depicted in the Figures. In Figure 2.2(a), the allocation rule is bossy since if we start from a region where alternative e^2 is chosen and agent 1 increases his value, then we can come to a region where alternative e^0 is chosen (i.e., agent 1 can change the outcome without changing his own outcome). However, such a problem is absent for the allocation rule in Figure 2.2(b).



(a) (b)

Figure 2.2: Bossy and non-bossy allocation rules

Lemma **2** *A strongly rationalizable allocation rule is non-bossy.*

Proof: Let f be a strongly rationalizable allocation rule with $\succ$ being the corresponding ordering on D. Fix an agent i and $v_{-i} \in V_{-i}$. Consider $v_i, v'_i \in V_i$ such that $f(v_i, v_{-i}) = e^j \neq$

e^i. By definition, $\mathbf{1}_{v_j} \succ \mathbf{1}_{v_k}$ for all $k \in (N \cup \{0\}) \setminus \{j\}$. Suppose $f(v'_i, v_{-i}) = e^l \neq e^i$. By definition, $\mathbf{1}_{v_l} \succ \mathbf{1}_{v_k}$ for all $k \in (N \cup \{0\}) \setminus \{l\}$. Assume for contradiction $e^l \neq e^j$. Then, we get that $\mathbf{1}_{v_j} \succ \mathbf{1}_{v_l}$ and $\mathbf{1}_{v_l} \succ \mathbf{1}_{v_j}$, which is a contradiction. ■

This leads to the formal connection between implementability and rationalizability.

THEOREM 1 *An allocation rule is implementable and non-bossy if and only if it is strongly rationalizable.*

The proof of Theorem 1 is in the appendix. Theorem 1 reveals a surprising connection between rationalizability and single object auction design. Such a connection of rationalizability and mechanism design was first established in Mishra and Sen (2012). They consider general quasi-linear environments with private values. They show that if the type space is a *multidimensional open interval*, then every implementable and *neutral* allocation rule is rationalizable. Note that rationalizability is weaker than strong rationalizability in the sense that it does not require the underlying ordering to be a strict linear ordering. Our results depart from those in Mishra and Sen (2012) in many ways. First, as discussed earlier, their domain condition is not satisfied in our model, and neutrality is vacuous in the single object auction models. Second, we show that implementability and non-bossiness is *equivalent to strong rationalizability*. Mishra and Sen (2012) do not provide any such equivalence. Indeed, the non-bossiness that we use, is a condition that is specific to private good allocation problems, and cannot be used in general mechanism design problems.

Notice that Theorem 1 does not require any restriction on V_i. If the strict linear ordering we constructed in the proof of Theorem 1 can be represented using a utility function, then the characterization will be even more direct. If for every agent $i \in N$, V_i is finite, then it is possible. But, as the next example illustrates, this is not always possible.

EXAMPLE 3

Suppose $N = \{1, 2\}$ and $V_1 = V_2 = \mathbb{R}_{++}$. Consider the allocation rule f such that for all valuation profiles (v_1, v_2), $f(v_1, v_2) = e^1$ if $v_1 \geq 1$, $f(v_1, v_2) = e^2$ if $v_1 < 1$ and $v_2 \geq 1$, and $f(v_1, v_2) = e^0$ otherwise. It can be verified that f is implementable (monotone) and non-bossy. By Theorem 1, f is strongly rationalizable. Now, consider the strict linear order defined in the proof of Theorem 1 that strongly rationalizes f - denote it by $\succ^f$. If $v_1 = v_2 = 1$, we have $f(v_1, v_2) = e^1$. Hence, $\mathbf{1}_{v_1} \succ^f \mathbf{1}_{v_2}$.

Now, consider the following definition.

DEFINITION 5 *An ordering $\succeq$ on the set D is separable if there exists a countable set $Z \subseteq D$ such that for every $x, y \in D$ with $x \succ y$, there exists $z \in Z$ such that $x \succeq z \succeq y$.*

It is well known that an ordering on D has a utility representation if and only if it is separable - the result goes back to at least Debreu (1954) (see also Fishburn (1970) for details). We show that $\succ^f$ is not separable. Consider $v_1 = v_2 = 1$. By definition of f,

$\mathbf{1}_{v_1} \succ^f \mathbf{1}_{v_2} \succ^f \mathbf{1}_{v_0}$. Note that since $\succ^f$ is monotone, any utility vector between $\mathbf{1}_{v_1}$ and $\mathbf{1}_{v_2}$ (according to $\succ^f$) will be of the form $\mathbf{1}_{v_2+\epsilon}$ or $\mathbf{1}_{v_1-\epsilon}$ for some $\epsilon > 0$. But, $f(v_1, v_2+\epsilon) = e^2$ implies that $\mathbf{1}_{v_2+\epsilon} \succ^f \mathbf{1}_{v_1}$ for all $\epsilon > 0$. Also, $f(v_1 - \epsilon, v_2) = e^2$ implies that $\mathbf{1}_{v_2} \succ^f \mathbf{1}_{v_1-\epsilon}$ for all $\epsilon > 0$. Hence, there cannot exist $z \in D$ such that $\mathbf{1}_{v_1} \succ^f z \succ^f \mathbf{1}_{v_2}$.

2.3.3 Simple Utility Maximization

We saw that the strict linear ordering that strongly rationalizes an allocation rule may not have a utility representation. The aim of this section is to explore minimal conditions that allow us to define a new ordering for any implementable and non-bossy allocation rule which has a utility representation. This allows us to sharpen our characterization, and relate it to a seminal result of Border (1991). Our extra condition is a *continuity* condition.

Definition 6 *An allocation rule f satisfies* **Condition $\mathcal{C}^*$** *if for every $i, j \in N$ ($i \neq j$) and for every v_{-ij}, for every $\epsilon > 0$, there exists a $\delta_{\epsilon, v_{-ij}} > 0$ such that for every v_i, v_j with $f(v_i, v_j, v_{-ij}) = e^i$, we have $f(v_i + \epsilon, v_j + \delta_{\epsilon, v_{-ij}}, v_{-ij}) = e^i$.*

Condition $\mathcal{C}^*$ requires some version of continuity of the allocation rule. It says that if some agent i is winning the object at a valuation profile, for every increase in value of agent i, there exists some increase in value of agent j such that agent i continues to win the object. Later, we provide an example to show that Condition $\mathcal{C}^*$ and non-bossiness do not imply implementability.

If f is monotone (implementable) and non-bossy, then Condition $\mathcal{C}^*$ implies that for every $i, j \in N$ ($i \neq j$) and for every v_{-ij}, for every $\epsilon > 0$, there exists a $\delta_{\epsilon, v_{-ij}} > 0$ such that for every v_i, v_j with $f(v_i, v_j, v_{-ij}) = e^i$, we have $f(v_i + \epsilon, v_j + \delta, v_{-ij}) = e^i$ for all $0 < \delta < \delta_{\epsilon, v_{-ij}}$. To see this, choose some $\delta \in (0, \delta_{\epsilon, v_{-ij}})$ and assume for contradiction, $f(v_i+\epsilon, v_j+\delta, v_{-ij}) = e^k$ for some $k \neq i$. If $k = j$, then by monotonicity, $f(v_i + \epsilon, v_j + \delta_{\epsilon, v_{-ij}}, v_{-ij}) = e^j$, which is a contradiction to Condition $\mathcal{C}^*$. If $k \neq \{i, j\}$, then by non-bossiness, $f(v_i+\epsilon, v_j+\delta_{\epsilon, v_{-ij}}, v_{-ij}) \in \{e^j, e^k\}$, again a contradiction to Condition $\mathcal{C}^*$. Since we will use Condition $\mathcal{C}^*$ along with implementability and non-bossiness, we can freely make use of this implication.

We will now introduce a new class of allocation rules.

Definition 7 *An allocation rule f is a* **simple utility maximizer (SUM)** *if there exists a non-decreasing function $U_i : V_i \to \mathbb{R}$ for every $i \in N \cup \{0\}$, where $U_0(0) = 0$, such that for every valuation profile $v \in V$, $f(v) = e^j$ implies that $j \in \arg\max_{i \in N \cup \{0\}} U_i(v_i)$.*

Notice that an SUM allocation rule is simpler to state and, hence, more suitable for practical use than a strongly rationalizable allocation rule. The aim of this section is to show that the SUM allocation rules are not much different from the strongly rationalizable allocation rules.

It can be easily seen that not every SUM allocation rule is non-bossy. For instance, consider the efficient allocation rule that allocates the good to an agent with the highest value. Suppose there are three agents with valuations $10, 10, 8$ respectively and suppose that the efficient allocation rule allocates the object to agent 1. Consider the valuation profile $(10, 10, 9)$ and suppose that the efficient allocation rule now allocates the object to agent 2. This violates non-bossiness. As we will show that such violations can happen in case of ties (as was the case here with ties between agents 1 and 2), and when ties are broken carefully, an SUM allocation rule becomes non-bossy.

Similarly, not every SUM allocation rule is implementable. For instance, consider an example with two agents $\{1, 2\}$ with $V_1 = V_2 = \mathbb{R}_{++}$. Let $U_1(v_1) = 1$ and $U_1(v_2) = v_2$. Now, suppose we pick agent 1 as the winner of the object at valuation profile $(1, 1)$ but pick agent 2 as the winner of the object at valuation profile $(2, 1)$. Note that this is consistent with simple utility maximization but violates monotonicity, and hence, not implementable.

Now, consider the following modification of the SUM allocation rule.

Definition 8 *An allocation rule f is a* **simple utility maximizer (SUM) with order-based tie-breaking** *if there exists a non-decreasing function $U_i : V_i \to \mathbb{R}$ for every $i \in N \cup \{0\}$, where $U_0(0) = 0$, and a monotone strict linear ordering $\succ$ on D such that for every valuation profile $v \in V$, $f(v) = e^j$ implies that $j \in \arg\max_{i \in N \cup \{0\}} U_i(v_i)$ and $\mathbf{1}_{v_j} \succ \mathbf{1}_{v_k}$ for all $k \neq j$ and $k \in \arg\max_{i \in N \cup \{0\}} U_i(v_i)$, i.e., $\mathbf{1}_{v_j}$ is the unique simple utility maximizer according to $\succ$.*

The tie-breaking rule that we specified is very general. It covers some intuitive tie-breaking rules such as having an ordering over $N \cup \{0\}$ and breaking the tie in simple utility maximization using this ordering.

Lemma 3 *An SUM allocation rule with order-based tie-breaking is implementable.*

Proof: Suppose f is an SUM allocation rule with order-based tie-breaking. Let the corresponding simple utility functions be $U_0, U_1, \ldots, U_n$ and $\succ$ be the ordering used to break ties. At any valuation profile v, let

$$W(v) = \{j \in N \cup \{0\} : U_j(v_j) \geq U_k(v_k) \ \forall \ k \in N \cup \{0\}\}.$$

Fix an agent i and the valuation profile of other agents at v_{-i}. Consider v_i, v'_i such that $v_i < v'_i$ and $f(v_i, v_{-i}) = e^i$. Then, by SUM maximization, $i \in W(v_i, v_{-i})$. Further, by order-based tie-breaking $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$ for all $j \in W(v_i, v_{-i})$. Since U_i is non-decreasing, $U_i(v'_i) \geq U_j(v_j)$ for all $j \in (N \cup \{0\}) \setminus \{i\}$. Hence, $i \in W(v'_i, v_{-i})$. Again, by order-based tie-breaking, $\mathbf{1}_{v'_i} \succ \mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$ for all $j \in W(v'_i, v_{-i})$. This implies that $f(v'_i, v_{-i}) = e^i$. So, f is monotone, and hence, implementable. ■

An SUM allocation rule with order-based tie-breaking is also non-bossy.

Lemma 4 *An SUM allocation rule with order-based tie-breaking is non-bossy.*

Proof: Let f be an SUM allocation rule with order-based tie-breaking and v be a valuation profile such that $f(v) \neq e^j$ for some $j \in N$. Suppose $f(v'_j, v_{-j}) \neq e^j$. Then, by definition, the unique simple utility maximizer of f remains the same in (v_j, v_{-j}) and (v'_j, v_{-j}). So, $f(v_j, v_{-j}) = f(v'_j, v_{-j})$, and hence, f is non-bossy. ■

We are now ready to state the main result of this section.

THEOREM **2** *Suppose $V_i = (0, \beta_i)$, where $\beta_i \in \mathbb{R}_{++} \cup \{\infty\}$, for all $i \in N$ and f is an allocation rule satisfying Condition $\mathcal{C}^*$. Then, the following statements are equivalent.*

1. *f is an implementable and non-bossy allocation rule.*
2. *f is a simple utility maximizer allocation rule with order-based tie-breaking.*

The proof of Theorem 2 is given in the Appendix. The non-trivial part of the proof is to establish that under Condition $\mathcal{C}^*$, implementability and non-bossiness imply simple utility maximization. This part of the proof is long and tedious, but reveals beautiful structure of implementable and non-bossy allocation rules. Once this is established, we use Theorem 1 to conclude how the ties must be broken. As we discussed earlier, the strict linear ordering induced by an implementable and non-bossy allocation rule on the set of utility vectors D may not have a utility representation. Hence, we cannot invoke Theorem 1 directly to show Theorem 2. The proof of Theorem 2 constructs another ordering (which is not a linear order) and shows that this has a utility representation under Condition $\mathcal{C}^*$. We provide some remarks on Theorem 2 below.

SOME SIMPLE UTILITY MAXIMIZERS. An efficient allocation rule is also an SUM allocation rule, where $U_i(v_i) = v_i$ for all $i \in N$ and for all $v_i \in V_i$. Similarly, we can define for every $i \in N$ and for every $v_i \in V_i$, $U_i(v_i) = \lambda_i v_i + \kappa_i$ for some $\lambda_i \geq 0$ and $\kappa_i \in \mathbb{R}$, and this SUM will correspond to the affine maximizer allocation rules of Roberts (1979). The simple utility function in Myerson (1981) takes the form $U_i(v_i) = v_i - \frac{1-F_i(v_i)}{f_i(v_i)}$, where F_i and f_i are respectively the cumulative density function and density function of the distribution of valuation of agent i.

PAYMENTS. It is well known that revenue equivalence (Myerson, 1981) implies that for any implementable allocation rule, the payments are determined uniquely up to an additive constant. Suppose V_i is an interval for all $i \in N$. For any implementable allocation rule f, define the cutoff for agent i and valuation profile v_{-i} as $\kappa_i^f(v_{-i}) = \inf\{\alpha \in V_i : f(\alpha, v_{-i}) = e^i\}$, where $\kappa_i^f(v_{-i}) = 0$ if $f(\alpha, v_{-i}) \neq e^i$ for all $\alpha \in V_i$. It is well known that for every $i \in N$ and for every $(v_i, v_{-i}) \in V$, $p_i^f(v_i, v_{-i}) = \kappa_i^f(v_{-i})$ if $f(v_i, v_{-i}) = e^i$ and $p_i^f(v_i, v_{-i}) = 0$ if $f(v_i, v_{-i}) \neq e^i$ is a payment rule which implements f. Further, by revenue equivalence, any payment rule p which implements f must satisfy for every $i \in N$ and for every (v_i, v_{-i}), $p_i(v_i, v_{-i}) = p_i^f(v_i, v_{-i}) + h_i(v_{-i})$, where $h_i : V_{-i} \to \mathbb{R}$ is any function. Thus, by characterizing implementable allocation rules, we characterize the class of dominant strategy incentive compatible mechanisms.

CONDITION $\mathcal{C}^*$. We give an example of an allocation rule which is non-bossy and satisfies Condition $\mathcal{C}^*$ but not implementable. The example illustrates that Condition $\mathcal{C}^*$ and non-bossiness do not make implementability a redundant condition. In other words, these two conditions together are not stronger than monotonicity.

EXAMPLE 4

Let $N = \{1, 2\}$. Suppose $V_1 = V_2 = \mathbb{R}_{++}$. Let $U_1(v_1) = -v_1$ and $U_2(v_2) = -v_2$. The allocation rule f is defined as follows. It chooses e^0 (not allocating the object) if $U_1(v_1)$ and $U_2(v_2)$ are less than -1. Else, it allocates the object to the agent with the highest $U_i(v_i)$, breaking ties in favor of agent 1.

Formally, if $\max(U_1(v_1), U_2(v_2)) \leq -1$, then $f(v_1, v_2) = e^0$. Else, if $U_1(v_1) \geq \max(U_2(v_2), -1)$, then $f(v_1, v_2) = e^1$ and if $U_2(v_2) > U_1(v_1)$ and $U_2(v_2) \geq -1$, then $f(v_1, v_2) = e^2$. Clearly, this allocation rule is not monotone, and hence, not implementable. However, it is non-bossy and satisfies Condition $\mathcal{C}^*$.

2.3.4 Randomization and Bayesian Implementation via Border's Hierarchical Allocation Rules

We relate our results to Border's *hierarchical allocation rules* (Border, 1991). [7] Border considered allocation rules which are not necessarily deterministic and Bayesian implementable. To describe his results, we consider randomized allocation rules in this section. A randomized allocation rule is a map $f : V \to \Delta A$, where ΔA denotes the convex hull of the $(n + 1)$ vectors $\{e^0, e^1, \dots, e^n\}$ in $\mathbb{R}^n$. Hence, $f_i(v)$ will now denote the probability of agent i getting the object at valuation profile v. Border (1991) considers independent private values setting. Each bidder i has a probability distribution G_i using which it draws its value from V_i. Denote by $G_{-i}(v_{-i}) \equiv \times_{j \neq i} G_j(v_j)$. The interim allocation probability of an allocation rule f for agent i is

$$a_i^f(v_i) = \int_{V_{-i}} f_i(v_i, v_{-i}) dG_{-i}(v_{-i}).$$

Border also considers Bayesian implementation. An allocation rule f is Bayesian implementable if there exists a payment rules $(p_1, \dots, p_n)$ such that for every $i \in N$, for every $v_i, v_i' \in V_i$

$$v_i a_i^f(v_i) - \int_{V_{-i}} p_i(v_i, v_{-i}) dG_{-i}(v_{-i}) \geq v_i a_i^f(v_i') - \int_{V_{-i}} p_i(v_i', v_{-i}) dG_{-i}(v_{-i}).$$

[7] I am grateful to Mallesh Pai for motivating the contents of this section.

Definition 9 *An allocation rule f^h is a* **hierarchical allocation rule** *if there exists non-decreasing functions $I_i : V_i \to \mathbb{R}$ for all $i \in N$ such that at every valuation profile $v \in V$*

$$f_i^h(v) = \begin{cases} \frac{1}{|\{j \in N : I_i(v_i) = I_j(v_j)\}|} & \text{if } I_i(v_i) \geq 0 \text{ and } I_i(v_i) \geq I_j(v_j) \text{ for all } j \in N \\ 0 & \text{otherwise} \end{cases}$$

In a seminal result, Border showed that for every Bayesian implementable allocation rule f, there exist a set of hierarchical allocation rules whose randomization gives the same interim allocation probability as f - see also Manelli and Vincent (2010); Mierendorffl (2011); Deb and Pai (2013). [8]

Now, notice that a hierarchical allocation rule is a randomization over simple utility maximizers (which are deterministic allocation rules). To see this, we define $(n+1)!$ order based tie-breaking rules. Take any strict linear ordering P of the set of alternatives in A. Define an ordering $\succ$ on the set of utility vectors D as follows. For any $i \in N$, if $\mathbf{1}_{v_i}, \mathbf{1}_{v'_i} \in D_i$ with $v_i > v'_i$, then $\mathbf{1}_{v_i} \succ \mathbf{1}_{v'_i}$. If $e^i P e^j$, then for every $\mathbf{1}_{v_i} \in D_i$ and every $\mathbf{1}_{v_j} \in D_j$, $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$. Note that $\succ$ can be defined exactly $(n+1)!$ ways, one for each P. Let $\mathcal{P}$ be the set of all such orderings of D. Now, given a hierarchical allocation rule with $(I_1, \ldots, I_n)$, we can construct $(n+1)!$ simple utility maximizers with $U_i = I_i$ for all $i \in N$ and taking as tie-breaking rule one of the orderings in $\mathcal{P}$. Clearly, uniform randomization over these simple utility maximizers produce the hierarchical allocation rule. Hence, randomization over the hierarchical allocation rules is equivalent to randomization over simple utility maximizers.

Thus, simple utility maximizers occupy a central role in the theory of private value single object auctions. By characterizing simple utility maximizers, Theorem 2 indirectly provides an axiomatic foundation for Border's hierarchical allocation rules. In particular, the interim allocation probability of any implementable allocation rule can be obtained by randomizing over the set of implementable and non-bossy allocation rules satisfying Condition $\mathcal{C}^*$.

2.3.5 Extension of Roberts' Theorem

Consider a general mechanism design set up with private values and quasi-linear utility. Let A be a finite set of alternatives. Suppose $|A| \geq 3$. The type of agent i is denoted as $v_i \in \mathbb{R}^{|A|}$ and $v_i(a)$ denotes the valuation of agent i for alternative a. Roberts (1979) shows that if type space of every agent is $\mathbb{R}^{|A|}$, then for every onto and implementable allocation rule f, there exists $\lambda_1, \ldots, \lambda_n \geq 0$, not all of them equal to zero, and $\kappa : A \to \mathbb{R}$ such that at every valuation profile v,

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

Such allocation rules are called affine maximizer allocation rules. Theorems 1 and 2 can be thought of as the analogue of Roberts' affine maximizer theorem in the single object auction

[8] Although Border (1991) does not consider incentive constraints, it is clear how his results can be modified in the presence of incentive constraints.

model (under non-bossiness). It shows how much the set of implementable allocation rule expands in a restricted domain like the single object auction domain.

2.4 The Complete Characterization

Theorems 1 and 2 characterize implementable allocation rules under additional assumptions. In this section, we drop these additional assumptions and provide a complete characterization of implementable allocation rules. These characterizations are in the spirit of extending the Roberts' affine maximizer theorem. In particular, we show that an implementable allocation rule is equivalent to a *generalized utility maximizer* allocation rule.

A **generalized utility function (GUF)** of agent $i \in N$ is a function $u_i : V \to \mathbb{R}$. Notice that the generalized utility of an agent may be negative also. Further, a simple utility function is a GUF. We will need the following version of single crossing property.

Definition 10 *The GUFs* $(u_1, \ldots, u_n)$ *satisfy* **top single crossing** *if for every* $i \in N$*, for every* $v_{-i} \in V_{-i}$*, and for every* $v_i, v'_i \in V_i$ *with* $v_i > v'_i$ *and* $u_i(v'_i, v_{-i}) \geq \max(0, \max_{k \in N \setminus \{i\}} u_k(v'_i, v_{-i}))$*, we have* $u_i(v_i, v_{-i}) > \max(0, \max_{k \in N \setminus \{i\}} u_k(v_i, v_{-i}))$*.*

The top single crossing condition is a very general inter-agent crossing condition. Such crossing conditions are extensively used in the literature of interdependent value auctions - see for instance, Cremer and McLean (1985); Maskin (1992); Dasgupta and Maskin (2000); Perry and Reny (2002). For the finite type space, Cremer and McLean (1985) use conditions similar to our top single crossing to establish implementation (in ex post equilibrium) of the efficient allocation rule in the interdependent values model.

The standard definition of a "single crossing" property, which implies top single crossing, is the following.

Definition 11 *GUFs* $(u_1, \ldots, u_n)$ *satisfy* **single crossing** *if for every* $i, j \in N$*, for every* $v_{-i} \in V_{-i}$*, for every* $v'_i, v_i \in V_i$ *with* $v_i > v'_i$*, we have* $u_i(v_i, v_{-i}) - u_i(v'_i, v_{-i}) > u_j(v_i, v_{-i}) - u_j(v'_i, v_{-i})$*.*

A GUF u_i is **increasing** if for every $v_{-i} \in V_{-i}$ and for every $v_i, v'_i \in V_i$ with $v_i > v'_i$ we have $u_i(v_i, v_{-i}) > u_i(v'_i, v_{-i})$.

Lemma 5 *If GUFs* $(u_1, \ldots, u_n)$ *satisfy single crossing and* u_i *is increasing for every* $i \in N$*, then they satisfy top single crossing.*

Proof: Consider $i \in N$ and $v_{-i} \in V_{-i}$. Let $v_i, v'_i \in V_i$ such that $v_i > v'_i$ and $u_i(v'_i, v_{-i}) \geq \max(0, \max_{k \in N \setminus \{i\}} u_k(v'_i, v_{-i}))$. Since u_i is increasing, $u_i(v_i, v_{-i}) > u_i(v'_i, v_{-i}) \geq 0$. Further, by single crossing, $u_i(v_i, v_{-i}) - u_i(v'_i, v_{-i}) > u_j(v_i, v_{-i}) - u_j(v'_i, v_{-i})$ for all $j \neq i$. Using the fact that $u_i(v'_i, v_{-i}) \geq u_j(v'_i, v_{-i})$ for all $j \neq i$, we get that $u_i(v_i, v_{-i}) > u_j(v_i, v_{-i})$ for all $j \neq i$. Hence, $u_i(v_i, v_{-i}) > \max(0, \max_{k \in N \setminus \{i\}} u_k(v_i, v_{-i}))$. ■

We are now ready to introduce a new class of implementable allocation rules.

