Theory of Mechanism Design

November 9, 2014

Debasis Mishra [1]

These notes are updated every Fall semester when I offer the course on "Theory of Mechanism Design". I thank numerous students who have taken this course for their input, questions, and suggestions. I am also thankful to Arunava Sen for various discussions that led to a major portion of the note.

[1]Indian Statistical Institute, Email: dmishra@isid.ac.in

1 Introduction

Consider a seller who owns an indivisible object, say a house, and wants to sell it to a set of buyers. Each buyer has a value for the object, which is the utility of the house to the buyer. The seller wants to design a selling procedure, an auction for example, such that he gets the maximum possible price (revenue) by selling the house. If the seller knew the values of the buyers, then he would simply offer the house to the buyer with the highest value and give him a "take-it-or-leave-it" offer at a price equal to that value. Clearly, the (highest value) buyer has no incentive to reject such an offer. Now, consider a situation where the seller is unaware of the values of the buyers. What selling procedure will give the seller the maximum possible revenue? A clear answer is impossible if the seller knows nothing about the values of the buyer. However, the seller may have some information about the values of the buyers. For example, the possible range of values, the probability of having these values etc. Given these information, is it possible to design a selling procedure that guarantees maximum (expected) revenue to the seller?

In this example, the seller had a particular objective in mind - maximizing revenue. Given his objective he wanted to *design* a selling procedure such that when buyers participate in the selling procedure and try to maximize their own payoffs within the rules of the selling procedure, the seller will maximize his expected revenue over all such selling procedures.

The study of mechanism design looks at such issues. A planner (mechanism designer) needs to design a *mechanism* (a selling procedure in the above example) where strategic agents can interact. The interactions of agents result in some outcome. While there are several possible ways to design the rules of the mechanism, the planner has a particular objective in mind. For example, the objective can be *utilitarian* (maximization of the total utility of agents) or maximization of his own utility (as was the case in the last example) or some *fairness* objective. Depending on the objective, the mechanism needs to be designed in a manner such that when strategic agents interact, the resulting outcome gives the desired objective. One can think of mechanism design as the *reverse engineering* of game theory. In game theory terminology, a mechanism induces a *game-form* whose equilibrium outcome is the objective that the mechanism designer has set.

1.1 Private Information and Utility Transfers

The main input to a mechanism design problem is the set of possible outcomes or alternatives. Agents have preferences over the set of alternatives. These preferences are unknown to the mechanism designer. Mechanism design problems can be classified based on the amount of

information asymmetry present between the agents and the mechanism designer.

1. Complete Information: Consider a setting where an accident takes place on the road. Three parties (agents) are involved in the accident. Everyone knows perfectly who is at fault, i.e., who is responsible to what extent for the accident. The traffic police comes to the site but is unaware of the information agents have. The mechanism design problem is to design a set of rules where the traffic police's objective (to punish the true offenders) can be realized. The example given here falls in a broad class of problems where agents perfectly know all the information between themselves, but the mechanism designer does not know this information.

   This class of problems is usually termed as the *implementation problem*. It is usually treated separately from mechanism design because of strong requirements in equilibrium properties in this literature. We will not touch on the implementation problem in this course.

2. Private Information and Interdependence: Consider the sale of a single object. The utility of an agent for the object is his private information. This utility information may be known to him completely, but usually not known to other agents and the mechanism designer. There are instances where the utility information of an agent may not be perfectly known to him. Consider the case where a seat in a flight is being sold by a private airlines. An agent who has never flown this airlines does not completely know his utility for the flight seat. However, there are other agents who have flown this airlines and have better utility information for the flight seat. So, the utility of an agent is influenced by the information of other agents. Still the mechanism designer is not aware of any information agents have.

Besides the type of information asymmetry, mechanism design problems can also be classified based on whether monetary transfers are involved or not. Transfers are a means to redistribute utility among agents.

1. Models without transfers. Consider a setting where a set of agents are deciding to choose a candidate in an election. There is a set of candidates in the election, and each of them is an alternative. Agents have preference over the candidates. Usually monetary transfers are not allowed in such voting problems.

2. Models with transfers and quasi-linear utility. The single object auction is a classic example where monetary transfers are allowed. If an agent buys the object

he is expected to pay an amount to the seller. The net utility of the agent in that case is his utility for the object minus the payment he has to make. Such net utility functions are linear in the payment component, and is referred to as the quasi-linear utility functions.

In this course, we will focus on (a) **voting models without transfers** and (b) **models with transfers and quasi-linear utility**. In voting models, we will mainly deal with **ordinal preferences**, i.e., intensities of preferences will not matter. We will mainly focus on the case where agents have **private information about their preferences over alternatives**. Note that such private information is completely known to the respective agents but not known to other agents and the mechanism designer.

1.2 Examples in Practice

The theory of mechanism design is probably the most successful story of game theory. Its practical applications are found in many places. Below, we will look at some of the applications.

1. *Matching.* Consider a setting where students need to be matched to schools. Students have preferences over schools and schools have preference over students. What mechanisms must be used to match students to schools? This is a model without any transfers. Lessons from mechanism design theory has been used to design centralized matching mechanisms for major US cities like Boston and New York. Such mechanisms and its variants are also used to match kidney donors to patients, doctors to hospitals, and many more.

2. *Sponsored Search Auction.* If you search for a particular keyword on Google, once the search results are displayed, one sees a list of advertisements on the right of the search results. Such slots for advertisements are dynamically sold to potential buyers (advertising companies) as the search takes place. One can think of the slots on a page of search result as a set of indivisible objects. So, the sale of slots on a page can be thought of as simultaneous sale of a set of indivisible objects to a set of buyers. This is a model where buyers make payments to Google. Google uses a variant of a well studied auction in the auction theory literature. Bulk of Google's revenues come from such auctions.

3. *Spectrum Auction.* Airwave frequencies are important for communication. Traditionally, Govt. uses these airwaves for defense communication. In late 1990s, various

Govts. started selling (auctioning) airwaves for private communication. Airwaves for different areas were sold simultaneously. For example, India is divided into various "circles" like Delhi, Punjab, Haryana etc. A communication company can buy the airwaves for one or more circles. Adjacent circles have synergy effects and distant circles have substitutes effects on utility. Lessons from auction theory were used to design auctions for such spectrum sale in US, UK, India, and many other European countries. The success of some of these auctions have become the biggest advertisement of game theory.

2 A General Model of Mechanism Design

We will now formally define a model that will set up some basic ingredients to study mechanism design. The model will be general enough to cover cases where transfers are permitted and where it is excluded.

Let $N := \{1, \ldots, n\}$ be a set of n agents. Let X be the set of possible outcomes. Every agent has some private information, which is called his **type**. The type of agent i is denoted by θ_i. Let Θ_i be the set of all possible types of agent i - Θ_i is usually referred to as the **type space** of agent i. The mechanism designer has complete information about the type space Θ_i of every agent i, but does not know the types of the agent. A type profile will be denoted as $\theta \equiv (\theta_1, \ldots, \theta_n)$. Let $\Theta := \Theta_1 \times \ldots \times \Theta_n$ be the set of type profiles of agents.

Given the type of an agent, it evaluates its utility from every outcome using a utility function. Let $u_i : X \times \Theta_i \to \mathbb{R}$ denote the utility function of agent i. Though the mechanism designer is unaware of the type of the agent, it knows the form of its utility function. In some models, it is customary to allow the utility of an agent to depend on the types of all the agents - such models are called **interdependent value** models. We will mainly focus on the case where utility of an agent only depends on his own type - this is called the **private values** model.

We give some classic examples to clarify the model before proceeding further.

1. Voting. In the voting model, X may represent the set of candidates in an election and N the set of voters. The type θ_i of agent i is a ranking (ordering) of the set of candidates. Then, the utility function u_i is any utility function that represents θ_i. For instance if $X = \{a, b, c\}$ and θ_i is an ordering $\succeq$ such that $a \succeq b, b \succeq c, a \succeq c$, then u_i can be any function satisfying $u_i(a, \theta_i) \geq u_i(b, \theta_i) \geq u_i(c, \theta_i)$.

2. Single object sale/allocation. In this model, we consider sale or allocation of a single indivisible object to a set of buyers. Usually, this also involves payment

or transfer. Hence, an outcome here consists of two components: (a) an allocation decision and (b) a payment amount decision. Abstractly, an outcome $x \in X$ consists of two vectors a and p, where $a \equiv (a_1, \ldots, a_n)$ the allocation vector with $a_i \in \{0, 1\}$ for all $i \in N$ and $\sum_{i \in N} a_i \leq 1$ and $p \equiv (p_1, \ldots, p_n)$ with p_i denoting the payment of agent i. The set of outcomes X is the set of all such (a, p) pairs.

The type θ_i here denotes the value of agent i for the object. A familiar utility function in this setting is

$$u_i((a, p), \theta_i) := a_i \theta_i - p_i.$$

This is called the **quasi-linear utility** function. Whenever transfers/payments are allowed, we will make use of this form of utility function.

3. Choosing a public project. In this model, citizens of a city are deciding whether to build a public project (park, bridge, museum etc.) or not. This is usually accompanied by a decision on the amount of tax each citizen must pay to finance the project. Formally, N is the set of citizens and X consists of outcomes which are pairs of the form (a, p) with $a \in \{0, 1\}$ and $p \equiv (p_1, \ldots, p_n)$ is the payment vector satisfying $\sum_{i \in N} p_i \geq C$, where C is the cost of the project. The type θ_i of agent i denotes the value from the public project. Assuming quasi-linear utility, the utility of agent i from an outcome (a, p) is given by $u_i((a, p), \theta_i) = a\theta_i - p_i$.

4. Choosing one out of many public projects. In this model, citizens of a city are deciding to choose one out of many public projects. Let A be the set of public projects. An outcome consists of a pair (f, p), where for every $a \in A$, $f_a \in \{0, 1\}$ denotes whether project a is chosen or not with $\sum_{a \in A} f_a = 1$ and p is the payment vector satisfying $\sum_{i \in N} p_i \leq C \cdot f$ with C being the cost vector of projects.

   The type θ_i of agent i is a vector in $\mathbb{R}^{|A|}$ in this model with $\theta_i(a)$ denoting the value of agent i for project $a \in A$. Type of this form are called **multidimensional** types. Assuming quasi-linear utility function, the utility of an outcome (f, p) to agent i is given by $u_i((f, p), \theta_i) = f \cdot \theta_i - p_i$.

   A similar model will can be used to describe a setting where a set of objects are being assigned to a set of buyers. Here, type of an agent will represent the values of the agent for each of the objects. An outcome will consist of a feasible assignment and a payment vector.

A **social choice function (SCF)** is a map $F : \Theta \to X$. Hence, an scf chooses an outcome for every type profile of agents. An SCF is supposed to capture all the objectives of

a mechanism designer. If the types of the agents were known to the designer, the outcome chosen by the SCF reflects what the designer would like to do at that type profile.

In settings where transfers are allowed, it is convenient to think of an SCF F as $(f, p_1, \ldots, p_n)$, where $f : \Theta \to A$ with A being a set of decisions/alternatives and $p_i : \Theta \to \mathbb{R}$ being the payment function of agent i. In settings where transfers are not permitted, $X \equiv A$ and $F \equiv f$.

Given an SCF F, the utility of an agent i with type θ_i when agents report $\hat{\theta}$ to the SCF is given by

$$U_i(\hat{\theta}, \theta_i; F) := u_i(F(\hat{\theta}), \theta_i).$$

A mechanism is a more complicated object than an SCF. The main objective of a mechanism is to set up rules of interaction between agents. These rules are often designed with the objective of realizing the outcomes of a social choice function. The basic ingredient in a mechanism is a **message**. A message is a communication between the agent and the mechanism designer. A mechanism must specify the **message space** - the set of all possible messages. Given a message profile, the mechanism must choose an outcome. Hence, a **mechanism** is defined as $\mathcal{M} \equiv (M_1, \ldots, M_n, g)$, where for every $i \in N$, M_i is the message space of agent i and $g : M_1 \times \ldots \times M_n \to X$ is the decision rule.

A special form of mechanism is a **direct mechanism**, where $M_i = \Theta_i$ for every $i \in N$. So, in a direct mechanism every agent communicates a type from his type space to the mechanism designer. Hence, the decision rule in a direct mechanism is an SCF.

However, the message space of a mechanism can be quite complicated. Consider the sale of a single object by a "price-based" procedure. The mechanism designer announces a price and asks every buyer to communicate if it wants to buy the object at the announced price. The price is raised if more than one buyer expresses interest in buying the object, and the procedure is repeated till exactly one buyer shows interest. The message space in such a mechanism is quite complicated.

3 Dominant Strategy Incentive Compatibility

The goal of mechanism design is to design the message space and outcome function in a way such that when agents participate in the mechanism they have (best) strategies (messages) that they can choose as a function of their private types such that the desired outcome is achieved. The most fundamental, though somewhat demanding, notion in mechanism design is the notion of dominant strategies. A strategy $m_i \in M_i$ is a **dominant strategy** at $\theta_i \in \Theta_i$

in a mechanism $(M_1, \ldots, M_n, g)$ if for every $m_{-i} \in M_{-i}$ [2] we have

$$u_i(g(m_i, m_{-i}), \theta_i) \geq u_i(g(\hat{m}_i, m_{-i}), \theta_i) \qquad \forall\ \hat{m}_i \in M_i.$$

Alternatively, a strategy $m_i \in M_i$ is a dominant strategy at $\theta_i \in \Theta_i$ in a mechanism $(M_1, \ldots, M_n, g)$ if for every $m_{-i} \in M_{-i}$

$$u_i(g(m_i, m_{-i}), \theta_i) = \max_{\hat{m}_i \in M_i} u_i(g(\hat{m}_i, m_{-i}), \theta_i).$$

Notice the strong requirement that m_i has to be the best strategy for *every* strategy profile of other agents. Such a strong requirement limits the settings where dominant strategies exist.

A social choice function F is **implemented** in dominant strategies by a mechanism $\mathcal{M} \equiv (M_1, \ldots, M_n, g)$ if there exists mappings for every agent $i \in N$, $m_i : \Theta_i \to M_i$ such that $m_i(\theta_i)$ is a dominant strategy at θ_i for every $\theta_i \in \Theta_i$ and $g(m(\theta)) = F(\theta)$ for all $\theta \in \Theta$.

A direct mechanism (or associated social choice function) is **strategy-proof** or **incentive compatible** if for every agent $i \in N$ and every $\theta_i \in \Theta_i$, θ_i is a dominant strategy at θ_i. In other words, F is strategy-proof if for every agent $i \in N$, every $\theta_{-i} \in \Theta_{-i}$, and every $\theta_i, \theta_i' \in \Theta_i$, we have

$$u_i(F(\theta_i, \theta_{-i}), \theta_i) \geq u_i(F(\theta_i', \theta_{-i}), \theta_i),$$

i.e., truth-telling is a dominant strategy.

So, to verify whether a social choice function is implementable or not, we need to search over infinite number of mechanisms whether any of them implements this SCF. A fundamental result in mechanism design says that one can restrict attention to the direct mechanisms.

PROPOSITION **1 (Revelation Principle)** *If a mechanism $\mathcal{M} \equiv (M_1, \ldots, M_n, g)$ implements a social choice function F in dominant strategies then the direct mechanism F is strategy-proof.*

Proof: Fix an agent $i \in N$. Consider two types $\theta_i, \theta_i' \in \Theta_i$. Consider θ_{-i} to be the report of other agents. Let $m_i(\theta_i) = m_i$ and $m_{-i}(\theta_{-i}) = m_{-i}$, where for all $j \in N$, m_j is the dominant strategy message function of agent $j \in N$. Similarly, $m_i(\theta_i') = m_i'$. Then, using the fact that F is implemented by $\mathcal{M} \equiv (M_1, \ldots, M_n, g)$ in dominant strategies, we get

$$\begin{aligned} u_i(F(\theta_i, \theta_{-i}), \theta_i) &= u_i(g(m_i, m_{-i}), \theta_i) \\ &\geq u_i(g(m_i', m_{-i}), \theta_i) \\ &= u_i(F(\theta_i', \theta_{-i}), \theta_i). \end{aligned}$$

[2] Here, m_{-i} is the profile of messages of agents except agent i and M_{-i} is the cross product of message spaces of agents except agent i.

Hence, F is strategy-proof. ■

Thus, a social choice function F is implementable in dominant strategies if and only if the direct mechanism F is strategy-proof. Revelation principle is a central result in mechanism design. One of its implications is that if we wish to find out what social choice functions can be implemented in dominant strategies, we can restrict attention to direct mechanisms. This is because, if some non-direct mechanism implements a social choice function in dominant strategies, revelation principle says that the corresponding direct mechanism is also strategy-proof.

Of course, a drawback is that a direct mechanism may leave out some equilibria of the main mechanism. The original mechanism may have some equilibria that may get ruled out because of restricting to the direct mechanism since it has smaller strategy space. In general, this is a criticism of the mechanism design theory. Even in a direct mechanism, incentive compatibility only insists that truth-telling is an equilibrium but there may be other equilibria of the mechanism which may not implement the given social choice function. These stronger requirement that every equilibria, truth-telling or non-truth-telling, must correspond to the social choice function outcome is the cornerstone of the implementation literature.

4 Bayesian Incentive Compatibility

Bayesian incentive compatibility was introduced in Harsanyi (1967-68). It is a weaker requirement than the dominant strategy incentive compatibility. While dominant strategy incentive compatibility required the equilibrium strategy to be the best strategy under all possible strategies of opponents, Bayesian incentive compatibility requires this to hold in *expectation*. This means that in Bayesian incentive compatibility, an equilibrium strategy must give the highest expected utility to the agent, where we take expectation over types of other agents. To be able to take expectation, agents must have information about the probability distributions from which types of other agents are drawn. Hence, Bayesian incentive compatibility is informationally demanding. In dominant strategy incentive compatibility the mechanism designer needed information on the type space of agents, and every agent required no prior information of other agents to compute his equilibrium. In Bayesian incentive compatibility, every agent and the mechanism designer needs to know the distribution from which agents' types are drawn.

To understand Bayesian incentive compatibility, fix a mechanism (M, g). A Bayesian strategy for such a mechanism is a vector of mappings $m_i : \Theta_i \to M_i$ for every $i \in N$. A

profile of such mapping $(m_1, \ldots, m_n)$ is a **Bayesian equilibrium** if for all $i \in N$, for all $\theta_i \in \Theta_i$, and for all $\hat{m}_i \in M_i$ we have

$$E_{-i}\big[u_i(g(m_{-i}(\theta_{-i}), m_i(\theta_i)), \theta_i)|\theta_i\big] \geq E_{-i}\big[u_i(g(m_{-i}(\theta_{-i}), \hat{m}_i)), \theta_i)|\theta_i\big],$$

where $E_{-i}[\cdot]$ denotes the expectation over type profile θ_{-i} *conditional* on the fact that i has type θ_i. If all θ_is are drawn independently, then we need not condition in the expectation.

A direct mechanism (social choice function) F is **Bayesian incentive compatible** if $m_i(\theta_i) = \theta_i$ for all $i \in N$ and for all $\theta_i \in T_i$ is a Bayesian equilibrium, i.e., for all $i \in N$ and for all $\theta_i, \hat{\theta}_i \in \Theta_i$ we have

$$E_{-i}\big[u_i(F(\theta_{-i}, \theta_i), \theta_i)|\theta_i\big] \geq E_{-i}\big[u_i(F(\theta_{-i}, \hat{\theta}_i), \theta_i)|\theta_i\big]$$

A dominant strategy incentive compatible mechanism is Bayesian incentive compatible. A mechanism (M, g) **realizes** a social choice function F in Bayesian equilibrium if there exists a Bayesian equilibrium $m : \Theta \to M$ of (M, g) such that $g(m(\theta)) = F(\theta)$ for all $i \in N$ and for all $\theta \in \Theta$. Analogous to the revelation principle for dominant strategy incentive compatibility, we also have a revelation principle for Bayesian incentive compatibility. The proof is similar to Proposition 1 and is skipped.

Proposition **2 (Revelation Principle)** *If a mechanism (M, g) realizes a social choice function F in Bayesian equilibrium, then the direct mechanism F is Bayesian incentive compatible.*

Like the revelation principle of dominant strategy incentive compatibility, the revelation principle for Bayesian incentive compatibility is not immune to criticisms for multiplicity of equilibria.