Definition 12 *An allocation rule f is a* **generalized utility maximizer** *if there exist GUFs $(u_1, \ldots, u_n)$ satisfying top single crossing such that for every $v \in V$, $f(v) = e^i$ implies that $i \in \arg\max_{i \in N \cup \{0\}} u_i(v)$, where $u_0(v) = 0$.*

Generalized utility maximizers are implementable. The proof is similar to the proof in Cremer and McLean (1985), who establish implementation (in ex post equilibrium) of efficient allocation rule in an interdependent values model.

Lemma 6 *If f is a generalized utility maximizer, then it is implementable.*

Proof: Fix a generalized utility maximizer f, and let $(u_1, \ldots, u_n)$ be the corresponding GUFs satisfying top single crossing. Consider agent i and $v_{-i} \in V_{-i}$. Also, consider any $v_i, v_i' \in V_i$ with $v_i > v_i'$ and $f(v_i', v_{-i}) = e^i$. By definition, $u_i(v_i', v_{-i}) \geq \max(0, \max_{k \in N \setminus \{i\}} u_k(v_i', v_{-i}))$. By top single crossing, $u_i(v_i, v_{-i}) > \max(0, \max_{k \in N \setminus \{i\}} u_k(v_i, v_{-i}))$. Hence, $f(v_i, v_{-i}) = e^i$. So, f is monotone, and hence, implementable. ■

This leads to the main result of this section.

Theorem 3 *Suppose $V_i \subseteq \mathbb{R}_{++}$ is bounded for every $i \in N$. Then, f is implementable if and only if it is a generalized utility maximizer.*

Proof: Lemma 6 showed that every GUF maximizer is implementable. Now, for the converse, suppose f is implementable. Fix an agent $i \in N$ and $v_{-i} \in V_{-i}$. If $f(v_i, v_{-i}) \neq e^i$ for all $v_i \in V_i$, then define $\kappa_i^f(v_{-i}) = \sup\{v_i : v_i \in V_i\}$. Else, define $\kappa_i^f(v_{-i}) = \inf\{v_i \in V_i : f(v_i, v_{-i}) = e^i\}$. Since V_i is bounded, $\kappa_i^f(v_{-i})$ is well defined. Further, since f is monotone, for every agent $i \in N$, for every v_{-i}, and for every $v_i \in V_i$, if $v_i > \kappa_i^f(v_{-i})$, we have $f(v_i, v_{-i}) = e^i$ and for every $v_i < \kappa_i^f(v_{-i})$ we have $f(v_i, v_{-i}) \neq e^i$. Define for every $i \in N$ and for every (v_i, v_{-i}), $u_i(v_i, v_{-i}) := v_i - \kappa_i^f(v_{-i})$. By definition, if $f(v) = e^i$, then $v_i - \kappa_i^f(v_{-i}) \geq 0$ and $v_j - \kappa_j^f(v_{-j}) \leq 0$ for all $j \neq i$. Hence, $i \in \arg\max_{k \in N \cup \{0\}} u_k(v)$, where $u_0(v) = 0$.

To show that $(u_1, \ldots, u_n)$ satisfy top single crossing, consider $i \in N$ and $v_{-i} \in V_{-i}$. Let $v_i, v_i' \in V_i$ such that $v_i > v_i'$ and $u_i(v_i', v_{-i}) \geq \max(0, \max_{k \in N \setminus \{i\}} u_k(v_i', v_{-i}))$. Notice that $u_i(v_i, v_{-i}) > u_i(v_i', v_{-i}) \geq 0$. By definition of $u_1, \ldots, u_n$, if $u_i(v_i, v_{-i}) > 0$, then $v_i > \kappa_i^f(v_{-i})$, and hence, $f(v_i, v_{-i}) = e^i$. But, this implies that $u_k(v_i, v_{-i}) = v_k - \kappa_k^f(v_{-k}) \leq 0$ for all $k \neq i$. Hence, $u_i(v_i, v_{-i}) > \max(0, \max_{k \in N \setminus \{i\}} u_k(v_i, v_{-i}))$. ■

Our characterization of implementability shows that implementability is equivalent to maximizing generalized utilities. Generalized utilities transform the original valuation of an agent to a new utility, which depends on the valuations of *all* the agents. In contrast to simple utility functions, generalized utility functions are much harder to construct. This illustrates how a natural axiom like non-bossiness helps to simplify the class of implementable allocation rules.

Generalized utility maximizers are similar to implementing the *efficient* allocation rule in an interdependent values model with the qualification that we allow generalized utilities to be negative, which is precluded in the standard interdependent value model. It is well known that the efficient allocation rule is not generally implementable in the interdependent values single object auction unless some inter agent crossing condition holds (Cremer and McLean, 1985; Maskin, 1992; Dasgupta and Maskin, 2000; Perry and Reny, 2002; Jehiel et al., 2006). Our top single crossing condition is similar to these conditions in the interdependent values literature. Our result reveals a surprising and interesting connection between these seemingly different models.

2.5 Discussions

We conclude by discussing some of the open questions that remain.

Randomization and Bayesian Implementation. Although we focus on deterministic dominant strategy implementation, randomization is a natural extension of our model. Indeed, the monotonicity characterization of Myerson (1981) extends to single object auctions with randomization. Extending characterizations of deterministic allocation rules to randomized allocation rules present several challenges. A natural way to think of randomization is that of domain restriction - the utility from a lottery alternative is restricted to be the expected utility from the deterministic alternatives in its support. Thus, the challenges of going from deterministic to randomized allocation rules is similar to that of going from a larger domain to a restricted domain. For instance, a counterpart of Roberts' seminal result with randomization is still not known in the unrestricted domain.

However, we provided a relationship of our simple utility maximizer and Border's hierarchical allocation rules that can be used to obtain interim allocation probability of every Bayesian and randomized allocation rule. Hence, our characterizations can be used in an indirect way to characterize interim allocation probabilities of Bayesian implementable randomized allocation rules. However, the direct characterization remains an open question.

Optimizing payments. A popular research theme in auction theory and mechanism design is to "optimize" over the set of incentive compatible mechanisms. This usually involves optimizing over payments and assumes some prior distribution over valuations of agents by the mechanism designer. The implications of such optimizations in the single object auctions is fairly well understood.

Clearly, our results do not contribute to this literature. Our characterizations are more tailored towards understanding the inherent structure of deterministic single object auctions in private values set up. They completely describe the set of "options" available to a mechanism designer (without bothering about the distributional assumptions) in the single object auctions. Our main characterizations provide axiomatic foundations to various commonly used auctions.

We also believe that this opens a door for carrying out similar exercises in multidimensional mechanism design models, including the multi-object auction model. The problem of finding an expected revenue maximizing mechanism in such models is considered a difficult problem Hart and Reny (2012); Hart and Nisan (2012). Perhaps, understanding the structure of incentive compatible mechanisms will allow us to simplify these problems.

Appendix: Omitted Proofs

Proof of Theorem 1

By virtue of Lemmas 1 and 2, we only need to show that if an allocation rule f is implementable and non-bossy then it is strongly rationalizable. We do the proof in several steps.

Step 1. For any $i, j \in N \cup \{0\}$ with $i \neq j$, consider $\mathbf{1}_{v_i}$ and $\mathbf{1}_{v_j}$ for some $v_i \in V_i$ and $v_j \in V_j$. Suppose for some v_{-ij}, we have $f(v_i, v_j, v_{-ij}) = e^i$. We will show that if f is non-bossy, then $f(v_i, v_j, v'_{-ij}) \neq e^j$ for all v'_{-ij}. Consider any $k \notin \{i, j\}$ and the profile $(v_i, v_j, v'_k, v_{-ijk})$. By non-bossiness, $f(v_i, v_j, v'_k, v_{-ijk}) \in \{e^i, e^k\}$. Repeating this argument for all $k \notin \{i, j\}$, we get $f(v_i, v_j, v'_{-ij}) \neq e^j$.

Step 2. We will first define a binary relations $\succ$ on $D \times D$ [9] using f as follows. For every $i, j \in N \cup \{0\}$ with $i \neq j$, $\mathbf{1}_{v_i} \in D_i$ and $\mathbf{1}_{v_j} \in D_j$, define

$$\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$$

if there is some v_{-ij} such that $f(v_i, v_j, v_{-ij}) = e^i$. Further, for every $i \in N$ and every $v_i \in V_i$, define

$$\mathbf{1}_{v_i+\epsilon} \succ \mathbf{1}_{v_i}$$

for all $\epsilon > 0$ such that $(v_i + \epsilon) \in V_i$. Using Step 1, if $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$, then $\mathbf{1}_{v_j} \not\succ \mathbf{1}_{v_i}$. Hence, $\succ$ is anti-symmetric. Further, $\succ$ is irreflexive by definition.

Step 3. Let $D^f := \{x \in D : G^f(v) = x \text{ for some } v \in V\}$. We now show that $\succ$ satisfies the following conditions:

1 for every $x, y \in D^f$, either $x \succ y$ or $y \succ x$ (but not both), where $D^f = \{x \in D : G^f(v) = x \text{ for some } v \in V\}$,

2 for every $x \in D^f$ and for every $y \notin D^f$, $x \succ y$,

3 for all $v \in V$, $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$ for all $j \in \{0, 1, \ldots, n\} \setminus \{i\}$, where $G^f(v) = \mathbf{1}_{v_i}$.

[9] To remind, D is the set of all utility vectors given the type space.

- Proof of (1). Pick $x, y \in D^f$. By definition, there is $v \in V$, such that $G^f(v) = x$. If $x = \mathbf{1}_{v_i}$, then $f(v) = e^i$. Suppose $y = \mathbf{1}_{v_i'}$. Then, by definition, either $x \succ y$ or $y \succ x$. Hence, suppose $y = \mathbf{1}_{v_j'}$ for some $j \neq i$. Then, by monotonicity and non-bossiness, $f(v_i, v_j', v_{-ij}) \in \{e^i, e^j\}$. Hence, either $x \succ y$ or $y \succ x$. Since $\succ$ is anti-symmetric, either $x \succ y$ or $y \succ x$ but not both.

- Proof of (2). Pick $x \in D^f$ but $y \notin D^f$. By definition, there is $v \in V$, such that $G^f(v) = x$. If $x = \mathbf{1}_{v_i}$, then $f(v) = e^i$. Suppose $y = \mathbf{1}_{v_i'}$. Then, if $v_i' > v_i$, we have $f(v_i', v_{-i}) = e^i$ by monotonicity, and this contradicts the fact that $y \notin D^f$. Hence, $v_i' < v_i$, and by definition, $x \succ y$.

  Suppose $y = \mathbf{1}_{v_j'}$ for some $j \neq i$. Then, by monotonicity and non-bossiness, $f(v_i, v_j', v_{-ij}) \in \{e^i, e^j\}$. Using the fact that $y \notin D^f$, we get that $f(v_i, v_j', v_{-ij}) = e^i$. Hence, $x \succ y$.

- Proof of (3). At any valuation profile $(v_1, \ldots, v_n)$, if $f(v_1, \ldots, v_n) = e^i$, then, by definition, $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$ for all $j \neq i$.

Step 4. We show that $\succ$ is transitive. Suppose for some $i \in N$, $\mathbf{1}_{v_i+\epsilon} \succ \mathbf{1}_{v_i}$ for some $\epsilon > 0$ such that $v_i + \epsilon \in V_i$. Also, for some $j \neq i$, $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$. Then, by definition, for some v_{-ij}, $f(v_i, v_j, v_{-ij}) = e^i$. By monotonicity, $f(v_i + \epsilon, v_j, v_{-ij}) = e^i$. Hence, $\mathbf{1}_{v_i+\epsilon} \succ \mathbf{1}_{v_j}$.

We also know that for some $i \in N$ and for some $\epsilon > 0, \delta > 0$, if $\mathbf{1}_{v_i+\epsilon+\delta} \succ \mathbf{1}_{v_i+\epsilon}$ and $\mathbf{1}_{v_i+\epsilon} \succ \mathbf{1}_{v_i}$, then $\mathbf{1}_{v_i+\epsilon+\delta} \succ \mathbf{1}_{v_i}$.

Finally, pick $v_i \in V_i, v_j \in V_j$ and $v_k \in V_k$ such that $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_j}$ and $\mathbf{1}_{v_j} \succ \mathbf{1}_{v_k}$, where i, j, k are distinct. This means, $f(v_i, v_j, v_{-ij}') = e^i$ for some v_{-ij}'. By monotonicity and non-bossiness, $f(v_i, v_j, v_k, v_{-ijk}') \in \{e^i, e^k\}$. But, $\mathbf{1}_{v_j} \succ \mathbf{1}_{v_k}$ implies that $f(v_i, v_j, v_k, v_{-ijk}') \neq e^k$. Hence, $f(v_i, v_j, v_k, v_{-ijk}') = e^i$. Hence, $\mathbf{1}_{v_i} \succ \mathbf{1}_{v_k}$. This shows that $\succ$ is transitive.

Step 5. We show that f is strongly rationalizable. Since $\succ$ is an anti-symmetric, irreflexive and transitive binary relation on $D \times D$, we can extend it to an anti-symmetric, irreflexive, complete, and transitive binary relation $\succ'$ on $D \times D$ due to Szpilrajn's extension theorem - see Fishburn (1970) for instance. By definition of $\succ'$ and Step 3, at any valuation profile $(v_1, \ldots, v_n)$, if $f(v_1, \ldots, v_n) = e^i$, then, $\mathbf{1}_{v_i} \succ' \mathbf{1}_{v_j}$ for all $j \neq i$. By definition, $\succ'$ is monotone. Hence, f is strongly rationalizable.

Proof of Theorem 2

By Lemmas 3 and 4, an SUM allocation rule with order-based tie-breaking is implementable and non-bossy. We show that every implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$ is an SUM allocation rule with order-based tie-breaking. We do the proof in various steps. Throughout we assume that $V_i = (0, \beta_i)$, where $\beta_i \in \mathbb{R}_{++} \cup \{\infty\}$, for all

$i \in N$.

Step 1. In this step, we show that if f is implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$, then there is an ordering $\succeq^f$ on D which rationalizes f. We construct this specific $\succeq^f$ in this step. [10]

Suppose f is an implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$. We first define the notion of a **winning set**. The winning set of allocation rule f at a valuation profile v is denoted as $W^f(v)$, and defined as follows. For any $i \in N$, we say $e^i \in W^f(v)$ if for all $\epsilon > 0$, we have $f(v_i + \epsilon, v_{-i}) = e^i$, where $(v_i + \epsilon) \in V_i$. We say that $e^0 \in W^f(v)$ if for all $\epsilon > 0$, we have $f(\{v_j - \epsilon\}_{j \in N}) = e^0$, where $(v_j - \epsilon) \in V_j$ for all $j \in N$. The first claim is that $W^f(v)$ is non-empty for all valuation profiles v.

Lemma 7 *If f is implementable and non-bossy, then for every value profile v, $f(v) \in W^f(v)$.*

Proof: Consider an implementable and non-bossy allocation rule f and a valuation profile v. If $f(v) = e^j \neq e^0$, then by monotonicity $f(v_j + \epsilon, v_{-j}) = e^j$ for all $\epsilon > 0$. Hence, $f(v) \in W^f(v)$.

If $f(v) = e^0$, then consider any $\epsilon > 0$ and a valuation profile v' such that $v'_i - \epsilon > 0$ for all $i \in N$. We argue that $f(v') = e^0$, and hence, $e^0 = f(v) \in W^f(v)$. Assume for contradiction that $f(v') = e^j \neq e^0$. Now, we go from v' to v by increasing the valuation of one agent at a time. By monotonicity, $f(v_j, v'_{-j}) = e^j$. Now, pick any $k \in N \setminus \{j\}$. Then, either $f(v_j, v_k, v'_{-jk}) = e^k$ or by non-bossiness $f(v_j, v_k, v'_{-jk}) = e^j$. In both cases, we see that $f(v_j, v_k, v'_{-jk}) \neq e^0$. Continuing in this manner, we will reach the valuation profile v and get that $f(v) \neq e^0$, a contradiction. ■

Step 1.1. In this step, we show that an implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$ satisfies a form of independence property.

Definition 13 *An allocation rule f satisfies* **binary independence** *if for any pair of alternatives $e^j, e^k \in A$ and any pair of valuation profiles v, v' such that $\mathbf{1}_{v_j} = \mathbf{1}_{v'_j}$ and $\mathbf{1}_{v_k} = \mathbf{1}_{v'_k}$, the following conditions hold.*

1. *if $e^k \in W^f(v)$ and $e^j \in W^f(v')$, then $e^k \in W^f(v')$,*
2. *if $e^j \in W^f(v)$ and $e^k \notin W^f(v)$, then $e^k \notin W^f(v')$.*

Intuitively, the binary independence property says that the comparison of any pair of utility vectors is independent of what the other utility vectors are.

[10] Notice that by Theorem 1, if f is implementable and non-bossy, then it is a strongly rationalizable allocation rule, and hence, a rationalizable allocation rule. The novelty of this step of the proof is to be able to construct a *specific* ordering which rationalizes f.

Proposition 1 *An implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$ satisfies binary independence.*

Proof: We will use the following lemma to prove the proposition.

Lemma 8 *Suppose v and v' are two distinct valuation profiles such that $v_i \geq v'_i$ for all $i \in N$. Let $B(v, v') = \{e^i \in A : v_i > v'_i\}$. If f is an implementable and non-bossy allocation rule, then $W^f(v) \setminus B(v, v') \subseteq W^f(v')$.*

Proof: Let f be an implementable and non-bossy allocation rule and v and v' be two distinct valuation profiles with $v_i \geq v'_i$ for all $i \in N$. We will go from v to v' by lowering one agent's value at a time. Pick any $e^j \in B(v, v')$. Consider a new type profile v'' such that the value of every agent $i \neq j$ remains v_i and the value of agent j is v'_j, which is strictly less than v_j. Pick any $e^k \in W^f(v)$ such that $e^k \neq e^j$. Then, we consider two cases.

Case 1: $e^k \neq e^0$. We argue that $e^k \in W^f(v'')$. Assume for contradiction that $e^k \notin W^f(v'')$. Then, for some $\epsilon > 0$, we have $f(v_k + \epsilon, v'_j, v_{-kj}) \neq e^k$. If $f(v_k + \epsilon, v'_j, v_{-kj}) = e^j$, then by monotonicity, we have $f(v_k + \epsilon, v_j, v_{-kj}) = e^j$. This is a contradiction since $e^k \in W^f(v)$. If $f(v_k + \epsilon, v'_j, v_{-kj}) = e^l \notin \{e^j, e^k\}$, then monotonicity and non-bossiness implies that $f(v_k + \epsilon, v_j, v_{-kj}) \in \{e^l, e^j\}$. But this contradicts $e^k \in W^f(v)$.

Case 2: $e^k = e^0$. Since $e^0 \in W^f(v)$, for any $\epsilon > 0$ such that $\bar{v}_i := v_i - \epsilon > 0$ for all $i \in N$, we have $f(\bar{v}_1, \ldots, \bar{v}_n) = e^0$. Note that $v'_i - \epsilon = v_i - \epsilon = \bar{v}_i$ for all $i \neq j$ for any ϵ. Now, fix any $\epsilon > 0$ such that $v'_j - \epsilon > 0$. Consider the valuation profile $(\bar{v}_{-j}, v'_j - \epsilon)$. Since $f(\bar{v}_1, \ldots, \bar{v}_n) = e^0$ and $\bar{v}_j = v_j - \epsilon > v'_j - \epsilon$, by monotonicity and non-bossiness, we have $f(v'_j - \epsilon, \bar{v}_{-j}) = e^0$. Hence, $e^0 \in W^f(v'')$.

This establishes that $e^k \in W^f(v'')$ for any $e^k \neq e^j$. Hence, $W^f(v) \setminus \{e^j\} \subseteq W^f(v'')$. Repeating this argument for other elements of $B(v, v')$ one by one, we conclude that $W^f(v) \setminus B(v, v') \subseteq W^f(v')$. ■

Now, let f be an implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$. Pick any pair of alternatives $e^j, e^k \in A$ and any pair of valuation profiles v, v' such that $\mathbf{1}_{v_j} = \mathbf{1}_{v'_j}$ and $\mathbf{1}_{v_k} = \mathbf{1}_{v'_k}$. We will show that f satisfies both (1) and (2) of Definition 13.

1. Suppose $e^k \in W^f(v)$ and $e^j \in W^f(v')$. We will show that $e^k \in W^f(v')$. Construct a new type profile v'' such that $v''_i = \min(v_i, v'_i)$ for all $i \in N$. Note that $\mathbf{1}_{v''_j} = \mathbf{1}_{v_j} = \mathbf{1}_{v'_j}$ and $\mathbf{1}_{v''_k} = \mathbf{1}_{v_k} = \mathbf{1}_{v'_k}$. By Lemma 8, $e^j, e^k \in W^f(v'')$. Now, assume for contradiction that $e^k \notin W^f(v')$. We now consider various cases.

   Case 1: $e^j, e^k \in A \setminus \{e^0\}$. Since $e^k \notin W^f(v')$, there exists $\epsilon > 0$ such that $f(v'_k + \epsilon, v'_{-k}) \neq e^k$. By monotonicity and non-bossiness, for all $\epsilon' > 0$ we have $f(v'_j + \epsilon', v'_k +$

$\epsilon, v'_{-jk}) \neq e^k$. Further, we show that $f(v'_j + \epsilon', v'_k + \epsilon, v'_{-jk}) = e^j$ for all $\epsilon' > 0$. To see this, suppose $f(v'_j + \epsilon', v'_k + \epsilon, v'_{-jk}) = e^l$ for some $e^l \notin \{e^j, e^k\}$. Then, by monotonicity and non-bossiness, we get $f(v'_j + \epsilon', v'_k, v'_{-jk}) = e^l$, and this contradicts $e^j \in W^f(v')$. Hence, $f(v'_j + \epsilon', v'_k + \epsilon, v'_{-jk}) = e^j$ for all $\epsilon' > 0$. Now, applying monotonicity and non-bossiness again, for all $\epsilon' > 0$, we have

$$f(v'_j + \epsilon', v'_k + \epsilon, v''_{-jk}) = e^j. \tag{2.1}$$

Since $e^k \in W^f(v'')$, we have $f(v'_j, v'_k + \frac{\epsilon}{2}, v''_{-jk}) = e^k$. By Condition $\mathcal{C}^*$, there is an $\epsilon' > 0$ such that $f(v'_j + \epsilon', v'_k + \epsilon, v''_{-jk}) = e^k$. This contradicts Equation 2.1.

Case 2: $e^j = e^0$. We have to show that $e^0 \in W^f(v')$ implies $e^k \in W^f(v')$. Assume for contradiction that $e^k \notin W^f(v')$ but $e^0 \in W^f(v')$. For this, we first show that there is some $\epsilon_i > 0$ for every $i \in N$ such that $f(v'_k + \epsilon_k, \{v'_i - \epsilon_i\}_{i \neq k}) = e^0$.

To see this, suppose $f(v'_k + \epsilon_k, \{v'_i - \epsilon_i\}_{i \neq k}) = e^k$ for all $\{\epsilon_i\}_{i \in N}$. Fix any $l \neq k$. Then, by Condition $\mathcal{C}^*$, for every ϵ there is a δ such that, $f(v'_k + \epsilon_k + \epsilon, (v'_l - \epsilon_l + \delta), \{v'_i - \epsilon_i\}_{i \neq k,l}) = e^k$ for all $\{\epsilon_i\}_{i \in N}$. Fix some $\epsilon > 0$. By, Condition $\mathcal{C}^*$, we can choose $\epsilon_l = \delta$. Also, let $\epsilon_k = \epsilon$. Hence, we get $f(v'_k + 2\epsilon, v'_l, \{v'_i - \epsilon_i\}_{i \neq k,l}) = e^k$. Repeating this, we reach $f(v'_k + (n-1)\epsilon, v'_{-k}) = e^k$. Since $n > 1$, we get that $e^k \in W^f(v')$. But this contradicts the fact that $e^k \notin W^f(v')$.

Similarly, suppose $f(v'_k + \epsilon_k, \{v'_i - \epsilon_i\}_{i \neq k}) = e^l$ for some $l \neq 0, k$. Then, by monotonicity and non-bossiness, we get that $f(\{v'_i - \epsilon_i\}_{i \in N}) = e^l$. This means $f(\{v'_i - \epsilon_i\}_{i \in N}) \neq e^0$. Now, choose $\epsilon' < \min_{i \in N} \epsilon_i$. Then, consider the profile $\{v'_i - \epsilon'\}_{i \in N}$. By repeated application of monotonicity and non-bossiness, $f(\{v'_i - \epsilon'\}_{i \in N}) \neq e^0$. This contradicts $e^0 \in W^f(v')$.

This shows that there is some $\epsilon_i > 0$ for all $i \in N$ such that $f(v'_k + \epsilon_k, \{v'_i - \epsilon_i\}_{i \neq k}) = e^0$. By monotonicity and non-bossiness, $f(v'_k + \epsilon_k, \{v''_i - \epsilon_i\}_{i \neq k}) = e^0$. But $e^k \in W^f(v'')$ implies that $f(v''_k + \epsilon_k, v''_{-k}) = e^k$ (to remind, $v'_k = v''_k$). But monotonicity and non-bossiness implies that $f(v'_k + \epsilon_k, \{v''_i - \epsilon_i\}_{i \neq k}) = e^k$. This gives us a contradiction.

Case 3: $e^k = e^0$. We have to show that if $e^j \in W^f(v')$ then $e^0 \in W^f(v')$. Assume for contradiction $e^0 \notin W^f(v')$. We first show that for some $\epsilon > 0$ and $\epsilon' > 0$, $f(v'_j - \epsilon, \{v'_i - \epsilon'\}_{i \neq j}) = e^j$.

To see this, suppose that $f(v'_j - \epsilon, \{v'_i - \epsilon'\}_{i \neq j}) = e^0$ for all ϵ, ϵ'. Then, by monotonicity and non-bossiness, we see that $f(\{v'_i - \min(\epsilon, \epsilon')\}_{i \in N}) = e^0$ for all ϵ, ϵ'. But this contradicts $e^0 \notin W^f(v')$.

Similarly, suppose that $f(v'_j - \epsilon, \{v'_i - \epsilon'\}_{i \neq j}) = e^l$ for some $l \in N \setminus \{j\}$ and for all ϵ, ϵ'. By Condition $\mathcal{C}^*$, there is some $\delta := \delta_{\epsilon', v'_{-lj}} < \epsilon'$ such that $f(v'_j - \epsilon + \delta, v'_l, \{v'_i - \epsilon'\}_{i \neq j,l}) = e^l$

for all ϵ, ϵ'. Since δ is independent of ϵ, we can choose $\epsilon = \frac{\delta}{2}$ for every ϵ'. Hence, we have $f(v'_j + \frac{\delta}{2}, v'_l, \{v'_i - \epsilon'\}_{i \neq j,l}) = e^l$ for every ϵ'. Further, since $e^j \in W^f(v')$, we know that $f(v'_j + \frac{\delta}{2}, v'_{-j}) = e^j$ for all ϵ'. By repeatedly applying monotonicity and non-bossiness, we get that $f(v'_j + \frac{\delta}{2}, v'_l, \{v'_i - \epsilon'\}_{i \neq j,l}) = e^j$ for every ϵ'. This gives us a contradiction.

This shows that $f(v'_j - \epsilon, \{v'_i - \epsilon'\}_{i \neq j}) = e^j$ for some $\epsilon > 0$ and $\epsilon' > 0$. By repeatedly applying monotonicity and non-bossiness, we get that $f(v'_j - \epsilon, \{v''_i - \epsilon'\}_{i \neq j}) = e^j$ for some $\epsilon > 0$ and $\epsilon' > 0$. Since $e^0 \in W^f(v'')$, we know that $f(\{v'_i - \min(\epsilon, \epsilon')\}_{i \in N}) = e^0$. By repeatedly applying monotonicity and non-bossiness, we get that $f(v'_j - \epsilon, \{v''_i - \epsilon'\}_{i \neq j}) = e^0$. This is a contradiction.

This concludes the proof of Property (1) in Definition 13.

2. Property (2) in Definition 13 follows by applying Property (1). To see this, pick any $e^j, e^k \in A$ and v, v' as in Definition 13. Suppose $e^j \in W^f(v)$ but $e^k \notin W^f(v')$. We need to show that $e^k \notin W^f(v')$. Assume for contradiction $e^k \in W^f(v')$. Then, by changing the role of v and v' in (1), we get that $e^k \in W^f(v)$, which is a contradiction.

■

Step 1.2. In this step, we define an ordering on the set of utility vectors D. We denote this ordering as $\succeq^f$. The anti-symmetric part of this ordering is denoted as $\succ^f$ and the symmetric part is denoted as $\sim^f$. For any $i \in N$ and for any $v_i, v'_i \in V_i$ with $v_i > v'_i$, we define $\mathbf{1}_{v_i} \succ^f \mathbf{1}_{v'_i}$. Further, for every $i \in N$ and every $v_i \in V_i$, we define $\mathbf{1}_{v_i} \sim^f \mathbf{1}_{v_i}$ (reflexive). For any $i, j \in N \cup \{0\}$ (with $i \neq j$) and any $v_i \in V_i$ and $v_j \in V_j$, we define

1. $\mathbf{1}_{v_i} \succ^f \mathbf{1}_{v_j}$, if there exists a valuation profile v' such that $\mathbf{1}_{v'_i} = \mathbf{1}_{v_i}$, $\mathbf{1}_{v'_j} = \mathbf{1}_{v_j}$, and $e^i \in W^f(v')$ but $e^j \notin W^f(v')$;

2. $\mathbf{1}_{v_i} \sim^f \mathbf{1}_{v_j}$, if (a) there exists a valuation profile v' such that $\mathbf{1}_{v'_i} = \mathbf{1}_{v_i}$, $\mathbf{1}_{v'_j} = \mathbf{1}_{v_j}$, and $e^i, e^j \in W^f(v')$ or (b) at every valuation profile v' such that $\mathbf{1}_{v'_i} = \mathbf{1}_{v_i}$, and $\mathbf{1}_{v'_j} = \mathbf{1}_{v_j}$, we have $e^i, e^j \notin W^f(v')$.

We show that the binary relation $\succeq$ is well defined.

Lemma **9** *Suppose f is implementable, non-bossy, and satisfies Condition $\mathcal{C}^*$. Then, $\succeq^f$ is well-defined.*

Proof: Fix some $x, y \in D$. If $x, y \in D_i$ for some $i \in N$, and $x = \mathbf{1}_{v_i}$ and $y = \mathbf{1}_{v'_i}$ with $v_i > v'_i$ then, by definition, $x \succ^f y$. Similarly, if $x \in D_i$ and $y \in D_j$ for some $i \neq j$, and for every valuation profile v with $\mathbf{1}_{v_i} = x$ and $\mathbf{1}_{v_j} = y$ we have $e^i, e^j \notin W^f(v)$, then, by definition, $x \sim^f y$.

So, we just need to consider the case where $x \in D_i$ and $y \in D_j$ for some $i \neq j$, and there is a valuation profile v with $\mathbf{1}_{v_i} = x$ and $\mathbf{1}_{v_j} = y$ with either e^i or e^j or both are in $W^f(v)$.

We consider two cases.

CASE 1. Suppose $e^i, e^j \in W^f(v)$. Now, consider any other valuation profile v' such that $\mathbf{1}_{v_i} = \mathbf{1}_{v'_i} = x$ and $\mathbf{1}_{v_j} = \mathbf{1}_{v'_j} = y$. By Proposition 1, $e^i \in W^f(v')$ if and only if $e^j \in W^f(v')$. This means that the relation $x \sim^f y$ is well-defined.

CASE 2. Suppose $e^i \in W^f(v)$ but $e^j \notin W^f(v)$. Now, consider any other valuation profile v' such that $\mathbf{1}_{v_i} = \mathbf{1}_{v'_i} = x$ and $\mathbf{1}_{v_j} = \mathbf{1}_{v'_j} = y$. By Proposition 1, $e^j \notin W^f(v')$. This means that the relation $x \succ^f y$ is well-defined. ■

STEP 1.3. In this step, we show that $\succeq^f$ is an ordering, i.e., the binary relation is reflexive, complete, and transitive. The fact that $\succeq^f$ is reflexive and complete is clear. We show that $\succeq^f$ is transitive.

PROPOSITION **2** *If f is an implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$, then $\succeq^f$ is transitive.*

Proof: For this, we will show that $\succ^f$ and $\sim^f$ are transitive, and this in turn will imply that $\succeq^f$ is transitive. Pick any $x, y, z \in D$ such that $x \neq y \neq z$. We consider three cases.

CASE 1. Suppose $x, y, z \in D_i$ for some $i \in N$ and $x = \mathbf{1}_{v_i}, y = \mathbf{1}_{v'_i}, z = \mathbf{1}_{v''_i}$. Suppose $x \succ^f y$ and $y \succ^f z$. Then, it must be $v_i > v'_i > v''_i$. By definition, we have $x \succ^f z$.

CASE 2. $x, y \in D_i$ but $z \in D_j$ for some i, j where $i \neq j$. Suppose $x = \mathbf{1}_{v_i}, y = \mathbf{1}_{v'_i}$, and $z = \mathbf{1}_{v_j}$. Suppose $x \succ^f y$ and $y \succ^f z$. Note that $x \succ^f y$ implies $v_i > v'_i$. We consider two subcases.

CASE 2A. Suppose $j \neq 0$. Since $y \succ^f z$, there is a valuation profile v'' such that $v''_i = v'_i$, $v''_j = v_j$, and $e^i \in W^f(v'')$ but $e^j \notin W^f(v'')$. Now consider the type profile $\bar{v}$, where $\bar{v}_k = v''_k$ if $k \neq i$ and $\bar{v}_i = v_i$. We show that $e^i \in W^f(\bar{v})$ and $e^j \notin W^f(\bar{v})$, and this will show that $x \succ^f z$. Since $e^i \in W^f(v'')$, we know that $f(v'_i + \epsilon, v_j, v''_{-ij}) = e^i$ for all $\epsilon > 0$. By monotonicity, $f(v_i + \epsilon, v_j, v''_{-ij}) = e^i$ for all $\epsilon > 0$. Hence, $e^i \in W^f(\bar{v})$. Since $e^j \notin W^f(v'')$, there is some $\epsilon > 0$ such that $f(v'_i, v_j + \epsilon, v''_{-ij}) \neq e^j$. By monotonicity and non-bossiness, $f(v_i, v_j + \epsilon, v''_{-ij}) \neq e^j$. Hence, $e^j \notin W^f(\bar{v})$.

CASE 2B. Suppose $j = 0$. So, z is the n-dimensional zero vector. Since $y \succ^f z$, there is a valuation profile $\bar{v}$ with $\mathbf{1}_{\bar{v}_i} = \mathbf{1}_{v'_i} = y$ and $e^i \in W^f(\bar{v})$ but $e^0 \notin W^f(\bar{v})$. Now, consider the valuation profile $v'' \equiv (v_i, \bar{v}_{-i})$. Since $v_i > v'_i$, by monotonicity, we have $e^i \in W^f(v'')$.

Since $e^0 \notin W^f(\bar{v})$, there is some $\epsilon > 0$ such that $f(\{\bar{v}_k - \epsilon\}_{k \in N}) \neq e^0$. Now, since $v_i > v'_i$, by monotonicity and non-bossiness, $f(v_i - \epsilon, \{\bar{v}_k - \epsilon\}_{k \neq i}) \neq e^0$. Hence, $e^0 \notin W^f(v'')$.

This completes the proof of Case 2.

Case 3. $x \in D_i$, $y \in D_j$, $z \in D_k$, where i, j, k are distinct. Suppose $x = \mathbf{1}_{v_i}, y = \mathbf{1}_{v_j}$, and $z = \mathbf{1}_{v_k}$. Here, we will consider transitivity of both $\succ^f$ and $\sim^f$.

Case 3a - Transitivity of $\succ^f$. Suppose $x \succ^f y$ and $y \succ^f z$. Since $x \succ^f y$, there is some valuation profile v'' where $\mathbf{1}_{v''_i} = x, \mathbf{1}_{v''_j} = y$, and $e^i \in W^f(v'')$ but $e^j \notin W^f(v'')$.

First, note that $i \neq 0$. To see this, since $y \succ^f z$ there is a valuation profile v' where $\mathbf{1}_{v'_j} = y, \mathbf{1}_{v'_k} = z$, and $e^j \in W^f(v')$ but $e^k \notin W^f(v')$. But $\mathbf{1}_{v'_i} = x$ implies that $y \succeq^f x$, which contradicts $x \succ^f y$. Hence, $i \neq 0$.

Suppose $v''_k < v_k$. Since $e^i \in W^f(v'')$, for every $\epsilon > 0$, $f(v''_i + \epsilon, v''_j, v''_k, v''_{-ijk}) = e^i$. By monotonicity and non-bossiness, $f(v''_i + \epsilon, v''_j, v_k, v''_{-ijk}) \in \{e^i, e^k\}$ for every $\epsilon > 0$. But $f(v''_i + \epsilon, v''_j, v_k, v''_{-ijk}) = e^k$ for any $\epsilon > 0$ will imply that $z \succeq^f y$, and this will contradict $y \succ^f z$. Hence, $f(v''_i + \epsilon, v''_j, v_k, v''_{-ijk}) = e^i$ for every $\epsilon > 0$. So, $e^i \in W^f(v''_i, v''_j, v_k, v''_{-ijk})$. Since $y \succ^f z$, $e^k \notin W^f(v''_i, v''_j, v_k, v''_{-ijk})$. Hence, $x \succ^f z$.

Suppose $v''_k \geq v_k$. As before, since $e^i \in W^f(v'')$, for every $\epsilon > 0$, $f(v''_i+\epsilon, v''_j, v''_k, v''_{-ijk}) = e^i$. By monotonicity and non-bossiness, $f(v''_i + \epsilon, v''_j, v_k, v''_{-ijk}) = e^i$ for every $\epsilon > 0$. Hence, $e^i \in W^f(v''_i, v''_j, v_k, v''_{-ijk})$. Since $y \succ^f z$, $e^k \notin W^f(v''_i, v''_j, v_k, v''_{-ijk})$. Hence, $x \succ^f z$.

Case 3b - Transitivity of $\sim^f$. Suppose $x \sim^f y$ and $y \sim^f z$. Suppose for every valuation profile v' such that $\mathbf{1}_{v'_i} = x$ and $\mathbf{1}_{v'_j} = y$, we have $e^i, e^j \notin W^f(v')$. Further, suppose for every valuation profile $\bar{v}$ with $\mathbf{1}_{\bar{v}_j} = y$ and $\mathbf{1}_{\bar{v}_k} = z$, we have $e^j, e^k \notin W^f(\bar{v})$. Consider any valuation profile v'' such that $\mathbf{1}_{v''_i} = x$ and $\mathbf{1}_{v''_k} = z$. Assume for contradiction $e^i \in W^f(v'')$. Consider the valuation profile $\hat{v}$ such that $\mathbf{1}_{\hat{v}_j} = y$ and $\hat{v}_l = v''_l$ for all $l \neq j$. Since $\mathbf{1}_{\hat{v}_k} = z$, by definition $e^j, e^k \notin W^f(\hat{v})$. By monotonicity and non-bossiness, $e^i \in W^f(\hat{v})$. But, this is not possible since $\mathbf{1}_{\hat{v}_i} = x$ implies that $e^i, e^j \notin W^f(\hat{v})$. This means that at every valuation profile v'' with $\mathbf{1}_{v''_i} = x$ and $\mathbf{1}_{v''_k} = z$ we must have $e^i, e^k \notin W^f(v'')$. Hence, $x \sim^f z$.

Now, consider the case where $y \sim^f z$ and there is some valuation profile v' such that $\mathbf{1}_{v'_j} = y$, $\mathbf{1}_{v'_k} = z$, and $e^j, e^k \in W^f(v')$. If $x = \mathbf{1}_{v_0}$, then by Proposition 1, $e^i \in W^f(v')$, and this immediately implies that $x \sim^f z$. Suppose $x = \mathbf{1}_{v_i}$ and $i \neq 0$. Then, either $j \neq 0$ or $k \neq 0$. We consider the case of $j \neq 0$ - the proof for $k \neq 0$ is similar. Since $e^j \in W^f(v')$, $f(v'_j + \epsilon, v'_{-j}) = e^j$ for all $\epsilon > 0$. By monotonicity and non-bossiness, $f(v'_j + \epsilon, v_i, v'_{-ij}) \in \{e^i, e^j\}$ for all $\epsilon > 0$. If $f(v'_j + \epsilon, v_i, v'_{-ij}) = e^i$, then by monotonicity and non-bossiness, $e^i \in W^f(v'_j, v_i, v'_{-ij})$. Since, $x \succ^f y$ and $y \succ^f z$, by repeated application of Proposition 1, we get that $e^j, e^k \in W^f(v'_j, v_i, v'_{-ij})$. This implies that $x \succ^f z$. Similarly, if $f(v'_j+\epsilon, v_i, v'_{-ij}) = e^j$, then $e^j \in W^f(v'_j, v_i, v'_{-ij})$. Again, using the fact that $x \succ^f y$ and $y \succ^f z$, by repeated application of Proposition 1, we get that $e^i, e^k \in W^f(v'_j, v_i, v'_{-ij})$. So, $x \succ^f z$. ■

Step 1.4. We conclude Step 1 by showing that f is a rationalizable allocation rule and $\succeq^f$ rationalizes f. Note that the ordering $\succeq^f$, defined in Steps 1.2 and 1.3, is a monotone ordering. By Lemma 7, for every valuation profile v, $f(v) \in W^f(v)$. Hence, by definition of $\succeq^f$, $G^f(v) \succeq^f \mathbf{1}_{v_i}$ for all $i \in N \cup \{0\}$. This shows that f is a rationalizable allocation rule and $\succeq^f$ rationalizes f.

STEP 2. In this step, we show that if f is a non-bossy allocation rule satisfying Condition $\mathcal{C}^*$, then it is implementable if and only if it is an SUM allocation rule. By Lemma 3, an SUM allocation rule is implementable. Suppose f is an implementable and non-bossy allocation rule satisfying Condition $\mathcal{C}^*$. By Step 1, f can be rationalized by the monotone ordering $\succeq^f$, defined as in Step 1.2. We say that $\succeq^f$ has a **utility representation** if there exists a utility function $U : D \to \mathbb{R}$ such that for all $x, y \in D$ we have $U(x) > U(y)$ if and only if $x \succ^f y$.

STEP 2.1. In this step, we will show that $\succeq^f$ is separable in the sense of Definition 5. Let $Z := \{x \in D : x = \mathbf{1}_{v_i}$ for some $i \in N \cup \{0\}$ and v_i is rational$\}$. Note that since the set of rational numbers is countable, Z is a countable subset of D. Now, pick $x, y \in D$ such that $x \succ^f y$. If $x, y \in D_i$ for some $i \in N$, then let $x = \mathbf{1}_{v_i}$ and $y = \mathbf{1}_{v'_i}$. By definition, $v_i > v'_i$. Then, we can find a rational v''_i such that $v_i > v''_i > v'_i$ (this is because the set of rational numbers is a dense set). Let $z = \mathbf{1}_{v''_i}$. By definition, $z \in Z$ and $x \succ^f z \succ^f y$. Now, assume that $x = \mathbf{1}_{v_i}$ and $y = \mathbf{1}_{v_j}$ for some $i, j \in N \cup \{0\}$ with $i \neq j$. We consider various cases.

CASE A. Suppose $i \neq 0$ and $j \neq 0$. Since $x \succ^f y$, there is a valuation profile $v \equiv (v_i, v_j, v_{-ij})$ such that $e^i \in W^f(v)$ but $e^j \notin W^f(v)$. Since $e^j \notin W^f(v)$, there is some $\epsilon > 0$ such that $f(v_i, v_j + \epsilon, v_{-ij}) \neq e^j$. This means that $e^j \notin W^f(v_i, v_j + \frac{\epsilon}{2}, v_{-ij})$. Consider any $\delta > 0$. Since $f(v_i, v_j + \frac{\epsilon}{2}, v_{-ij}) \neq e^j$, by monotonicity and non-bossiness, $f(v_i + \delta, v_j + \frac{\epsilon}{2}, v_{-ij}) \neq e^j$. Since $e^i \in W^f(v)$, $f(v_i + \delta, v_j, v_{-ij}) = e^i$. By monotonicity and non-bossiness, $f(v_i + \delta, v_j + \frac{\epsilon}{2}, v_{-ij}) \in \{e^i, e^j\}$. This implies that $f(v_i + \delta, v_j + \frac{\epsilon}{2}, v_{-ij}) = e^i$. Hence, $e^i \in W^f(v_i, v_j + \frac{\epsilon}{2}, v_{-ij})$. Then, $x = \mathbf{1}_{v_i} \succ \mathbf{1}_{v_j + \frac{\epsilon}{2}} \succ \mathbf{1}_{v_j} = y$. Since the set of rational numbers is dense, we can find a $z \in Z$ arbitrarily close to $\mathbf{1}_{v_j + \frac{\epsilon}{2}}$ such that $x \succ^f z \succ^f y$.

CASE B. Suppose $i \neq 0$ and $j = 0$. Since $x \succ^f y$, there is a valuation profile (v_i, v_{-i}) such that $e^i \in W^f(v_i, v_{-i})$ but $e^0 \notin W^f(v_i, v_{-i})$. This means for some $\delta > 0$, we have $f(\{v_j - \delta\}_{j \in N}) \neq e^0$. Suppose $f(\{v_j - \delta\}_{j \in N}) = e^k$ for some $k \neq 0$. Then, $\mathbf{1}_{v_k - \delta} \succeq^f y$. Since $e^i \in W^f(v_i, v_{-i})$, we get that $x = \mathbf{1}_{v_i} \succeq^f \mathbf{1}_{v_k} \succ^f \mathbf{1}_{v_k - \delta}$. Hence, $x \succ^f \mathbf{1}_{v_k - \delta} \succeq^f y$. Since the set of rational numbers is dense, we can choose a $z \in Z$ arbitrarily close to $\mathbf{1}_{v_k - \delta}$ such that $x \succ^f z \succeq^f y$.

CASE C. Suppose $i = 0$ and $j \neq 0$. Since $x \succ^f y$, there is a valuation profile (v_j, v_{-j}) such that $e^j \notin W^f(v_j, v_{-j})$ but $e^0 \in W^f(v_j, v_{-j})$. Then, for some $\epsilon > 0$, we have $f(v_j + \epsilon, v_{-j}) = e^k$, where $k \neq j$. This implies that $\mathbf{1}_{v_k} \succeq^f \mathbf{1}_{v_j + \epsilon} \succ^f \mathbf{1}_{v_j} = y$. But $e^0 \in W^f(v_j, v_{-j})$ implies that $x \succeq^f \mathbf{1}_{v_k}$. Hence, $x \succeq^f \mathbf{1}_{v_j + \epsilon} \succ^f y$. Since the set of rational numbers is dense, we can find $z \in Z$ arbitrarily close to $\mathbf{1}_{v_j+\epsilon}$ such that $x \succeq^f z \succ^f y$.

This shows that $\succeq^f$ is separable. Using Debreu (1954), $\succeq^f$ has a utility representation. Let $U : D \to \mathbb{R}$ be a utility function representing $\succeq^f$. Without loss of generality, we can assume $U(\mathbf{1}_{v_0}) = 0$. Now, for every $i \in N \cup \{0\}$, define $U_i : V_i \to \mathbb{R}$ as follows: $U_i(v_i) = U(\mathbf{1}_{v_i})$ for all $v_i \in V_i$. Note that by the definition of $\succeq^f$, each U_i is well-defined and increasing.

Since U represents $\succeq^f$ and f is a rationalizable allocation rule with $\succeq^f$ being the corresponding ordering, we get that for all valuation profiles v, $f(v) \in \arg\max_{i \in N \cup \{0\}} U_i(v_i)$. Hence, f is an SUM allocation rule.

By Theorem 1, f is a strongly rationalizable allocation rule. Let $\succ$ be the strict linear ordering that strongly rationalizes f. By definition, for all $x \in D^f$ and for all $y \notin D^f$, $x \succ y$. Further, for all $v \in V$ if $f(v) = e^j$, then $\mathbf{1}_{v_j} \succ \mathbf{1}_{v_i}$ for all $i \neq j$. In that case, $\mathbf{1}_{v_j} \succ \mathbf{1}_{v_k}$ for all $k \neq j$ and $k \in \arg\max_{i \in N \cup \{0\}} U_i(v_i)$. Hence, f is an SUM allocation rule with order-based tie-breaking.

Chapter 3

Mechanism Design in Single Dimensional Type Spaces

3.1 Introduction

We study a mechanism design problem over a connected graph in quasi-linear and private value environment. The planner needs to use a subset of edges of the graph to connect to all its vertices. For instance, consider a planner who wants to send data over a mobile network. The nodes in the mobile network are different geographic locations and the planner must send data to all such locations. These locations (nodes) on the network are connected by edges which are communication links. Communication links are privately held by different agents (firms), who incur costs when their links are used.

The cost of each agent is a private information. A mechanism consists of an allocation rule and a payment rule for every agent. Our objective is to identify the set of all dominant strategy incentive compatible mechanisms that can be designed in this environment. We identify a rich class of allocation rules for this problem called *generalized utility maximizers*, and show that there exist payment rules such that the generalized utility maximizers and the corresponding payment rules are dominant strategy incentive compatible. Moreover, every dominant strategy incentive compatible mechanism consists of an allocation rule which is a generalized utility maximizer. [1]

A generalized utility maximizer assigns a map, which we call a generalized utility function (GUF), to every agent. A GUF assigns a real number to every profile of types. A generalized utility maximizer chooses a subset of agents at every type profile such that (a) all the nodes are connected (b) the sum of generalized utilities of agents is maximized.

We extend the characterization of this model to a model where the planner is choosing a *base* of a matroid and the matroid is defined by the set of agents as the ground set and a family of subsets of agents being *independent set*. This model allows us to capture

[1] The well-known revenue equivalence result holds in this environment. Hence, identifying the allocation rule of a dominant strategy incentive compatible mechanism also pins down the payment rules.

the connected graph model and many well studied models in the literature, including the single object auction model, the multi-unit homogeneous goods model with unit demand and the heterogeneous good auction with dichotomous preferences. Thus, we provide a complete characterization of dominant strategy incentive compatible mechanisms for a family of problems where the type of an agent is a single number (i.e., type space is one dimensional) and the set of alternatives exhibits certain structure.

We sharpen our characterization by adding two reasonable axioms, a suitable version of *anonymity* and *non-bossiness*, on top of incentive compatibility. We show that only dominant strategy incentive compatible mechanisms are the Groves mechanism, i.e., uses an efficient allocation. This extends and unifies the similar results which have been proved in specific models. We discuss these results later.