5 The Strategic Voting Model

We now discuss a general model of voting and examine the consequence of incentive compatibility in this model. The model is very general and introduces us to the rich literature on strategic voting models where monetary transfers are excluded.

Let A be a finite set of alternatives with $|A| = m$. Let N be a finite set of individuals or agents or voters with $|N| = n$. Every agent has a preference over the set of alternatives. Let P_i denote the preference of agent i. The preferences can be represented in many ways. Here is one plausible way of representing the preferences. A preference relation R_i of agent i is called an **ordering** if it satisfies the following properties:

- COMPLETENESS: For all $a, b \in A$ either aR_ib or bR_ia.
- REFLEXIVITY: For all $a \in A$, aR_ia.
- TRANSITIVITY: For all $a, b, c \in A$, $\left[aR_ib, bR_ic\right] \Rightarrow \left[aR_ic\right]$.

We will denote the set of all orderings over A as $\mathcal{R}$. By definition, an ordering gives ordered pairs of A. An ordering R_i is a **linear ordering** if for any $a, b \in A$, aR_ib and bR_ia means $a = b$, i.e., indifference is not allowed.

We assume that the preference ordering of every agent is a **linear ordering**. Given a preference ordering P_i of agent i, we say aP_ib if and only if a is strictly preferred to b under P_i. Further, the top ranked element of this ordering is denoted by $P_i(1)$, the second ranked element by $P_i(2)$, and so on. Let $\mathcal{P}$ be the set of all strict preference orderings over A. A profile of preference orderings (or simply a preference profile) is denoted as $P \equiv (P_1, \ldots, P_n)$. So, $\mathcal{P}^n$ is the set of all preference profiles. A **social choice function (SCF)** is a mapping $f : \mathcal{P}^n \to A$. Note that this definition of a social choice function implicitly assumes that all possible profiles of linear orderings are permissible. This is known as the **unrestricted domain** assumption in the strategic voting (social choice) literature. Later, we will study some interesting settings where the domain of the social choice function is restricted.

Every agent knows his own preference ordering but does not know the preference ordering of other agents, and the mechanism designer (planner) does not know the preference orderings of agents. This is a very common situation in many voting scenarios: electing a candidate among a set of candidates, selecting a project among a finite set of projects for a company, selecting a public facility location among a finite set of possible locations, etc. Monetary transfers are precluded in these settings. The objective of this section is to find out which social choice functions are implementable in dominant strategies in such strategic voting scenarios.

We first describe several desirable properties of an SCF. The first property is an efficiency property. We say an alternative $a \in A$ is **Pareto dominated** at a preference profile P if there exists an alternative $b \in A$ such that bP_ia for all $i \in N$. Efficiency requires that no Pareto dominated alternative must be chosen.

DEFINITION **1** *A social choice function f is* **efficient** [3] *if for every profile of preferences P and every $a \in A$, if a is Pareto dominated at P then $f(P) \neq a$.*

The next property requires to respect unanimity.

[3]Such a social choice function is also called Pareto optimal or Pareto efficient or ex-post efficient.

DEFINITION **2** *A social choice function f is* **unanimous** *if for every preference profile $P \equiv (P_1, \ldots, P_n)$ with $P_1(1) = P_2(1) = \ldots = P_n(1) = a$ we have $f(P) = a$.*

Note that this version of unanimity is a stronger version than requiring that if the *preference ordering* of all agents is the same, then the top ranked alternative must be chosen. This definition requires only the top to be the same, but other alternatives can be ranked differently by different agents.

Next, we define the strategic property of a social choice function.

DEFINITION **3** *A social choice function f is* **manipulable by agent i at profile $P \equiv (P_i, P_{-i})$** [4] **by preference ordering P_i'** *if $f(P_i', P_{-i}) P_i f(P)$. A social choice function f is* **strategy-proof** *if it is not manipulable by any agent i at any profile P by any preference ordering P_i'.*

This notion of strategy-proofness is the dominant strategy requirement since no manipulation is possible for every agent for every possible profile of other agents.

Finally, we define a technical property on the social choice function.

DEFINITION **4** *A social choice function f is* **onto** *if for every $a \in A$ there exists a profile of preferences $P \in \mathcal{P}^n$ such that $f(P) = a$.*

5.1 Examples of Social Choice Functions

We give some examples of social choice functions.

- CONSTANT SCF. A social choice function f^c is a constant SCF if there is some alternative $a \in A$ such that for every preference profile P, we have $f^c(P) = a$. This SCF is strategy-proof but not unanimous.

- DICTATORSHIP SCF. A social choice function f^d is a **dictatorship** if there exists an agent i, called the dictator, such that for every preference profile P, we have $f^d(P) = P_i(1)$. Dictatorship is strategy-proof and onto. Moreover, as we will see later, they are also efficient and unanimous.

- PLURALITY SCF (WITH FIXED TIE-BREAKING). Plurality is a popular way of electing an alternative. Here, we present a version that takes care of tie-breaking carefully. For every preference profile P and every alternative $a \in A$, define the score of a in P as

[4] We use the standard notation P_{-i} to denote the preference profile of agents other than agent i.

$s(a, P) = |\{i \in N : P_i(1) = a\}|$. Define $\tau(P) = \{a \in A : s(a, P) \geq s(b, P) \; \forall \; b \in A\}$ for every preference profile P, and note that $\tau(P)$ is non-empty. Let $\succ^T$ be a linear ordering over alternatives A that we will use to break ties. . A social choice function f^p is called a plurality SCF with tie-breaking according to $\succ^T$ if for every preference profile P, $f^p(P) = a$, where $a \in \tau(P)$ and $a \succ^T b$ for all $b \in \tau(P) \setminus \{a\}$.

Though the plurality SCF is onto, it is not strategy-proof. To see this, consider an example with three agents $\{1, 2, 3\}$ and three alternatives $\{a, b, c\}$. Let $\succ^T$ be defined as: $a \succ^T b \succ^T c$. Consider two preference profiles shown in Table 1. We note first that $f(P) = a$ and $f(P') = b$. Since bP_3a, agent 3 can manipulate at P by P'_3.

P_1	P_2	P_3	$P'_1 = P_1$	$P'_2 = P_2$	P'_3
a	b	c	a	b	b
b	c	b	b	c	a
c	a	a	c	a	c

Table 1: Plurality SCF is manipulable.

- Borda SCF (with fixed tie-breaking). The Borda SCF is a generalization of the Plurality voting SCF. The tie-breaking in this SCF is defined similar to Plurality SCF. Let $\succ^T$ be a linear ordering over alternatives A that we will use to break ties. Fix a preference profile P. For every alternative $a \in A$, the *rank* of a in P_i for agent i is given by $r(a, P_i) = k$, where $P_i(k) = a$. From this, the score of alternative a in preference profile P is computed as $s(a, P) = \sum_{i \in N}[|A| - r(a, P_i)]$. Define for every preference profile P, $\tau(P) = \{a \in A : s(a, P) \geq s(b, P) \; \forall b \in A\}$. A social choice function f^b is called a Borda SCF with tie-breaking according to $\succ^T$ if for every preference profile P, $f^b(P) = a$ where $a \in \tau(P)$ and $a \succ^T b$ for all $b \in \tau(P) \setminus \{a\}$.

  Like the Plurality SCF, the Borda SCF is onto but manipulable. To see this, consider an example with three agents $\{1, 2, 3\}$ and three alternatives $\{a, b, c\}$. Let $\succ^T$ be defined as: $c \succ^T b \succ^T a$. Consider two preference profiles shown in Table 2. We note first that $f(P) = b$ and $f(P') = c$. Since cP_1b, agent 1 can manipulate at P by P'_1.

5.2 Implications of Properties

We now examine the implications of these properties. We start out with a simple characterization of strategy-proof social choice functions using the following monotonicity property.

P_1	P_2	P_3	P_1'	$P_2' = P_2$	$P_3' = P_3$
a	b	b	c	b	b
c	c	c	a	c	c
b	a	a	b	a	a

Table 2: Borda SCF is manipulable.

Such monotonicity properties are heart of every incentive problem - though the nature of monotonicity may differ from problem to problem.

For any alternative $a \in A$, let $B(a, P_i)$ be the set of alternatives below a in preference ordering P_i. Formally, $B(a, P_i) := \{b \in A : aP_ib\}$.

DEFINITION **5** *A social choice function f is* **monotone** *if for any two profiles P and P' with $B(f(P), P_i) \subseteq B(f(P), P_i')$ for all $i \in N$, we have $f(P) = f(P')$.*

Note that in the definition of monotonicity when we go from a preference profile P to P' with $f(P) = a$, whatever was below a in P for every agent continues to be below it in P' also, but other relations may change. For example, the following is a valid P and P' in the definition of monotonicity with $f(P) = a$ (see Table 3).

P_1	P_2	P_3	P_1'	P_2'	P_3'
a	b	c	a	a	a
b	a	a	b	c	c
c	c	b	c	b	b

Table 3: Two valid profiles for monotonicity

THEOREM **1** *A social choice function $f : \mathcal{P}^n \to A$ is strategy-proof if and only if it is monotone.*

Proof: Consider social choice function $f : \mathcal{P}^n \to A$ which is strategy-proof. Consider two preference profiles P and P' such that $f(P) = a$ and $B(a, P_i) \subseteq B(a, P_i')$ for all $i \in N$. We define $(n-1)$ new preference profiles. Define preference profile P^1 as follows: $P_1^1 = P_1'$ and $P_i^1 = P_i$ for all $i > 1$. Define preference profile P^k for $k \in \{1, \ldots, n-1\}$ as $P_i^k = P_i'$ if $i \leq k$ and $P_i^k = P_i$ if $i > k$. Set $P^0 = P$ and $P^n = P'$. Note that if we pick two preference profiles P^k and P^{k+1} for any $k \in \{0, \ldots, n-1\}$, then preference of all agents other than agent $(k+1)$ are same in P^k and P^{k+1}, and preference of agent $(k+1)$ is changing from P_{k+1} in P^k to P_{k+1}' in P^{k+1}.

We will show that $f(P^k) = a$ for all $k \in \{0, \ldots, n\}$. We know that $f(P^0) = f(P) = a$, and consider $k = 1$. Assume for contradiction $f(P^1) = b \neq a$. If bP_1a, then agent 1 can manipulate at P^0 by P^1. If aP_1b, then aP_1^1b, and agent 1 can manipulate at P^1 by P_1^0. This is a contradiction since f is strategy-proof.

We can repeat this argument by assuming that $f(P^q) = a$ for all $q \leq k < n$, and showing that $f(P^{k+1}) = a$. Assume for contradiction $f(P^{k+1}) = b \neq a$. If $bP_{k+1}a$, then agent $(k+1)$ can manipulate at P^k by P^{k+1}. If $aP_{k+1}b$ then $aP'_{k+1}b$. This means agent $(k+1)$ can manipulate at P^{k+1} by P_{k+1}^k. This is a contradiction since f is strategy-proof.

Hence, by induction, $f(P^n) = f(P') = a$, and f is monotone.

Now suppose, $f : \mathcal{P}^n \to A$ is a monotone social choice function. Assume for contradiction that f is not strategy-proof. In particular, agent i can manipulate at preference profile P by a preference ordering P'_i. Let $P' \equiv (P'_i, P_{-i})$. Suppose $f(P) = a$ and $f(P') = b$, and by assumption bP_ia. Consider a preference profile $P'' \equiv (P''_i, P_{-i})$, where P''_i is any preference ordering satisfying $P''_i(1) = b$ and $P''_i(2) = a$. By monotonicity, $f(P'') = f(P') = b$ and $f(P'') = f(P) = a$, which is a contradiction. ■

Theorem 1 is a strong result. The necessity of monotonicity is true in any domain - even if a subset of all possible preference profiles are permissible. Even for sufficiency, we just need a domain where we are able to rank any pair of alternatives first and second.

We now explore the implications of other properties.

LEMMA **1** *If an SCF f is monotone and onto then it is efficient.*

Proof: Consider $a, b \in A$ and a preference profile P such that aP_ib for all $i \in N$. Assume for contradiction $f(P) = b$. Since f is onto, there exists a preference profile P' such that $f(P') = a$. We construct another preference profile $P'' \equiv (P''_1, \ldots, P''_n)$ as follows. For all $i \in N$, let $P''_i(1) = a$, $P''_i(2) = b$, and $P''_i(j)$ for $j > 2$ can be set to anything. Since f is monotone, $f(P'') = f(P) = b$, and also, $f(P'') = f(P') = a$. This is a contradiction. ■

LEMMA **2** *If an SCF f is efficient then it is unanimous.*

Proof: Consider a preference profile $P \equiv (P_1, \ldots, P_n)$ with $P_1(1) = P_2(1) = \ldots = P_n(1) = a$. Consider any $b \neq a$. By definition, aP_ib for all $i \in N$. By efficiency, $f(P) \neq b$. Hence, $f(P) = a$. ■

LEMMA **3** *If a social choice function is unanimous then it is onto.*

Proof: Take any alternative $a \in A$ and a social choice function f. Consider a profile P such that $P_i(1) = a$ for all $i \in N$. Then $f(P) = a$ by unanimity. So, f is onto. ■

We can summarize these results in the following proposition.

PROPOSITION **3** *Suppose $f : \mathcal{P}^n \to A$ is a strategy-proof social choice function. Then, f is onto if and only if it is efficient if and only if it is unanimous.*

Proof: Suppose f is strategy-proof. By Theorem 1, it is monotone. Then, Lemmas 1, 2, and 3 establish the result. ■

5.3 The Gibbard-Satterthwaite Theorem

THEOREM **2 (Gibbard-Satterthwaite Theorem)** *Suppose $|A| \geq 3$. A social choice function $f : \mathcal{P}^n \to A$ is onto and strategy-proof if and only if it is a dictatorship.*

Before we discuss the proof, we make the following observations about the Gibbard-Satterthwaite (GS) theorem.

1. $|A| = 2$. The GS theorem fails when there are only two alternatives. An example of a non-dictatorial social choice function which is onto and strategy-proof is the plurality social choice function with a fixed tie-breaking. (The proof of this fact is an exercise.)

2. **Unrestricted domain.** The assumption that the type space of each agent consists of all possible strict orderings over A is critical in the GS theorem. Before we proceed to discuss restricted domains in the next section, we discuss here (in full generality) the implications of restricting domains. Suppose $F : \Theta \to X$ be an arbitrary direct mechanism (social choice function). Consider another type space $\Theta' \subseteq \Theta$. Let F' be the restriction of F to Θ', i.e., $F'(\theta) := F(\theta)$ for all $\theta \in \Theta'$. Now, it is easy to see that if F is incentive compatible, F' is also incentive compatible. However, the set of incentive compatible social choice functions may expand as we restrict our type space. For instance, in the degenerate case, where the type space consists of a singleton element, the mechanism designer perfectly knows the type of the agent and can implement any social choice function.

   The intuition about why the set of strategy-proof social choice functions become larger as we restrict the type space is very simple. In a smaller type space, agents have less

opportunity to manipulate a social choice functions and, hence, it is easier for incentive constraints to hold.

It is because of this reason, the GS theorem fails in various *restricted domains.* In particular, a domain $\mathcal{D} \subseteq \mathcal{R}$ is called a restricted domain if $\mathcal{P} \nsubseteq \mathcal{D}$. This will be the focus of discussion in the next section. We will show that various non-dictatorial social choice functions can be strategy-proof in these domains.

3. **Indifference**. Suppose every agent has a preference ordering which is not necessarily anti-symmetric, i.e., there are ties between alternatives. Let $\mathcal{R}$ be the set of all preference orderings. Note that $\mathcal{P} \subsetneq \mathcal{R}$. Now, consider a domain $\mathcal{D} \subseteq \mathcal{R}$ such that $\mathcal{P} \subseteq \mathcal{D}$. Call such a domain **admissible**. A social choice function $f : \mathcal{D}^n \to A$ is **admissible** if $\mathcal{D}$ is admissible. In other words, if the domain of preference orderings include *all possible* linear orderings, then such a domain is admissible. The GS theorem is valid in admissible domains, i.e., if $|A| \geq 3$ and $f : \mathcal{D}^n \to A$ is admissible, onto, and strategy-proof, then it is a dictatorship. The proof follows from the observation that the proof of GS-Theorem only requires existence of certain strict preference orderings. So, as long as such preference orderings exist, the GS-Theorem proof goes through.

   However, dictatorship may not be strategy-proof when indifference is permitted. For instance, consider the dictatorship SCF f as follows. It always selects an alternative in agent 1's top - so, agent 1 is the dictator. However, if there are more than one alternative in agent 1's top, then the following tie-breaking rule is followed. Let $\succ$ be a linear ordering over A. Consider a profile P such that $P_1(1)$ has more than one element [5]. Then, consider $P_2(1)$. If $P_2(1) \cap P_1(1)$ is non-empty, choose an element from $P_2(1) \cap P_1(1)$ using $\succ$, i.e., breaking ties according to $\succ$. Else, choose an alternative from $P_1(1)$ using $\succ$. As an example, suppose agent 1's top consists of $\{a, b, c\}$. Agent 2's top consists of b and c. The tie-breaking is done using $\succ$, and it has $b \succ c$. So, the outcome at this profile must be b. If agent 2's top did not have an element in $\{a, b, c\}$ and $a \succ b \succ c$, then the outcome will be a.

   Such dictatorship SCFs are manipulable. To see this, consider a setting with three alternatives $\{a, b, c\}$ and two agents. Suppose we use the dictatorship of the previous example with $a \succ b \succ c$. Consider a profile where agent 1's top consists of b and c. But agent 2's top has a followed by c, and then followed by b at the bottom. Then, according to the SCF, b will be the outcome. Note that b is the worst alternative for agent 2. He can improve it by reporting c as his unique top since the outcome will now

[5] Since we allow for indifference, $P_i(1)$ for any agent i is a subset of alternatives.

change to c.

4. **Cardinalization.** Instead of types to be strict orderings, we can think of types to be utility numbers on alternatives consistent with some ordering. For instance, let $\mathcal{U}$ be the set of all utility functions $u : A \to \mathbb{R}$. We can assume that the type space of each agent is $\mathcal{U}$ instead of $\mathcal{P}$. In that case, the scf is no longer *ordinal* since it considers cardinal utility of each agent.

   The following simple argument illustrates that as long as the solution concept is dominant strategies, considering cardinal scfs does not expand the set of strategy-proof scfs. To see this, consider agent i and let the utility functions of other agents be fixed at u_{-i}. Suppose f is a strategy-proof scf. Let u_i and u_i' be to utility functions of agent i representing the same strict ordering over alternatives, i.e., for any $a, b \in A$, we have $u_i(a) > u_i(b)$ if and only if $u_i'(a) > u_i'(b)$. We will argue that $f(u_i, u_{-i}) = f(u_i', u_{-i})$. Assume for contradiction $f(u_i, u_{-i}) = a \neq b = f(u_i', u_{-i})$. If $u_i(a) < u_i(b)$, then agent i manipulates at (u_i, u_{-i}) by u_i'. If $u_i(a) > u_i(b)$, then $u_i'(a) < u_i'(b)$, and agent i manipulates at (u_i', u_{-i}) by u_i. This is a contradiction. This shows that f must be *ordinal*, i.e., must ignore cardinal intensities. Hence, it is without loss of generality to assume that the type space of each agent is a set of preference orderings.

5.4 Proof of the Gibbard-Satterthwaite Theorem

We do the proof using induction on number of agents. We first analyze the case when $n = 2$.

Lemma **4** *Suppose $|A| \geq 3$ and $N = \{1, 2\}$. Suppose f is an onto and strategy-proof social choice function. Then for every preference profile P, $f(P) \in \{P_1(1), P_2(1)\}$.*

Proof: Fix a preference profile $P = (P_1, P_2)$. If $P_1(1) = P_2(1)$, the claim is due to unanimity (Proposition 3). Else, let $P_1(1) = a$ and $P_2(1) = b$, where $a \neq b$. Assume for contradiction $f(P) = c \notin \{a, b\}$. We will use the preference profiles shown in Table 4.

P_1	P_2	P_1	P_2'	P_1'	P_2'	P_1'	P_2
a	b	a	b	a	b	a	b
$\cdot$	$\cdot$	$\cdot$	a	b	a	b	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 4: Preference profiles required in proof of Lemma 10.