3.1.1 Relation of Our Result to the Literature

In our connected graph model, any allocation rule *implementable* by a dominant strategy incentive compatible mechanism can be characterized by a simple monotonicity condition (Myerson, 1981). Monotonicity is the following requirement; fixing the type of other agents if an agent is chosen by the allocation rule at a type, then he continues to be chosen by the allocation rule at a higher type. While monotonicity is a simple condition to interpret and use, it is only an implicit characterization of an implementable allocation rule. In contrast, our characterization of generalized utility maximizer explicitly describes the parameters of the mechanism in these problems.

Our characterization is in the spirit of a seminal contribution of Roberts (1979) [2]. Roberts shows that with at least three alternatives and in the multi-dimensional unrestricted domain, an allocation rule is implementable if an only if it is an *affine maximizer*. Our characterizations are for the restricted single dimensional type spaces and it provides a richer class of implementable allocation rules than Roberts (1979).

In single dimensional type spaces, several papers show that a mechanism is *anonymous in utility*, dominant strategy incentive compatible, individually rational and has non-negative payments if and only if it is the VCG (Vickrey, 1961; Clarke, 1971; Groves, 1973) mechanism. For instance, Ashlagi and Serizawa (2011) characterize the VCG mechanism for homogeneous goods with unit demand with all these axioms and Mukherjee (2014) characterizes the VCG mechanism for single object auction considering all these axioms. However, we characterize all dominant strategy incentive compatible mechanisms in a class of single dimensional type spaces without imposing any condition on mechanism.

A closely related paper that deserves special attention is of Bikhchandani et al. (2011). They consider selling *bases* of a matroid using an ascending Vickrey auction. In our matroid model, we also consider the set of alternatives as the set of all bases of a matroid. But

[2] See Mishra and Sen (2012); Carbajal et al. (2013); Mishra and Quadir (2014) for various generalizations of Roberts' theorem for restricted domains.

our model differs from theirs in the following ways, (a) we construct our bases of a matroid considering subsets of agents while they construct bases of a matroid considering subsets of objects, (b) we characterize all dominant strategy incentive compatible mechanisms while they design an ascending Vickrey auction and show that truthful bidding is an ex post equilibrium.

Our paper also contribute to the growing literature of characterizing the fair mechanisms. A fairness criteria that is very common in the literature is anonymity. In this spirit, we have provided a characterization of Groves mechanisms using a suitable version of anonymity and non-bossiness along with incentive compatibility. Recently, using a suitable version of anonymity along with some other conditions, fair mechanisms have been characterized both in quasi-linear (Ashlagi and Serizawa, 2011; Andersson et al., 2014) and non-quasilinear setting (Sakai, 2008; Saitoh and Serizawa, 2008; Morimoto and Serizawa, 2014) in specific settings.

3.2 The Connected Graph Model

Let $N := \{1, 2, \ldots, n\}$ be a finite set of agents. We consider a connected graph $G = (M, E)$ where M is the set of nodes and E is the set of edges of the graph. We assume that $|N| = |E|$ and each edge is owned by an unique agent in N. For notational simplicity, we will refer to an edge by the agent who owns it. Each agent has a private valuation which can be thought of as the weight of the edge of the graph owned by that agent.

Let the set of all possible valuations of agent i be $V_i = (0, \beta_i)$ where $\beta_i \in \mathbb{R}_{++} \cup \{\infty\}$. Note that we do not allow for zero valuations. We use the standard notation $v = (v_1, v_2, \ldots, v_n)$ as the vector of valuations of agents and v_{-i} as the vector of valuations of all agents except agent i. Similarly V_{-i} is the set of all profiles of valuations without agent i. Let $V := V_1 \times V_2 \times \cdots \times V_n$.

A tree is a acyclic connected graph. A *spanning tree* of the graph $G = (M, N)$ is a subgraph (M, N') such that it is a tree i.e, a tree that connects all the nodes of the graph. There can be many spanning trees of the graph G. Let X be the set of all the spanning trees of G. Without loss of generality, we will denote a spanning tree by the set of edges only.

We consider a social alternative as a spanning tree. We did not consider a cycle as an alternative because a cycle is an inefficient way to connect all vertices of a graph. Thus, spanning tree ensures some minimal efficiency. Another possibility is to consider the set of alternatives as the set of all trees. We did not consider this because our objective is to connect all vertices of a graph. In particular, we take into our consideration the case of telecommunication network where an outsider wants to use the entire network. It is interesting exercise to consider the set of alternatives as the set of all trees but it constitutes a new model.

An allocation rule is a mapping $f : V \to X$ - note that $f(v)$ is a spanning tree. Notice that we do not allow for the fact that no spanning tree is selected at some valuation profile.

Payments are allowed in this model. We define the payment function of an agent $i \in N$ as a mapping $p_i : V \to \mathbb{R}$. A mechanism is a tuple (f, p) where $p = (p_1, p_2, \ldots, p_n)$ is a collection of payment functions. Net utility of an agent $i \in N$ from the mechanism (f, p) when his true type is v_i and the reported profile is (v'_i, v_{-i}) is

$$v_i \mathbb{1}_i^{f(v'_i, v_{-i})} - p_i(v'_i, v_{-i}),$$

where for every agent $i \in N$ and for every spanning tree $x \in X$, we denote by $\mathbb{1}_i^x$ to indicate if agent i is in the spanning tree x i.e., $\mathbb{1}_i^x = 1$ if $i \in x$ and $\mathbb{1}_i^x = 0$ if $i \notin x$.

Definition 14 *An allocation rule f is* **implementable** *(in dominant strategies) if there exist payment functions $(p_1, p_2, \ldots, p_n)$ such that for every agent $i \in N$ and for every $v_{-i} \in V_{-i}$*

$$v_i \mathbb{1}_i^{f(v_i, v_{-i})} - p_i(v_i, v_i) \geq v_i \mathbb{1}_i^{f(v'_i, v_{-i})} - p_i(v'_i, v_{-i}) \qquad \forall v_i, v'_i \in V_i$$

In this case, we say $(p_1, p_2, \ldots, p_n)$ implement f and the mechanism $(f, p_1, p_2, \ldots, p_n)$ is **incentive compatible***.*

It is well-known that implementability is equivalent to the following monotonicity condition - see (Myerson, 1981; Nisan, 2007).

Definition 15 *An allocation rule f is* monotone *if for every $i \in N$, for every $v_{-i} \in V_{-i}$, for every $v_i, v'_i \in V_{-i}$ with $v_i > v'_i$ and $\mathbb{1}_i^{f(v'_i, v_{-i})} = 1$, we have $\mathbb{1}_i^{f(v_i, v_{-i})} = 1$.*

3.2.1 The Complete Characterization

We now provide a complete characterization of implementable allocation rules for the connected graph model. Unlike the monotonicity characterization, our characterization is in the spirit of Roberts (1979) by describing the parameters of the mechanism explicitly.

A **cut** of a graph is the partitioning of the nodes of the graph into two parts. Formally, a **cut** in the graph $G = (M, N)$ is the partitioning of the nodes $(S, M \setminus S)$ with $S \neq \emptyset$ and $S \neq M$. Let the set of edges crossing this cut be $N(S)$. At a valuation profile (v_i, v_{-i}), an edge is called the **heavy edge** of a cut if it has the highest valuation among all edges crossing this cut of the graph.

Define a **generalized utility function (GUF)** for agent i as a mapping $u_i : V \to \mathbb{R}$. A GUF assigns a real number to every valuation profile. We impose the following condition on the collection of GUFs $(u_1, u_2, \ldots, u_n)$.

Definition 16 *The GUFs $(u_1, u_2, \ldots, u_n)$ satisfy* **top single crossing** *if for every $i \in N$, for every $v_{-i} \in V_{-i}$, for every $v_i, v'_i \in V_i$ with $v_i > v'_i$, for some cut of the graph $(S', M \setminus S')$ such that $i \in N(S')$ and $u_i(v'_i, v_{-i}) \geq \max_{k \in N(S') \setminus \{i\}} u_k(v'_i, v_{-i})$, there exists a cut $(S, M \setminus S)$ such that $i \in N(S)$ and $u_i(v_i, v_{-i}) > \max_{k \in N(S) \setminus \{i\}} u_k(v_i, v_{-i})$.*

The top single crossing condition requires that for every valuation profile of other agents, if agent i is one of the heavy edges in some cut when his valuation is v'_i, then he must be unique heavy edge for some cut when his valuation is increased to v_i.

The standard inter-agent crossing condition that is used in the literature is "single crossing" which implies top single crossing.

Definition **17** *GUFs* $(u_1, \ldots, u_n)$ *satisfy* **single crossing** *if for every* $i, j \in N$*, for every* $v_{-i} \in V_{-i}$*, for every* $v_i, v'_i \in V_i$ *with* $v_i > v'_i$*, we have* $u_i(v_i, v_{-i}) - u_i(v'_i, v_{-i}) > u_j(v_i, v_{-i}) - u_j(v'_i, v_{-i})$.

A GUF u_i is **increasing** if for every $v_{-i} \in V_{-i}$ and for every $v_i, v'_i \in V_i$ with $v_i > v'_i$, we have $u_i(v_i, v_{-i}) > u_i(v'_i, v_{-i})$.

Lemma **10** *If GUFs* $(u_1, u_2, \ldots, u_n)$ *satisfy single crossing and* u_i *is increasing for every* $i \in N$*, then they satisfy top single crossing.*

Proof: Consider an agent $i \in N$ and a cut of the graph $(S, M \setminus S)$ such that $i \in N(S)$. Let $v_i, v'_i \in V_i$ with $v_i > v'_i$ and $u_i(v'_i, v_{-i}) \geq \max_{k \in N(S) \setminus \{i\}} u_k(v'_i, v_{-i})$. Since u_i is increasing, we have $u_i(v_i, v_{-i}) > u_i(v'_i, v_{-i})$. Further by single crossing, we have $u_i(v_i, v_{-i}) - u_i(v'_i, v_{-i}) > u_j(v_i, v_{-i}) - u_j(v'_i, v_{-i})$ for all $j \neq i$. Since this is true for all $j \neq i$, this is also true for the cut $(S, M \setminus S)$ such that $i \in N(S)$ and $u_i(v_i, v_{-i}) - u_i(v'_i, v_{-i}) > u_j(v_i, v_{-i}) - u_j(v'_i, v_{-i})$ for all $j \in N(S)$. Using the fact that $u_i(v'_i, v_{-i}) \geq u_j(v_i, v_{-i})$ for all $j \in N(S) \setminus \{i\}$, we have $u_i(v_i, v_{-i}) > u_j(v_i, v_{-i})$ for all $j \in N(S) \setminus \{i\}$. Thus, there exists a cut $(S, M \setminus S)$ such that $i \in N(S)$ and we have $u_i(v_i, v_{-i}) > \max_{k \in N(S) \setminus \{i\}} u_k(v_i, v_{-i})$. ■

We now introduce a class of implementable allocation rules.

Definition **18** *An allocation rule* f *is a* **generalized utility maximizer** *if there exist GUFs* $(u_1, u_2, \ldots, u_n)$ *satisfying top single crossing such that for every* $v \in V$*,* $f(v) \in \arg\max_{x \in X} \sum_{i \in N} u_i(v_i, v_{-i}) \mathbb{1}_i^x$.

A generalized utility maximizer considers the connected graph $G = (M, N)$ and at every valuation profile (v_i, v_{-i}) assigns a weight u_i to the edge owned by agent i. Then, it selects the maximum weight spanning tree of this weighted graph. In other words, it selects a spanning tree which maximizes the sum of generalized utility functions.

Note that the utility in GUF has nothing to do with the utility in a mechanism. It is an artificial construct used by the mechanism designer.

The generalized utility maximizer includes a rich class of allocation rules. The following examples illustrate this.

Example **5**

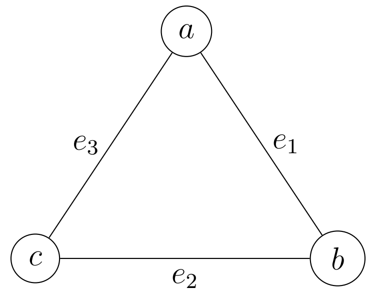


Figure 3.1: Illustration of generalized utility maximizer.

Consider the following graph of three edges in Figure 3.1,

Let the edges be $\{e_1, e_2, e_3\}$. Every agent has some valuation for owning an edge. Define a generalized utility function at valuation profile (v_i, v_{-i}) as $u_i : V \to \mathbb{R}_+$, for every agent $i = 1, 2, 3$ where $u_1(v_1, v_{-1}) = v_1^2$, $u_2(v_2, v_{-2}) = 2v_2$ and $u_3(v_3, v_{-3}) = v_3$. Notice that these generalized utility functions do not depend on other agents' valuations. We can assign the weights of u_1, u_2 and u_3 to the edges e_1, e_2 and e_3 of the graph in Figure 3.1 respectively. It can be very easily verified that these GUFs satisfy single crossing. Hence, by Lemma 10 they satisfy top single crossing. Therefore, these utility functions define a generalized utility maximizer.

We give another example where the generalized utility functions of agents depend on the valuations of other agents.

Example **6**

Consider the graph in Figure 3.1 where there are three edges. Every agent has some valuation for owning the edge. Define a generalized utility function at valuation profile (v_i, v_{-i}) for every agent $i = 1, 2, 3$ as $u_1(v_1, v_{-1}) = v_1^2 + v_2 + v_3$, $u_2(v_2, v_{-2}) = v_1 + 2v_2 + v_3$ and $u_3(v_3, v_{-3}) = v_1 + v_2 + 3v_3$. Notice that these GUFs also depend on other agents' valuations. Now assign the weights u_1, u_2 and u_3 to edges e_1, e_2 and e_3 of the graph in Figure 3.1 respectively. It can easily be verified that these GUFs satisfy single crossing. Hence, by Lemma 10 they satisfy top single crossing. Therefore, they define a generalized utility maximizer.

First, we show that a generalized utility maximizer is implementable.

Lemma **11** *Every generalized utility maximizer is implementable.*

Proof: Consider a generalized utility maximizer f and let $(u_1, u_2, \ldots, u_n)$ be the corresponding GUFs satisfying top single crossing. Fix an agent $i \in N$ and $v_{-i} \in V_{-i}$. Let $v_i, v'_i \in V_i$ with $v_i > v'_i$. Suppose $i \in f(v'_i, v_{-i})$. This implies that i must be one of the heavy edges for some cut $(S', M \setminus S')$. By the top single crossing, there exists a cut $(S, M \setminus S)$ such that $i \in N(S)$ and $u_i(v_i, v_{-i}) > \max_{k \in N(S) \setminus \{i\}} u_k(v_i, v_{-i})$. Hence, i is the unique heavy edge of the cut $(S, M \setminus S)$. Since a maximum weight spanning tree must choose the unique heavy edge from every cut, $i \in f(v_i, v_{-i})$. ■

Our main result shows that generalized utility maximizers are the only implementable allocation rule in the connected graph model.

THEOREM 4 *An allocation rule is implementable if and only if it is a generalized utility maximizer.*

Proof: Lemma 11 showed that every GUF maximizer is implementable. To prove the converse, suppose f is implementable. Fix an agent i and $v_{-i} \in V_{-i}$. Define $\kappa_i^f(v_{-i}) := \inf\{v_i \in V_i : \mathbb{1}_i^{f(v_i,v_{-i})} = 1\}$. Since V_i is a bounded interval, κ_i^f is well-defined. Further, since f is implementable and hence monotone, for every agent $i \in N$, for every $v_{-i} \in V_{-i}$ and for every $v_i \in V_i$ if $v_i \geq \kappa_i^f(v_{-i})$, we have $\mathbb{1}_i^{f(v_i,v_{-i})} = 1$ and for every $v_i < \kappa_i^f(v_{-i})$ we have $\mathbb{1}_i^{f(v_i,v_{-i})} = 0$.

Define for every $i \in N$ and for every (v_i, v_{-i}),

$$u_i(v_i, v_{-i}) := v_i - \kappa_i^f(v_{-i}).$$

By definition, for all $i \in f(v_i, v_{-i})$, we have $u_i(v_i, v_{-i}) = v_i - \kappa_i^f(v_{-i}) \geq 0$ and for all $i \notin f(v_i, v_{-i})$, we have $u_i(v_i, v_{-i}) \leq 0$. Hence, we have $f(v_i, v_{-i}) \in \arg\max_{x \in X} \sum_{i \in N} u_i(v_i, v_{-i})\mathbb{1}_i^x$.

Now, we prove that the GUFs $(u_1, u_2, \ldots, u_n)$ as constructed above satisfy top single crossing. Fix an agent $i \in N$ and $v_i, v_i' \in V_i$ with $v_i > v_i'$. Consider $v_{-i} \in V_{-i}$ and a cut of the graph $(S', M \setminus S')$ such that $i \in N(S')$. Let $u_i(v_i', v_{-i}) \geq \max_{k \in N(S') \setminus \{i\}} u_k(v_i', v_{-i})$. By the construction of the GUF u_i, we have $u_i(v_i, v_{-i}) = v_i - \kappa_i^f(v_{-i}) > v_i' - \kappa_i^f(v_{-i}) = u_i(v_i', v_{-i})$. Hence, $u_i(v_i, v_{-i}) > u_i(v_i', v_{-i})$. This implies that u_i is an increasing function. We know that $u_i(v_i', v_{-i}) \geq \max_{k \in N(S') \setminus \{i\}} u_k(v_i', v_{-i})$ for the cut $(S', M \setminus S')$ of the graph. Suppose $i \in f(v_i', v_{-i})$, then we have $u_i(v_i', v_{-i}) \geq 0$. Suppose $i \notin f(v_i', v_{-i})$, then some other agent $j \in N(S')$, we have $j \in f(v_i', v_{-i})$ (This is because every spanning tree must choose at least one edge from every cut). This implies that $u_i(v_i', v_{-i}) \geq u_j(v_i, v_{-i}) \geq 0$. Therefore, for both the cases, we have $u_i(v_i', v_{-i}) \geq 0$. Since u_i is increasing, we have $u_i(v_i, v_{-i}) > u_i(v_i', v_{-i}) \geq 0$. This implies that $v_i > \kappa_i^f(v_{-i})$ and hence $i \in f(v_i, v_{-i})$.

Since $f(v_i, v_{-i}) \in \arg\max_{x \in X} \sum_{i \in N} u_i(v_i, v_{-i})\mathbb{1}_i^x$, i belongs to the maximum weight spanning tree chosen by f at valuation profile (v_i, v_{-i}). This implies that there exists a cut $(S, M \setminus S)$ such that for every $j \in N(S) \setminus \{i\}$, we have $j \notin f(v_i, v_{-i})$.[3] This implies that for every $j \in N(S) \setminus \{i\}$, we have $u_j(v_i, v_{-i}) = v_j - \kappa_j^f(v_{-j}) \leq 0 < u_i(v_i, v_{-i})$. This proves the claim. ■

[3]To see this, suppose for every cut $(S, M \setminus S)$, there is an edge $j \in N(S) \setminus \{i\}$ such that $j \in f(v_i, v_{-i})$, then we can remove i from $f(v_i, v_{-i})$ and the resulting graph will still be a spanning tree. This contradicts the fact that $f(v_i, v_{-i})$ is a spanning tree.

3.2.2 Payments and Revenue Equivalence

In Theorem 4, we characterize only the implementable allocation rules. However, this also characterizes the set of dominant strategy incentive compatible mechanisms by using the revenue equivalence principle.

Definition **19** *An allocation rule f satisfies revenue equivalence if for every $(p_1, \ldots, p_n)$ and $(p'_1, \ldots, p'_n)$ that implement f, we have for every $i \in N$ and for every $v_{-i} \in V_{-i}$, a mapping $h_i : V_{-i} \to \mathbb{R}$ such that*

$$p'_i(v_i, v_{-i}) = p_i(v_i, v_{-i}) + h_i(v_{-i})$$

for all $v_i \in V_i$.

Revenue equivalence holds in our model since V_i is connected for every $i \in N$ (Nisan, 2007; Heydenreich et al., 2009).

An implication of revenue equivalence is that identifying one payment rule that implements an allocation rule also identifies the entire class of payment rules (upto an additive constant). Given an implementable allocation rule f (a generalized utility maximizer), define for every $i \in N$ and for every (v_i, v_{-i}), $p_i(v_i, v_{-i}) = 0$ if $i \notin f(v_i, v_{-i})$ and $p_i(v_i, v_{-i}) = \kappa_i^f(v_{-i})$ if $i \in f(v_i, v_{-i})$ where κ_i^f is as defined in the proof of Theorem 4. It is easy to verify that this payment rule implements f if f is monotone. By the revenue equivalence principle, we can characterize the entire class of payment rules which implement monotone allocation rule f.

3.3 The Matroid Model

We now generalize the ideas we used in the connected graph model to a more general model. We use the matroid theory to formulate a general model which captures a class of mechanism design problems with single dimensional type spaces like multi-unit homogeneous goods and heterogeneous good auction with dichotomous preferences etc.

A matroid is a set system that has specific requirements.

Definition **20** *A matroid $\mathcal{M}$ is an ordered pair $(N, \mathcal{I})$ where N is a finite ground set and $\mathcal{I}$ is a set of subsets of N satisfying the following axioms,*

1. *$\emptyset \in \mathcal{I}$. (Non-emptiness)*
2. *If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$. (Heredity)*
3. *If $I_1, I_2 \in \mathcal{I}$ and $|I_1| < |I_2|$, then there exists an element $i \in I_2 - I_1$ such that $I_1 \cup \{i\} \in \mathcal{I}$. (Exchange property)*

The family of subsets of N that belongs to $\mathcal{I}$ is called **independent set**; all other subsets of N are called **dependent set**. Maximal independent set of matroid $\mathcal{M}$ is called a **basis** of $\mathcal{M}$, set of all bases is denoted as $\mathcal{B}(\mathcal{M})$

To define a matroid model, we take the set of agents $N := \{1, 2, \ldots, n\}$ as the ground set of a matroid. A base B of a matroid is the maximal independent set of the matroid $\mathcal{M}$. There are many bases of the matroid $\mathcal{M}$. One of the salient features of the bases of a matroid is that they have the same number of elements. Let $\mathcal{B}$ be the set of all bases of the matroid $\mathcal{M}$. The set of bases of our matroid form the set of alternatives. Thus, a base is an alternative, which is a subset of agents. Defining the set $\mathcal{B}$ appropriately defines a specific problem.

Every agent $i \in N$ has a weight that we call the valuation of the agent. The valuation of the agent is his private information. Like in the previous section, let $V_i = (0, \beta_i)$, where $\beta_i \in \mathbb{R}_{++} \cup \{\infty\}$ be the set of all possible valuations of agent i and $V := V_1 \times V_2 \times \cdots \times V_n$ be the space of valuation profiles. As before v_{-i} and V_{-i} are the vector of valuations of all agents and valuation profiles of agents except agent i respectively.

Define an allocation rule as a mapping $f : V \rightarrow \mathcal{B}$. Notice that now an allocation rule picks a base of the matroid $\mathcal{M}$. Payment for an agent $i \in N$ is a mapping $p_i : V \rightarrow \mathbb{R}$. Thus, a matroid mechanism is a tuple (f, p) where $p = (p_1, \ldots, p_n)$ is a collection of payment functions. The definitions of implementability and monotonicity are exactly the same as in the connected graph model. Only the interpretation of allocation rule has been changed. Now it is a base of a matroid, earlier it was a spanning tree.

3.3.1 The Complete Characterization For The Matroid Model

In this section, we provide a complete characterization of implementable allocation rules for the matroid model. Like the connected graph model, we define top single crossing and generalized utility maximizer for this model.

Define a **co-circuit** C^* of a matroid $\mathcal{M}$ as follows: A set C^* is a co-circuit of $\mathcal{M}$ if and only if C^* has a minimal non-empty intersection with every basis of $\mathcal{M}$. Notice the connection between co-circuit of a matroid and a cut of a connected graph. Every cut of a graph intersects minimally with a spanning tree[4] like every co-circuit of a matroid intersects with every basis of a matroid. An element is called a **heavy element** of a co-circuit C^* if it has the highest weight among all the elements of the co-circuit. There is an algorithm which uses this property of co-circuits to find out the maximum weight base (it was developed by Dawson (1980)). We will use this also for our proof.

Define a generalized utility function (GUF) $u_i : V \rightarrow \mathbb{R}$ and call it **generalized utility function (GUF)**. Note that the the utility function is dependent on every agent's valua-

[4] A base corresponds to a spanning tree for the the connected graph model.

tions and may be negative. We impose the following condition on the collection of GUFs $(u_1, u_2, \dots, u_n)$.

Definition **21** *The GUFs* $(u_1, u_2, \dots, u_n)$ *satisfy* **top single crossing** *if for every* $i \in N$*, for every* $v_{-i} \in V_{-i}$*, for every* $v_i, v_i' \in V_i$ *with* $v_i > v_i'$*, for some co-circuit* C^* *of a matroid* $\mathcal{M}$ *such that* $i \in C^*$ *and* $u_i(v_i', v_{-i}) \geq \max_{k \in C^* \setminus \{i\}} u_k(v_i', v_{-i})$*, then there exists a* C^{**} *such that* $i \in C^{**}$ *and* $u_i(v_i, v_{-i}) > \max_{k \in C^{**} \setminus \{i\}} u_k(v_i, v_{-i})$*.*

Similar to the connected graph, we introduce a class of implementable allocation rules for general matroid model.