Consider a preference ordering P_2' for agent 2 where $P_2'(1) = b$, $P_2'(2) = a$, and the remaining ordering can be anything. By efficiency, $f(P_1, P_2') \in \{a, b\}$. Further $f(P_1, P_2') \neq b$ since agent 2 can then manipulate at P by P_1. So, $f(P_1, P_2') = a$.

Now, consider a preference ordering P_1' for agent 1 where $P_1'(1) = a$, $P_1'(2) = b$, and the remaining ordering can be anything. Using an analogous argument, we can show that $f(P_1', P_2) = b$. Now, consider the preference profile (P_1', P_2'). By monotonicity (implied by strategy-proofness - Theorem 1) , $f(P_1', P_2') = f(P_1, P_2') = a$ and $f(P_1', P_2') = f(P_1', P_2) = b$. This is a contradiction. ■

LEMMA **5** *Suppose* $|A| \geq 3$ *and* $N = \{1, 2\}$. *Suppose* f *is onto and strategy-proof social choice function. Consider a profile* P *such that* $P_1(1) = a \neq b = P_2(1)$. *Consider a preference profile* $P' = (P_1', P_2')$ *with* $P_1'(1) = c$ *and* $P_2'(1) = d$. *If* $f(P) = a$, *then* $f(P') = c$ *and if* $f(P) = b$ *then* $f(P') = d$.

Proof: We can assume that $c \neq d$, since the claim is true due to unanimity when $c = d$. We do the proof for different possible cases.

CASE 1: $c = a$, $d = b$. This case establishes a *tops-only* property. From Lemma 10, $f(P') \in \{a, b\}$. Assume for contradiction $f(P') = b$ (i.e., agent 2's top is chosen). Consider a preference profile $\hat{P} \equiv (\hat{P}_1, \hat{P}_2)$ such that $\hat{P}_1(1) = a, \hat{P}_1(2) = b$ and $\hat{P}_2(1) = b, \hat{P}_2(2) = a$ (See Table 5). By monotonicity, $f(\hat{P}) = f(P') = f(P)$, which is a contradiction.

P_1	P_2	P_1'	P_2'	$\hat{P}_1$	$\hat{P}_2$
a	b	a	b	a	b
$\cdot$	$\cdot$	$\cdot$	$\cdot$	b	a
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 5: Preference profiles required in Case 1.

CASE 2: $c \neq a$, $d = b$. Consider any profile $\hat{P} = (\hat{P}_1, \hat{P}_2)$, where $\hat{P}_1(1) = c \neq a$, $\hat{P}_1(2) = a$, and $\hat{P}_2(1) = b$ (See Table 6).

By Lemma 10, $f(\hat{P}) \in \{b, c\}$. Suppose $f(\hat{P}) = b$. Then, agent 1 can manipulate by reporting any preference ordering where his top is a, and this will lead to a as the outcome (Case 1). Hence, $f(\hat{P}) = c = \hat{P}_1(1)$. Using Case 1, $f(P') = c$.

P_1	P_2	P_1'	P_2'	$\hat{P}_1$	$\hat{P}_2$
a	b	$c \neq a$	$d = b$	c	b
$\cdot$	$\cdot$	$\cdot$	$\cdot$	a	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 6: Preference profiles required in Case 2.

Case 3: $c \notin \{a, b\}$, $d \neq b$ [6]. Consider a preference profile $\hat{P}$ such that $\hat{P}_1(1) = c, \hat{P}_2(1) = d, \hat{P}_2(2) = b$ (See Table 7).

P_1	P_2	P_1'	P_2'	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_1$	$\hat{P}_2'$
a	b	$c \neq \{a, b\}$	$d \neq b$	c	d	c	b
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	b	$\cdot$	d
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 7: Preference profiles required in Case 3.

By Lemma 10, $f(\hat{P}) \in \{c, d\}$. Suppose $f(\hat{P}) = d$. Then consider $\hat{P}_2'$ such that $\hat{P}_2'(1) = b$ and $\hat{P'}_2(2) = d$ (See Table 7). By Case 2, $f(\hat{P}_1, \hat{P}_2') = c$. Since $d\hat{P}_2'c$, agent 2 will manipulate at $(\hat{P}_1, \hat{P}_2')$ by $\hat{P}_2$. Hence, $f(\hat{P}) = c$. Using Lemma Case 1, $f(P') = c$.

Case 4: $c = a$, $d \neq b$. By Lemma 10, $f(P') \in \{a, d\}$. Assume for contradiction $f(P') = d$. Consider a preference ordering $\hat{P}_2$ such that $\hat{P}_2(1) = b$ and $\hat{P}_2(2) = d$ (See Table 8).

P_1	P_2	P_1'	P_2'	P_1'	$\hat{P}_2$
a	b	$c = a$	$d \neq b$	a	b
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	d
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 8: Preference profiles required in Case 4.

Now, by Case 1, $f(P_1', \hat{P}_2) = a$. But $d\hat{P}_2a$. Hence, agent 2 can manipulate at $(P_1', \hat{P}_2)$ by P_2', which is a contradiction. Using Case 1, $f(P') = a$.

Case 5: $c = b$, $d \neq a$. By Lemma 10, $f(P') \in \{b, d\}$. Assume for contradiction $f(P') = d$. Consider a preference ordering $\hat{P}_1$ such that $\hat{P}_1(1) = b$ and $\hat{P}_1(2) = a$, and $\hat{P}_2$ such that $\hat{P}_2(1) = d$. Consider another preference ordering $\hat{P'}_1$ such that $\hat{P'}_1(1) = a$ (See Table 9).

[6] This case actually covers two cases: one where $d = a$ and the other where $d \notin \{a, b\}$.

P_1	P_2	P'_1	P'_2	$\hat{P}_1$	$\hat{P}_2$	$\hat{P'}_1$	$\hat{P}_2$
a	b	$c = b$	$d \neq a$	b	d	a	d
$\cdot$	$\cdot$	$\cdot$	$\cdot$	a	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 9: Preference profiles required in Case 5.

By Cases 1 and 4, $f(\hat{P'}_1, \hat{P}_2) = a$. But $a\hat{P}_1 d$. So, agent 1 can manipulate $(\hat{P}_1, \hat{P}_2)$ by $\hat{P'}_1$. This is a contradiction. Using Case 1, $f(P') = b$

Case 6: $c = b$, $d = a$. Since there are at least three alternatives, consider $x \notin \{a, b\}$. Consider a preference ordering $\hat{P}_1$ such that $\hat{P}_1(1) = b$ and $\hat{P}_1(2) = x$ (See Table 10).

P_1	P_2	P'_1	P'_2	$\hat{P}_1$	P'_2	$\hat{P'}_1$	P'_2
a	b	$c = b$	$d = a$	b	a	x	a
$\cdot$	$\cdot$	$\cdot$	$\cdot$	x	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 10: Preference profiles required in Case 6.

By Lemma 10, $f(\hat{P}_1, P'_2) \in \{b, a\}$. Assume for contradiction $f(\hat{P}_1, P'_2) = a$. Consider a preference ordering $\hat{P'}_1$ such that $\hat{P'}_1(1) = x$ (See Table 10). By Case 3, $f(\hat{P'}_1, P'_2) = x$. But $x\hat{P}_1 a$. Hence, agent 1 can manipulate $(\hat{P}_1, P'_2)$ by $\hat{P'}_1$. This is a contradiction. Hence, $f(\hat{P}_1, P'_2) = b$. By Case 1, $f(P') = b$. ■

Proposition **4** *Suppose $|A| \geq 3$ and $n = 2$. A social choice function is onto and strategy-proof if and only if it is dictatorship.*

Proof: This follows directly from Lemmas 10 and 11 and unanimity (implied by onto and strategy-proofness - Proposition 3). ■

Once we have the theorem for $n = 2$ case, we can apply induction on the number of agents. In particular, we prove the following proposition.

Proposition **5** *Let $n \geq 3$. Consider the following statements.*

(a) For all positive integer $k < n$, we have if $f : \mathcal{P}^k \to A$ is onto and strategy-proof, then f is dictatorial.

(b) If $f : \mathcal{P}^n \to A$ is onto and strategy-proof, then f is dictatorial.

Statement (a) implies statement (b).

Proof: Suppose statement (a) holds. Let $f : \mathcal{P}^n \to A$ be an onto and strategy-proof social choice function. We construct another social choice function $g : \mathcal{P}^{n-1} \to A$ from f by merging agents 1 and 2 as one agent. In particular, $g(P_1, P_3, P_4, \ldots, P_n) = f(P_1, P_1, P_3, P_4, \ldots, P_n)$ for all preference profiles $(P_1, P_3, P_4, \ldots, P_n)$. So agents 1 and 2 are "coalesced" in social choice function g, and will be referred to as agent 1 in SCF g.

We do the proof in two steps. In the first step, we show that g is onto and strategy-proof. We complete the proof in the second step, i.e., show that f is dictatorship.

STEP 1: It is clear that agents 3 through n cannot manipulate in g (if they can manipulate in g, they can also manipulate in f, which is a contradiction). Consider an arbitrary preference profile of $n-1$ agents $(P_1, P_3, P_4, \ldots, P_n)$. Suppose

$$f(P_1, P_1, P_3, P_4, \ldots, P_n) = g(P_1, P_3, P_4, \ldots, P_n) = a.$$

Consider any arbitrary preference ordering $\bar{P}_1$ of agent 1. Let

$$f(P_1, \bar{P}_1, P_3, P_4, \ldots, P_n) = b.$$

Let

$$f(\bar{P}_1, \bar{P}_1, P_3, P_4, \ldots, P_n) = g(\bar{P}_1, P_3, P_4, \ldots, P_n) = c.$$

If $a = c$, then agent 1 cannot manipulate g at $(P_1, P_3, P_4, \ldots, P_n)$ by $\bar{P}_1$. So, assume $a \neq c$. Suppose $a = b \neq c$. Then, agent 1 cannot manipulate f at $(P_1, \bar{P}_1, P_3, P_4, \ldots, P_n)$ by $\bar{P}_1$. So, $a = bP_1c$. Hence, agent 1 cannot manipulate g at $(P_1, P_3, P_4, \ldots, P_n)$ by $\bar{P}_1$. A similar logic works for the case when $b = c$.

Now, assume that a, b, and c are distinct. Since f is strategy-proof, agent 2 cannot manipulate f at $(P_1, P_1, P_3, P_4, \ldots, P_n)$ by $\bar{P}_1$. So, aP_1b. Similarly, agent 1 cannot manipulate f at $(P_1, \bar{P}_1 P_3, P_4, \ldots, P_n)$ by $\bar{P}_1$. So, bP_1c. By transitivity, aP_1c. Hence, agent 1 cannot manipulate g at $(P_1, P_3, P_4, \ldots, P_n)$ by $\bar{P}_1$. This shows that g is strategy-proof.

It is straightforward to show that if f is onto, then g is onto (follows from unanimity of f).

STEP 2: By our induction hypothesis, g is dictatorship. Suppose j is the dictator. There are two cases to consider.

CASE A: Suppose $j \in \{3, 4, \ldots, n\}$ is the dictator in g. We claim that j is also the dictator in f. Assume for contradiction that there is a preference profile $P \equiv (P_1, P_2, \ldots, P_n)$ such that

$$f(P) = b \text{ and } P_j(1) = a \neq b.$$

Since g is dictatorship, we get

$$f(P_1, P_1, P_3, P_4, \ldots, P_n) = g(P_1, P_3, P_4, \ldots, P_n) = a,$$
$$f(P_2, P_2, P_3, P_4, \ldots, P_n) = g(P_2, P_3, P_4, \ldots, P_n) = a.$$

We get bP_1a, since f is strategy-proof, and agent 1 cannot manipulate f at $(P_1, P_2, P_3, P_4, \ldots, P_n)$ by P_2. Similarly, agent 2 cannot manipulate at $(P_1, P_1, P_3, P_4, \ldots, P_n)$ by P_2. So, aP_1b. This is a contradiction.

CASE B: Suppose $j = 1$ is the dictator in g. In this case, we construct a 2-agent social choice function h as follows: for every preference profile $(P_1, P_2, \ldots, P_n)$, we define

$$h^{P_{-12}}(P_1, P_2) = f(P_1, P_2, \ldots, P_n).$$

Since agent 1 is the dictator in g, $h^{P_{-12}}$ is onto. Moreover, $h^{P_{-12}}$ is strategy-proof: if any of the agents can manipulate in $h^{P_{-12}}$, they can also manipulate in f. By our induction hypothesis, $h^{P_{-12}}$ is dictatorship. But $h^{P_{-12}}$ was defined for every $n-2$ agent profile $P_{-12} \equiv (P_3, P_4, \ldots, P_n)$. We show that the dictator does not change across two $n-2$ agent profiles.

Assume for contradiction that agent 1 is the dictator for profile $(P_3, P_4, \ldots, P_n)$ but agent 2 is the dictator for profile $(\bar{P}_3, \bar{P}_4, \ldots, \bar{P}_n)$. Now, progressively change the preference profile $(P_3, P_4, \ldots, P_n)$ to $(\bar{P}_3, \bar{P}_4, \ldots, \bar{P}_n)$, where in each step, we change the preference of one agent j from P_j to $\bar{P}_j$. Then, there must exist a profile $(\bar{P}_3, \bar{P}_4, \bar{P}_{j-1}, P_j, P_{j+1}, \ldots, P_n)$ where agent 1 dictates and another profile $(\bar{P}_3, \bar{P}_4, \bar{P}_{j-1}, \bar{P}_j, P_{j+1}, \ldots, P_n)$ where agent 2 dictates with $3 \leq j \leq n$. Consider $a, b \in A$ such that aP_jb. Pick P_1 and P_2 such that $P_1(1) = b$ and $P_2(1) = a$ with $a \neq b$. By definition,

$$f(P_1, P_2, \bar{P}_3, \bar{P}_4, \bar{P}_{j-1}, P_j, P_{j+1}, \ldots, P_n) = P_1(1) = b,$$
$$f(P_1, P_2, \bar{P}_3, \bar{P}_4, \bar{P}_{j-1}, \bar{P}_j, P_{j+1}, \ldots, P_n) = P_2(1) = a.$$

This means agent j can manipulate in SCF f at $(P_1, P_2, \bar{P}_3, \bar{P}_4, \bar{P}_{j-1}, P_j, P_{j+1}, \ldots, P_n)$ by $\bar{P}_j$. This is a contradiction since f is strategy-proof. This shows that f is also a dictatorship.

This completes the proof of the proposition. ■

The proof of the Gibbard-Satterthwaite theorem follows from Propositions 4 and 5, and from the fact that the proof is trivial for $n = 1$.

Note that the induction step must start at $n = 2$, and not $n = 1$, since the induction argument going from k to $k + 1$ works for $k \geq 2$ only.

6 Single Peaked Domain of Preferences

We will now study an important restricted domain where the Gibbard-Satterthwaite theorem does not apply. This is the domain where preferences of agents exhibit *single-peaked* property. To understand single-peaked preferences, consider an election with several candidates (possibly infinite). Candidates are ordered on a line so that candidate on left is the most leftist, and candidates become more and more right wing as we move to right. Now, it is natural to assume that every voter has an ideal political position. As one moves away from his ideal political position, either to left or to right, his preference decreases.

To be more precise, let $\{a, b, c\}$ be three candidates, with a to extreme left, b in the center, and c to extreme right. Now, suppose a voter's ideal position is b. Then, he likes b over a and b over c, but can have any preference over a and c. On the other hand, suppose a voter likes a the most. Then, the only possible ordering is a better than b better than c. Hence, when a is on top, c cannot be better than b. This restriction shows that this is a domain which is restricted, and the Gibbard-Satterthwaite theorem does not apply.

We now formally define the single-peaked preferences. Let $N = \{1, \ldots, n\}$ be the set of agents. Let A be a set of alternatives. All the results we state will hold for A finite or infinite. Consider the linear order $<$ induced by less than relation on $[0, 1]$. We map the set of alternatives in A onto $[0, 1]$ (this is only for convenience - we can pick any linear order $\succ$ on A and work with that). A preference ordering P_i (a linear order over the set of alternatives A) of agent i is **single peaked** with respect to $<$ if there exists an alternative $p_i \in A$, called the **peak**, such that

- for all $b, c \in A$ with $b < c < p_i$ we have $p_i P_i c$ and $c P_i b$, and
- for all $b, c \in A$ with $p_i > b > c$ we have $p_i P_i b$ and $b P_i c$.

So, preferences away from peak decreases, but no restriction is put for comparing alternatives when one of them is on the left to the peak, but the other one is on the right of the peak. We show some preference relations in Figure 1, and color the single-peaked ones in green.

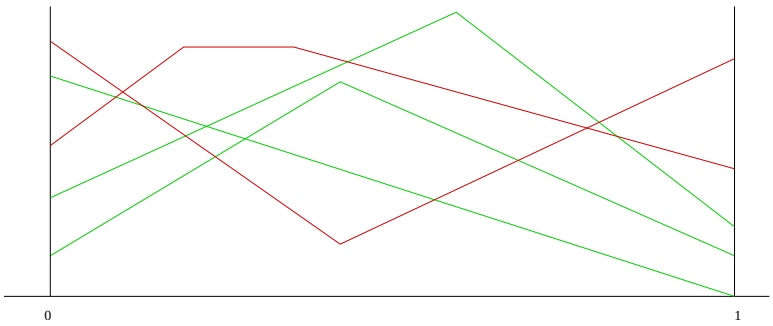

Figure 1: Examples of single-peaked preferences

Since we fix the order $<$ on $[0,1]$, we will just say single-peaked preferences instead of single-peaked with respect to $<$. Note that if we have some finite set of alternatives, then single-peaked preference first maps each of the alternative onto $[0,1]$, which induces the ordering $<$, and then single-peaked preferences can be defined in the usual way. We illustrate the idea with four alternatives $A = \{a, b, c, d\}$. Let us put the alternatives on $[0,1]$ such that $a < b < c < d$. With respect to $<$, we give the permissible single peaked preferences in Table 11. There are sixteen more preference orderings that are not permissible here. For example, $bP_idP_iaP_ic$ is not permissible since c, d are on the same side of peak, and in that case c is nearer to b than d is to b. So, cP_id, which is not the case here.

a	b	b	b	c	c	c	d
b	a	c	c	d	b	b	c
c	c	d	a	b	a	d	b
d	d	a	d	a	d	a	a

Table 11: Single-peaked preferences

We now give some more examples of single-peaked preferences.

- An amount of public good (number of buses in the city) needs to be decided. Every agent has an optimal level of public good that needs to be consumed. The preferences decrease as the difference of the decided amount and optimal level increases.

- If we are locating a facility along a line then agents can have single-peaked preferences. For every agent, there is an optimal location along a line where he wants the facility, and the preference decreases as the distance from the optimal location increases in one direction.

- Something as trivial as setting the room temperature of a building by a group of agents exhibit single-peaked preferences. Everyone has an ideal temperature, and as the difference from the ideal temperature increases, the preference decreases.

Let $\mathcal{S}$ be the set of all single-peaked preferences. A social choice function f is a mapping $f : \mathcal{S}^n \to A$. An SCF f is manipulable by i at (P_i, P_{-i}) if there exists another single-peaked preference $\hat{P}_i$ such that $f(\hat{P}_i, P_{-i}) P_i f(P_i, P_{-i})$. An SCF is strategy-proof if it is not manipulable by any agent at any preference profile.

6.1 Possibility Examples in Single-Peaked Domains

We start with an example to illustrate that many non-dictatorial social choice functions are strategy-proof in this setting. For any single-peaked preference ordering P_i, we let $P_i(1)$ to denote its peak. Now, consider the following SCF f: for every preference profile P, $f(P)$ is the minimal element with respect to $<$ among $\{P_1(1), P_2(1), \ldots, P_n(1)\}$. First, this is not a dictatorship since at every profile, a different agent can have its peak to the left. Second, it is strategy-proof. To see this, note that the agent whose peak coincides with the chosen alternative has no incentive to deviate. If some other agent deviates, then the only way to change the outcome is to place his peak to the left of chosen outcome. But that will lead to an outcome which is even more left to his peak, which he prefers less than the current outcome. Hence, no manipulation is possible.

One can generalize this further. Pick an integer $k \in \{1, \ldots, n\}$. In every preference profile, the SCF picks the k-th lowest peak. Formally, $f(P_1, \ldots, P_n)$ chooses among $\{P_1(1), \ldots, P_n(1)\}$ the k-th lowest alternative according to $<$. To understand why this SCF is manipulable, note that those agents whose peak coincides with the k-th lowest peak have no incentive to manipulate. Consider an agent i, which lies to the left of k-th lowest peak. The only way he can change the outcome is to move to the right of the k-th lowest peak. In that case, an outcome which is even farther away from his peak will be chosen. According to single-peaked preferences, he prefers this less. A symmetric argument applies to the agents who are on to the right of k-th lowest peak.

6.2 Median Voter Result

We now define the notion of a *median voter*. Consider any sequence of points $B \equiv (x_1, \ldots, x_{2k+1})$ such that for all $j \in \{1, \ldots, 2k+1\}$, we have $x_j \in A$. Now $b \in B$ is the median if $|\{x \in B : x < b \text{ or } x = b\}| \geq k+1$ and $|\{x \in B : x > b \text{ or } x = b\}| \geq k+1$. The median

of a sequence of points B will be denoted as $med(B)$. Also, for any profile $(P_1, \ldots, P_n)$, we denote the sequence of peaks as $peak(P) \equiv (P_1(1), \ldots, P_n(1))$.

DEFINITION **6** *A social choice function $f : \mathcal{S}^n \to A$ is a* **median voter** *social choice function if there exists $B = (y_1, \ldots, y_{n-1})$ such that $f(P) = med(B, peak(P))$ for all preference profiles P. The alternatives in B are called the peaks of* **phantom voters**.

Note that by adding $(n-1)$ phantom voters, we have $(2n-1)$ (odd) peaks, and a median exists. We give an example to illustrate the ideas. Figure 2 shows the peak of 4 agents (in green). Then, we add 3 phantom voters, whose peaks are shown (in brown). The median voter SCF chooses the median of this set, which is shown to be the peak of the 3rd phantom voter in Figure 2.

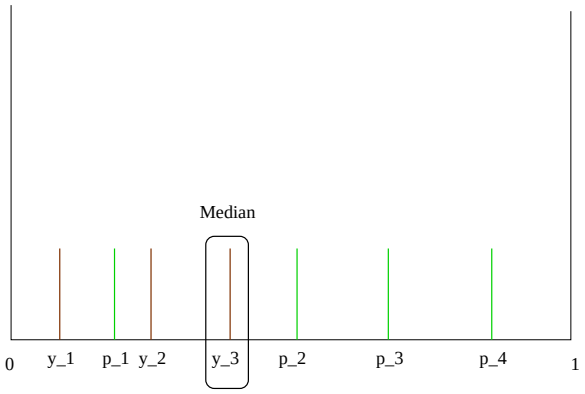

Figure 2: Phantom voters and the median voter

Of course, the median voter SCF is a class of SCFs. A median voter SCF must specify the peaks of the phantom voters (it cannot change across profiles). We can simulate the k-th lowest peak social choice function that we described earlier by placing the phantom voters suitably. In particular, place peaks of $(n-k)$ phantom voters at 0 (or lowest alternative according to $<$) and the remaining $(k-1)$ peaks of phantom voters at 1 (or highest alternative according to $<$). It is clear that the median of this set lies at the kth lowest peak of agents.

PROPOSITION **6** *Every median voter social choice function is strategy-proof.*

Proof: Consider any profile of single-peaked preferences $P = (P_1, \ldots, P_n)$. Let f be a median voter SCF, and $f(P) = a$. Consider agent i. Agent i has no incentive to manipulate if $P_i(1) = a$. Suppose agent i's peak is to the left of a. The only way he can change the outcome is by changing the median, which he can only do by changing his peak to the right

of a. But that will shift the median to the right of a which he does not prefer to a. So, he cannot manipulate. A symmetric argument applies if i's peak is to the right of a. ■

One may wonder if one introduces an arbitrary number of phantom voters. Will the corresponding social choice function be still strategy-proof? We assume that whenever there are even number of agents (including the phantom voters), we pick the minimum of two medians. Along the lines of proof of Proposition 6, one can show that even this social choice function is strategy-proof.

We then ask what is unique about the median voter social choice function (where we take $n-1$ phantom voters). We next intend to characterize the median voter social choice function.

6.3 Properties of Social Choice Functions

We first define some desirable properties of a social choice function. Most of these properties have already been discussed earlier for the Gibbard-Satterthwaite result.

Definition **7** *A social choice function $f : \mathcal{S}^n \to A$ is* **onto** *if for every $a \in A$, there exists a profile $P \in \mathcal{S}^n$ such that $f(P) = a$.*

Onto rules out constant social choice functions.

Definition **8** *A social choice function $f : \mathcal{S}^n \to A$ is* **unanimous** *if for every profile P with $P_1(1) = P_2(1) = \ldots = P_n(1) = a$ we have $f(P) = a$.*

Definition **9** *A social choice function $f : \mathcal{S}^n \to A$ is* **efficient** *if for every profile of preferences P and every $b \in A$, if there exists $a \neq b$ such that aP_ib for all $i \in N$, then $f(P) \neq b$.*

Denote by $[a, b]$, the set of all alternatives which lie between a and b (including a and b) according to $<$.

Lemma **6** *For every preference profile P, let p^{min} and p^{max} denote the smallest and largest peak (according to $<$) respectively in P. A social choice function $f : \mathcal{S}^n \to A$ is efficient if and only if for every profile P, $f(P) \in [p^{min}, p^{max}]$.*

Proof: Suppose f is efficient. Fix a preference profile P. If $f(P) < p^{min}$, then choosing p^{min} is better for all agents. Similarly, if $f(P) > p^{max}$, then choosing p^{max} is better for all

agents. Hence, by efficiency, $f(P) \in [p^{min}, p^{max}]$. For the converse, if $f(P) \in [p^{min}, p^{max}]$, then any alternative other than $f(P)$ will move it away from either p^{min} or p^{max}. Hence, f is efficient. ■

Median voting with arbitrary number of phantom voters may be inefficient. Consider the median voting with $(3n-1)$ phantom voters. Suppose we put all the phantoms at zero, and consider the instance where the peaks of the agents are arbitrarily close to 1. The outcome in this case is zero. But choosing one of the agents' peaks make every agent better off.

DEFINITION **10** *A social choice function $f : \mathcal{S}^n \to A$ is* **monotone** *if for any two profiles P and P' with $f(P) = a$ and for all $b \neq a$, $aP_i'b$ if aP_ib we have $f(P') = a$.*

Like in the unrestricted domain, strategy-proofness implies monotonicity.

LEMMA **7** *If a social choice function $f : \mathcal{S}^n \to A$ is strategy-proof, then it is monotone.*

Proof: The proof is exactly similar to the necessary part of Theorem 1. We take two preference profiles $P, P' \in \mathcal{S}^n$ such that $f(P) = a$ and $aP_i'b$ if aP_ib for all $b \neq a$. As in the proof of Theorem 1, we can consider P and P' to be different in agent j's preference ordering *only* (else, we construct a series of preference profiles each different from the previous one by just one agent's preference). Assume for contradiction $f(P') = b \neq a$.

If bP_ja, then agent j can manipulate at P by P'. Hence, aP_jb. But that means $aP_j'b$. In such a case, agent j will manipulate at P' by P. This is a contradiction. ■

Like in the unrestricted domain, some of these properties are equivalent in the presence of strategy-proofness.

PROPOSITION **7** *Suppose $f : \mathcal{S}^n \to A$ is a strategy-proof social choice function. Then, f is onto if and only if it is unanimous if and only if it is efficient.*

Proof: Consider a strategy-proof social choice function $f : \mathcal{S}^n \to A$. We do the proof in three steps.

UNANIMITY IMPLIES ONTO. Fix an alternative $a \in A$. Consider a single peaked preference profile P where every agent has his peak at a. By unanimity, $f(P) = a$.

ONTO IMPLIES EFFICIENCY. Consider a preference profile P such that $f(P) = b$ but there exists a $a \neq b$ such that aP_ib for all $i \in N$. By single-peakedness, there is an alternative c

which is a *neighbor* of b in $<$ and cP_ib for all $i \in N$. [7] Since f is onto, there exists a profile P' such that $f(P') = c$. Consider another preference profile P'' such that the peaks of every agent is c, but the second ranked alternative is b - such a preference is possible in a single-peaked domain. By Lemma 7, f is monotone. By monotonicity, we get $f(P'') = f(P') = c$ and $f(P'') = f(P) = b$. This is a contradiction.

Efficiency implies unanimity. In any profile, where peaks are the same, efficiency will imply that the peak is chosen. ■

We now define a new property which will be crucial. For this, we need some definitions. A permutation of agents is denoted by a bijective mapping $\sigma : N \to N$. We apply a permutation σ to a profile P to construct another profile as follows: the preference ordering of agent i goes to agent $\sigma(i)$ in the new preference profile. We denote this new preference profile as P^σ.

Table 12 shows a pair of profiles, one of which is obtained by permuting the other. We consider $N = \{1, 2, 3\}$ and σ as $\sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1$.

P_1	P_2	P_3	P_1^σ	P_2^σ	P_3^σ
a	b	b	b	a	b
b	a	c	c	b	a
c	c	a	a	c	c

Table 12: Example of permuted preferences

Definition **11** *A social choice function $f : \mathcal{S}^n \to A$ is* **anonymous** *if for every profile P and every permutation σ such that $P^\sigma \in \mathcal{S}^n$, we have $f(P^\sigma) = f(P)$.*

Anonymous social choice functions require that the identity of agents are not important, and does not discriminate agents on that basis. Dictatorial social choice functions are not anonymous (it favors the dictator). Any social choice function which *ignores* the preferences of some agent is not anonymous.

6.4 Characterization Result

We show now that the only strategy-proof social choice function which is onto and anonymous is the median voter.

[7] Two alternatives x and y are neighbors in $<$ if $x < y$ and there is no alternative z such that $x < z < y$ or $x > y$ and there is no alternative z such that $x > z > y$.

THEOREM **3** *A strategy-proof social choice function is onto and anonymous if and only if it is the median voter social choice function.*

We discuss the necessity of all the properties. First, a dictatorial social choice function is onto and strategy-proof. So, anonymity is crucial in the characterization. Second, putting arbitrarily large number of phantoms at the lowest alternative according to $<$, and then taking the median is anonymous and strategy-proof, but it is not onto - it always selects the lowest alternative according to $<$. Hence, all the conditions are necessary in the result. We now give the proof.

Proof: It is clear that the median voter social choice function is strategy-proof (Proposition 6), onto (all the peaks in one alternative will mean that is the median), and anonymous (it does not distinguish between agents). We now show the converse.

Suppose $f : \mathcal{S}^n \to A$ is a strategy-proof, onto, and anonymous social choice function. The following two preference orderings are of importance for the proof:

- P_i^0: this is the unique preference ordering where the peak of agent i is at the lowest alternative according to $<$.
- P_i^1: this is the unique preference ordering where the peak of agent i is at the highest alternative according to $<$.

FINDING THE PHANTOMS. For any $j \in \{1, \ldots, n-1\}$, define y_j as follows:

$$y_j = f(P_1^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1).$$

So, y_j is the chosen alternative, when $(n-j)$ agents have their peak at the lowest alternative and the remaining j agents have their peak at the highest alternative. Notice that which of the j agents have their peaks at the highest alternative does not matter due to anonymity of f.

Further, we show that $y_j = y_{j+1}$ or $y_j < y_{j+1}$ for any $j \in \{1, \ldots, n-1\}$. To see this consider two profiles $P = (P_1^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1)$ and $P' = (P_1^0, \ldots, P_{n-j-1}^0, P_{n-j}^1, \ldots, P_n^1)$. Only preference ordering of agent $k \equiv n-j$ is changing from P to P'. Note that $f(P) = y_j$ and $f(P') = y_{j+1}$. Since f is strategy-proof, $y_j P_k^0 y_{j+1}$. But the peak of agent k in P_k^0 is at the lowest alternative according to $<$. So, either $y_j = y_{j+1}$ or $y_j < y_{j+1}$.

Now, we consider a preference profile $P = (P_1, \ldots, P_n)$, where $P_i(1) = p_i$. We wish to show that

$$f(P) = med(p_1, \ldots, p_n, y_1, \ldots, y_{n-1}).$$

Assume without loss of generality (due to anonymity) that $p_1 \leq p_2 \leq \ldots \leq p_n$. We let $a = med(p_1, \ldots, p_n, y_1, \ldots, y_{n-1})$, and consider two possible cases.

MEDIAN IS PHANTOM PEAK. Suppose $a = y_j$ for some $j \in \{1, \ldots, n-1\}$. Since we are taking median of $2n-1$ points, and exactly $j-1$ phantom voters are to left of a and $n-j-1$ phantom voters to right (monotonicity of y_js), we must have $n-j$ agent peaks to the left and the remaining to the right. This means, $p_{n-j} \leq a = y_j \leq p_{n-j+1}$ due to monotonicity of agent peaks.

Now, we consider two preference profiles where preference ordering of agent 1 is different: $P' = (P_1^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1)$ and $P'' = (P_1, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1)$. Note that $f(P') = y_j$. Let $f(P'') = b$. Since f is strategy-proof, $y_j P_1^0 b$ or $y_j \leq b$. Also, strategy-proofness implies that $bP_1 y_j$. But $p_1 \leq y_j$. This implies that $b \leq y_j$. Hence, $b = y_j$.

Now, we consider another preference profile $P''' = (P_1, P_2, P_3^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1)$, and repeat the previous argument for P'' and P'''. Repeating this way, we get $y_j = f(P_1, \ldots, P_{n-j}, P_{n-j+1}^1, \ldots, P_n^1)$.

Now, let $P' = (P_1, \ldots, P_{n-j}, P_{n-j+1}^1, \ldots, P_n^1)$ and $P'' = (P_1, \ldots, P_{n-j}, P_{n-j+1}^1, \ldots, P_n)$. By assumption, $y_j = f(P')$. Let $f(P'') = b$. Since f is strategy-proof, $y_j P_n^1 b$, which implies that $y_j \geq b$. Again, applying f to be strategy-proof, we get that $bP_n y_j$. But, by assumption, $y_j \leq p_n$. This implies that $y_j \leq b$. This shows that $b = y_j$. Repeating this argument for all agents greater than j, we conclude that $f(P) = y_j$.

MEDIAN IS AGENT PEAK. We do this part of the proof for two agents. Suppose $N = \{1, 2\}$. We first show a claim that shows *peaks-only* property of a strategy-proof and efficient social choice function.

CLAIM **1** *Suppose $N = \{1, 2\}$ and f is a strategy-proof and efficient social choice function. Let P and P' be two profiles such that $P_i(1) = P_i'(1)$ for all $i \in N$. Then, $f(P) = f(P')$.*

Proof: Consider preference profiles P and P' such that $P_1(1) = P_1'(1) = a$ and $P_2(1) = P_2'(1) = b$. Consider the preference profile (P_1', P_2), and let $f(P) = x$ but $f(P_1', P_2) = y$. By strategy-proofness, $xP_1 y$ and $yP_1' x$. This implies, if x and y belong to the same side of a, then $x = y$. Then, the only other possibility is x and y belong to the different sides of a. We will argue that this is not possible. Assume without loss of generality $x < a < y$. Suppose, without loss of generality, $b < a$. Then, by efficiency (Lemma 6) at profile P_1', P_2, we must have $y \in [b, a]$. This is a contradiction since $a < y$. Hence, it is not possible that x and y belong to the different sides of a. Thus, $x = y$ or $f(P_1, P_2) = f(P_1', P_2)$.

Now, we can replicate this argument by going from (P_1', P_2) to (P_1', P_2'). This will show that $f(P_1', P_2') = x = f(P_1, P_2)$. ■

Now, consider a profile (P_1, P_2) such that $P_1(1) = a$, $P_2(1) = b$, and y_1 is the phantom peak. By our assumption, the median of (a, b, y_1) is an agent peak. Suppose that peak is a. Let $f(P_1, P_2) = c$. By efficiency, $c \in [a, b]$. Assume for contradiction that $c > a$. Consider another single-peaked preference ordering P_1' for agent 1 such that $P_1'(1) = a = P_1(1)$ and $y_1 P_1' c$ - this is possible since c and y_1 are on different sides of a. By Claim 1, $f(P_1', P_2) = c$. Now, consider the preference profile (P_1^0, P_2). By definition, the median of $P_1^0(1)$, $P_2(1) = b$, and y_1 is y_1. By the earlier case of the proof, $f(P_1^0, P_2) = y_1$. Since $y_1 P_1' c$, agent 1 will manipulate at (P_1', P_2) via P_1^0. This is a contradiction since f is strategy-proof. ■

The peaks of the phantom voters reflect the degree of compromise the social choice function has when agents have *extreme* preferences. If j agents have the highest alternative as the peak, and the remaining $n - j$ agents have the lowest alternative as the peak, then which alternative is chosen? A true median will choose the peak which has more agents, but the median voter social choice function may do something intermediate.

7 Private Good Allocation

The private good allocation problem allocates a set of private goods to agents. In the abstract problem formulation, let A be a set of private goods. Each agent has a preference ordering over the set of private goods A. Denote by $\succ_i$ the preference of agent i over A. A social choice function in this case chooses a private good in A for every agent. In other words, an outcome consists of an element in A^n, where n is the number of agents. Of course, not every element in A^n may be feasible - for instance, when we are allocating indivisible objects, we cannot allocate the same object to more than one agent. So, the feasible set of outcomes will be a subset $F \subseteq A^n$, where F will vary from problem to problem.

Notice that agents have preference over A, but the set of possible outcomes is F. This preference induces a preference over F in a natural way because we will assume *no externalities*, i.e., every agent only cares about his allocation and not what others get. However, we get many indifferences. For instance, let $a, b \in F$ be two possible outcomes, where the allocation of an agent is the same in a and b. By the no externality assumption, whatever the preference of this agent be, he will be indifferent between a and b. For this particular reason, the private good allocation problems have inherent domain restrictions and the results for the public good problems (e.g., the Gibbard-Satterthwaite theorem) cannot be directly

applied.

7.1 Allocating a Divisible Commodity

We begin by discussing a private good problem, which is an extension of the single-peaked domain problem we discussed earlier. The single-peaked domain can be considered to be an instance of a domain where a public good is being allocated. Consider a problem where an infinitely divisible private good has to be allocated among n agents.

A classical application of this problem is time sharing. Suppose a task needs to be completed. Without loss of generality, assume that the task takes one unit of time. There are n agents. The problems is to assign each agent i with a share of time $s_i \in [0,1]$ such that the sum of times assigned to all the agents is one, i.e., the task is completed or $\sum_j s_j = 1$.

Each agent has single peaked preference over his share of time. This is usually the case if there is some trade off over cost and value for the time share. For instance, suppose working for $\theta_i \in [0,1]$ unit gives agent i wage of $\theta_i w$, where w is per unit time wage and he incurs a cost of $\kappa\theta_i^2$. Then, his net utility is $w\theta_i - \kappa\theta_i^2$, whose peak is at $\theta_i^* = \frac{w}{2\kappa}$ and preferences are single peaked from the peak.

However, as we show next, this does not translate to a single-peaked preference ordering over the set of alternatives. Hence, earlier results cannot be applied.

Here, every alternative is a vector $s = (s_1, \ldots, s_n)$ such that $\sum_{i\in N} s_i = 1$ and $s_i \geq 0$ for all $i \in N$. So, the set of alternatives is

$$A = \{(s_1, \ldots, s_n) : s_i \geq 0 \ \forall \ i \in N, \ \sum_{i\in N} s_i = 1\}.$$

Suppose agents preferences are not known (but only known to be single-peaked), and agents care only about their own shares. If agents only care about their own shares, then the preferences over A cannot be single-peaked because two alternatives with same share to an agent must be same for that agent. Hence, the earlier results on single-peaked domains do not apply.

However, for various possible shares of agent i, that agent has a preference ordering $\succ_i$ over $[0,1]$ with a peak at $p_i(\succ_i)$, and single-peaked. Denote by $\mathcal{S}$ the set of all single-peaked preferences over $[0,1]$. A social choice function is a mapping $f : \mathcal{S}^n \to A$. The share allocation to agent i at preference profile $\succ$ is denoted by $f_i(\succ) \in [0,1]$.