Definition **22** *An allocation rule* f *is a* **generalized utility maximizer** *if there exist GUFs* $(u_1, u_2, \dots, u_n)$ *satisfying top single crossing such that for every* $v \in V$*,* $f(v) \in \arg\max_{B \in \mathcal{B}} \sum_{i \in N} u_i(v_i, v_{-i}) \mathbb{1}_i^B$*, where* $\mathbb{1}_i^B = 1$ *if* $i \in B$ *and* $\mathbb{1}_i^B = 0$ *if* $i \notin B$*.*

We can also prove a lemma like Lemma 10 showing that if GUFs satisfy single crossing and u_i is increasing, then they satisfy top single crossing for the matroid model.

Now we state the main result which shows that generalized utility maximizers are the only implementable allocation rules in the matroid model.

Theorem **5** *An allocation rule is implementable if and only if it is a generalized utility maximizer.*

Proof: The proof of this theorem is similar to the proof of Theorem 4. See the details in Appendix B. ■

Our result says that given the valuations of the agents, we can construct utility functions for every agent such that these utility functions depend not only on an agent's own valuation but also on the valuations of all other agents. After constructing the utility functions, the allocation rule maximizes the sum of utilities and this generalized utility maximizer is equivalent to implementable allocation rules. Generalized utility maximizers are similar to implementing the efficient allocation rules in the interdependent values setting (Cremer and McLean, 1985; Maskin, 1992; Dasgupta and Maskin, 2000; Perry and Reny, 2002).

Like in the previous section, revenue equivalence principle holds here too. Therefore, by characterizing the implementable allocation rules, we characterize the set of dominant strategy incentive compatible mechanisms by the revenue equivalence principle.

This is a very general result in a sense that it covers many single dimensional type spaces like multi-unit auction model with unit demand, the connected graph model, heterogeneous good auction model with dichotomous preferences and single object auction model. In the literature, the VCG mechanism has been characterized in various single dimensional domains considering the following conditions - *anonymity in utility*, implementability, *individual rationality* and *non-negative payments*. We characterize all dominant strategy incentive compatible mechanisms without any extra conditions. We discuss various applications of this model in the next subsection.

3.3.2 Applications

The matroid model unifies many single dimensional mechanism design problems. Now, we discuss various applications of our result.For instance, the connected graph model, the multi-unit auction model, heterogeneous good auction with dichotomous preferences model all can be formulated as a specific types of matroid problem.

3.3.2.1 The connected Graph Model

We can formulate the connected graph model as a matroid known as the *graphic matroid*. Let $G = (M, N)$ be a connected graph and define a set $\mathcal{I} = \{F \subseteq N : (M, F) \text{ is a tree}\}$. It can easily be verified that the set $\mathcal{I}$ satisfies all the three axioms of matroid. Thus, the set $\mathcal{I}$ constitutes an independent set of the matroid where N is the ground set which is the set of edges. Therefore, a *graphic matroid* is the matroid $(N, \mathcal{I})$ where N is the set of edges and $\mathcal{I}$ is the set of all trees of the graph G. Since we have assumed that each agent owns a unique edge, N also denotes the set of agents.

Notice that the bases of the graphic matroid is the set of all spanning trees of the graph G. Hence, our result applies here and Theorem 4 is a corollary to Theorem 5.

3.3.2.2 Multi-unit Auction With Unit Demand

Consider the problem where k homogeneous units of a good are available to be sold and there are n agents. Let the set of agents be $N = \{1, 2, \ldots, n\}$. Every agent i demands a single unit and has some private value v_i for the unit. Assume $n \geq k$. Our objective is to characterize incentive compatible mechanisms to allocate the objects to agents.

We show that this problem can be formulated as a matroid known as the *uniform* matroid. Let N be a ground set. Define the following collection of subsets of N by:

$$\mathcal{I} = \{X \subseteq N : |X| \leq k\} \tag{3.1}$$

It can be easily verified that the set $\mathcal{I}$ satisfies the three axioms of independent set. Hence, $\mathcal{I}$ is an independent set of the matroid on N.

Thus, $(N, \mathcal{I})$ is a matroid. This is called the uniform matroid. A basis here is a subset of N of size k and the set of bases is $\mathcal{B} = \{X \subseteq N : |X| = k\}$. Notice that a basis is a subset of agents who can be allocated all the units. Hence, this auction setting is a special type of a matroid problem and our result applies here.

A very special case of this problem is when only a single unit is for sale. In this case, consider the set $\mathcal{B}$ as the collection of 1-element subsets of N. Clearly, it is a matroid and $\mathcal{B}$ is the set of bases. So, our result also applies to the single object auction model.

3.3.2.3 *Heterogeneous Good Auction With Dichotomous Preferences*

In this model, there is a finite set of items, denoted by $M = \{1, \ldots, m\}$. Each agent $i \in N$, has a set $D_i \subseteq M$ that he desires. The interpretation of D_i is that agent i realizes his valuation v_i if and only if he gets an item from D_i. If agent i is given an item outside D_i, then he gets zero utility from it.

We assume that for every agent i, his desired set of items D_i is common knowledge. An alternative in this model is a **matching** of items to agents such that no item is assigned to more than one agent and no agent is assigned more than one item. We assume that the collection of desired set of items $\{D_1, \ldots, D_n\}$ is such that there exists a matching where each item in M can be allocated to an agent who desires it. We call such a matching a **perfect matching**. The set of all perfect matchings is the set of alternatives in this model. So, an allocation rule chooses a perfect matching in this problem.

We argue now that the set of alternatives correspond to the set of bases of a matroid. The fact that the set of perfect matchings form a matroid is well known (Oxley, 2006)) - these are called **transversal matroids**. We formally define transversal matroids in Appendix A.

The following example explains the ideal of perfect matching.

Example **7**

Consider the problem where two non-identical items are for sale and there are three buyers. Thus, $M = \{a, b\}$ and $N = \{1, 2, 3\}$. Suppose that the desired set of items for agents $1, 2$ and 3 are $D_1 = \{a, b\}$, $D_2 = \{a\}$ and $D_3 = \{a, b\}$ respectively. We can construct a bipartite graph like in Figure 3.2.

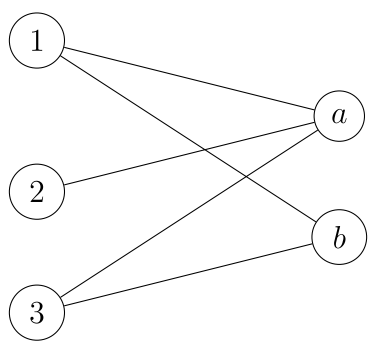


Figure 3.2: Bipartite graph to explain perfect matching.

We denote when agent i is matched with item a by ia. In Figure 3.2, there are four perfect matchings.They are as follows, $\{1a, 3b\}$, $\{1b, 2a\}$, $\{1b, 3a\}$ and {2a, 3b}. Thus, we see that as we determine a perfect matching, we also determine the subset of agents who are matched in it.

Notice that in our matroid model, each base corresponded to a unique subset of agents but in this matroid, each base correspond to a unique perfect matching. This means that the same subset of agents may be involved in two different bases (alternatives) of the matroid. For example, in Figure 3.2, we see that the subset of agents $\{1, 3\}$ is matched in to ways $\{1a, 3b\}$ and $\{1b, 3a\}$. But this does not create additional complications since two perfect matchings that involve the same set of agents will give the exact same utility to the agents. So, these bases can be treated as equivalent. Thus, we can apply Theorem 5 to this model also.

Note that this problem can not be formulated in the uniform matroid. To see this, modify the desired set of agent 3 in the above example as $D_3 = \{a\}$ keeping all other features same. We have the following bipartite graph with this modification.

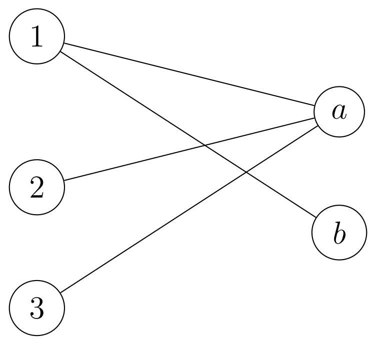


Figure 3.3: Bipartite graph for two items and three agents.

But in the graph in Figure 3.3, we see that $\{2, 3\}$ can not be matched perfectly. Therefore, it can not be an alternative in this model. However, it forms a basis in the uniform matroid. This is because we have defined the set of bases in the uniform matroid as the family of all subsets of N which has m elements.

3.4 Anonymity and Efficient Allocation Rule

In this section, we characterize a specific class of dominant strategy incentive compatible mechanisms. In particular, we characterize the Groves mechanisms by imposing two more plausible axioms on mechanisms on top of incentive compatibility.

A standard exercise in mechanism design is to characterize the mechanism imposing some "good" criteria. One of the good criteria is fairness. Thus, we want to impose *anonymity in utility*, a fairness notion on mechanism.

Several papers in the literature focus on characterizing the VCG mechanism in specific models by imposing axioms like *anonymity in utility*, incentive compatibility and *individual rationality* - (Ashlagi and Serizawa, 2011; Mukherjee, 2014). But in our setting, especially for the connected graph model, there is incompatibility between efficiency and anonymity in utility. We discuss this tension later. We only give the characterization for the connected graph model. The extension to a matroid model can be done straightforwardly.

In auctioning of k multi-unit homogeneous goods, we pick the k highest valuation bidders in an efficient allocation rule. But in the connected graph problem, though an allocation rule picks $(m-1)$ agents at every valuation profile (m is the number of nodes), the efficient allocation rule need not pick the $(m-1)$ highest valuation agents. The following example illustrates this:

EXAMPLE **8**

Consider the graph in Figure 3.4 with four nodes and five edges where the number on edges is the valuation of agents who own them.

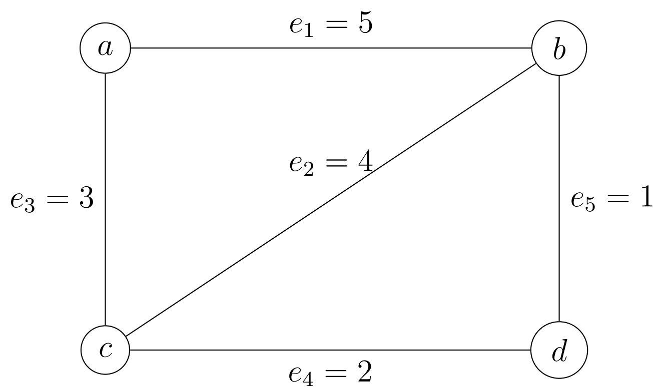


Figure 3.4: Illustration of not top 3 edges will be chosen.

Let $v = (5, 4, 3, 2, 1)$ be a valuation profile of agents. We have to pick exactly 3 edges to get a spanning tree in this graph because it has four nodes. If we pick the top three valuations of agents in this graph, then they form a cycle. Thus, efficient allocation can not pick top three valuations in this graph.

However, if we consider that there are three copies of a good and 5 agents who are interested in buying at most one unit. A valuation profile of agents is $v = (5, 4, 3, 2, 1)$. Then, efficient allocation rule will allocate one copy each to the agents who have valuations 5, 4 and 3 respectively.

This particular property of a maximum value weight spanning tree makes implementability, *anonymity in utility* and *efficiency* incompatible.

An efficient allocation rule f^* is a spanning tree that carries the maximum weight for every valuation profile v.

DEFINITION **23** *An allocation rule f^* is efficient if for every $v \in V$*

$$\sum_{i \in N} v_i \mathbb{1}_i^{f^*(v)} \geq \sum_{i \in N} v_i \mathbb{1}_i^x \qquad \textit{for all } x \in X$$

Note that we can have more than one efficient allocation rule if agents do not have distinct valuations.

DEFINITION **24** *A mechanism (f, p) is anonymous in utility if for every $v = (v_1, v_2, \dots, v_n)$, for every $i, j \in N$ and for every v' such that $v'_i = v_j$, $v'_j = v_i$, $v'_k = v_k$ for all $k \notin \{i, j\}$, we have*

$$v'_i \mathbb{1}_i^{f(v')} - p_i(v') = v_j \mathbb{1}_j^{f(v)} - p_i(v)$$

and

$$v'_j \mathbb{1}_j^{f(v')} - p_j(v') = v_i \mathbb{1}_i^{f(v)} - p_i(v).$$

Anonymity in utility implies that if two agents exchange their valuations, then their net utilities are also exchanged.

DEFINITION **25** *A mechanism (f, p) is individually rational if for every $i \in N$ and every $v \in V$, we have $v_i \mathbb{1}_i^{f(v)} - p_i(v) \geq 0$.*

We allow only non-negative payment for every agent, i.e., $p_i(v) \geq 0$ for every $i \in N$. To show incompatibility, we need the following lemmas which are consequences of anonymity in utility, individual rationality, incentive compatibility, and non-negative payment.

LEMMA **12** *Let (f, p) be an individually rationality and incentive compatible mechanism. For every $v \in V$ if $\mathbb{1}_i^{f(v_i, v_{-i})} = 0$, then $p_i(v_i, v_{-i}) = 0$.*

Proof: From individual rational of mechanism (f, p), we have $v_i \mathbb{1}_i^{f(v)} - p_i(v) \geq 0$. Since $\mathbb{1}_i^{f(v)}$, we have $p_i(v) \leq 0$. From the assumption of non-negative payment, we have $p_i(v) \geq 0$. Hence, we have $p_i(v) = 0$. ■

LEMMA **13** *If a mechanism (f, p) is incentive compatible, then for every $i \in N$, for every $v \in V$ and for every $v'_i \in V_i$ with $\mathbb{1}_i^{f(v'_i, v_{-i})} = \mathbb{1}_i^{f(v_i, v_{-i})}$, we have $p_i(v'_i, v_{-i}) = p_i(v_i, v_{-i})$.*

It is a well-know result in the literature. We give its proof in Appendix B.

LEMMA **14** *Let (f, p) be an anonymous in utility, individually rational and incentive compatible mechanism. For any $i, j \in N$, $v_i \in V_i$, $v_j \in V_j$ and $v_{-ij} \in V_{-ij}$ with $v_i < v_j$ if $\mathbb{1}_i^{f(v_i, v_j, v_{-ij})} = 1$, then $\mathbb{1}_j^{f(v'_i, v_j, v_{-ij})} = 1$ when $v'_i = v_j$.*

Proof: Consider an anonymous in utility, individually rational and incentive compatible mechanism (f, p). Fix $i, j \in N$, $v_i \in V_i$, $v_j \in V_j$, $v_{-ij} \in V_{-ij}$ with $v_i < v_j$ and denote $v'_i, = v_j$. Suppose $\mathbb{1}_i^{f(v_i, v_j, v_{-ij})} = 1$.

Incentive compatibility implies that f is monotone. Therefore, we have $\mathbb{1}_i^{f(v'_i, v_{-i})} = 1$ by monotonicity. Thus, $p_i(v'_i, v_j, v_{-i}) = p_i(v_i, v_j, v_{-ij})$ by Lemma 13. Assume for contradiction that $\mathbb{1}_j^{f(v_i',v_j,v_{-ij})} = 0$. By individual rationality and non-negative payment, $v_j\mathbb{1}_j^{f(v_i',v_j,v_{-ij})} - p_j(v_i', v_j, v_{-ij}) = 0$. Since $v_i' = v_j$, by anonymity in utility we have

$$v_i'\mathbb{1}_i^{f(v_i',v_{-i})} - p_i(v_i', v_{-i}) = v_j\mathbb{1}_j^{f(v_i',v_j,v_{-ij})} - p_j(v_i', v_j, v_{-ij}) = 0. \tag{3.2}$$

Define $v_j' = v_i$. We have $\mathbb{1}_j^{f(v_i',v_{-i})} = 0$ by assumption. Thus, $\mathbb{1}_j^{f(v_i',v_j',v_{-ij})} = 0$ by monotonicity (Since $v_j' = v_i > v_j$). By Lemma 12, we have $v_j'\mathbb{1}_j^{f(v_i',v_j',v_{-ij})} - p_j(v_i', v_j', v_{-ij}) = 0$. Exchanging v_i' with v_j' and applying anonymity in utility we have,

$$v_j'\mathbb{1}_i^{f(v_j',v_i',v_{-ij})} - p_i(v_j', v_i', v_{-ij}) = v_j'\mathbb{1}_j^{f(v_i',v_j',v_{-ij})} - p_j(v_i', v_j', v_{-ij}) = 0 \tag{3.3}$$

Substituting the value $v_j' = v_i$ and $v_i' = v_j$, we have from equation 3,

$$v_i\mathbb{1}_i^{f(v_i,v_j,v_{-ij})} - p_i(v_i, v_j, v_{-ij}) = 0. \tag{3.4}$$

From equations 2 and 4, we have $v_i = v_i'$, a contradiction. ■

Now, we use these lemmas and show that anonymity in utility, incentive compatibility and efficiency are incompatible. Consider the following example which shows this incompatibility.

EXAMPLE **9**

Consider the graph in Figure 3.5 with four nodes and five edges. The numbers on edges are valuations of agents. Let $v = (5, 5, 5, 3, 2)$.

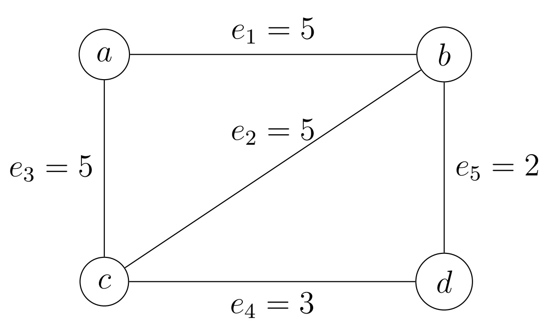


Figure 3.5: Illustration of incompatibility of anonymity in utility and efficiency

An efficient allocation rule f^* must choose $\{e_1, e_2, e_4\}$ at valuation profile v. Now, if we increase the valuation of edge e_4 and equate it with valuation of edge e_3, then by Lemma 14 edge e_3 will be in allocation rule at the new valuation profile, $\bar{v} = (5, 5, 5, 5, 2)$. Since the valuation of edges e_1, e_2 and e_3 are equal, by a similar argument as in Lemma 14, we conclude that edges e_1 and e_2 are chosen by the efficient allocation rule f^* at $\bar{v}$. But this is not possible because $f^*(\bar{v})$ forms a cycle.

Anonymity in utility has been used for characterizing the VCG mechanism in various setting like multi-unit auction with unit demand (Ashlagi and Serizawa, 2011) and single object auction (Mukherjee, 2014). As Example 9 illustrated, this is not possible in our model. Hence, we define a different version of anonymity which we call *restricted anonymity*.

First, we may consider weakening anonymity in utility to anonymity in allocation. One natural way to define anonymity in allocation is that if agent i swaps his valuation with the valuation of agent j, then their allocations are swapped. But defining anonymity in allocation like this is not consistent with every permutation as for some permutation, we may not get a spanning tree. The following example illustrates this.

Example **10**

Consider the graph in Figure 3.5. Suppose that at some valuation profile $(v_1, v_2, v_3, v_4, v_5)$, $f(v_1, v_2, v_3, v_4, v_5) = \{e_1, e_2, e_4\}$. Now, consider another valuation profile after exchanging the valuation of agent 3 with agent 5, i.e., $(v_1, v_2, v_5, v_4, v_3)$. By anonymity in allocation, we have to choose edge e_5 in the allocation rule at this new valuation profile. Now, we have $f(v_1, v_2, v_5, v_4, v_3) = \{e_1, e_2, e_5\}$ at this new valuation profile. But it is not a spanning tree.

Now, we introduce our version of anonymity that we call restricted anonymity. A **permutation** of the set of agents is a map $\sigma : N \to N$ such that it is is one-to-one and onto. Denote by $\sigma(v) = (v_{\sigma(1)}, \dots, v_{\sigma(n)})$ the valuation vector of agents after the permutation. Given allocation rule f, for every $v \in V$ we say that a permutation σ is **valid** at v if $\sigma(f(v))$ is a spanning tree where we denote by $\sigma(f(v)) = (\mathbb{1}^{f(v)}_{\sigma(1)}, \dots, \mathbb{1}^{f(v)}_{\sigma(n)})$, the collection of edges. Note that $\sigma(f(v))$ need not be a spanning tree.

Definition **26** *An allocation rule f is* **restricted anonymous** *if for every $v \in V$ such that $\sigma(v)$ is a valid permutation and $v \neq \sigma(v)$, we have $f(\sigma(v)) = \sigma(f(v))$.*

This says that if agent i exchanges his valuation with the valuation of agent j, then their allocations are also exchanged if this allocation exchange does form a spanning tree. This implies that the identity of agents does not matter.

We also need the following non-bossiness condition for our characterization.

Definition **27** *An allocation rule f is* **non-bossy** *if for every $i \in N$, for every $v_{-i} \in V_{-i}$ and for every $v_i, v'_i \in V_i$ with $\mathbb{1}_i^{f(v_i, v_{-i})} = \mathbb{1}_i^{f(v'_i, v_{-i})}$, we have $f(v_i, v_{-i}) = f(v'_i, v_{-i})$.*

This assumption requires that if an agent cannot change his allocation in the allocation rule by changing his valuation, then he should not be able to change the allocation of other agents. Now, we present a proposition that helps us in characterizing the Groves mechanism.

As our definition indicates that we only consider the non-bossiness in allocation. We are not considering non-bossiness in outcome as first proposed by Satterthwaite and Sonnenschein (1981).

Proposition **3** *If an allocation rule is implementable, non-bossy and, restricted anonymous, then it is an efficient allocation rule.*

Proof: We use the following lemmas to prove the proposition. The proofs of the lemmas are in Appendix B.

LEMMA **15** *A maximum weight spanning tree contains at least one heavy edge for every cut of the graph. Moreover, every edge in maximum weight spanning tree is a heavy edge for some cut.*

We call a cut "loose" at v if $f(v)$ does not choose any heavy edge from the cut.

LEMMA **16** *For every valuation profile $v \in V$ and for any loose cut $(S, M \backslash S)$ at v if $i \in N(S)$ and $i \in f(v_i, v_{-i})$, there exists $j \in N(S)$ such that j is a heavy edge and $f(v) \setminus \{i\} \cup \{j\}$ is a spanning tree.*

Consider an allocation rule f which is implementable, non-bossy and, restricted anonymous. Picking an efficient allocation is equivalent to picking a maximum weight spanning tree for this problem. We know by Lemma 15 that a maximum weight spanning tree is the collection of heavy edges of different cuts of the graph. Therefore, if we can show that an allocation rule f contains a heavy edge of every cut, then we are done.

If the valuations of all the agents are the same, then we are trivially done. Else, consider valuation profile (v_i, v_{-i}). Consider a cut of the graph $(S, M \setminus S)$. Assume for contradiction that no heavy edge of the cut $(S, M \setminus S)$ is chosen by allocation rule f at valuation profile (v_i, v_{-i}). Let i be one of the heavy edges of the cut $(S, M \setminus S)$. Thus, $i \notin f(v_i, v_{-i})$. For f to be a spanning tree at (v_i, v_{-i}), there must be an edge $j \in N(S)$ such that j is not a heavy edge and $j \in f(v)$.

We can apply permutation σ at v in the cut $(S, M \setminus S)$, if there exists an edge $j \in N(S)$ such that j is not heavy edge and permutation of the valuation of agent j with the valuation of agent i is valid. By Lemma 16 such an edge $j \in N(S)$ exists and it is not one of the heavy edges.

Since i is a heavy edge, suppose $v_i = \alpha > v_j = \beta$. Since $j \in f(v_i = \alpha, v_j = \beta, v_{-ij})$ and $\alpha > \beta$, by monotonicity $j \in f(v_i = \alpha, v'_j = \alpha, v_{-ij})$ for $v'_j = \alpha$. By non-bossiness, $i \notin f(v_i = \alpha, v'_j = \alpha, v_{-ij})$ and $j \in f(v_i = \alpha, v'_j = \alpha, v_{-ij})$.

By restricted anonymity, we have $i \in f(v'_i = \beta, v'_j = \alpha, v_{-ij})$ where $v'_i = \beta$ and $j \notin f(v'_i = \alpha, v'_j = \beta, v_{-ij})$. Since $\beta < \alpha$, we have $i \in f(v_i = \alpha, v'_j = \alpha, v_{-ij})$ by monotonicity. By non-bossiness, we have $j \notin f(v_i = \alpha, v'_j = \alpha, v_{-ij})$. But this contradicts the fact that $j \in f(v_i = \alpha, v'_j = \alpha, v_{-ij})$. Hence, f must select one of the heavy edges from every cut of the graph. By Lemma 15, this is a maximum weight spanning tree. ■

Using revenue equivalence and Proposition 3, we can now characterize the Groves mechanisms.

DEFINITION **28** *A mechanism (f, p) is a Groves mechanism if for every $v \in V$, f is an efficient allocation rule and for every $i \in N$ $p_i(v_i, v_{-i}) = \sum_{j \neq i} v_j \mathbb{1}_j^{f^*(v_i, v_{-i})} + h_i(v_{-i})$, where $h_i : V_{-i} \to \mathbb{R}$ is any function.*

Now, we state the main result of this section.

Theorem **6** *A mechanism (f, p) is incentive compatible where f is non-bossy and restricted anonymous if and only if it is the Groves mechanisms.*

Proof: By Proposition 3, every allocation rule satisfying implementability, non-bossiness and, restricted anonymity must be an efficient allocation rule. By revenue equivalence (Holmstrom, 1979), the only efficient incentive compatible mechanisms are the Groves mechanisms. ■

Remark 1: Proposition 3 and Theorem 6 can be easily generalized to the matroid model as in section 2. We only have to define the restricted anonymity properly. Like in the connected model, we need valid permutation to define the restricted anonymity in the matroid model. We can adapt the terminology of the matroid theory to define it. After performing a permutation on the valuations of agents if we get a base, then the permutation is valid. The definition of non-bossy is straightforward in the matroid model. With this adaptation of the restricted anonymity, Proposition 3 can easily be proved and hence Theorem 6.

Remark 2: The requirements of incentive compatibility of mechanism and and allocation rule satisfying restricted anonymity is crucial for proving Proposition 3. We can easily construct examples showing that they are independent. As far as the non-bossiness axiom is concerned, we can not establish its independence for proving efficient allocation rule.

3.5 Conclusion

We characterize all dominant strategy incentive compatible mechanisms for a class of mechanism design problems in single dimensional type spaces. In particular, we consider two mechanism design problems, the connected graph model and the matroid model and find a broader class of implementable allocation rules. By virtue of revenue equivalence principle, we pin down the entire class of dominant strategy incentive compatible mechanisms in both the problems. The matroid model unifies many results like the connected graph model, multi-unit auction with unit demand and heterogeneous good auction with dichotomous preferences, which have been looked in specific models. We also characterize the Groves mechanisms with two extra conditions of restricted anonymity and non-bossiness along with dominant strategy incentive compatibility.

There are many future research directions. One important question is to explore a suitable version of anonymity in the matroid model and characterize the VCG mechanism. Another question is to look for a parallel result of our Theorem 4 and 5 in multi-dimensional mechanism design problem.

Appendix A

We provide some elementary concept of matroid theory and use the standard notation (see Oxley (2006)). You can also find a very nice introduction of matroid theory in Vohra (2004) and an intuitive description in Wilson (1973).

A minimal dependent set in an arbitrary matroid $\mathcal{M}$ is called a **circuit**. The set of circuits is denoted as $\mathcal{C}(\mathcal{M})$. We need the following lemma about a property of the bases,

Lemma **17** *For every $B_i, B_j \in \mathcal{B}$ for $i \neq j$, we have $|B_i| = |B_j|$.*

This lemma says that every base of a matroid has the same cardinality. The proof of this lemma is pretty standard in matroid theory and can be found in standard matroid book like Oxley (2006), Vohra (2004) etc.

The dual of a matroid also exists and in fact it is a matroid. There is a strong relationship between matroid and its dual.

Theorem **7** *Let $\mathcal{M}$ be a matroid and $\mathcal{B}^*(\mathcal{M}) = \{N(\mathcal{M}) - B : B \in \mathcal{B}(\mathcal{M})\}$. Then $\mathcal{B}^*(\mathcal{M})$ is the set of bases of a matroid on $N(\mathcal{M})$.*

Proof: See Oxley (2006), page number 68. ∎

The matroid generated in the last theorem, whose ground set is N and whose set of bases $\mathcal{B}^*(\mathcal{M})$, is called the **dual** of matroid $\mathcal{M}$ and is denoted by $\mathcal{M}^*$. As we have defined basis and circuit of matroid $\mathcal{M}$, we can also define the basis and circuit of the matroid $\mathcal{M}^*$. The bases and circuits of the matroid $\mathcal{M}^*$ are known co-basis and co-circuit of the matroid $\mathcal{M}$ respectively. We denote an arbitrary base and co-circuit as B^* and C^*. The following property pins down a very nice relationship between base and co-circuit of a matroid $\mathcal{M}$,

Theorem **8** *Let $\mathcal{M}$ be a matroid.*

1. *A set C^* is a co-circuit of the matroid $\mathcal{M}$ if and only if C^* is a minimal set having non-empty intersection with every basis of $\mathcal{M}$.*

2. *A set B is a basis of $\mathcal{M}$ if and only if B is a minimal set having non-empty intersection with every co-circuit of $\mathcal{M}$.*

Transversal Matroids

Let $N = \{x_1, x_2, \ldots, x_n\}$ be a finite set. A family of subsets of the set N is a finite sequence $(A_1, A_2, \ldots, A_m)$ such that for all $j \in M = \{1, 2, \ldots, m\}$, $A_j \subseteq N$. Note that the terms in this sequence may not be distinct.

We define *partial transversal* in terms of a *matching* in a bipartite graph. Let $\mathcal{A}$ be the family $(A_1, A_2, \ldots, A_m)$ of subsets of N and M. A bipartite graph $G(\mathcal{A})$ associated with $\mathcal{A}$ has vertex set $N \cup M$. Each edge of $G(\mathcal{A})$ is $\{x_i j : x_i \in N \text{and } j \in M\}$.

A *matching* in a graph is a set of edges of the graph no two of which have a common end point. A subset X of N is a **partial transversal** of $\mathcal{A}$ if and only if there is a matching in $G(\mathcal{A})$ in which every edge has one end point in X. If the matching is perfect, then we have a transversal.

The following example explains the idea of matching.

EXAMPLE **11**

Let $N = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $A_1 = \{x_1, x_2, x_6\}$, $A_2 = \{x_3, x_5, x_6\}$, $A_3 = \{x_2, x_3\}$ and $A_4 = \{x_2, x_4, x_6\}$. For $\mathcal{A} = (A_1, A_2, A_3, A_4)$, the bipartite graph is given in Figure 3.6 below.

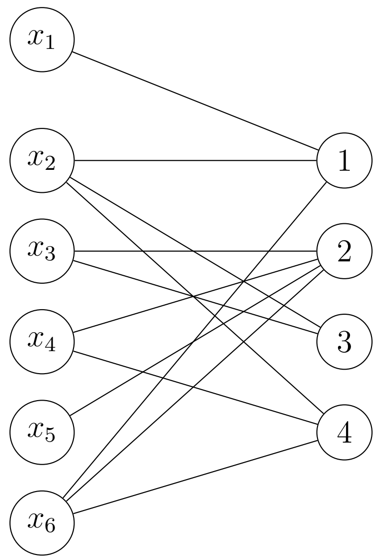


Figure 3.6: Bipartite graph.

We can see that $\{x_1, x_2, x_3, x_4\}$ is a transversal of $\mathcal{A}$. To verify that it is a transversal, we need only to check that $\{x_1 1, x_4 2, x_3 3, x_2 4\}$ is a matching in $G(\mathcal{A})$. There are many partial transversals as well. For instance, $\{x_1, x_2, x_6\}$ and $\{x_6, x_2, x_4\}$ are partial transversals of $\mathcal{A}$ because $\{x_1 1, x_6 2, x_2 3\}$ and $\{x_6 1, x_2 3, x_4 2\}$ are matchings in $G(\mathcal{A})$. There are some other partial transversals as well.

The following theorem from Oxley (2006) shows that partial transversals are the independent sets.

THEOREM **9** *Let $\mathcal{A}$ be a family of $(A_1, A_2, \ldots, A_m)$ of subsets of N. Let $\mathcal{I}$ be the set of partial transversal of $\mathcal{A}$. Then, $\mathcal{I}$ is a collection of independent sets of a matroid on N.*

Such matroids are called *transversal matroids*.

Appendix B

Proof of Theorem 5

First we prove that generalized utility maximizer is implementable. Consider a generalized utility maximizer f and let $(u_1, u_2, \ldots, u_n)$ be the corresponding GUFs satisfying top single crossing. Fix an agent $i \in N$ and $v_{-i} \in V_{-i}$. Let $v_i, v_i' \in V_i$ with $v_i > v_i'$ and consider a co-circuit C^* of the matroid $\mathcal{M}$ such that $i \in C^*$. Suppose $i \in f(v_i', v_{-i})$. This implies that i must be one of the heavy elements for some co-circuit C^*. By the top single crossing, there exists a co-circuit C^{**} such that $i \in C^{**}$ and $u_i(v_i, v_{-i}) > \max_{k \in C^{**} \setminus \{i\}} u_k(v_i, v_{-i})$. Hence, i is the unique heavy element of the co-circuit C^{**}. Since a maximum weight base must choose heavy element from every co-circuit, $i \in f(v_i, v_{-i})$ - see Oxley (2006).

To prove the converse, suppose f is implementable. Fix an agent $i \in N$ and $v_{-i} \in V_{-i}$. Define $\kappa_i^f(v_{-i}) := \inf \{v_i \in V_i : \mathbb{1}_i^f(v_i, v_{-i}) = 1\}$. Since V_i is a bounded interval, κ_i^f is well-defined. Further, since f is implementable and hence monotone, for every $i \in N$, for every $v_{-i} \in V_{-i}$ and for every $v_i \in V_i$ if $v_i \geq \kappa_i^f(v_{-i})$, we have $\mathbb{1}_i^f(v_i, v_{-i}) = 1$ and for every $v_i < \kappa_i^f(v_{-i})$, we have $\mathbb{1}_i^f(v_i, v_{-i}) = 0$.

Define for every $i \in N$ and for every (v_i, v_{-i}),

$$u_i(v_i, v_{-i}) = v_i - \kappa_i^f(v_{-i}).$$

By definition, for every $i \in f(v_i, v_{-i})$, we have $u_i(v_i, v_{-i}) \geq 0$ and for all $i \notin f(v_i, v_{-i})$, we have $u_i(v_i, v_{-i}) < 0$. Hence, we have $f(v_i, v_{-i}) \in \arg\max_{B \in \mathcal{B}} \sum_{i \in N} u_i(v_i, v_{-i}) \mathbb{1}_i^B(v_i, v_{-i})$.

Now, we show that the GUFs $(u_1, u_2, \ldots, u_n)$ constructed as above satisfy top single crossing. Fix an agent $i \in N$ and $v_i, v_i' \in V_i$. Consider $v_{-i} \in V_{-i}$ and a co-circuit C^* of the matroid $\mathcal{M} = (N, \mathcal{I})$ such that $i \in C^*$. Let $u_i(v_i', v_{-i}) \geq \max_{k \in C^* \setminus \{i\}} u_k(v_i', v_{-i})$. By the construction of GUF u_i, we have $u_i(v_i, v_{-i}) = v_i - \kappa_i^f(v_{-i}) > v_i' - \kappa_i^f(v_{-i}) = u_i(v_i', v_{-i})$. Hence, $u_i(v_i, v_{-i}) > u_i(v_i', v_{-i})$. This implies that u_i is an increasing function. We know that for some C^* such that $i \in C^*$, we have $u_i(v_i', v_{-i}) \geq \max_{k \in C^* \setminus \{i\}} u_k(v_i', v_{-i})$. Suppose $i \in f(v_i', v_{-i})$, then we have $u_i(v_i', v_{-i}) \geq 0$. Suppose $i \notin f(v_i', v_{-i})$, then some other element $j \in C^*$, we have $j \in f(v_i', v_{-i})$ (This is because every basis must choose at least one element from every co-circuit - see Oxley (2006)). This implies that $u_i(v_i', v_{-i}) \geq u_j(v_i', v_{-i}) \geq 0$. Therefore, for both the cases, we have $u_i(v_i', v_{-i}) \geq 0$. Since $u_i(v_i, v_{-i}) > u_i(v_i', v_{-i}) \geq 0$. This implies that $v_i > \kappa_i^f(v_{-i})$ and hence $i \in f(v_i, v_{-i})$.

Since $f(v_i, v_{-i}) \in \arg\max_{B \in \mathcal{B}} \sum_{i \in N} u_i(v_i, v_{-i}) \mathbb{1}_i^B$, i belongs to a maximum weight base chosen by f at valuation profile (v_i, v_{-i}). This implies that there exists a co-circuit C^{**} such that for every $j \in C^{**} \setminus \{i\}$, we have $j \notin f(v_i, v_{-i})$.[5] This implies that for every $j \in C^{**} \setminus \{i\}$, we have $u_j(v_i, v_{-i}) = v_j - \kappa_i^f(v_{-j}) \leq 0 < u_i(v_i, v_{-i})$. This proves the claim.

[5]To see this, suppose for every co-circuit C^{**}, there is an element $j \in C^{**} \setminus \{i\}$ such that $j \in f(v_i, v_{-i})$, then we can remove i from $f(v_i, v_{-i})$ and the resulting independent set will still be a basis. This contradicts the fact that $f(v_i, v_{-i})$ is a basis.

Proof of Lemma 13

Consider an incentive compatible mechanism (f, p) and choose $v \in V$, $v_i, v'_i \in V_i$ and $i \in N$ with $\mathbb{1}_i^{f(v_i,v_{-i})} = \mathbb{1}_i^{f(v'_i,v_{-i})}$. Consider v_i as true value of agent i, then by incentive compatibility we have

$$v_i \mathbb{1}_i^{f(v_i,v_{-i})} - p_i(v_i, v_{-i}) \geq v_i \mathbb{1}_i^{f(v'_i,v_{-i})} - p_i(v'_i, v_{-i}) \tag{3.5}$$

Consider v'_i as true value of agent i, then by incentive compatibility we have

$$v'_i \mathbb{1}_i^{f(v'_i,v_{-i})} - p_i(v'_i, v_{-i}) \geq v'_i \mathbb{1}_i^{f(v_i,v_{-i})} - p_i(v_i, v_{-i}) \tag{3.6}$$

From equation 2, we have $p_i(v_i, v_{-i}) \leq p_i(v'_i, v_{-i})$ and from equation 3, we have $p_i(v_i, v_{-i}) \geq p_i(v'_i, v_{-i})$. Hence, we have $p_i(v'_i, v_{-i}) = p_i(v_i, v_{-i})$.

Proof of Lemma 15

Consider a a cut of the graph $(S, M \setminus S)$ and a spanning tree x. Let $N(S)$ be the set of all edges that cross the cut $(S, M \setminus S)$. Suppose no edge from this cut is chosen in spanning tree x, i.e., $x \cap \{N(S)\} = \emptyset$. But this is not possible because the remaining set of edges $x \setminus \{N(S)\}$ cannot span all the vertices of the graph. Hence, we have to choose at least one edge from every cut of the graph.

Now suppose no heavy edge from the cut $(S, M \setminus S)$ is chosen in x. Then, some edge $j \in N(S)$ such that j is not a heavy edge and $j \in x$. Now delete edge j and include one of the heavy edges, we will get a spanning tree. We can do this for every cut of the graph and get a spanning tree which is a collection of heavy edges of different cuts.

Proof of Lemma 16

Consider a valuation profile v and a loose cut $(S, M \setminus S)$. Since $(S, M \setminus S)$ is a loose cut, no heavy edge of this cut belongs to $f(v)$. For f to be a spanning tree at v, we have to choose at least one edge $k \in N(S)$ such that $k \in f(v_i, v_{-i})$. By definition k is not a heavy edge.

Let i be one of the heavy edges of $(S, M \setminus S)$. Now, if we add edge i to the spanning tree chosen by f at v, it will create a cycle, i.e., $f(v) \cup \{i\}$ is a cycle. Since $f(v) \cup \{i\}$ is a cycle, there exists an edge $j \in N(S) \setminus \{i\}$ such that $j \in f(v)$ and j is part of the cycle. Thus, if we remove j, we have $f(v) \setminus \{j\} \cup \{i\}$ which is a spanning tree.

Chapter 4

Single Object Auctions With Externalities: A Tractable Model

4.1 Introduction

In many situations, individuals enjoy some utility even if they do not own a particular object. For instance, objects like patents and paintings may have positive externalities on agents who do not own them. We study such a single object auction model where externalities are modeled in a specific manner.

We restrict attention to deterministic single object auction. An allocation rule for single object auction is implementable if there exists payments such that truth-telling is a dominant strategy for every agent. We identify a necessary and sufficient condition for implementable allocation rules in our model. Using revenue equivalence, we characterize all the dominant strategy incentive compatible mechanisms. We use our characterization to design a revenue maximizing auction (optimal auction) for this model. In the optimal auction, every agent makes some payment irrespective of whether he is getting the object or not.

The innovative feature of this paper lies in the way we model externalities. Imagine a situation where there are certain features of every agent known to everyone that allows one to infer how he will use the object. [1] Such features directly influence the utility other agents will have from him owning the object. We model this aspect by assuming that each agent has a a strict ranking over the set of all agents (including himself) and the seller, where he keeps himself at the top and the seller at the bottom of the ranking. A commonly known real number is assigned to each position, with the top position getting 1, the bottom position getting zero, and each intermediate position getting a number strictly between 0 and 1 with the numbers decreasing with position. The utility for an agent when an agent gets the object or the seller keeps the object is a product of his own valuation and the real number associated with the position of the winning agent in his ranking. For instance, if agent i ranks agent j at the third position, then the utility of agent i when agent j wins the object is $\alpha_3 v_i$, where v_i

[1] In case of patents, a company's past use of patents may reflect how he will use any patent in the market.

is the valuation of agent i for the object and $\alpha_3 \in (0,1)$ is the third position-specific number.

We identify a condition on the allocation rule, which we call the *interval property*, that is necessary and sufficient for implementability of the allocation rule. The interval property requires the following requirement: considering an arbitrary agent i and fixing the valuation of other agents, if agent j wins the object at v_i and agent k wins the object at v'_i, where $v'_i > v_i$, then agent k is higher than agent j in the ranking of agent i.

This interval property allows us to pin down one payment rule that implements an implementable allocation rule. Using revenue equivalence, we can then pin down the entire class of payment rules that can implement an implementable allocation rule. These ideas are then used to derive a revenue maximizing auction for this model using Myersonian techniques.

In the standard single object auction, if agents are symmetric, then when the object is allocated, it will be allocated efficiently. However, we see that in our optimal auction with externalities model, the object may be allocated inefficiently even for symmetric agents case. Thus, we have another source of inefficiency due to externalities.

Throughout, we assume that the ranking of each agent is common knowledge. This makes the type space *one dimensional*. However, the specific nature of the utility function makes our analysis non-trivial. We point out some specific difficulties we encounter if the ranking of each agent is also a private information.

4.1.1 Related Literature

Specific models of externalities have been studied in the mechanism design literature by Jehiel et al. (1999). Jehiel et al. (1999) model externalities by considering a multi-dimensional type space, where each agent has a private valuation for every agent participating in the auction. This paper contributed to the multi-dimensional mechanism design literature. In particular, Jehiel et al. (1999) designed a revenue maximizing mechanism for the two bidders case and some other specific settings. One take-away from their analysis is that designing revenue optimal auctions is hard in general models of externalities. We circumvent this problem by analyzing a simpler model of externalities.

Verma (2002) and Aseff and Chade (2008) analyze equilibrium bidding behavior in standard auctions when there are externalities. Verma (2002) analyzes the equilibrium bidding behavior in open ascending-bid auction with identity-dependent externalities. Like us, Aseff and Chade (2008) consider a specific model of externalities to design an optimal auction for multiple unit allocation. Our paper is different from Aseff and Chade (2008) since the way we model externalities is quite different leading to different insights.

Our model falls into the general models of one-dimensional mechanism design (Nisan, 2007). These models usually assume a *binary* outcome space, where each agent gets some positive utility (captured by the valuation) from one of the outcomes and zero utility from the other outcome. As a consequence, natural monotonicity conditions can be used to characterize implementability (Myerson, 1981). However, in our model, an agent gets positive utility from various outcomes. As a result, our analysis is different from the traditional one-dimensional models. [2] We could have resorted to results in the multidimensional mechanism design literature (Rochet, 1987; Bikhchandani et al., 2006) to get necessary and sufficient conditions for implementability in our model and then used that to get to our result. But our proofs are more transparent and illustrate the delicate interplay between incentive constraints and our characterization condition (interval property).

In a multi-dimensional environment, Carbajal and Muller (2014) discuss one application of their model in the presence of externalities.

We describe the model in Section 2 and present the main result of this paper for the case of known ranking in Section 3. We design a revenue maximizing mechanism in Section 4. Finally, we discuss the difficulty when we relax the assumption of known ranking in Section 5.

4.2 The Model

A seller is interested in selling a single indivisible object to n potential agents (buyers). The set of agents is denoted by $N := \{1, 2, \ldots, n\}$. Every agent i has private valuation v_i for the object. The set of all possible valuations of agent i is an open interval, $(0, \beta_i)$ where $\beta_i \in \mathbb{R}_{++} \cup \{\infty\}$ - note that we do not allow zero valuations. Denote by $V_i = (0, \beta_i)$. We will use the usual notation $v = (v_1, v_2, \ldots, v_n)$ as a profile of valuations for all agents and v_{-i} for the valuation profile of agents other than i. Let $V := V_1 \times V_2 \times \cdots \times V_n$ and $V_{-i} := \Pi^n_{j \neq i} V_{-j}$.

We define the set of alternatives as $A := \{a_0, a_1, \ldots, a_n\}$, where a_i is the alternative when agent i receives the object and a_0 is the alternative when the object remains unsold.

In our model, an agent may derive some utility if some other agent is assigned the object. We assume that if the seller keeps the object every agent gets zero utility and the seller also gets zero utility. However, assigning the object to any agent in N results in some utility for all the agents. We capture such externalities in our model by considering a strict ranking of each agent over the set of alternatives. Note that the set of alternatives is nothing but the set of agents and the seller. Each of these rankings satisfy the property that every agent keeps himself at the top and the seller at the bottom in his ranking. For instance, agent i keeps alternative a_i at the top and a_0 at the bottom in his ranking. We assume that the rankings of agents are known to everyone and denote this (public) ranking of an agent i as P_i.

We denote by $P_i(k)$ the k-th ranked alternative of agent i. We assign some weight to every position in the ranking P_i. These weights determine the utility due to externalities from other agents. Denote by α_k the weight assigned to the k-th ranked alternative. Then, agent i with valuation v_i will get a utility of $\alpha_k v_i$ if the alternative $P_i(k)$ is assigned the object. With a slight abuse of notation, we will denote by α_{n+1} the weight on the last ranked

[2] For instance, unlike the standard single object auction model, in our model, efficiency does not imply that the object must go to the agent with the highest valuation.

alternative (the seller getting the object). Further, we assume that $\alpha_1 = 1, \alpha_{n+1} = 0$, and $\alpha_1 > \alpha_2 > \ldots > \alpha_n > \alpha_{n+1}$ where every $\alpha_j \in (0,1)$ for all $j \in \{2,3,\ldots,n\}$. We assume that these weights are common knowledge. Given that the rankings of agents are also common knowledge, the designer can infer the utilities of the agents once he knows their valuations.

As an example, consider the case where there are three agents $\{1,2,3\}$. Suppose $\alpha_1 = 1, \alpha_2 = 0.8, \alpha_3 = 0.5, \alpha_4 = 0$. Then, suppose the ranking of agent 1 is as follows: $a_1 P_1 a_3 P_1 a_2 P_1 a_0$. In that case if agent 1 has valuation v_1, then his utility from different alternatives are given by

$$u(a_0, v_1) = 0, u(a_1, v_1) = v_1, u(a_2, v_1) = \alpha_3 v_1 = 0.5 v_1,\ u(a_3, v_1) = \alpha_2 v_1 = 0.8 v_1,$$

where we used the notation $u(a, v_1)$ to denote the utility of agent 1 from an alternative a when he has valuation v_1.

An allocation rule is a mapping $f : V \to A$. Note that we do not consider randomized allocation rules in our model. We consider only deterministic allocation rules.

We allow payments in this model. A payment function of agent $i \in N$ is a mapping $\pi_i : V \to \mathbb{R}$.

We also assume that agents have quasi-linear utility function over payments and are risk neutral. We define $P_i^{-1}(a_k)$ as the position of alternative a_k in i^{th} agent ranking P_i. Net utility of an agent i for every $v_{-i} \in V_{-i}$ when his true valuation is v_i and he reports v_i' is,

$$u_i(v_i', v_{-i}) = v_i \alpha_{P_i^{-1}(f(v_i', v_{-i}))} - \pi_i(v_i', v_{-i})$$

where $\alpha_{P_i^{-1}(f(v_i', v_{-i}))} = \alpha_1 = 1$ if $f(v_i', v_{-i}) = a_i$ and and $\alpha_{P_i^{-1}(f(v_i', v_{-i}))} = \alpha_{P_i^{-1}(a_k)}$ if $f(v_i', v_{-i}) = a_k$.

Note that every agent gets some utility unlike the standard single object auction where the utility is binary ,i.e, only winner gets utility, other gets zero utility.

Definition **29** *An allocation rule f is* **implementable** *(in dominant strategies) if for every $i \in N$, there exists a payment function π_i such that for every $v_{-i} \in V_{-i}$, $v_i \in V_i$, and $v_i' \in V_i$,*

$$v_i \alpha_{P_i^{-1}(f(v_i, v_{-i}))} - \pi_i(v_i, v_{-i}) \geq v_i \alpha_{P_i^{-1}(f(v_i', v_{-i}))} - \pi_i(v_i', v_{-i})$$

In this case, we say that $\pi = (\pi_1, \pi_2 \ldots, \pi_n)$ implement f and the mechanism (f, π) is **incentive compatible**.

We state below a well known fact that will be useful for us later. If $(f, \pi_1, \ldots, \pi_n)$ is incentive compatible, then for every $i \in N$, for every (v_i, v_{-i}) and (v_i', v_{-i}) such that $f(v_i, v_{-i}) = f(v_i', v_{-i})$, we have $\pi_i(v_i, v_{-i}) = \pi_i(v_i', v_{-i})$ - this can be shown by writing down the two associated incentive constraints for v_i and v_i'. As a result, for every $i \in N$, π_i can be written as a map $\pi_i : A \times V_{-i} \to \mathbb{R}$.