We first look at the implication of *efficiency* in this setting. Without formally defining it, an allocation is efficient if there does not exist another allocation which makes everyone better off with at least one agent getting strictly better. If $\sum_{i\in N} p_i(\succ_i) = 1$, then efficiency

directs us to allocate $p_i(\succ_i)$ to agents i for all $i \in N$. If $\sum_{i \in N} p_i(\succ_i) > 1$, then for some agent $k \in N$, $f_k(\succ) < p_k(\succ_k)$. In that case, no agent $j \neq k$ must be getting $f_j(\succ) > p_j(\succ_j)$. Because in that case, decreasing agent j's share and increasing agent k's share makes both of them better off. So, in this case, we must have $f_j(\succ) \leq p_j(\succ_j)$ for all $j \in N$. Similarly, if $\sum_{i \in N} p_i(\succ_i) < 1$, we must have $f_j(\succ) \geq p_j(\succ_j)$ for all $j \in N$.

There are many possible social choice functions that one can imagine for this situation. We give some examples first.

1. **Serial Dictatorship.** In this social choice function, agents are ordered using some permutation $\sigma : N \to N$. The first agent in the permutation $\sigma(1)$ chooses his peak amount. The next agent chooses minimum of his peak amount and left over amount and so on. This is a very standard method to allocate private goods. It is fairly clear that such a social choice function is strategy-proof and Pareto efficient (we will discuss this in detail later). However, it heavily favors agents who are earlier in the order σ.

2. **Proportional.** The proportional social choice function looks at the peak of each agent and assigns every agent a fraction of the divisible good in proportion of their peaks. If all the peaks are at zero then, it assigns equal amount to all the agents. Although this social choice function looks *fair*, it can be manipulated (argue why).

We now define a social choice function that is strategy-proof and satisfies some other desirable properties. It is referred to as the **uniform rule** social choice function and denoted as f^u. For any profile $\succ$, we define for every $i \in N$,

$$
\begin{aligned}
f_i^u(\succ) &= p_i(\succ_i) \qquad \text{if } \sum_{i \in N} p_i(\succ_i) = 1 \\
f_i^u(\succ) &= \max(p_i(\succ_i), \mu(\succ)) \qquad \text{if } \sum_{i \in N} p_i(\succ_i) < 1 \\
f_i^u(\succ) &= \min(p_i(\succ_i), \lambda(\succ)) \qquad \text{if } \sum_{i \in N} p_i(\succ_i) > 1,
\end{aligned}
$$

where $\mu(\succ)$ solves $\sum_{i \in N} \max(p_i(\succ_i), \mu(\succ)) = 1$ in the second case and $\lambda(\succ)$ solves $\sum_{i \in N} \min(p_i(\succ_i), \lambda(\succ)) = 1$ in the third case. It can be verified that these quantities have a unique solution.

The uniform rule SCF has a nice interpretation. Every agent has a bucket of 1 unit capacity. There is a mark at $p_i(\succ_i)$ in every bucket i. If the sum of these marks are equal to 1, we fill the buckets with water till their marks. If the sum of these marks are greater than 1, we fill water in the buckets at equal rate, till one of the buckets hits the mark. We stop filling that bucket, but fill the other buckets at equal rate, till we hit another mark, and

so on till the sum of water in the buckets is 1. The water level in the buckets at the end indicate the final allocation.

When sum of the marks is less than 1, we fill the buckets completely and empty them uniformly till the sum is equal to 1. We stop filling a bucket once we hit the mark. Then, we continue emptying the other buckets at a uniform rate, and so on till the sum of water in the buckets is 1. The water level in the buckets at the end indicate the final allocation.

PROPOSITION 8 *The uniform rule social choice function is efficient, anonymous, and strategy-proof.*

Proof: The uniform rule social choice function is anonymous since only the peaks of agents matter but not the identity of the "owners" of peaks. Consider a preference profile $\succ$. Efficiency is equivalent to verifying the following two cases.

- When $\sum_{i\in N} p_i(\succ_i) < 1$, then $f_i^u(\succ) \geq p_i(\succ_i)$ for all $i \in N$. This is true because in this case, we empty the buckets uniformly, and stop as soon as a bucket hits the peak.
- When $\sum_{i\in N} p_i(\succ_i) > 1$, then $f_i^u(\succ) \leq p_i(\succ_i)$ for all $i \in N$. This is true because in this case, we fill the buckets uniformly, and stop as soon as a bucket hits the peak.
- When $\sum_{i\in N} p_i(\succ_i) = 1$, then $f_i^u(\succ) = p_i(\succ_i)$ for all $i \in N$. This is true by definition of f^u.

To verify strategy-proofness, consider agent j at a preference profile $\succ$. We consider three cases separately.

- If $\sum_{i\in N} p_i(\succ_i) = 1$, then $f_j^u(\succ) = p_j(\succ_j)$. Hence, agent j has no incentive to manipulate.
- If $\sum_{i\in N} p_i(\succ_i) < 1$, then $f_j^u(\succ) \geq p_j(\succ_j)$. He will like to manipulate if $f_j^u(\succ) > p_j(\succ_j)$. Since f^u only depends on the peaks of agents, the only way to manipulate is to change the peak. Suppose agent j reports $\succ_j'$ with peak $p_j(\succ_j')$. If $p_j(\succ_j') \leq f_j^u(\succ)$, then we will still have $\sum_{i\neq j} p_i(\succ_i) + p_j(\succ_j') \leq \sum_{i\neq j} f_i^u(\succ_j, \succ_{-j}) + f_j^u(\succ_j, \succ_{-j}) = 1$. Hence, we will again empty the buckets. Since $p_j(\succ_j') \leq f_j^u(\succ)$, the outcome will not change.

  If $p_j(\succ_j') > f_j^u(\succ)$, the share of j will only increase, which he prefers less. To see why j's share will increase we consider two cases.

  - $\sum_{i\neq j} p_i(\succ_i) + p_j(\succ_j') < 1$. This implies that we will again empty the buckets. By efficiency, $f_j^u(\succ_j', \succ_{-j}) \geq p_j(\succ_j') > f_j^u(\succ) \geq p_j(\succ_j)$. Hence, agent j does not like this outcome compared to $f_j^u(\succ)$.

– $\sum_{i \neq j} p_i(\succ_i) + p_j(\succ'_j) > 1$. In this case, we fill the buckets. If we hit the peak of agent j, then clearly $f^u_j(\succ'_j, \succ_{-j}) = p_j(\succ'_j) > f^u_j(\succ) \geq p_j(\succ_j)$. So, agent j does not like $f^u_j(\succ'_j, \succ_{-j})$ compared to $f^u_j(\succ)$.

If we do not hit the peak, then it is the highest share amongst all agents in profile $(\succ'_j, \succ_{-j})$. More specifically, $1 = \sum_{i \neq j} f^u_i(\succ'_j, \succ_{-j}) + f^u_j(\succ'_j, \succ_{-j}) \leq n f^u_j(\succ'_j, \succ_{-j}$). On the other hand, since $f^u_j(\succ) > p_j(\succ_j)$, it is the lowest share amongst all agents in profile $(\succ)$. More specifically, $1 = \sum_{i \neq j} f^u_i(\succ) + f^u_j(\succ) \geq n f^u_j(\succ)$. Hence, we again get $f^u_j(\succ'_j, \succ_{-j}) \geq f^u_j(\succ)$. As a result, $f^u_j(\succ'_j, \succ_{-j}) > f^u_j(\succ) \geq p_j(\succ_j)$, and again, agent j does not like $f^u_j(\succ'_j, \succ_{-j})$ compared to $f^u_j(\succ)$

- If $\sum_{i \in N} p_i(\succ_i) > 1$, then $f^u_i(\succ) \leq p_i(\succ_i)$ for all $i \in N$. Using a symmetric argument as the previous case, we can show that no agent j can manipulate.

■

The converse of Proposition 8 is known to be true also. In particular, the following theorem is true, whose proof is skipped.

THEOREM **4** *A social choice function is strategy-proof, efficient, and anonymous if and only if it is the uniform rule.*

8 One Sided Matching - Object Allocation Mechanisms

In this section, we look at an important model where the GS theorem does not hold. There is a finite set of objects $M = \{a_1, \ldots, a_m\}$ and a finite set of agents $N = \{1, \ldots, n\}$. We assume that $m \geq n$. The objects can be houses, jobs, projects, positions, candidates or students etc. Each agent has a linear order over the set of objects, i.e., a complete, transitive, and anti-symmetric binary relation. In this model, this ordering represents the preference of agents, and is the private information of agents. The preference ordering of agent i will be denoted as $\succ_i$. A profile of preferences will be denoted as $\succ \equiv (\succ_1, \ldots, \succ_n)$. The set of all preference orderings over M will be denoted as $\mathcal{M}$. The top element amongst a set of objects $S \subseteq M$ according to ordering $\succ_i$ is denoted as $\succ_i (1, S)$, and the k-th ranked object by $\succ_i (k, S)$.

The main departure of this model is that agents do not have direct preference over alternatives. We need to extract their preference over alternatives from their preference over objects. What are the alternatives? An alternative is a *feasible matching*, i.e., an injective mapping from N to M. The set of alternatives will be denoted as A, and this is the set of

all injective mappings from N to M. For a given alternative $a \in A$, if $a(i) = j \in M$, then we say that agent i is assigned object j (in a).

Consider two alternatives a and b. Suppose agent 1 is is assigned the same object in both a and b (this is possible if there are at least three objects). Then, it is reasonable to assume that agent 1 will **always** be indifferent between a and b. Hence, for any preference ordering of agent 1, aP_1b and bP_1a are not *permissible*. This restriction implies that the domain of preference orderings over alternatives is not the unrestricted domain, which was the case in the GS theorem. Because of this reason, we cannot apply the GS theorem. Indeed, we will show that non-dictatorial social choice functions are strategy-proof in these settings.

A social choice function f is a mapping $f : \mathcal{M}^n \to A$. We now define a **fixed priority (serial dictatorship)** mechanism. We call this a mechanism but not a social choice function since it is not a direct revelation mechanism. A **priority** is a bijective mapping $\sigma : N \to N$, i.e., an ordering over the set of agents. The fixed priority mechanism is defined inductively. Fix a preference profile $\succ$. We now construct an alternative a as follows:

$$
\begin{aligned}
a(\sigma(1)) &=\succ_{\sigma(1)} (1, N) \\
a(\sigma(2)) &=\succ_{\sigma(2)} (1, N \setminus \{a(\sigma(1))\}) \\
a(\sigma(3)) &=\succ_{\sigma(3)} (1, N \setminus \{a(\sigma(1)), a(\sigma(2))\}) \\
&\cdots\cdots \\
a(\sigma(i)) &=\succ_{\sigma(i)} (1, N \setminus \{a(\sigma(1)), \ldots, a(\sigma(i-1))\} \\
&\cdots\cdots \\
a(\sigma(n)) &=\succ_{\sigma(n)} (1, N \setminus \{a(\sigma(1)), \ldots, a(\sigma(n-1))\}.
\end{aligned}
$$

Now, the fixed priority mechanism (and the underlying SCF) assigns $f^\sigma(\succ) = a$.

Let us consider an example. We start with an example. The ordering over houses $\{a_1, a_2, \ldots, a_6\}$ of agents $\{1, 2, \ldots, 6\}$ is shown in Table 13. Fix a priority σ as follows:

$\succ_1$	$\succ_2$	$\succ_3$	$\succ_4$	$\succ_5$	$\succ_6$
a_3	a_3	a_1	a_2	a_2	a_1
a_1	a_2	a_4	a_1	a_1	a_3
a_2	a_1	a_3	a_5	a_6	a_2
a_4	a_5	a_2	a_4	a_4	a_4
a_5	a_4	a_6	a_3	a_5	a_6
a_6	a_6	a_5	a_6	a_3	a_5

Table 13: An example for housing model

$\sigma(i) = i$ for all $i \in N$. According to this priority, the fixed priority mechanism will let agent 1 choose his best object first, which is a_3. Next, agent 2 chooses his best object among remaining objects, which is a_2. Next, agent 3 gets his best object among remaining objects $\{a_1, a_4, a_5, a_6\}$, which is a_1. Next, agent 4 gets his object among remaining objects $\{a_4, a_5, a_6\}$, which is a_5. Next, agent 5 gets his best object among remaining objects $\{a_4, a_6\}$, which is a_6. So, agent 6 gets a_4.

Note that a fixed priority mechanism is a generalization of dictatorship. We show below (quite obvious) that a fixed priority mechanism is strategy-proof. Moreover, it is efficient in the following sense.

DEFINITION **12** *A social choice function f is* **efficient** *(in the house allocation model) if for all preference profiles $\succ$ and all matchings a, if there exists another matching $a' \neq a$ such that either $a'(i) \succ_i a(i)$ or $a'(i) = a(i)$ for all $i \in N$, then $f(\succ) \neq a$.*

PROPOSITION **9** *Every fixed priority social choice function (mechanism) is strategy-proof and efficient.*

A word of caution here about strategy-proof notion of the fixed priority social choice function. The fixed priority mechanism is not a direct mechanism. However, using revelation principle, one can think of the associated direct mechanism - agents report their entire ordering, and the mechanism designer executes the fixed priority SCF on this ordering. Whenever, we say that the fixed priority mechanism is strategy-proof, we mean that the underlying direct mechanism is strategy-proof.

Proof: Fix a priority σ, and consider f^σ- the associated fixed priority mechanism. The strategy of any agent i is any ordering over M. Suppose agent i wants to deviate. When agent i is truthful, let M^{-i} be the set of objects allocated to agents who have higher priority than i (agent j has higher priority than agent i if and only if $\sigma(j) < \sigma(i)$). So, by being truthful, agent i get $\succ_i (1, M \setminus M^{-i})$. When agent i deviates, any agent j who has a higher priority than agent i continues to get the same object that he was getting when agent i was truthful. So, agent i gets an object in $M \setminus M^{-i}$. Hence, deviation cannot be better.

To show efficiency, assume for contradiction that f^σ is not efficient. Consider a profile $\succ$ such that $f(\succ) = a$. Let a' be another matching satisfying $a'(i) \succ_i a(i)$ or $a'(i) = a(i)$ for all $i \in N$. Then, consider the first agent j in the priority σ such that $a'(j) \succ_j a(j)$. Since agents before j in priority σ got the objects of matching a', object $a'(j)$ was still available to agent j. This is a contradiction since agent j chose $a(j)$ with $a'(j) \succ_j a(j)$. ■

Note that every fixed priority mechanism f^σ is a dictatorship. In the fixed priority mechanism f^σ corresponding to priority σ, agent $\sigma(1)$ gives his top house, and hence, his top alternative. So, $\sigma(1)$ is a dictator in f^σ. As we have already seen, not every dictatorship is strategy-proof when indifference is allowed in preference orderings. However, Proposition 9 shows that fixed priority mechanism is strategy-proof in the housing allocation model.

One can construct social choice functions which are strategy-proof but not a fixed priority mechanism in this model. We show this by an example. Let $N = \{1, 2, 3\}$ and $M = \{a_1, a_2, a_3\}$. The social choice function we consider is f, and is *almost* a fixed priority SCF. Fix a priority σ as follows: $\sigma(i) = i$ for all $i \in N$. Another priority is σ': $\sigma'(1) = 2, \sigma'(2) = 1, \sigma'(3) = 3$. The SCF f generates the same outcome as f^σ whenever $\succ_2 (1, M) \neq a_1$. If $\succ_2 (1, M) = a_1$, then it generates the same outcome as $f^{\sigma'}$. To see that this is strategy-proof, it is clear that agents 1 and 3 cannot manipulate since they cannot change the priority. Agent 2 can change the priority. But, can he manipulate? If his top ranked house is a_1, he gets it, and he cannot manipulate. If his top ranked house is $\in \{a_2, a_3\}$, then he cannot manipulate without changing the priority. If he does change the priority, then he gets a_1. But being truthful, either he gets his top ranked house or second ranked house. So, he gets a house which is either a_1 or some house which he likes more than a_1. Hence, he cannot manipulate.

8.1 Top Trading Cycle Mechanism with Fixed Endowments

The top trading cycle mechanism (TTC) with fixed endowment is a class of general mechanisms which are strategy-proof, and has some nice properties. We will study them in detail here.

We assume here $m = n$ for simplicity. In the next subsection, we show how to relax this assumption. To explain the mechanism, we start with the example in Table 13. In the first step of the TTC mechanism, agents are endowed with a house each. Suppose the *fixed endowment* for this example is a^*: $a^*(1) = a_1, a^*(2) = a_3, a^*(3) = a_2, a^*(4) = a_4, a^*(5) = a_5, a^*(6) = a_6$.

The TTC mechanism goes in steps. In each step, a set of houses are assigned to a set of agents, and they are excluded from the subsequent steps of the mechanism. Hence, the mechanism maintains a set of "remaining agents" and a set of "remaining houses" in each step.

At every step, a directed graph is constructed. The set of nodes in this directed graph is the same as the set of remaining agents. Initially, the set of remaining agents is N. Then, there is a directed edge from agent i to agent j if and only if agent j is endowed with agent i's top ranked house amongst the remaining houses (initially, all houses are remaining houses).

Formally, if $H \subseteq M$ is the set of remaining houses in any step, then the directed graph in this iteration has an edge from agent i to agent j (i can be j also) if and only if $\succ_i (1, H) = a^*(i)$. Note that such a graph will have exactly one outgoing edge from every node (though possibly many incoming edges to a node). Further, there may be an edge from a node to itself (this will be treated as cycle, and called a loop). It is clear that such a graph will always have a cycle.

Figure 3 shows the directed graph for the first step of the example in Table 13. The only cycle in this graph is a loop involving agent 2. So, agent 2 gets his endowment, which is house a_3. Agent 2 is eliminated from the graph, and house a_3 is eliminated from the problem. Now, the graph for the next step is constructed. Now, every agent points to his top ranked house amongst houses remaining (which is the houses except house a_3). This graph is shown in Figure 4. Here, the only cycle is a loop involving agent 1. So, agent 1 gets his endowment a_1. Agent 1 and house a_1 is eliminated from the problem. Next, the graph for the next step is constructed, which is shown in Figure 5. There is a cycle involving agents 3 and 4. So, agent 3 gets the endowment of agent 4 (a_4) and agent 4 gets the endowment of agent 3 (a_2). These agents and houses are eliminated from the problem, and the next graph is constructed as shown in Figure 6. This graph has a loop involving agent 6. So, agent 6 gets his endowment a_6, and the only remaining house a_5 goes to agent 5.

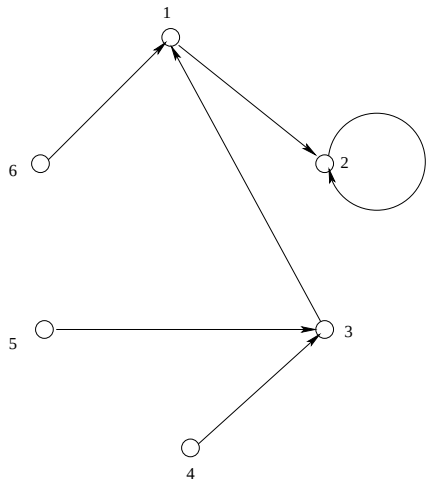

Figure 3: Cycle in Step 1 of the TTC mechanism

We now formally describe the TTC mechanism. Fix an endowment of agents a^*. The mechanism maintains the remaining set of houses M^k and remaining set of agent N^k in every Step k of the mechanism.

- Step 1: Set $M^1 = M$ and $N^1 = N$. Construct a directed graph G^1 with nodes N^1.

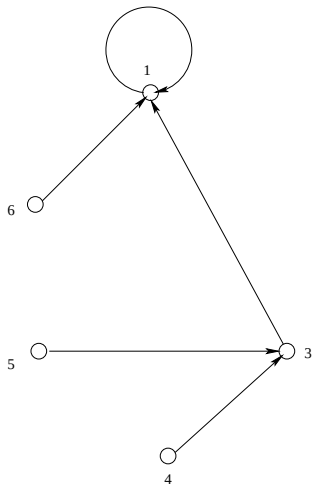

Figure 4: Cycle in Step 2 of the TTC mechanism

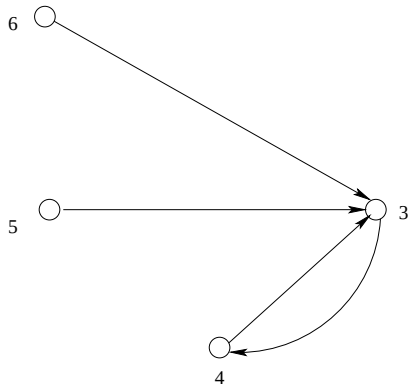

Figure 5: Cycle in Step 3 of the TTC mechanism

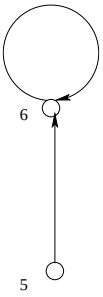

Figure 6: Cycle in Step 4 of the TTC mechanism

There is a directed edge from node (agent) $i \in N^1$ to agent $j \in N^1$ if and only if $\succ_i (1, M^1) = a^*(j)$.

Allocate houses along every cycle of graph G^1. Formally, if $(i^1, i^2, \ldots, i^p, i^1)$ is a cycle

in G^1 then set $a(i^1) = a^*(i^2), a(i^2) = a^*(i^3), \ldots, a(i^{p-1}) = a^*(i^p), a(i^p) = a^*(i^1)$. Let $\hat{N}^1$ be the set of agents allocated in such cycles in G^1, and $\hat{M}^1$ be the set of houses assigned in a to N^1.

Set $N^2 = N^1 \setminus \hat{N}^1$ and $M^2 = M^1 \setminus \hat{M}^1$.

- STEP k: Construct a directed graph G^k with nodes N^k. There is a directed edge from node (agent) $i \in N^k$ to agent $j \in N^k$ if and only if $\succ_i (1, M^k) = a^*(j)$.

  Allocate houses along every cycle of graph G^k. Formally, if $(i^1, i^2, \ldots, i^p, i^1)$ is a cycle in G^k then set $a(i^1) = a^*(i^2), a(i^2) = a^*(i^3), \ldots, a(i^{p-1}) = a^*(i^p), a(i^p) = a^*(i^1)$. Let $\hat{N}^k$ be the set of agents allocated in such cycles in G^k, and $\hat{M}^k$ be the set of houses assigned in a to N^k.

  Set $N^{k+1} = N^k \setminus \hat{N}^k$ and $M^{k+1} = M^k \setminus \hat{M}^k$. If N^{k+1} is empty, STOP, and a is the final matching chosen. Else, repeat.

PROPOSITION **10** *TTC with fixed endowment mechanism is strategy-proof and efficient.*

Proof: Consider agent i who wants to deviate. Suppose agent i is getting assigned in Step k of the TTC mechanism if he is truthful. Given the preferences of the other agents, suppose agent i reports a preference ordering different from his true preference ordering. Let H^{k-1} be the set of houses assigned in Steps 1 through $k-1$ when agent i is truthful. If the deviation of agent i results in no change of his strategy (pointing to the most preferred remaining house) before Step k, then the allocation of houses in H^{k-1} will not change due to his deviation. As a result agent i will get an object from $M \setminus H^{k-1}$. Since agent i gets his most preferred object from $M \setminus H^{k-1}$ if he is truthful, this is not a successful manipulation. Hence, we focus on the case where the deviation of agent i result in a change of his strategy before Step k.

Suppose $r < k$ is the first step in the TTC mechanism where the underlying allocation in that step changes due to this deviation. Notice that the only change in graph G^r in cases where agent i is truthful and where he is deviating is the outgoing edge of agent i. Consider the case when agent i is truthful. In that case since agent i is not allocated in Step r, he is not involved in any cycle in G^r. But there may be sequence of nodes of the nature $(i^1, i^2, \ldots, i^p, i)$, where i^1 has no incoming edge, but edges exist from i^1 to i^2, and i^2 to i^3, and so on. Call such sequence of nodes i-paths. Let P_i be the set of all nodes in all the i-paths - P_i includes i also.

Figure 7 gives an illustration. Here, $P_i = \{i^1, i^2, i^3, i^5, i^6, i\}$.

Note that if agent i's deviation does not lead agent i to point to an agent in P_i, then the allocations in Step r is unchanged because of his deviation. This follows from the fact that

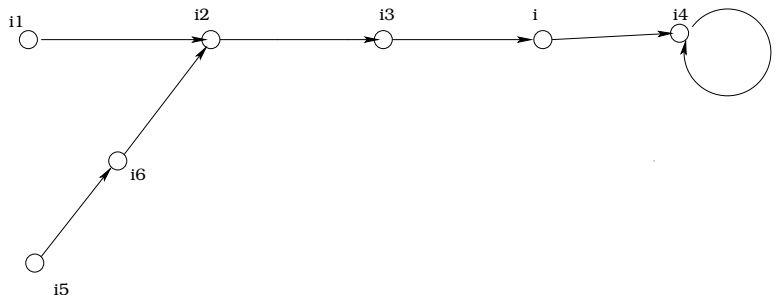

Figure 7: i-Paths in a Step

the only way i can change allocation in Step r is by creating a new cycle involving himself - he cannot break cycles which does not involve him. As a result, the only way to change the allocation in Step r is to deviate by pointing to an agent in P_i. In that case, a subset of agents in P_i which includes i, call them C^r, will form a cycle, and get assigned in Step r. We argue that agents in C^r must be unassigned (i.e., part of the "remaining agents") in Step k when agent i is truthful. To see this, consider any agent $i^1 \in P_i$. By definition, there is a path from i^1 to i - say, $(i^1, i^2, \ldots, i^p, i)$. Since house of i is available till Step k, i^p will continue to point to i. Hence, the house of i^p is available till Step k. As a result, i^{p-1} will continue to point to i^p till Step k, and so on. Hence, the path $(i^1, i^2, \ldots, i^p, i)$ will continue to exist in Step k. This shows that agent in C^r are unassigned in Step k. Hence, the allocation achieved by agent i by his deviation in Step r can also achieved by deviating in Step k. But, we know that if he deviates in Step k, then it is not a successful manipulation. So, the only possibility is that he deviates by pointing to an agent not in P_i, in which case he does not alter the allocation in Step r. As a result, the cycles in subsequent rounds also do not change due to deviations.

Hence, all the agents who were assigned in Steps 1 through $(k-1)$ still get assigned the same houses. By definition, agent k gets his top ranked object amongst $M \setminus H^{k-1}$ if he is truthful. By deviating he will get an object in $M \setminus H^{k-1}$. Hence, deviation cannot be better.

Now, we prove efficiency. Let a be a matching produced by the TTC mechanism for preference profile $\succ$. Assume for contradiction that this matching is not efficient, i.e., there exists a different matching a' such that $a'(i) \succ_i a(i)$ or $a'(i) = a(i)$ for all $i \in N$. Consider the first step of the TTC mechanism where some agent i gets $a(i) \neq a'(i)$. Since all the agents get the same object in a and a' before this step, object $a'(i)$ is available in this step, and since $a'(i) \succ_i a(i)$, agent i cannot have an edge from i to the "owner" of $a(i)$ in this step. This means that agent i cannot be assigned to $a(i)$. This gives a contradiction. ■

Note that a TTC mechanism need not be a dictatorship. To see this, suppose there are

three agents and three houses. Fix an endowment a^* as $a^*(i) = a_i$ for all $i \in \{1, 2, 3\}$. Let us examine the TTC mechanism corresponding to a^*. Consider the profile $(\succ_1, \succ_2, \succ_3)$ such that $\succ_i (1, N) = a_1$ for all $i \in \{1, 2, 3\}$, i.e., every agent has object a_1 as his top ranked object. Clearly, only agent 1 gets one of this top ranked alternatives (matchings) in this profile according to this TTC mechanism. Now, consider the profile $(\succ'_1, \succ'_2, \succ'_3)$ such that $\succ'_i (1, N) = a_2$ for all $i \in \{1, 2, 3\}$, i.e., every agent has object a_2 as his top ranked object. Then, only agent 2 gets one of his top ranked alternatives (matchings) according to this TTC mechanism. Hence, this TTC mechanism is not a dictatorship.

8.2 Stable House Allocation with Existing Tenants

We consider a variant of the house allocation problem. In this model, each agent already has a house that he owns - if an agent i owns house j then he is called the tenant of j. Immediately, one sees that the TTC mechanism can be applied in this setting with initial endowment given by the house-tenant relationship. This is, as we have shown, strategy-proof and efficient (Proposition 10).

We address another concern here, that of *stability*. In this model, agents own resources that are allocated. So, it is natural to impose some sort of stability condition on the mechanism. Otherwise, a group of agents can break away and trade their houses amongst themselves.

Consider the example in Table 13. Let the existing tenants of the houses be given by matching a^*: $a^*(1) = a_1, a^*(2) = a_3, a^*(3) = a_2, a^*(4) = a_4, a^*(5) = a_5, a^*(6) = a_6$. Consider a matching a as follows: $a(i) = a_i$ for all $i \in N$. Now consider the coalition of agents $\{3, 4\}$. In the matching a, we have $a(3) = a_3$ and $a(4) = a_4$. But agents 3 and 4 can reallocate the houses they own among themselves in a manner to get a better matching for themselves. In particular, agent 3 can get a_4 (house owned by agent 4) and agent 4 can get a_2 (house owned by agent 3. Note that $a_4 \succ_3 a_3$ and $a_2 \succ_4 a_4$. Hence, both the agents are better off trading among themselves. So, they can potentially *block* matching a. We formalize this idea of blocking below.

Let a^* denote the matching reflecting the initial endowment of agents. We will use the notation a^S for every $S \subseteq N$, to denote a matching of agents in S to the houses owned by agents in S. Whenever we write a matching a without any superscript we mean a matching of all agents. Formally, a coalition (group of agents) $S \subseteq N$ can **block** a matching a at a preference profile $\succ$ if there exists a matching a^S such that $a^S(i) \succ_i a(i)$ or $a^S(i) = a(i)$ for all $i \in S$ with $a^S(j) \succ_j a(j)$ for some $j \in S$. A matching a is in the **core** at a preference profile $\succ$ if no coalition of agents can block a at $\succ$. A social choice function f is **stable** if

for all preference profile $\succ$, $f(\succ)$ is in the core at preference profile $\succ$. Note that stability implies efficiency - efficiency requires that the grand coalition cannot block.

We will now analyze if the TTC mechanism is stable. Note that when we say a TTC mechanism, we mean the TTC mechanism where the initial endowment is the endowment given by the house-tenant relationship.

THEOREM **5** *The TTC mechanism is stable. Moreover, there is a unique core matching for every preference profile.*

Proof: Assume for contradiction that the TTC mechanism is not stable. Then, there exists a preference profile $\succ$, where the matching a produced by the TTC mechanism at $\succ$ is not in the core. Let coalition S block this matching a at $\succ$. This means there exists another matching a^S such that $a^S(i) \succ_i a(i)$ or $a^S(i) = a(i)$ for all $i \in S$, with equality not holding for all $i \in S$. Let $T = \{i \in S : a^S(i) \succ_i a(i)\}$. Assume for contradiction $T = \emptyset$.

To remind notation, we denote $\hat{N}^k$ to be the set of agents allocated houses in Step k of the TTC mechanism, and $\hat{M}^k$ be the set of these houses. Clearly, agents in $S \cap \hat{N}^1$ are getting their respective top ranked houses. So, $(S \cap \hat{N}^1) \subseteq (S \setminus T)$. Define $S^k = S \cap \hat{N}^k$ for each stage k of the TTC mechanism. We now complete the proof using induction. Suppose $(S^1 \cup \ldots \cup S^{k-1}) \subseteq (S \setminus T)$ for some stage k. We show that $S^k \subseteq (S \setminus T)$. Now, agents in $S \cap \hat{N}^k$ are getting their respective top ranked houses amongst houses in $M \setminus (\hat{M}^1 \cup \ldots \cup \hat{M}^k)$. Given that agents in $(S^1 \cup \ldots \cup S^{k-1})$ get the same set of houses in a^S and a, any agent in S^k cannot be getting a better house in a^S than his house in a. Hence, again $S^k \subseteq (S \setminus T)$. By induction, $S \subseteq (S \setminus T)$ or $T = \emptyset$, which is a contradiction.

Finally, we show that the core matching returned by the TTC mechanism is the unique one. Suppose the core matching returned by the TTC mechanism is a, and let a' be another core matching for preference profile $\succ$. Note that (a) in every Step k of the TTC mechanism agents in $\hat{N}^k$ get allocated to houses owned by agents in $\hat{N}^k$, and (b) agents in $\hat{N}^1$ get their top ranked houses. Hence, if $a(i) \neq a'(i)$ for any $i \in \hat{N}^1$, then agents in $\hat{N}^1$ will block a'. So, $a(i) = a'(i)$ for all $i \in \hat{N}^1$.

Now, we use induction. Suppose, $a(i) = a'(i)$ for all $i \in \hat{N}^1 \cup \ldots \cup \hat{N}^{k-1}$. We will argue that $a(i) = a'(i)$ for all $i \in \hat{N}^k$. Agents in $\hat{N}^k$ get their highest ranked house from $M \setminus \hat{M}^1 \cup \ldots \cup \hat{M}^{k-1}$. So, given that agents in $\hat{N}^1 \cup \ldots \cup \hat{N}^{k-1}$ get the same houses in a and a', if some agent $i \in \hat{N}^k$ get different houses in a and a', then it must be $a(i) \succ_i a'(i)$. This means, agents in $\hat{N}^k$ will block a'. This contradicts the fact that a' is a core matching.

This shows that $a = a'$, a contradiction. ■

The TTC mechanism with existing tenants has another nice property. Call a mechanism

f **individually rational** if at every profile $\succ$, the matching $f(\succ) \equiv a$ satisfies $a(i) \succ_i a^*(i)$ or $a(i) = a^*(i)$ for all $i \in N$, where a^* is the matching given by the initial endowment or existing tenants.

Clearly, the TTC mechanism is individually rational. To see this, consider a profile $\succ$ and let $f(\succ) = a$. Note that the TTC mechanism has this property that if the house owned by an agent i is matched in Step k, then agent i is matched to a house in Step k too. If $a(i) \neq a^*(i)$ for some i, then agent i must be part of a trading cycle where he is pointing to a house better than $a^*(i)$. Hence, $a(i) \succ_i a^*(i)$.

This also follows from the fact that the TTC mechanism is stable and stability implies individual rationality - individual rationality means no coalition of single agent can block.

In the model of house allocation with existing tenants, the TTC mechanism satisfies three compelling properties along with stability - it is strategy-proof, efficient, and individually rational. Remarkably, these three properties characterize the TTC mechanism in the existing tenant model. We skip the proof.

THEOREM **6** *A mechanism is strategy-proof, efficient, and individually rational if and only if it is the TTC mechanism.*

Note that the serial dictatorship with a fixed priority is strategy-proof and efficient but not individually rational. The "status-quo mechanism" where everyone is assigned the houses they own is strategy-proof and individually rational but not efficient. So, the properties of individual rationality and efficiency are crucial for the characterization of Theorem 6.

8.3 Generalized TTC Mechanisms

In this section, we generalize the TTC mechanisms in a natural way so that one extreme covers the TTC mechanism we discussed and the other extreme covers the fixed priority mechanism. We can now handle the case where the number of objects is not equal to the number of agents. We now define **fixed priority TTC (FPTTC)** mechanisms. In a FPTTC mechanism, each house a_j is endowed with a priority $\sigma_j : N \to N$ over agents. This generates a profile of priorities $\sigma \equiv (\sigma_1, \ldots, \sigma_n)$.

The FPTTC mechanism then goes in stages, with each stage executing a TTC mechanism but the endowments in each stage changing with the fixed priority profile σ.

We first illustrate the idea with the example in Table 14.

Consider two priorities σ_1 and σ_2, where $\sigma_1(i) = i$ for all $i \in N$ and σ_2 is defined as $\sigma_2(1) = 2, \sigma_2(2) = 1, \sigma_2(3) = 4, \sigma_2(4) = 3$. Suppose houses a_1 and a_2 are assigned priority σ_1 but houses a_3 and a_4 are assigned priority σ_2.

$\succ_1$	$\succ_2$	$\succ_3$	$\succ_4$
a_3	a_2	a_2	a_1
a_2	a_3	a_4	a_4
a_1	a_4	a_3	a_3
a_4	a_1	a_1	a_2

Table 14: An example for housing model

In stage 1, the endowments are derived from the priorities of houses. Since houses a_1 and a_2 have agent 1 as top in their priority σ_1, agent 1 is endowed with these houses. Similarly, agent 2 is endowed houses a_3 and a_4 by priority σ_2. Now, the TTC phase of stage 1 begins. By the preferences of agents, each agent points to agent 1, except agent 1, who points to agent 2 (agent 2 is endowed house a_3, which is agent 1's top ranked house). So, trade takes place between agents 1 and 2. This is shown in Figure 8 - the endowments of agents are shown in square brackets. The edges also reflect which object it is pointing to.

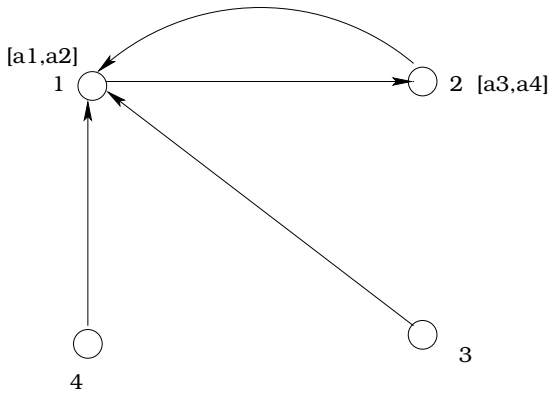

Figure 8: Cycle in stage 1 of the FPTTC mechanism

In the next stage, only agents 3 and 4 remain. Also, only houses a_1 and a_4 remain. We look at the priority of σ_1 of house a_1. Of the remaining agents, agent 3 is the top. Then, for priority σ_2 of house a_4, the top agent among remaining agent is agent 4. So, the new endowment is agent 3 gets a_1 and agent 4 gets a_4. We run the TTC phase now. Agent 3 points to agent 4 and agent 4 points to agent 3. So, they trade, and the FPTTC mechanism gives the following matching $\bar{a}$: $\bar{a}(1) = a_3, \bar{a}(2) = a_2, \bar{a}(3) = a_4, \bar{a}(3) = a_1$. This is shown in Figure 9.

If all the houses have the same fixed priority, then we recover the fixed priority mechanism. To see this, notice that because of identical priority of houses, all the houses are endowed to the same agent in every stage of the FPTTC mechanism. As a result, at stage i, the ith

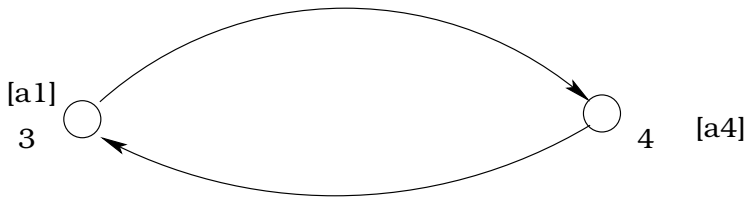

Figure 9: Cycle in stage 2 of the FPTTC mechanism

agent in the priority gets his top-ranked house. Hence, we recover the fixed priority (serial dictatorship) mechanism.

On the other extreme, if all the houses have priorities such that the top ranked agents in the priorities are distinct (i.e., for any two houses a_j, a_k with priorities σ_j and σ_k, we have $\sigma_j(1) \neq \sigma_k(1)$), then the endowments of the agents do not change over stages if the number of houses is equal to the number of agents. If there are more houses than number of agents, the endowment of each agent increases (in terms of set inclusion) across stages. So, we recover the traditional TTC mechanism for the case of equal number of agents and houses.

The following proposition can now be proved using steps similar to Proposition 10.

Proposition **11** *The FPTTC mechanism is strategy-proof and efficient.*

9 The Two-sided Matching Model

The house allocation model is a model of one-sided matching - only agents (one side of the market) had preference over the houses. In many situations, the matching market can be partitioned into two sides, and an agent on one side will have preference over agents on the other side. For instance, consider the scenario where students are matched to schools. It is plausible that not only students have preferences over the schools but schools also have a preferences over students. Other applications of two-sided matching include job applicants matched to firms, doctoral students matched to faculty etc.

Let M be a set of men and W be a set of women. For simplicity, we will assume that $|M| = |W|$ - but this is not required to derive the results. Every man $m \in M$ has a *strict* preference ordering $\succ_m$ over the set of women W. So, for $x, y \in W$, $x \succ_m y$ will imply that m ranks x over y. A matching is a bijective mapping $\mu : M \to W$, i.e., every man is assigned to a unique woman. If μ is a matching, then $\mu(m)$ denotes the woman matched to man m and $\mu^{-1}(w)$ denotes the man matched to woman w. This model is often called the "marriage market" model or "two-sided matching" model. We first discuss the stability aspects of this model, and then discuss the strategic aspects.