4.3 Implementable Allocation Rules

In this section, we characterize the implementable allocation rules using a property called the *interval property*. This is a natural generalization of the monotonicity condition used to characterize implementability in the standard single object auction model (Myerson, 1981).

Before defining the interval property formally, we illustrate this with a simple example.

Example **12**

Suppose $N = \{1,2,3\}$. Consider agent 1 and fix the valuations of other agents at v_{-1}. Suppose the ranking of alternatives of agent 1 is $a_1 P_1 a_3 P_1 a_2 P_1 a_0$. If f is an implementable allocation rule, we will show that the type space $V_1 \equiv (0, \beta_1)$ can be divided into four *subintervals*, whose interiors are denoted by $(0 = \beta_1^0, \beta_1^1)$, (β_1^1, β_1^2), (β_1^2, β_1^3), and $(\beta_1^3, \beta_1^4 = \beta_1)$, where $\beta_1^0 \leq \beta_1^1 \leq \beta_1^2 \leq \beta_1^3 \leq \beta_1^4$. It is possible that some of these subintervals are empty. For each $j \in \{0,1,2,3\}$, alternative $P_1(j)$ is chosen by f in the subinterval $(\beta_1^j, \beta_1^{j+1})$. For instance, for any $v_i \in (\beta_1^2, \beta_1^3)$, $f(v_1, v_{-1}) = P_i(2) = a_3$.

This can be illustrated in Figure 4.1 where we assume that none of the above subintervals is empty. We can see that for given other agents' valuation v_{-1} if the valuation of agent 1, v_1 is in the subinterval $[\beta_1^0, \beta_1^1)$, then allocation rule $f(v_1, v_{-1})$. Similarly if $v_1 \in [\beta_1^3, \beta_1^4)$, then $f(v_1, v_{-1}) = a_1$.

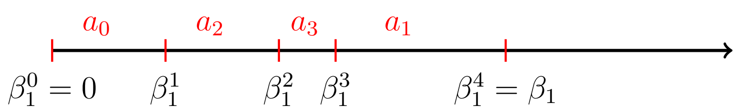


Figure 4.1: Illustration of the interval property.

We now formally define the interval property.

Definition **30** *An allocation rule f satisfies the* **interval property** *if for every $i \in N$, for every $v_{-i} \in V_{-i}$, for every $v_i, v_i' \in V_i$ with $v_i' > v_i$ if $f(v_i', v_{-i}) \neq f(v_i, v_{-i})$, then $P_i^{-1}(f(v_i', v_{-i})) < P_i^{-1}(f(v_i, v_{-i}))$*

We state the following fact that is a consequence of the interval property and is used for defining a payment function later.

Fact 1: Suppose an allocation rule f satisfies the interval property. Then for every $i \in N$, for every $v_i \in V_i$, for every $v_i, v_i' \in V_i$ with $v_i' > v_i$ if $f(v_i, v_{-i}) = f(v_i', v_{-i}) = a_k$, then $f(\hat{v}_i, v_{-i}) = a_k$ for every $\hat{v}_i \in [v_i, v_i']$.

Proof: Let f be an allocation rule satisfying the interval property. Fix an agent $i \in N$ and $v_{-i} \in V_{-i}$. Consider $v_i, v_i' \, inV_i$ with $v_i' > v_i$ and $f(v_i, v_{-i}) = f(v_i', v_{-i}) = a_k$. Assume for contradiction that there exists $\hat{v}_i \in (v_i, v_i')$ such that $f(\hat{v}_i, v_{-i}) = a_j \neq a_k$. Since $v_i < \hat{v}_i < v_i'$, we have $P_i^{-1}(a_k) < P_i^{-1}(a_j) < p_i^{-1}(a_k)$ by the interval property. This is a contradiction. ■

Interval property is implied by implementability.

Lemma **18** *Suppose f is an implementable allocation rule. Then, f satisfies the interval property.*

Proof: Suppose f is an implementable allocation rule. Consider $i \in N$ and $v_{-i} \in V_{-i}$, $v_i, v_i' \in V_i$ with $v_i' > v_i$ such that $f(v_i, v_{-i}) \neq f(v_i', v_{-i})$. Then, adding the incentive constraints from v_i to v_i' and from v_i' to v_i, we have

$$\alpha_{P_i^{-1}(f(v_i',v_{-i}))}(v_i' - v_i) \geq \alpha_{P_i^{-1}(f(v_i,v_{-i}))}(v_i' - v_i)$$

Since $v_i' > v_i$ and $f(v_i') \neq f(v_i)$, we have $\alpha_{P_i^{-1}(f(v_i'))} > \alpha_{P_i^{-1}(f(v_i))}$. This implies that $P_i^{-1}(f(v_i', v_{-i})) < P_i^{-1}(f(v_i, v_{-i}))$. ■

We now provide the main result of this section.

Theorem **10** *An allocation rule is implementable if and only if it satisfies the interval property.*

Proof: From Lemma 18, interval property is necessary for implementation. For sufficiency, consider an allocation rule that satisfies the interval property. We will now construct payment functions that implement f. To do so, fix agent i and types of other agents at v_{-i}. By the interval property, we can divide the interval $V_i = (0, \beta_i)$ into subintervals such that each of those subintervals correspond to a unique alternative. For any alternative $P_i(k)$, let the interior of the unique subinterval where $P_i(k)$ is the outcome be $(\underline{v}_i^{P_i(k)}, \underline{v}_i^{P_i(k+1)})$, i.e., for all $v_i \in (\underline{v}_i^{P_i(k)}, \underline{v}_i^{P_i(k+1)})$, we have $f(v_i, v_{-i}) = P_i(k)$.

We use this to define a payment function $\pi_i^* : A \times V_{-i} \to \mathbb{R}$ as follows. First, we set $\pi_i^*(a_0, v_{-i}) = 0$. Now, we define the payment function recursively. Having defined the payment for the $(k+1)^{th}$ ranked alternative, we define the payment for the k^{th} ranked alternative as

$$\pi_i^*(P_i(k), v_{-i}) := (\alpha_k - \alpha_{k+1})\underline{v}_i^{P_i(k)} + \pi_i^*(P_i(k+1), v_{-i}).$$

Note that this simplifies to

$$\pi_i^*(P_i(k), v_{-i}) := \sum_{d=k}^{n} (\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)}. \tag{4.1}$$

To show that π_i^* implements f, consider valuations $v_i, v_i' \in V_i$. Let $f(v_i, v_{-i}) = a_j$ and $f(v_i', v_{-i}) = a_k$. We have to show that

$$\alpha_{P_i^{-1}(a_j)} v_i - \pi_i^*(a_j, v_{-i}) \geq \alpha_{P_i^{-1}(a_k)} v_i - \pi_i^*(a_k, v_{-i}) \tag{4.2}$$

$$\text{or } \alpha_{P_i^{-1}(a_j)} v_i - \sum_{d=P_i^{-1}(a_j)}^{n} (\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)} \geq \alpha_{P_i^{-1}(a_k)} v_i - \sum_{d=P_i^{-1}(a_k)}^{n} (\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)} \tag{4.3}$$

If $a_j = a_k$, then there is nothing to prove. Therefore, assume $a_j \neq a_k$. We consider the following two cases:

CASE 1: $P_i^{-1}(a_j) < P_i^{-1}(a_k)$. The position of alternative a_j is above the position of alternative a_k in ranking P_i. Without loss of generality assume that $P_i^{-1}(a_j) = j$ and $P_i^{-1}(a_k) = k$. Hence, simplifying Inequality 4.2, we have to show the following inequality,

$$(\alpha_j - \alpha_k)v_i - \sum_{d=j}^{k-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)} \geq 0$$

To show this, we have the following sequence of inequalities,

$$\begin{aligned}(\alpha_j - \alpha_k)v_i - \sum_{d=j}^{k-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)} &\geq (\alpha_j - \alpha_k)\underline{v}_i^k - \sum_{d=j}^{k-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)} \\ &\geq (\alpha_j - \alpha_k)\underline{v}_i^k - \sum_{d=j}^{k-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^k \\ &= 0\end{aligned}$$

Using $j < k$, we see that first inequality follows because $v_i \geq \underline{v}_i^j$ and $\underline{v}_i^j \geq \underline{v}_i^k$. Second inequality follows because $\underline{v}_i^{P_i(d)} \geq \underline{v}_i^k$ for any $d \geq k$ and $(\alpha_d - \alpha_{d+1}) > 0$ for any d. Hence, we are done.

CASE 2: $P_i^{-1}(a_j) > P_i^{-1}(a_k)$. The position of alternative a_j is below the position of alternative a_k. Again, take $P_i^{-1}(a_j) = j$ and $P_i^{-1}(a_k) = k$. Since now $j > k$, simplifying Inequality 4.2, we have to show that

$$(\alpha_j - \alpha_k)v_i + \sum_{d=k}^{j-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)} \geq 0$$

Now we have the following sequence of inequalities

$$\begin{aligned}(\alpha_j - \alpha_k)v_i + \sum_{d=k}^{j-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)} &\geq (\alpha_j - \alpha_k)v_i + \sum_{d=k}^{j-1}(\alpha_d - \alpha_{d+1})v_i \\ &= 0\end{aligned}$$

Here, we used the fact $j > k$, to conclude the first inequality since $v_i \leq \underline{v}_i^{P_i(d)}$ for all $d \in \{k, k+1, \ldots, j-1\}$. Hence, we are done. ■

Theorem 10 not only characterizes all the implementable allocation rules but also provides an explicit formula for computing a payment function that implements it. Using revenue equivalence, we can write the payment function as follows. [3] So, given an allocation rule satisfying the interval property, for every agent $i \in N$, for every $v_{-i} \in V_{-i}$, and for every $a \in A$,

$$\pi_i(a, v_{-i}) = h_i(v_{-i}) + \sum_{d=P_i^{-1}(a)}^{n} (\alpha_d - \alpha_{d+1}) \underline{v}_i^{P_i(d)},$$

where h_i is a mapping $h_i : V_{-i} \to \mathbb{R}$. We summarize this in the theorem below.

Theorem **11** *A mechanism $(f, \pi_1, \ldots, \pi_n)$ is incentive compatible if and only if f satisfies the interval property and $(\pi_1, \ldots, \pi_n)$ is computed using Equation 4.1 by appropriately choosing h_i for each $i \in N$.*

4.4 Revenue Maximization For The Known Ranking

In this section, we design an optimal auction using the characterization results in the last section. We assume that the valuation of each agent i is drawn using distribution g_i with cumulative density function G_i. The density function is positive everywhere, i.e. $g_i(v_i) > 0$ for every $v_i \in V_i$. We assume that the valuation of each agent is drawn independently. We also assume that hazard rate $\frac{g_i(v_i)}{1-G_i(v_i)}$ is non-decreasing in v_i. Let $\omega_i : V_i \to \mathbb{R}$ be the *virtual valuation function* of agent i. It is defined as

$$\omega_i(v_i) = v_i - \frac{1 - G_i(v_i)}{g_i(v_i)} \qquad \forall v_i \in V_i.$$

Note that the assumption of non-decreasing hazard rate implies that virtual valuation is increasing in v_i for every agent $i \in N$.

The expected revenue in a mechanism $(f, \pi \equiv (\pi_1, \ldots, \pi_n))$ is defined as,

$$\Pi(f, \pi) = \sum_{i \in N} E_v[\pi_i(f(v), v_{-i})]$$

where $E_v[\cdot]$ denotes the expectation over all valuation profiles.

Definition **31** *A mechanism (f, π) is* **individually rational** *if for every $i \in N$ and every valuation profile $v \in V$, we have*

$$\alpha_{P_i^{-1}(f(v))} v_i - \pi_i(f(v), v_{-i}) \geq 0$$

Our objective is to find a mechanism that yields the maximum expected revenue among all incentive compatible and individually rational mechanisms.

[3] Revenue equivalence holds in this model, since utility for an agent i from an alternative a given his type v_i can be written as $u(a, v_i)$ and this function is linear in v_i. Further type space is convex. Hence, we can apply the standard envelope theorem to conclude revenue equivalence - see Vohra (2011) for details.

Definition 32 *A mechanism* (f, π) *is an* **optimal mechanism** *if it is incentive compatible, individually rational, and there does not exist another mechanism* (f', π') *such that* (f', π') *is incentive compatible, individually rational, and* $\Pi(f', \pi') > \Pi(f, \pi)$.

Suppose f is an implementable allocation rule. By Theorem 11, the payment of agent i at valuation profile v is given by

$$\pi_i(f(v), v_{-i}) = h_i(v_{-i}) + \sum_{d=P_i^{-1}(f(v))}^{n} (\alpha_d - \alpha_{d+1})\underline{v}_i^{P_i(d)}. \tag{4.4}$$

We will write this equation in a way such that we can express the expected revenue in terms of virtual valuation. Then, we can apply Myersonian techniques to conclude. To write the revenue in the form of virtual valuation, we have to convert the summation part in the payment function into integral form. For this, we use a new function to capture the summation part and then write everything in integral form. Consider an agent $i \in N$ and a valuation profile v. Let $P_i^{-1}(f(v)) = k$. Now, we can re-write the payment function as

$$\begin{aligned}\pi_i(f(v), v_{-i}) &= h_i(v_{-i}) + \alpha_k \underline{v}_i^{P_i(k)} - \sum_{d=k+1}^{n-1} \alpha_d(\underline{v}_i^{P_i(d-1)} - \underline{v}_i^{p_i(d)}) + \alpha_{n+1}\underline{v}_i^{P_i(n)} \\ &= h_i(v_{-i}) + \alpha_k v_i - \alpha_k[v_i - \underline{v}_i^{P_i(d)}] - \sum_{d=k+1}^{n-1} \alpha_d(\underline{v}_i^{P_i(d-1)} - \underline{v}_i^{P_i(d)}) + \alpha_{n+1}\underline{v}_i^{p_i(n)}\end{aligned}$$

Define an indicator function as $\mathbb{1}_{[\gamma,\gamma']}(t) = 1$ if $t \in [\gamma, \gamma']$ and $\mathbb{1}_{[\gamma,\gamma']}(t) = 0$ if $t \notin [\gamma, \gamma']$ for any $\gamma < \gamma'$. Now, define a mapping, $H^i_{v_{-i}} : [0, \beta_i] \mapsto \mathbb{R}_+$ as follows

$$H^i_{v_{-i}}(t) \equiv \sum_{d=1}^{n} \alpha_d \mathbb{1}_{[\underline{v}_i^{P_i(d)}, \underline{v}_i^{P_i(d-1)}]}(t)$$

Now, we can write

$$\int_0^{v_i} H^i_{v_{-i}}(t)dt = \alpha_k\big[v_i - \underline{v}_i^{P_i(k)}\big] + \sum_{d=k+1}^{n} \alpha_d\big[\underline{v}_i^{P_i(d-1)} - \underline{v}_i^{P_i(d)}\big]$$

Note that the integral has been computed over all the sub-intervals till the valuation v_i of agent i.

Now, we can write the payment function in Equation 4.4 as follows

$$\pi_i(v_i, v_{-i}) = h_i(v_{-i}) + \alpha_k v_i - \int_0^{v_i} H^{(i)}_{v_{-i}}(t)dt$$

Once we have the payment in this form, we can apply the methodology of Myerson (1981) to solve for an optimal auction. The expected payment of an agent i in the dominant strategy incentive compatible mechanism (f, π) is given by

$$\Pi_i(f, \pi) = E_{v_{-i}}\Big[h_i(v_{-i}) + \int_0^{\beta_i} \alpha_{P_i^{-1}(f(v))} v_i g_i(v_i)dv_i - \int_0^{\beta_i}\Big(\int_0^{v_i} H^i_{v_{-i}}(t)dt\Big) g_i(v_i)dv_i\Big]$$

By changing the order of integration in the second part of the above equation, we have

$$\int_0^{\beta_i} \left(\int_0^{v_i} H^i_{v_{-i}}(t)dt \right) g_i(v_i)dv_i = \int_0^{\beta_i} \left(\int_{v_i}^{\beta_i} g_i(t)dt \right) H^i_{v_{-i}}(v_i)dv_i$$
$$= \int_0^{\beta_i} \big(1 - G_i(v_i)\big) H^i_{v_{-i}}(v_i)dv_i$$

Observe that $H^i_{v_{-i}}(v_i) = \alpha_{p_i^{-1}(f(v))}$. Therefore, we have

$$\Pi_i(f,\pi) = E_{v_{-i}}\left[h_i(v_{-i}) + \int_0^{\beta_i} \alpha_{p_i^{-1}(f(v))} v_i g_i(v_i)dv_i - \int_0^{\beta_i} \alpha_{P_i^{-1}(f(v))}\big(1 - G_i(v_i)\big)dv_i\right]$$
$$= E_{v_{-i}}\left[h_i(v_{-i}) + \int_0^{\beta_i} \alpha_{P_i^{-1}(f(v))}\left(v_i - \frac{1 - G_i(v_i)}{g_i(v_i)}\right) g_i(v_i)dv_i\right]$$

Hence, the total expected revenue in the dominant strategy incentive compatible mechanism (f,π) is given by

$$\Pi(f,\pi) = \sum_{i\in N} E_{v_{-i}}\left[h_i(v_{-i}) + \int_0^{\beta_i} \alpha_{P_i^{-1}(f(v))}\left(v_i - \frac{1 - G_i(v_i)}{g_i(v_i)}\right) g_i(v_i)dv_i\right]$$

Since we are considering individually rational mechanism, it implies that $h_i(v_{-i}) \le 0$ for all $i \in N$ and for all $v_{-i} \in V_{-i}$. Since payment is non-negative, we have $\pi_i(0, v_i) \ge 0$ for all $v_{-i} \in V_{-i}$. This implies that $h_i(v_{-i}) \ge 0$. Hence, $h_i(v_{-i}) = 0$ for every $i \in N$. Now, we can write the total expected revenue simply as

$$\Pi(f,\pi) = E_{v_{-i}}\left[\sum_{i\in N} \int_0^{\beta_i} \alpha_{P_i^{-1}(f(v))}\omega_i(v_i)g_i(v_i)dv_i\right]$$
$$= E_v\left[\sum_{i\in N} \alpha_{P_i^{-1}(f(v))}\omega_i(v_i)\right]$$

Let us define a function for each $v \in V$ as $O_v : A \to \mathbb{R}$ such that

$$O_v(a_k) = \sum_{i\in N} \alpha_{P_i^{-1}(a_k)}\omega_i(v_i) \qquad \forall a_k \in A$$

This is the weighted sum of virtual valuation of all the agents for alternative $a_k \in A$ (notice that each alternative corresponds to an agent) and for a given valuation profile v. In other words, it computes "virtual" utilities of agents for a given alternative.

Remark 1: We have $O_v(a_0) = 0$ by definition.

Now the objective is to choose an allocation rule that maximize the expected revenue. In other words, we want to maximize the function $O_v(a_k)$ for all $a_k \in A$. If we sidestep the incentive compatibility requirement of allocation rule f, then we have to do point-wise maximization.

Thus, point-wise maximization means that at every valuation profile v we want to compute the value of function $O_v(a_k)$ for each alternative a_k. We take f is equal to the alternative that maximize the value of the function $O_v(a_k)$.

There may be more than one alternatives that maximize the function $O_v(a_k)$. In that case we break the ties arbitrarily.

Formally, an optimal allocation f^* is defined as follows. For any valuation profile v, f^* is

$$f^*(v) \in \arg\max_{a_k \in A} O_v(a_k) \tag{4.5}$$

Note that if $\omega_i(v_i) < 0$ for all $i \in N$, we have $f^*(v) = a_0$ because a_0 maximizes the function $O_v(a_k)$ for all $a_k \in A$. In other words, if the maximum virtual valuation of agents is negative, then the object remains unallocated.

Now, it only remains to prove that f^* is an implementable allocation rule. The following proposition takes care of the implementability of f^*.

PROPOSITION **4** *The allocation rule f^* is implementable.*

Proof: The proof is in the Appendix. ■

This shows that f^* along with payment function defined in Theorem 10 is the optimal mechanism. It is summarized in the following theorem.

THEOREM **12** *The mechanism given by (f^*, π^*) where f^* is defined as in Equation 4.5 and π^* is defined as in Equation 4.1 is optimal.*

Remark: There is a tension between optimality and efficiency. Our optimal allocation is not efficient even for the symmetric agents case unlike the Myerson' optimal auction. This has been discussed in Example 14.

Another source of inefficiency emerges when all agents have negative virtual valuation. This has been discussed in Example 13.

The following example explains how our optimal auction works and how it is different from the standard single object auction.

EXAMPLE **13**

Consider a setting with three agents whose valuations are distributed uniformly over the interval $(0, 1)$. Since there are three agents, we have four alternatives $\{a_0, a_1, a_2, a_3, a_4\}$. Let the rankings of agents 1, 2 and 3 be $a_1P_1a_2P_1a_3P_1a_0$, $a_2P_2a_3P_2a_1P_2a_0$ and $a_3P_3a_2P_3a_1P_3a_0$ respectively. Suppose $\alpha_1 = 1$, $\alpha_2 = 0.8$, $\alpha_3 = 0.5$ and $\alpha_4 = 0$.

The virtual valuation of every agent $i \in \{1, 2, 3\}$ is $\omega_i(v_i) = 2v_i - 1$. Suppose the valuations of agents 1, 2 and 3 are $v_1 = 0.8$, $v_2 = 0.7$ and $v_3 = 0.6$ respectively. Let us call this valuation profile v. Thus, the virtual valuations of agents are $\omega_1(v_1) = 0.6$, $\omega_2(v_2) = 0.4$ and $\omega_3(v_3) = 0.2$. Thus, we have non-negative virtual valuation for every agent at the valuation profile v. To apply our result, we do the following calculation at the valuation profile v:

$$O_v(a_1) = \sum_{i=1}^{3} \alpha_{P_i^{-1}(a_1)} \omega_i(v_i) = 1 \times 0.6 + 0.5 \times 0.4 + 0.5 \times 0.2 = 0.9.$$

$$O_v(a_2) = \sum_{i=1}^{3} \alpha_{P_i^{-1}(a_2)} \omega_i(v_i) = 0.8 \times 0.6 + 1 \times 0.4 + 0.8 \times 0.2 = 1.04.$$

$$O_v(a_3) = \sum_{i=1}^{3} \alpha_{P_i^{-1}(a_3)} \omega_i(v_i) = 0.5 \times 0.6 + 0.8 \times 0.4 + 1 \times 0.2 = 0.82.$$

Thus, we have $f^*(v) = a_2$ in our model.

Consider another valuation profile v' as $v'_1 = 0.4$, $v'_2 = 0.3$ and $v'_3 = 0.4$. Thus, we have negative virtual valuation of every agent at v'. Hence, $f^*(v) = a_0$. The object is not allocated to anyone.

In the Myerson auction, the reservation valuation is 0.5. Thus, Myerson optimal auction gives the object to agent 1 at the valuation profile v because agent 1 has the highest virtual valuation. Therefore, $f^{*M}(v) = a_1$ where f^{*M} is the Myerson optimal allocation rule. For the valuation profile v', we have $f^{*M}(v') = a_0$ because the virtual valuation of every agent is negative and moreover every one has valuation below the reservation price.

Note that if every agent has negative virtual valuation, then we have the same allocation in our mechanism as well as in Myerson' mechanism. We see that in our optimal mechanism, the allocation rule may not allocate the object to the agent who has the highest virtual valuation.

We show in the following example that we may allocate the object inefficiently even in symmetric agents setting. We contrast this with the optimal auction in standard single object auction model where if the agents are symmetric, then when the object is allocated, it will be allocated efficiently. Here, we see that even for symmetric agents case, the object may be allocated inefficiently.

Example **14**

Consider three agents whose valuations are uniformly distributed over the interval $(0, 1)$. We have four alternatives $\{a_0, a_1 a_2, a_3, a_4\}$. Let the rankings of agents 1, 2 and 3 be $a_1 P_1 a_2 P_1 a_3 P_1 a_0$, $a_2 P_2 a_3 P_2 a_1 P_2 a_0$ and $a_3 P_3 a_2 P_3 a_1 P_3 a_0$ respectively. Suppose $\alpha_1 = 1$, $\alpha_2 = 0.9$, $\alpha_3 = 0.5$ and $\alpha_4 = 0$.

Since agents are symmetric, their virtual valuation is $\omega(v_i) = 2v_i - 1$. Consider the following profile $\bar{v}$ such that $v_1 = 0.3$, $v_2 = 0.8$ and $v_3 = 0.5$. Thus, the virtual valuations for the valuation profile $\bar{v}$ are $\omega(v_1) = -0.4$, $\omega(v_2) = 0.6$ and $\omega(v_3) = 0$. To apply our result, we do the following computation at the valuation profile $\bar{v}$:

$$O_{\bar{v}}(a_1) = \sum_{i=1}^{3} \alpha_{P_i^{-1}(a_1)} \omega(v_i) = 1 \times (-0.4) + 0.5 \times 0.6 + 0.5 \times 0 = -0.1.$$

$$O_{\bar{v}}(a_2) = \sum_{i=1}^{3} \alpha_{P_i^{-1}(a_2)} \omega(v_i) = 0.9 \times (-0.4) + 1 \times 0.6 + 0.9 \times 0 = 0.24.$$

$$O_{\bar{v}}(a_3) = \sum_{i=1}^{3} \alpha_{P_i^{-1}(a_3)} \omega(v_i) = 0.5 \times (-0.4) + 0.9 \times 0.6 + 1 \times 0 = 0.34.$$

Thus, we have $f^*(v) = a_3$ in our model.