9.1 Stable Matchings in Marriage Market

As in the house allocation model with existing tenants, the resources to be allocated to agents in the marriage market model are owned by agents themselves. Hence, stability becomes an important criteria for designing any mechanism.

We consider an example with three men and three women. Let $M = \{m_1, m_2, m_3\}$ and $W = \{w_1, w_2, w_3\}$. Their preferences are shown in Table 15.

$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$	$\succ_{w_1}$	$\succ_{w_2}$	$\succ_{w_3}$
w_2	w_1	w_1	m_1	m_3	m_1
w_1	w_3	w_2	m_3	m_1	m_3
w_3	w_2	w_3	m_2	m_2	m_2

Table 15: Preference orderings of men and women

Consider the following matching μ: $\mu(m_1) = w_1, \mu(m_2) = w_2, \mu(m_3) = w_3$. This matching is *unstable* in the following sense. The pair $(m_1, \mu(m_2) = w_2)$ will *block* this matching (ex post) since m_1 likes w_2 over $\mu(m_1) = w_1$ and w_2 likes m_1 over $\mu^{-1}(w_2) = m_2$. So, (m_1, w_2) will break away, and form a new pair. This motivates the following definition of stability.

Definition **13** *A matching μ is* **pairwise unstable** *at preference profile $(\succ)$ if there exists $m, m' \in M$ such that (a) $\mu(m') \succ_m \mu(m)$ and (b) $m \succ_{\mu(m')} m'$. The pair $(m, \mu(m'))$ is called a* **blocking pair** *of μ at $(\succ)$. If a matching μ has no blocking pairs at a preference profile $\succ$, then it is called a* **pairwise stable matching** *at $\succ$.*

The following matching μ' is a pairwise stable matching at $\succ$: $\mu'(m_1) = w_1, \mu'(m_2) = w_3, \mu'(m_3) = w_2$ for the example in Table 15. The question is: Does a pairwise stable matching always exist? The answer to this question is remarkably yes, as we will show next.

One can imagine a stronger requirement of stability, where groups of agents block instead of just pairwise blocking. We say that a coalition $S \subseteq (M \cup W)$ **blocks** a matching μ at a profile $\succ$ if there exists another matching μ' such that (i) for all $m \in M \cap S$, $\mu'(m) \in W \cap S$ and for all $w \in W \cap S$, $\mu'^{-1}(w) \in M \cap S$, and (ii) for all $m \in M \cap S$, $\mu'(m) \succ_m \mu(m)$ and for all $w \in M \cap S$, $\mu'^{-1}(w) \succ_w \mu^{-1}(w)$. We say a matching μ is in core at a profile $\succ$ if no coalition can block μ at $\succ$. The following theorem suggests that this notion of stability is equivalent to the pairwise notion of stability we have initially defined.

Theorem **7** *A matching is pairwise stable at a profile if and only if it belongs to the core at that profile.*

Proof: Consider a matching μ which is pairwise stable at $\succ$. Assume for contradiction that μ is not in the core at $\succ$. Then, there must exist $S \subseteq (M \cup W)$ and a matching $\hat{\mu}$ such that for all $m \in M \cap S$ and for all $w \in W \cap S$ with $\hat{\mu}(m), \hat{\mu}^{-1}(w) \in S$ we have $\hat{\mu}(m) \succ_m \mu(m)$ and $\hat{\mu}^{-1}(w) \succ_w \mu^{-1}(w)$. This means for some $m \in S$ we have $\hat{\mu}(m) \in W \cap S$. Let $\hat{\mu}(m) = w$. We know $w \succ_m \mu(m)$. Then, we have $m \succ_w \mu^{-1}(w)$. Hence, (m, w) is a blocking pair at $\succ$ for μ. This implies that μ is not pairwise stable, which is a contradiction.

The other direction of the proof is trivial. ■

For this reason, we will say a matching is **stable** at a preference profile if it is pairwise stable at that preference profile. We will also drop that qualified "at a preference profile" at some places where the preference profile in question is clear from the context.

9.2 Deferred Acceptance Algorithm

In this section, we show that a stable matching always exists in the marriage market model. The fact that a stable matching always exists is proved by constructing an algorithm to find such a matching (this algorithm is due to David Gale and Lloyd Shapley, and also called the Gale-Shapley algorithm). There are two versions of this algorithm. In one version men propose to women and women either accept or reject the proposal. In another version, women propose to men and men either accept or reject the proposal. We describe the men-proposal version.

- S1. First, every man proposes to his top ranked woman.
- S2. Then, every woman who has at least one proposal keeps (tentatively) the top man amongst these proposals and rejects the rest.
- S3. Then, every man who was rejected in the last round, proposes to the top woman amongst those women who have not rejected him in earlier rounds.
- S4. Then, every woman who has at least two proposals, including any proposal tentatively kept from earlier rounds, keeps (tentatively) the top man amongst these proposals and rejects the rest. The process is then repeated from Step S3 till each woman has a proposal, at which point, the tentative proposal accepted by a woman becomes permanent.

Since each woman is allowed to keep only one proposal in every round, no woman will be assigned to more than one man. Since a man can propose only one woman at a time, no

man will be assigned to more than one woman. Since the number of men and women are the same, this algorithm will terminate at a matching. Also, the algorithm will terminate finitely since in every round, the set of women a man can propose does not increase, and strictly decreases for at least one man.

We illustrate the algorithm for the example in Table 15. A proposal from $m \in M$ to $w \in W$ will be denoted by $m \to w$.

- In the first round, every man proposes to his best woman. So, $m_1 \to w_2, m_2 \to w_1, m_3 \to w_1$.
- Hence, w_1 has two proposals: $\{m_2, m_3\}$. Since $m_3 \succ_{w_1} m_2$, w_1 rejects m_2 and keeps m_3.
- Now, m_2 is left to choose from $\{w_2, w_3\}$. Since $w_3 \succ_{m_2} w_2$, m_2 now proposes to w_3.
- Now, every woman has exactly one proposal. So the algorithm stops with the matching μ_m given by $\mu_m(m_1) = w_2, \mu_m(m_2) = w_3, \mu_m(m_3) = w_1$.

It can be verified that μ_m is a stable matching. Also, note that μ_m is a different stable matching than the stable matching μ' which we discussed earlier. Hence, there can be more than one stable matching.

One can also state a women proposing version of the deferred acceptance algorithm. Let us run the women proposing version for the example in Table 15. As before, a proposal from $w \in W$ to $m \in M$ will be denoted by $w \to m$.

- In the first round, every woman proposes to her top man. So, $w_1 \to m_1, w_2 \to m_3, w_3 \to m_1$.
- So, m_1 has two proposals: $\{w_1, w_3\}$. We note that $w_1 \succ_{m_1} w_3$. Hence, m_1 rejects w_3 and keeps w_1.
- Now, w_3 is left to choose from $\{m_2, m_3\}$. Since $m_3 \succ_{w_3} m_2$, w_3 proposes to m_3.
- This implies that m_3 has two proposals: $\{w_2, w_3\}$. Since $w_2 \succ_{m_3} w_3$, m_3 rejects w_3 and keeps w_2.
- Now, w_3 is left to choose only m_2. So, the algorithm terminates with the matching μ_w given by $\mu_w(m_1) = w_1, \mu_w(m_2) = w_3, \mu_w(m_3) = w_2$.

Note that μ_w is a stable matching and $\mu_m \neq \mu_w$.

9.3 Stability and Optimality of Deferred Acceptance Algorithm

Theorem **8** *At every preference profile, the Deferred Acceptance Algorithm terminates at a stable matching for that profile.*

Proof: Consider the Deferred Acceptance Algorithm where men propose (a similar proof works if women propose) for a preference profile $\succ$. Let μ be the final matching of the algorithm. Assume for contradiction that μ is not a stable matching. This implies that there exists a pair $m \in M$ and $w \in W$ such that (m, w) is a blocking pair. By definition $\mu(m) \neq w$ and $w \succ_m \mu(m)$. This means that w rejected m earlier in the algorithm (else m would have proposed to w at the end of the algorithm). But a woman rejects a man only if she gets a better proposal, and her proposals improve in every round. This implies that w must be assigned to a better man than m, i.e., $\mu^{-1}(w) \succ_w m$. This contradicts the fact that (m, w) is a blocking pair. ■

The men-proposing and the women-proposing versions of the Deferred Acceptance Algorithm may output different stable matchings. Is there a reason to prefer one of the stable matchings over the other? Put differently, should we use the men-proposing version of the algorithm or the women-proposing version?

To answer this question, we start with some notations. A matching μ is **men-optimal stable** matching if μ is stable and for every other stable matching μ' we have $\mu(m) \succ_m \mu'(m)$ or $\mu(m) = \mu'(m)$ for all man $m \in M$. Similarly, a matching μ is **women-optimal stable** matching if μ is stable and for every other stable matching μ' we have $\mu(w) \succ_m \mu'(w)$ or $\mu(w) = \mu'(w)$ for all woman $w \in W$.

Note that by definition, a men-optimal stable matching is unique - if there are two men optimal stable matchings μ, μ', then they must differ by at least one man's match and this man must be worse in one of the matchings. Similarly, there is a unique women-optimal stable matching.

Theorem **9** *The men-proposing version of the Deferred Acceptance Algorithm terminates at the unique men-optimal stable matching and the women-proposing version of the Deferred Acceptance Algorithm terminates at the unique women-optimal stable matching.*

Proof: We do the proof for men-proposing version of the algorithm. The proof is similar for the women-proposing version. Let $\hat{\mu}$ be the stable matching obtained at the end of the

men-proposing Deferred Acceptance Algorithm. Assume for contradiction that $\hat{\mu}$ is not men-optimal. Then, there exists a stable matching μ such that for some $m \in M$, $\mu(m) \succ_m \hat{\mu}(m)$. Let $M' = \{m \in M : \mu(m) \succ_m \hat{\mu}(m)\}$. Hence, $M' \neq \emptyset$.

Now, for every $m \in M'$, since $\mu(m) \succ_m \hat{\mu}(m)$, we know that m is rejected by $\mu(m)$ in some round of the algorithm. Denote the round in which $m \in M'$ is rejected by $\mu(m)$ by t_m. Choose $m' \in \arg\min_{m \in M'} t_m$, i.e., choose a man m' who is the first to be rejected by $\mu(m')$ among all men in M'. Since $\mu(m')$ rejects m', she must have got a better proposal from some other man m'', i.e.,

$$m'' \succ_{\mu(m')} m'. \tag{1}$$

Now, consider $\mu(m')$ and $\mu(m'')$. If $m'' \notin M'$, then $\hat{\mu}(m'') = \mu(m'')$ or $\hat{\mu}(m'') \succ_{m''} \mu(m'')$. Since m'' is eventually assigned to $\hat{\mu}(m'')$, it must be the last woman that m'' must have proposed in DAA. The fact that m'' proposed to $\mu(m')$ earlier means $\mu(m') \succ_{m''} \hat{\mu}(m'')$. Using, $\hat{\mu}(m'') = \mu(m'')$ or $\hat{\mu}(m'') \succ_{m''} \mu(m'')$, we get

$$\mu(m') \succ_{m''} \mu(m'').$$

If $m'' \in M'$, then, since $t_{m''} > t_{m'}$, m'' has not been rejected by $\mu(m'')$ till round $t_{m'}$. This means, again, m'' proposed to $\mu(m')$ before proposing to $\mu(m'')$. Hence, as in the earlier case, we get

$$\mu(m') \succ_{m''} \mu(m''). \tag{2}$$

By Equations 1 and 2, $(m'', \mu(m'))$ forms a blocking pair. Hence, μ is not stable. This is a contradiction. ■

The natural question is then whether there exists a stable matching that is optimal for both men and women. The answer is no. The example in Table 15 has two stable matchings, one is optimal for men but not for women and one is optimal for women but not for men. Also, there is a unique men-optimal stable matching and a unique women-optimal stable matching (the proof of this fact is skipped).

9.4 Strategic Issues in Deferred Acceptance Algorithm

We next turn to strategic properties of the Deferred Acceptance Algorithm (DAA). We first consider the men-proposing version. We define the notion of strategyproofness informally here. Strategyproofness is with respect to the direct revelation mechanism. The DAA is

strategy-proof if reporting a non-truthful preference ordering does not result in a better outcome for an agent for any reported preferences of other agents.

We first show that the men-proposing version of the Deferred Acceptance Algorithm is not strategyproof for women (i.e., women can manipulate). Let us return to the example in Table 15. We know if everyone is truthful, then the matching is: $\mu(m_1) = w_2, \mu(m_2) = w_3, \mu(m_3) = w_1$. We will show that w_1 can get a better outcome by not being truthful. We show the steps here.

- In the first round, every man proposes to his best woman. So, $m_1 \rightarrow w_2, m_2 \rightarrow w_1, m_3 \rightarrow w_1$.
- Next, w_2 only has one proposal (from m_1) and she accepts it. But w_1 has two proposals: $\{m_2, m_3\}$. If she is truthful, she should accept m_3. We will see what happens if she is not truthful. So, she accepts m_2.
- Now, m_3 has two choices: $\{w_2, w_3\}$. He likes w_2 over w_3. So, he proposes to w_2.
- Now, w_2 has two proposals: $\{m_1, m_3\}$. Since she likes m_3 over m_1, she accepts m_3.
- Now, m_1 has a choice between w_1 and w_3. Since he likes w_1 over w_3, he proposes to w_1.
- Now, w_1 has two proposal: $\{m_1, m_2\}$. Since she prefers m_1 over m_2 she accepts m_1.
- So, m_2 is only left with $\{w_2, w_3\}$. Since he likes w_3 over w_2 he proposes to w_3, which she accepts. So, the final matching $\hat{\mu}$ is given by $\hat{\mu}(m_1) = w_1, \hat{\mu}(m_2) = w_3, \hat{\mu}(m_3) = w_2$.

Hence, w_1 gets m_1 in $\hat{\mu}$ but was getting m_3 earlier. The fact that $m_1 \succ_{w_1} m_3$ shows that not being truthful helps w_1. However, the same result does not hold for men. Similarly, the women-proposing DAA is not strategy-proof for men.

THEOREM **10** *The men-proposing version of the Deferred Acceptance Algorithm is strategyproof for men. The women-proposing version of the Deferred Acceptance Algorithm is strategyproof for women.*

Proof: Suppose there is a profile $\pi = (\succ_{m_1}, \ldots, \succ_{m_n}, \succ_{w_1}, \ldots, \succ_{w_n})$ such that man m_1 can misreport his preference to be $\succ_*$, and obtain a better matching. Let this preference profile be π'. Let μ be the stable matching obtained by the men-proposing deferred acceptance algorithm when applied to π. Let ν be the stable matching obtained by the men-proposing

algorithm when applied to π'. We show that if $\nu(m_1) \succ_{m_1} \mu(m_1)$, then ν is not stable at π', which is a contradiction.

Let $R = \{m : \nu(m) \succ_m \mu(m)\}$. Since $m_1 \in R$, R is not empty. We show that $\{w : \nu^{-1}(w) \in R\} = \{w : \mu^{-1}(w) \in R\}$. Take any $\nu^{-1}(w) \in R$, we will show that $\mu^{-1}(w) \in R$, and this will establish the claim. If $\mu^{-1}(w) = m_1$, then we are done by definition. Else, let $w = \nu(m)$ and $m' = \mu^{-1}(w)$. Since $w \succ_m \mu(m)$, stability of μ at π implies that $m' \succ_w m$. Stability of ν at π' implies that $\nu(m') \succ_{m'} w$. Therefore, $m' \in R$. Let $S = \{w : \nu^{-1}(w) \in R\} = \{w : \mu^{-1}(w) \in R\}$.

By definition $\nu(m) \succ_m \mu(m)$ for any $m \in R$. By stability of μ, we then have $\mu^{-1}(w) \succ_w \nu^{-1}(w)$ for all $w \in S$. Now, pick any $w \in S$. By definition, $w \succ_{\nu^{-1}(w)} \mu(\nu^{-1}(w))$. This implies that during the execution of the men-proposing deferred acceptance algorithm at π, $\nu^{-1}(w) \in R$ must have proposed to w which she had rejected. Let $m \in R$ be the last man in R to make a proposal during the execution of the men-proposing deferred acceptance algorithm at π. Suppose this proposal is made to $w = \mu(m) \in S$. As argued, w rejected $\nu^{-1}(w)$ earlier. This means that when m proposed to w, she had some proposal, say from m', which she rejected. By definition, m' cannot be in R. This means that $m' \neq \nu^{-1}(w)$, and hence, $m' \succ_w \nu^{-1}(w)$. Since $m' \notin R$, $\mu(m') \succ_{m'} \nu(m')$ or $\mu(m') = \nu(m')$. Also, since w rejects m', $w \succ_{m'} \mu(m')$. This shows that $w \succ_{m'} \nu(m')$. This shows that (m', w) form a blocking pair for ν at π'. ■

Does this mean that no mechanism can be both stable and be strategyproof to all agents? The answer is yes.

THEOREM **11** *No mechanism which gives a stable matching can be strategy-proof for both men and women.*

However, one can trivially construct strategy-proof mechanisms for both men and women. Consider a mechanism which ignores all men (or women) orderings. Then, it can run a fixed priority mechanism for men (or women) or a TTC mechanism with fixed endowments for men (or women) to get a strategy-proof mechanism.

9.5 Extensions with Quotas and Individual Rationality

The deferred acceptance algorithm can be suitably modified to handle some generalizations. One such generalization is used in school choice problems. In a school choice problem, a set of students (men) and a set of schools (women) have preference ordering over each other. Each school has a quota, i.e., the maximum number of students it can take. In particular,

every school i has a quota of $q_i \geq 1$. Now, colleges need to have preferences over sets of students. For this, we need to extend preferences over students to subsets of students. There are many ways to do it. The standard restriction is **responsive** preferences: suppose S is a set of students and $s \notin S$ but $t \in S$, then $S \setminus \{t\} \cup \{s\}$ is preferred to S if and only if s is preferred to t. Usually, colleges do not like some students. This is modeled by allowing only acceptable students preferences. In the preference relation, we put the $\emptyset$ symbol to reflect this - i.e., any students who are worse than this are not acceptable. Again, a set of students S is worse than $S \cup \{s\}$ if and only if s is acceptable.

Students, on the other hand, have a set of schools that are acceptable and another set which is not acceptable, i.e., on top of the usual linear order over the set of schools, each student also has a *cut-off school*, below which he prefers to not attend any school. The preferences of agents are handled by adding a **dummy** school 0, whose quota is the number of students (so this school can admit possibly all students). An admission in the dummy school indicates that the student is not assigned any school. Now, each student has a preference ordering over the set of schools and the dummy school. All the schools below dummy school are never preferred by the student.

The deferred acceptance algorithm can be modified in a straightforward way in these settings. Each student proposes to its favorite remaining acceptable school. A proposal to the dummy school is always accepted. Any other school k evaluates the set of proposals it has, and accepts the top $\min(q_k, \text{number of proposals})$ acceptable proposals. The procedure is repeated as was described earlier. One can extend the stability, student-optimal stability, and strategy-proofness results of previous section to this setting in a straightforward way.

Another important property of a mechanism in such a set up is **individual rationality**. Individual rationality says that no student should get a school lower than the dummy school. It is clear that the deferred acceptance algorithm produces an individually rational matching.

10 Applications of Various Matching Models

The matching theory is one of those theories which have been applied extensively in practice. We give some examples.

- Deferred Acceptance Algorithm. Deferred acceptance algorithm (DAA) has been successfully used in assigning students to schools in New York City (high school) and Boston (all public schools). It is also used in assigning medical interns (doctors) to hospitals in US medicine schools. The US medical community has been at the forefront of implementing DAA - it is used in residents matching, doctor assignments to jobs,

and many other markets.

- Versions of Serial Dictatorship. Some (random) version of serial dictatorship (priority) mechanism is widely used in US Universities like Yale, Princeton, CMU, Harvard, Duke, Michigan to allocate graduate housing to graduate students. The version that is used is called *random serial dictatorship with squatting rights*. In this version, first existing tenants are given the option of entering the mechanism or going away with their existing house. After everyone announces their willingness to participate in the mechanism, an ordering of (participating) students is done uniformly at random. Then, serial dictatorship is applied on this ordering.