An allocation rule is **efficient** if for every $v \in V$, we have

$$f^{\mathrm{e}} \in \arg\max_{a_k \in A} \sum_{i \in N} \alpha_{P_i^{-1}}(a_k) v_i \qquad \forall a_k \in A.$$

Now, to see the efficient allocation, we do the following calculation at the valuation profile $\bar{v}$:

$$\sum_{i=1}^{3} \alpha_{P_i^{-1}}(a_1) v_i = 1 \times 0.3 + 0.5 \times 0.8 + 0.5 \times 0.5 = 0.95.$$

$$\sum_{i=1}^{3} \alpha_{P_i^{-1}}(a_2) v_i = 0.9 \times 0.3 + 1 \times 0.8 + 0.9 \times 0.5 = 1.52.$$

$$\sum_{i=1}^{3} \alpha_{P_i^{-1}}(a_3) v_i = 0.5 \times 0.3 + 0.9 \times 0.8 + 1 \times 0.5 = 1.37.$$

Hence, $f^{\mathrm{e}}(\bar{v}) = a_2$. Note that efficient allocation allocate the object to the person who has the highest valuation but our allocation does not allocate the object to the agent who has the highest valuation.

Remark: Agents are symmetric in our example in the sense that they have same distribution. The asymmetry among agents' ranking is an integral part of our model. The asymmetry between the ranking of two agents is emerging because of the externality. Therefore, we have a different kind of asymmetry because of the externality.

4.5 Private Rankings

In this section, we consider the model in the previous section by assuming that the rankings of each agent of over the set of alternatives is also a private information. As a result, the type of an agent consists of both the valuation and the private ranking. This considerably complicates the model.

As before, let P_i denote the ranking of agent i of the set of alternatives. Let $\mathcal{P}_i$ be all the possible rankings of agent i and $\mathcal{P} := \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_n$. We use the usual notations P_{-i} and $\mathcal{P}_{-i}$ as the ranking and profile of all possible rankings of agents other than agent i respectively. Now, a type profile is a tuple (v, P), where v is the valuation profile and P is the profile of rankings.

An allocation rule is a mapping $f : V \times \mathcal{P} \to A$. It means for every $(v, P) \in V \times \mathcal{P}$, $f(v, P)$ is the alternative chosen. A payment function of agent $i \in N$ is a mapping $\pi_i : V \times \mathcal{P} \to \mathbb{R}$. For convenience, we fix other agents' type profile (v_{-i}, P_{-i}) for our analysis. We can also modify the definition of implementability appropriately for this case.

We show below some natural payment functions and show that they cannot implement an implementable allocation rules. Our analysis hints that there are no "nice" payment functions associated with an incentive compatible mechanism. This illustrates the fundamental challenge going from a one-dimensional type space to a type space that is no longer single dimensional.

Before going into detail, we point out that we are dealing with a non-convex type space now. The following example explains this.

Example **15**

Let $N = \{1, 2, 3\}$. There are three agents and the set of alternatives is $A = \{a_0, a_1, a_2, a_3\}$. Externalities are captured in numbers by $\alpha_1 = 1 > \alpha_2 > \alpha_3 > \alpha_4 = 0$. Every agent has two rankings. Consider a type of agent 1 where his rankings are

P_1	P'_1
a_1	a_1
a_2	a_3
a_3	a_2
a_0	a_0

Now take v_1 and v'_1. Let the type of agent 1 be (v_1, P_1) and (v'_1, P'_1). We have $s_1 = (\alpha_1 v_1, \alpha_2 v_1, \alpha_3 v_1)$ with respect to type (v_1, P_1) and $t_1 = (\alpha_1 v'_1, \alpha_3 v'_1, \alpha_2 v'_1)$ with respect to type (v'_1, P'_1). Take $w_1 = \frac{s_1 + t_1}{2} = \big(\alpha_1(v_1 + v'_1), \alpha_2 v_1 + \alpha_3 v'_1, \alpha_3 v_1 + \alpha_2 v'_1\big)$. w_1 does not produce either of the rankings P_1 or P'_1. Therefore, it is not a convex domain.

Non-convex type spaces pose a major challenge in characterization of implementable allocation rules (Ashlagi et al., 2010). Now, we discuss various plausible payment rules and show how they do not work for establishing implementability.

4.5.1 Implementability and Payment Rules

In this subsection, we construct some payment functions and discuss issues with them. If we fix the ranking of an agent, then Lemma 18 works. This means that we can divide the valuation space of agent into different sub-intervals for a fixed ranking. Then, by Lemma 18, we have $\underline{v}_i^{P_i(d)} = \inf \{v_i \in V_i : f(v_i, P_i) = p_i(d)\}$ for a fixed ranking P_i. But this infimum value depends on the ranking of agent i and it is a private information.

To make the infimum value independent from the ranking of agent i, we define another infimum over all rankings. To see this, fix a position d in a ranking P_i. Now define the infimum for the alternative that is at position d in P_i. Similarly, define the infimum value for the alternative at position d in ranking P_i' and similarly define the infimum value for the alternative at position d in every ranking of agent i. In this way, we have a collection of least values at position d corresponding to all rankings. Then, take another infimum over all these least values. Since they are finites, the second infimum is well-defined. In other words, we can take the minimum of $\underline{v}_i^{P_i(d)}$ over all rankings. Formally, we have

$$\underline{v}_i^d = \min_{P_i \in \mathcal{P}_i} \{\underline{v}_i^{P_i(d)}\}$$

where d denotes the position of alternatives for which these infimums has been considered. The following example explains it.

Example **16**

Consider $N = \{1, 2, 3, 4,\}$. There are four agents who are interested in buying the object. The set of alternatives is $A = \{a_0, a_1, a_2, a_3, a_4, a_5\}$ where a_0 is the alternative where the object is not assigned to anyone and a_i is the alternative when agent $i \in N$ gets allocated the object. Fix agent 1. Then we know that agent 1 has two dimensional private information (v_1, P_1) for some $v_1 \in V_1$ and $P_1 \in \mathcal{P}_1$ where $\mathcal{P}_1 = \{P_1^1, P_1^2, P_1^3, P_1^4, P_1^5, P_1^6\}$. It is given in the following table where agent 1 keeps himself at the top in each of his rankings and a_0 at the bottom in each of his rankings.

	P_1^1	P_1^2	P_1^3	P_1^4	P_1^5	P_1^6
1	a_1	a_1	a_1	a_1	a_1	a_1
2	a_2	a_3	a_4	...	...	...
3	a_3	a_4	a_3	...	...	...
4	a_4	a_2	a_2	...	...	...
5	a_0	a_0	a_0	...	...	...

By Lemma 18, we can construct $\underline{v}_1^{P_1^1(k)} = \{v_1 \in V_i : f(v_1, P_1^1) = a_k)\}$. This is infimum calculated for agent 1, for the alternative which is at the position k and for a fixed ranking P_1^1. We see that $\underline{v}_1^{P_1^1(k)}$ depends on his ranking. To have $\underline{v}_1^d$ independent from his ranking, we take another infimum of $\underline{v}_1^{P_1^1(k)}$ over all rankings for position k. For instance, we define

$$\underline{v}_1^1 = \min\{\underline{v}_1^{P_1^1(1)}, \underline{v}_1^{P_1^2(1)} \ldots, \underline{v}_1^{P_1^6(1)}\}.$$

Thus, $\underline{v}_1^1$ is the value of first position. Similarly, we can construct the infimum values for other positions too.

We need one more piece of notation for defining a payment function of an agent. Define,

$$\bar{P}_i^{-1}(a_k) = \max_{P_i \in \mathcal{P}_i} \{P_i^{-1}(a_k) \in N : f(v_i, P_i) = a_k\}.$$

This is the maximum of positional numbers for alternative a_k where $f(v_i, P_i) = a_k$ at $(v_i, P_i) \in V_i \times \mathcal{P}_i$. Now we define the following payment function for an agent $i \in N$ and for an alternative a_k such that $f(v_i, P_i) = a_k$ at type profile (v_i, P_i),

$$\pi_i^*(f(v_i, P_i)) = \sum_{d=\bar{P}_i^{-1}(f(v_i,P_i))}^{n} (\alpha_d - \alpha_{d+1})\underline{v}_i^d$$

Note that $\underline{v}_i^d$ is just the positional value of the interval for alternative at position d. This is a natural generalization of the payment function defined in the previous section for single dimensional type space. Observe that if $f(v_i, P_i) = f(v_i', P_i')$ at $(v_i, P_i) \in V_i \times \mathcal{P}_i$ and $(v_i', P_i') \in V_i \times \mathcal{P}_i$, then $\pi_i^*(f(v_i, P_i)) = \pi_i^*(f(v_i', P_i'))$. Only the weight changes as agent changes his types. The weights play a very crucial role here in determining payments.

In the following example, we explain why this payment function does not achieve implementability.

Example **17**

Fix an agent i and consider that he has only two rankings as his private information along with his valuation for the object. His two rankings are given below.

	P_i	P_i'
	$\vdots$	$\vdots$
	$\vdots$	$\vdots$
k	a_k	$\vdots$
	$\vdots$	$\vdots$
j	a_j	$\vdots$
k'	$\vdots$	a_k

Let us consider P_i and P_i' as the true and misreported rankings of agent i respectively. Let $f(v_i, P_i) = a_j$ and $f(v_i', P_i') = a_k$. Suppose the position of alternative a_j in P_i is j and the position of alternative a_k in P_i and P_i' is k and k' respectively as indicated in the above table. The position of a_j cannot be above k' in P_i' because otherwise we have a contradiction with 2-cycle monotonicity. Now assume that $f(\hat{v}_i, P_i') \neq a_j$ for any $\hat{v}_i \in V_i$. Then, we have to satisfy the following inequality to show implementability,

$$\alpha_{P_i^{-1}(a_j)} v_i - \sum_{d=\bar{P}_i^{-1}(a_j)}^{n} (\alpha_d - \alpha_{d+1})\underline{v}_i^d \geq \alpha_{P_i^{-1}(a_k)} v_i - \sum_{d=\bar{P}_i^{-1}(a_k)}^{n} (\alpha_d - \alpha_{d+1})\underline{v}_i^d$$

Since $\bar{P}_i^{-1}(a_j) = j$ and $k' > j$, we can write the payment function of agent i with respect to alternative a_j in recursive form. Thus, we have

$$\alpha_{P_i^{-1}(a_j)} v_i - \sum_{d=j}^{k'-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^d - \sum_{d=k'}^{n}(\alpha_d - \alpha_{d+1})\underline{v}_i^d \geq \alpha_{P_i^{-1}(a_k)} v_i - \sum_{d=k'}^{n}(\alpha_d - \alpha_{d+1})\underline{v}_i^d$$

Therefore, we have to show the following inequality for implementability,

$$(\alpha_j - \alpha_{k'})v_i - \sum_{d=j}^{k'-1}(\alpha_d - \alpha_{d+1})\underline{v}_i^d \geq 0$$

But it is not satisfied because both the first and second terms are negative in the above inequality.

We construct one another payment function and show that it also does not ensure implementability. As before, we have $\underline{v}_i^{P_i}(a_k) = \inf\{v_i \in V_i : f(v_i, P_i) = a_k\}$. This is the infimum value where f chooses alternative a_k in ranking P_i. Now we collect all the least values in various rankings where alternative a_k is picked by allocation rule. Formally,

$$\underline{v}_i(a_k) = \min_{P_i \in \mathcal{P}_i} \{\underline{v}_i^{P_i}(a_k)\}.$$

This is the minimum value of alternative a_k across all P_i for an agent $i \in N$.

It cannot be the case that $\underline{v}_i(a_k)$ alone is a payment function because we multiply value v_i of agent i by α. Therefore, we have to assign some weight to $\underline{v}_i(a_k)$. Thus, we define a payment function of an agent i if $f(v_i, P_i) = a_k$ as

$$\pi_i^*(a_k) = \alpha_{\bar{P}_i^{-1}(a_k)}\underline{v}_i(a_k)$$

where $\bar{P}_i^{-1}(a_k) = \max_{P_i \in \mathcal{P}_i}\{P_i^{-1}(a_k) : f(v_i, P_i) = a_k\}$. The following example shows why this payment does not work.

Example **18**

Consider an allocation rule $f(v_i, P_i) = a_i$ when true type of an agent i is (v_i, P_i). Now if he misreports a type (v_i', P_i') such that $f(v_i', P_i') = a_k$, then we have to show for implementability that,

$$\alpha_{P_i^{-1}(a_i)}(v_i - \underline{v}_i(a_i)) \geq \alpha_{P_i^{-1}(a_k)}v_i - \alpha_{\bar{P}_i^{-1}(a_k)}\underline{v}_i(a_k)$$
$$\text{or } v_i - \underline{v}_i(a_i) \geq \alpha_{P_i^{-1}(a_k)}v_i - \alpha_{\bar{P}_i^{-1}(a_k)}\underline{v}_i(a_k)$$

where $\alpha_{P_i^{-1}(a_i)} = 1$ by definition. Agent i can misreport in such a manner that a_k has a lower position in P_i and higher position in P_i'. That is, $P_i^{-1}(a_k)$ is very close to a_i and $P_i'^{-1}(a_k)$ is very close to a_0. Then, there will be no way to maintain the above inequality.

We observe that the intuitive and straightforward payment functions will not ensure implementability of the allocation rule for this two dimensional mechanism design problem. Either, we have to restrict the dimensionality of type space or impose restrictions on allocation rules or type space to come up with some simple condition for implementability. We employ the former case and restrict our analysis to single dimensional type space considering rankings of all agents as common knowledge.

Jehiel et al. (1999) consider a multi-dimensional type space mechanism design problem with externalities and design a revenue maximizing mechanism only for the case where either agents have same type vector or only for two agents. From our discussion, it also seems that it is difficult to design a revenue maximizing mechanism even for a two dimensional type space.

4.6 Conclusion

We model externalities in a particular way that makes our model tractable. We show that an allocation rule is implementable if and only if it satisfies the interval property when the rankings of agents are known. By virtue of the revenue equivalence, we also characterize the entire class of dominant strategy incentive compatible mechanisms. We also derive an optimal auction which is very intuitive. Further, we argue by some examples that it is difficult to find out some simple payment function or condition for ensuring implementability for the case where both valuation for the object and ranking of agents over other agents are private information. In future, we will like to investigate this model in detail.

Appendix: Omitted Proofs

Proof Of Proposition 4

By Theorem 10, it is enough to show that f^* satisfies the interval property. Fix an agent $i \in N$ and valuation profiles, v_{-i}, of other agents. Consider $v_i, v_i' \in V_i$ such that $v_i > v_i'$. Let $f^*(v_i, v_{-i}) = a_k$ and $f^*(v_i', v_{-i}) = a_l$, where $a_k \neq a_l$. We are required to show that $P_i^{-1}(a_k) < P_i^{-1}(a_l)$. Assume for contradiction that $P_i^{-1}(a_k) > P_i^{-1}(a_l)$. By definition of f^*, we have

$$\alpha_{P_i^{-1}(a_k)}\omega_i(v_i) + \sum_{j \neq i} \alpha_{P_j^{-1}(a_l)}\omega_j(v_j) \geq \alpha_{P_i^{-1}(a_l)}\omega_i(v_i) + \sum_{j \neq i} \alpha_{P_i^{-1}(a_l)}\omega_j(v_j)$$

Re-arranging above inequality, we have

$$\sum_{j \neq i} \alpha_{P_j^{-1}(a_l)}\omega_j(v_j) - \sum_{j \neq i} \alpha_{P_i^{-1}(a_l)}\omega_j(v_j) \geq [\alpha_{P_i^{-1}(a_l)} - \alpha_{P_i^{-1}(a_k)}]\omega_i(v_i) \tag{4.6}$$

The virtual valuation is strictly increasing in v_i due to the assumption that hazard rate is non-decreasing. Hence, we have $\omega_i(v_i) > \omega_i(v_i')$. It is given by our assumption that $P_i^{-1}(a_k) > P_i^{-1}(a_l)$. This implies that $\alpha_{P_i^{-1}(a_k)} < \alpha_{P_i^{-1}(a_l)}$. Therefore, we have

$$\big[\alpha_{P_i^{-1}(a_l)} - \alpha_{P_i^{-1}(a_k)}\big]\omega_i(v_i) > \big[\alpha_{P_i^{-1}(a_l)} - \alpha_{P_i^{-1}(a_k)}\big]\omega_i(v_i')$$

Coupled with Inequality 4.6, we have

$$\sum_{j \neq i} \alpha_{P_j^{-1}(a_k)}\omega_j(v_j) - \sum_{j \neq i} \alpha_{P_i^{-1}(a_l)}\omega_j(v_j) > \big[\alpha_{P_i^{-1}(a_l)} - \alpha_{P_i^{-1}(a_k)}\big]\omega_i(v_i')$$

Re-arranging the terms in the above inequality, we have

$$\alpha_{P_i^{-1}(a_k)}\omega_i(v_i') + \sum_{j \neq i} \alpha_{P_j^{-1}(a_k)}\omega_j(v_j) > \alpha_{P_i^{-1}(a_l)}\omega_i(v_i') + \sum_{j \neq i} \alpha_{P_i^{-1}(a_l)}\omega_j(v_j)$$

This implies that $a_l \notin \arg\max_{a_m \in A} O_v^*(a_m)$. Thus, $f^*(v_i', v_{-i}) \neq a_l$, a contradiction. Hence f^* satisfies the interval property and therefore it is implementable.

Bibliography

Andersson, T., L. Ehlers, and L. Svensson (2014): "Budget-balance, Fairness and Minimal Manipulability," *Forthcoming, Theoretical Economics.*

Archer, A. and E. Tardos (2002): "Frugal Path Mechanisms," in *Proceedings of the Annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, SIAM Press.

Aseff, J. and H. Chade (2008): "An Optimal Auction with Identity-Dependent Externalities," *Rand Journal of Economics*, 731–746.

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

Ashlagi, I. and S. Serizawa (2011): "Characterizing Vickrey Allocation Rule by Anonymity," *Social Choice and Welfare*, 38, 1–12.

Barbera, S., F. Gul, and E. Stacchetti (1993): "Generalized Median Voter Schemes and Committees," *Journal of Economic Theory*, 61, 262–289.

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

Bikhchandani, S., J. de Vries S., Schummer, and R. Vohra (2011): "An Ascending Vickrey Auction for Selling Bases of a Matroid," 59, 1109–1132.

Border, K. (1991): "Implementation of Reduced Form Auctions: A geometric Approach," *Econometrica*, 59, 1175–1187.

Carbajal, J. C., A. McLennan, and R. Tourkey (2013): "Truthful Implementation and Preference aggregation in Restricted Domains," *Journal of Economic Theory*, 148, 1074–1101.

Carbajal, J. C. and R. Muller (2014): "Implementability under Monotonic Transformations in Differences," Working Paper.

Clarke, E. (1971): "Multipart Pricing of Public Goods," *Public Choice*, 11, 17–33.

Cremer, J. and R. McLean (1985): "Optimal Selling Strategies Under Uncertainityfor a discriminatingMonopolist whenDemands are Interdependent," *Econometrica*, 53, 345–361.

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

Dasgupta, P. and E. Maskin (2000): "Efficient Auctions," *The Quarterly Journal of Economics*, 115, 341–388.

Dawson, J. (1980): "Optimal Matroid Bases: An Algorithm Based on Cocircuit," *Quart. Journal of Mathematics*, 31, 65–69.

Deb, R. and M. Pai (2013): "Symmetric Auctions," Working Paper, University of Pennsylvania and University of Toronto.

Debreu, G. (1954): "Representation of a Preference Ordering by a Numerical Function," in *Decision Processes*, John Wiley and Sons, 159–165.

Dhangwatnotai, P., T. Roughgarden, and Q. Yan (2010): "Revenue Maximization with a Single Sample," in *Proceedings of the 11th ACM conference on Electronic commerce*, ACM, 129–138.

Dobzinski, S. and N. Nisan (2009): "A Modular Approach to Roberts' Theorem," in *In Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT 2009)*, Springer (Lecture Notes in Computer Science).

Ehlers, L. (2002): "Coalitional strategy-proof house allocation," *Journal of Economic Theory*, 105, 298–317.

Fishburn, P. C. (1970): *Utility Theory for decision Making*, John Wiley and Sons, New York.

Gibbard, A. (1973): "Manipulation of Voting Schemes: A General Result," *Econometrica*, 41, 587–602.

Groves, T. (1973): "Incentives in Teams," *Econometrica*, 41, 617–631.

Hart, S. and N. Nisan (2012): "Approximate Revenue Maximization for Multiple Items," The Hebrew University of Jerusalem, Center for Rationality DP-606.

Hart, S. and P. J. Reny (2012): "Maximizing Revenue with Multiple Goods: Nonmonotonicity and Other Observations," The Hebrew University of Jerusalem, Center for Rationality DP-630.

Hartline, J. and T. Roughgarden (2009): "Simple versus optimal mechanisms," in *Proceedings of the 10th ACM conference on Electronic commerce*, ACM, 225–234.

Hatfield, J. W. (2009): "Strategy-proof, efficient, and nonbossy quota allocations," *Social Choice and Welfare*, 33, 505–515.

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

Holmstrom, B. (1979): "Groves' Scheme on Restricted Domains," *Econometrica*, 47.

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

Jehiel, P., M. M. ter Vehn, and B. Moldovanu (2006): "The limits of Ex-Post Implementation," *Econometrica*, 74, 485–515.

——— (2008): "Ex-Post Implementation and Preference Aggregation via Potentials," *Economic Theory*, 37, 469–490.

Laffont, J. J. and E. Maskin (1980): "A Differential Approach to Dominant Strategy Mechanisms," *Econometrica*, 48, 1507–1520.

Lavi, R. (2007): *Algorithmic Game Theory*, Cambridge University Press, chap. Computationally-Efficient Approximate Mechanisms, 301–330, editors: Noam Nisan and Tim Roughgarden and Eva Tardos and Vijay Vazirani.

Lavi, R., A. Mualem, and N. Nisan (2003): "Towards a Characterization of Truthful Combinatorial Auctions," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03)*, IEEE Press.

Manelli, A. M. and D. R. Vincent (2010): "Bayesian and Dominant Strategy Implementation in the Independent Private Values Model," *Econometrica*, 78, 1905–1938.

Marchant, T. and D. Mishra (2012): "Mechanism Design with Two Alternatives in Quasilinear Environments," Working Paper, Indian Statistical Institute.

Maskin, E. (1992): *Privatization*, Mohr, Tubingen, chap. Auctions and Privatization, 115–136, editor: Horst Siebert.

Mierendorffl, K. (2011): "Asymetric Reduced form auctions," *Economics letters*, 110, 41–44.

Mishra, D. and A. Quadir (2014): "Non-bossy Single Object Auctions," *Economic Theory Bulletin*, 2, 93–110.

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

Mishra, D. and A. Sen (2012): "Roberts' Theorem with Neutrality: A Social Welfare Ordering Approach," *Games and Economic Behavior*, 75, 283–298.

Morimoto, S. and S. Serizawa (2014): "Strategyproofness and Efficiency with Non-quailinear Preferences: A Characterization of Minimum Price Walrasian Rule," *Forthcoming, Theoritical Economics.*

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

Mukherjee, C. (2014): "Fair and Group Strategy-proof Good Allocation with Money," *Social Choice and Welfare*, 42, 289–311.

Muller, E. and M. A. Satterthwaite (1977): "The Equivalence of Strong Positive Association ans Strategy-proofness," *Journal of Economic Theory*, 14, 412–418.

Mutuswami, S. (2005): "Strategyproofness, Non-Bossiness and Group Strategyproofness in a Cost Sharing Model," *Economics Letters*, 89, 83–88.

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

Nisan, N. (2007): "Introduction to Mechanism Design," in *Algorithmic Game Theory*, Cambridge University Press.

Oxley, J. (2006): *Matroid Theory*, Oxford University Press.

Papai, S. (2000): "Strategyproof Assignment by Hierarchical Exchange," *Econometrica*, 68, 1403–1433.

Perry, M. and P. Reny (2002): "An Efficient Auction," *Econometrica*, 70, 1199–1212.

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

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

Saitoh, H. and S. Serizawa (2008): "Vickery Rule Allocation With Income Effect," *Economic Theory*, 35, 391–401.

Sakai, T. (2008): "Second Price Auctions On General Preference Domains: Two Charaterization," *Economic Theory*, 37, 347–356.

——— (2013): "Axiomatizations of second price Auctions With a Reserve Price," *International Journal of Economic Theory*, 9, 255–265.

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

Satterthwaite, M. (1975): "Strategy-proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions," *Journal of Economic Theory*, 10, 187–217.

Satterthwaite, M. and H. Sonnenschein (1981): "Strategy-Proof Allocation Mechanisms at Differentiable Points," *Review of Economic Studies*, 48, 587–597.

Svensson, L.-G. (1999): "Strategy-proof Allocation of Indivisible Goods," *Social Choice and Welfare*, 16, 557–567.

Thomson, W. (2014): "Non-bossiness," Working Paper.

Verma, G. D. (2002): "Standard Auctions with Identity-dependent externalities," *Rand Journal of Economics*, 33, 689–708.

Vickrey, W. (1961): "Counterspeculation, Auctions, and Competitive Sealed Tenders," *Journal of Finance*, 16, 8–37.

Vohra, R. (2004): *Advanced Mathematical Economics*, Routledge.

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

Wilson, R. (1973): "An Introduction To Matroid Theory," *The American Mathematical Monthly*, 80, 500–525.