- Kindney Exchange. The kidney exchange problem can be modeled as a house allocation problem with existing tenants. In a kidney exchange problem, each patient (agent) can come with an incompatible donor agent (house which is endowed to him), and there is a set of donor agents (vacant houses). Patients have preference over donors (houses). A matching in this case is an assignment of patients to donors. There are two major differences from the model of house allocation with existing tenants: (i) not all houses have tenants (ii) number of houses is more than the number of agents. Variants of top trading cycle algorithm has been proposed, and run in US hospital systems to match kidney patients to donors.

11 Randomized Social Choice Function

Randomization is a way of expanding the set of possible strategy-proof social choice function. Lotteries are also common in practice. So, it makes sense to study the effects of randomization on strategy-proofness.

As before let $A = \{a, b, c, \ldots\}$ be a finite set of alternatives with $|A| = m$ and $N = \{1, \ldots, n\}$ be the set of agents. Let $\mathcal{L}(A)$ denote the set of all probability distributions over A. We will refer to this set as the set of **lotteries** over A. A particular element $\lambda \in \mathcal{L}(A)$ is a probability distribution over A, and λ_a denotes the probability of alternative a. Of course $\lambda_a \geq 0$ for all $a \in A$ and $\sum_{a \in A} \lambda_a = 1$. As before, every agent i has a linear order over A, which is his preference ordering. A randomized social choice function picks a lottery over A at every profile of preference orderings. Hence, the set of outcomes is the set of all lotteries over A, i.e., $\mathcal{L}(A)$. Note that we have not defined a preference ordering over $\mathcal{L}(A)$. Hence, a crucial component of analyzing randomized social choice functions is

how should two lotteries $\lambda, \lambda' \in \mathcal{L}(A)$ be compared given a preference ordering over A?

We discuss below a very basic way of making such a comparison. Let $\mathcal{P}$ is the set of all linear orders over A. The domain of interest may be any subset $\mathcal{D} \subseteq \mathcal{P}$. A **randomized social choice function (RSCF)** f is a mapping $f : \mathcal{D}^n \to \mathcal{L}(A)$. [8] We let $f_a(P)$ to denote the probability of alternative a being chosen at profile P. To avoid confusion, we refer to $f : \mathcal{D}^n \to A$ as a **deterministic** social choice function (DSCF).

11.1 Defining Strategy-proof RSCF

There are several meaningful ways to define strategy-proofness in this setting. We follow one of the first-proposed approaches (by Gibbard). It requires that an RSCF be non-manipulable for every *utility representation* of linear orders when lotteries are evaluated using the **expected utility criteria**.

A utility function $u : A \to \mathbb{R}$ represents a preference ordering $P_i \in \mathcal{D}$ if for all $a, b \in A$, $u(a) > u(b)$ if and only if aP_ib. Given a utility representation u of P_i, the utility from a lottery $\lambda \in \mathcal{A}$ is computed using the expected utility criteria, and is given by

$$\sum_{a \in A} \lambda_a u(a).$$

Notice that this is a *domain restriction* - the utility of a lottery outcome is *restricted* to be the expected utility of the alternatives in its support. Hence, analysis of randomized social choice function is similar to analyzing restricted domains, and therefore, we hope to uncover more social choice functions than in the deterministic case.

Now, it is easy to define the notion of strategy-proofness. An RSCF is strategy-proof if for *every possible representation of orderings*, the expected utility of telling the truth is not less than the expected utility of lying.

Definition **14** *An RSCF $f : \mathcal{D}^n \to \mathcal{L}(A)$ is* **strategy-proof** *if for all $i \in N$, all $P_{-i} \in \mathcal{D}^{n-1}$, for all $P_i \in \mathcal{D}$, and for all utility functions $u : A \to \mathbb{R}$ representing P_i, we have*

$$\sum_{a \in A} u(a) f_a(P_i, P_{-i}) \geq \sum_{a \in A} u(a) f_a(P_i', P_{-i}) \qquad \forall\ P_i' \in \mathcal{D}.$$

For the strategy-proofness of DSCF, we did not require this utility representation. However, it is easy to verify that a DSCF is strategy-proof in the sense of Definition 3 if and only if it

[8] Though, we assumed that the domain of every agent is the same $\mathcal{D}$, this is not required for the results we state here, and assumed for simplicity of notation.

is strategy-proof in the sense of Definition 14. Also, the qualifier "for all utility functions" in the above definitions is extremely important. It underlines the fact that we are considering *ordinal* social choice functions. If we were using "cardinal" social choice functions, then we will elicit utility functions from the agents instead of preference orderings.

It is well known that the above formulation of strategy-proofness is equivalent to first-order stochastic dominance. To define this formally, let $B(a, P_i) = \{b \in A : b = a \text{ or } bP_ia\}$. We can define the strategy-proofness in the following equivalent way.

DEFINITION **15** *An RSCF $f : \mathcal{D}^n \to \mathcal{L}(A)$ is* **strategy-proof** *if for all $i \in N$, all $P_{-i} \in \mathcal{D}^{n-1}$, for all $P_i \in \mathcal{D}$, and for all $a \in A$, we have*

$$\sum_{b \in B(a,P_i)} f_b(P_i, P_{-i}) \geq \sum_{b \in B(a,P_i)} f_b(P_i', P_{-i}) \qquad \forall\ P_i' \in \mathcal{D}.$$

The necessity of this first-order stochastic dominance is easy to derive. Fix some $i \in N$, some P_{-i}, some P_i and some alternative $a \in A$. A particular u that represents P_i is of the following form: $u(b) \to 1$ for all $b \in B(a, P_i)$ and $u(b) \to 0$ for all $b \notin B(a, P_i)$. Then, strategy-proofness gives that for every P_i', we must have

$$\sum_{b \in A} u(b) f_b(P_i, P_{-i}) \geq \sum_{b \in A} u(b) f_b(P_i', P_{-i}).$$

Substituting for u, we get

$$\sum_{b \in B(a,P_i)} f_b(P_i, P_{-i}) \geq \sum_{b \in B(a,P_i)} f_b(P_i', P_{-i}).$$

It can also be shown that the first-order stochastic dominance condition is sufficient for strategy-proofness (see Chapter 6 in Mas-Collel-Whinston-Green).

To understand this definition a little better let us take an example with two agents $\{1, 2\}$ and three alternatives $\{a, b, c\}$. The preference of agent 2 is fixed at P_2 given by aP_2bP_2c. Let us consider two preference orderings of agent 1: $P_1 : bP_1cP_1a$ and $P_1' : cP_1aP_1b$. Denote $P = (P_1, P_2)$ and $P' = (P_1', P_2)$. Suppose $f_a(P) = 0.6$ and $f_b(P) = 0.1$ and $f_c(P) = 0.3$. First order stochastic dominance requires the following.

$$f_b(P) = 0.1 \geq f_b(P')$$
$$f_b(P) + f_c(P) = 0.4 \geq f_b(P') + f_c(P').$$

Summarizing, we consider randomization but ordinal social choice functions. Agents have preferences over alternatives and use that to evaluate lotteries. Our idea of truthfulness says

that the lottery given by the scf from truthtelling must first-order stochastically dominate *every* other lottery that this agent can get from lying. This notion of strategy-proofness is equivalent to preventing manipulation for *all* cardinalization of preferences when agents use expected utility to evaluate lotteries.

Of course, we can think of other notions of strategy-proofness. For instance, fix agent i and fix the preferences of other agents at P_{-i}. We can say that agent i manipulates f at (P_i, P_{-i}) if there exists P_i' such that the lottery $f(P_i', P_{-i})$ first order stochastically dominates $f(P_i, P_{-i})$. Then, we can say that f is strategy-proof if no agent can manipulate it at any profile. This will be a weaker notion of strategy-proofness than what we use.

Another method of defining strategy-proofness is *lexicographic*. Again, fix agent i and fix the preferences of other agents at P_{-i}. Take two preferences P_i, P_i' of agent i. Then, defines a binary relation over every pair of lotteries using P_i in a lexicographic manner. It evaluates lotteries $f(P_i, P_{-i})$ and $f(P_i', P_{-i})$ in the following way: it first looks at $P_i(1)$ - the top ranked alternative in P_i, and compares the two lotteries; if they are the same, then it looks at $P_i(2)$, and so on. We can define strategy-proofness easily now - $f(P_i, P_{-i})$ must be lexicographically better than $f(P_i', P_{-i})$, where the lexicographic comparison is done using P_i.

11.2 Randomization over DSCFs

A natural way to construct an RSCF is to take a collection of DSCFs and randomize over them. We show a general result on strategy-proofness of RSCFs which can be expressed as a convex combination of other strategy-proof RSCFs.

Proposition **12** *Let $f^1, f^2, \ldots, f^k$ be a set of k strategy-proof RSCFs, all defined on the domain $\mathcal{D}^n$. Let $f : \mathcal{D}^n \to \mathcal{L}(A)$ be defined as: for all $P \in \mathcal{D}^n$ and for all $a \in A$, $f_a(P) = \sum_{j=1}^k \lambda_j f_a^j(P)$, where $\lambda_j \in [0,1]$ for all $j \in \{1, \ldots, k\}$ and $\sum_{j=1}^k \lambda_j = 1$. Then, f is strategy-proof.*

Proof: Fix an agent i and a profile P_{-i}. For some preference P_i consider a utility representation $u : A \to \mathbb{R}$. Then, for any P_i',

$$\begin{aligned}\sum_{a\in A} u(a)f_a(P) &= \sum_{a\in A} u(a)\sum_{j=1}^{k}\lambda_j f_a^j(P) = \sum_{j=1}^{k}\lambda_j\sum_{a\in A} u(a)f_a^j(P)\\ &\geq \sum_{j=1}^{k}\lambda_j\sum_{a\in A} u(a)f_a^j(P_i', P_{-i}) = \sum_{a\in A} u(a)\sum_{j=1}^{k}\lambda_j f_a^j(P_i', P_{-i})\\ &= \sum_{a\in A} u(a)f_a(P_i', P_{-i}).\end{aligned}$$

■

Another way to interpret Proposition 12 is that the set of strategy-proof RSCFs form a convex set. Since a DSCF cannot be written as convex combination of other social choice functions, a strategy-proof DSCF forms an *extreme point* of the set of strategy-proof RSCFs. Knowing the deterministic strategy-proof social choice functions automatically gives you a class of strategy-proof RSCFs.

11.3 The Counterpart of Gibbard-Satterthwaite Theorem

To understand the implication of randomization, we go back to the complete domain model in the Gibbard-Satterthwaite theorem. First, we define the notion of unanimity that we will use in this model. [9] The notion of unanimity we use is the exact version of unanimity we used in the deterministic social choice functions.

Definition **16** *An RSCF $f : \mathcal{P}^n \to \mathcal{L}(A)$ satisfies* **unanimity** *if for all $i \in N$, all $P \in \mathcal{P}^n$ such that $P_1(1) = P_2(1) = \ldots = P_n(1) = a$, we have $f_a(P) = 1$.*

As in the deterministic SCF case, we can see that the constant social choice function is not unanimous. But there is even a bigger class of RSCFs which are strategy-proof but not unanimous.

Definition **17** *An RSCF f is a* **unilateral** *if there exists an agent i and $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_{|A|}$ with $\alpha_j \in [0,1]$ and $\sum_{j=1}^{|A|}\alpha_j = 1$ such that for all P we have $f_{P_i(j)} = \alpha_j$ for all $j \in \{1, \ldots, |A|\}$.*

[9] In the deterministic model, there was an equivalence between unanimity, efficiency, and ontoness under strategy-proofness - this is no longer true in the model with randomization.

In a unilateral RSCF, there is a **weak dictator** i such that top ranked alternative of i gets probability α_1, second ranked alternative of i gets probability α_2, and so on. Notice that every unilateral is strategy-proof, but not unanimous.

We now define another broad class of RSCFs which are strategy-proof and unanimous.

DEFINITION **18** *An RSCF $f : \mathcal{P}^n \to \mathcal{L}(A)$ is a* **random dictatorship** *if there exists weights $\beta_1, \ldots, \beta_n \in [0,1]$ with $\sum_{i \in N} \beta_i = 1$ such that for all $P \in \mathcal{P}^n$,*

$$f_a(P) = \sum_{i \in N : P_i(1) = a} \beta_i.$$

If a particular agent i has $\beta_i = 1$, then such a random dictatorship is the usual dictatorship. A random dictatorship can be thought to be a randomization over deterministic dictatorships, where β_i reflects the probability with which agent i is a dictator. For example, if $N = \{1,2,3\}$ and $A = \{a,b,c\}$ and $\beta_1 = \frac{1}{2}$, $\beta_2 = \beta_3 = \frac{1}{4}$, then at a profile P where $P_1(1) = a, P_2(1) = a, P_3(1) = c$, the output of this random dictatorship will be $f_a(P) = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$ and $f_c(P) = \frac{1}{4}$.

Random dictatorship can be thought of as a convex combination of dictatorships, where β_i is the probability with which agent i is the dictator. Since dictatorship is strategy-proof, one can show that random dictatorship is also strategy-proof. As a corollary of Proposition 12, we get the following.

COROLLARY **1** *Every random dictatorship is strategy-proof.*

Proof: A random dictatorship is a convex combination of dictatorships. Hence, it is strategy-proof by Proposition 12. ■

We are now ready to state the counterpart of the Gibbard-Satterthwaite theorem for RSCFs. This was proved by Gibbard.

THEOREM **12** *Suppose $|A| \geq 3$. An RSCF is unanimous and strategy-proof if and only if it is a random dictatorship.*

The proof of this theorem is more involved than the Gibbard-Satterthwaite theorem. We only do the case with two agents.

Proof: We have already shown that a random dictatorship is strategy-proof (Corollary 1). It is also unanimous - if all agents have the same alternative as top ranked, βs will sum to 1 for that alternative. We now prove that any RSCF which is unanimous and strategy-proof must be a random dictatorship for $n = 2$ case. We do the proof by showing two claims. Let f be a strategy-proof and unanimous RSCF.

CLAIM **2** *Let $P \in \mathcal{P}^2$ be a preference profile such that $P_1(1) \neq P_2(1)$. If $f_a(P) > 0$ then $a \in \{P_1(1), P_2(1)\}$.*

Proof: Consider a preference profile P such that $P_1(1) = a \neq b = P_2(1)$. Let $f_a(P) = \alpha$ and $f_b(P) = \beta$. Consider a preference ordering P_1' such that $P_1'(1) = P_1(1) = a$ and $P_1'(2) = P_2(1) = b$. Similarly, consider a preference ordering P_2' such that $P_2'(1) = P_2(1) = b$ and $P_2'(2) = P_1(1) = a$.

Strategy-proofness implies that $f_a(P_1', P_2) = \alpha$. Also, by unanimity the outcome at (P_2, P_2) is b. So, strategy-proofness implies that $f_a(P_1', P_2) + f_b(P_1', P_2) \geq f_a(P_2, P_2) + f_b(P_2, P_2) = 1$. Hence, $f_a(P_1', P_2) + f_b(P_1', P_2) = 1$.

Using a symmetric argument, we can conclude that $f_b(P_1, P_2') = \beta$ and $f_a(P_1, P_2') + f_b(P_1, P_2') = 1$.

Strategy-proofness implies that $f_b(P_1', P_2') = f_b(P_1', P_2) = 1 - \alpha$. and $f_a(P_1', P_2') = f_a(P_1, P_2') = 1 - \beta$. But $f_a(P_1', P_2') + f_b(P_1', P_2') \leq 1$ implies that $\alpha + \beta \geq 1$ and $f_a(P) + f_b(P) \leq 1$ implies $\alpha + \beta \leq 1$. Hence, $\alpha + \beta = 1$. ■

CLAIM **3** *Let $P, \bar{P} \in \mathcal{P}^2$ be such that $P_1(1) = a \neq b = P_2(1)$ and $\bar{P}_1(1) = c \neq d = \bar{P}_2(1)$. Then $f_a(P) = f_c(\bar{P})$ and $f_b(P) = f_d(\bar{P})$.*

Proof: We consider various cases.

CASE 1: $c = a$ and $d = b$. Strategy-proofness implies that $f_a(P_1, P_2) = f_a(\bar{P}_1, P_2)$. By Claim 2, $f_a(P_1, P_2) + f_b(P_1, P_2) = f_a(\bar{P}_1, P_2) + f_b(\bar{P}_1, P_2) = 1$. Hence, $f_b(P_1, P_2) = f_b(\bar{P}_1, P_2)$. Repeating this argument for agent 2 while going from $(\bar{P}_1, P_2)$ to $(\bar{P}_1, \bar{P}_2)$, we get that $f_a(\bar{P}) = f_a(P)$ and $f_b(\bar{P}) = f_b(P)$.

CASE 2: $c = a$ or $d = b$. Suppose $c = a$. Consider a preference profile $(P_1, \hat{P}_2)$ such that $\hat{P}_2(1) = d \notin \{a, b\}$ and $\hat{P}_2(2) = b$. Assume without loss of generality that $P_2(1) = b$ and $P_2(2) = d$. Then, strategy-proofness implies that $f_b(P_1, \hat{P}_2) + f_d(P_1, \hat{P}_2) = f_b(P) + f_d(P)$. By Claim 2, $f_b(P_1, \hat{P}_2) = f_d(P) = 0$. Hence, $f_b(P) = f_d(P_1, \hat{P}_2)$. This further implies that $f_a(P) = f_a(P_1, \hat{P}_2)$. By Case 1, $f_a(P) = f_a(\bar{P})$ and $f_b(P) = f_d(\bar{P})$. An analogous proof works if $d = b$.

CASE 3: $c = b$ and $d \notin \{a, b\}$. Let $\hat{P} = (P_1, \bar{P}_2)$. By Case 2, $f_a(P) = f_a(\hat{P})$ and $f_b(P) = f_d(\hat{P})$. Again, applying Case 2, we get $f_a(P) = f_a(\hat{P}) = f_b(\bar{P})$ and $f_b(P) = f_d(\hat{P}) = f_d(\bar{P})$.

Case 4: $c \notin \{a, b\}$ and $d = a$. A symmetric argument to Case 3 can be made.

Case 5: $c = b$ and $d = a$. Since there are at least three alternatives there is a $x \notin \{a, b\}$. We construct a profile $\hat{P} = (\hat{P}_1, \bar{P}_2)$ such that $\hat{P}_1(1) = x$. By Case 4, $f_x(\hat{P}) = f_a(P)$ and $f_b(P) = f_a(\hat{P})$. Now, applying Case 2, we can conclude that $f_x(\hat{P}) = f_b(\bar{P})$ and $f_a(\hat{P}) = f_a(\bar{P})$.

Case 6: $c \notin \{a, b\}$ and $d \notin \{a, b\}$. Consider a profile $\hat{P} = (\hat{P}_1, P_2)$ such that $\hat{P}_1(1) = c$. By Case 2, $f_c(\hat{P}) = f_a(P)$ and $f_b(\hat{P}) = f_b(P)$. Applying Case 2 again, we get $f_c(\bar{P}) = f_c(\hat{P}) = f_a(P)$ and $f_d(\bar{P}) = f_b(\hat{P}) = f_b(P)$. ■

Claims 2 and 3 establishes that f is a RSCF. ■

As we have seen a unilateral SCF is not unanimous but strategy-proof. Hence, unanimity is a crucial assumption in Theorem 12.

12 Mechanism Design with Transfers

We will now discuss mechanism design where transfers are allowed. We will continue to restrict attention to private values setting, where the value of an agent for an alternative depends on his own type only. If transfers are allowed, then we have two decisions to make - (a) what alternative to choose (b) how much payment to make. In general, an agent i, can have a general utility function $U_i : T_i \times A \times \mathbb{R} \to \mathbb{R}$, where A is the set of alternatives and T_i is the type space of agent i. Here, $U_i(t_i, a, p_i)$ will denote the payment of agent i with type t_i on alternative a when he makes a payment of p_i. In this case, we can treat (a, p_i) to be the outcome. If U_i can be any function such that any ranking among the outcomes is possible, then we will be back in the Gibbard-Satterthwaite framework.

Usually, it is plausible to assume that U_i has some particular structure (that is common knowledge). We will make one such assumption. The crucial assumption we will make on the utility function is that net utility is the value from the alternative minus the payment. Suppose $v_i(a, t_i)$ is the value of agent i from an alternative a when his type is t_i and he makes a payment of p_i, then his net utility is

$$U_i(t_i, a, p_i) := v_i(a, t_i) - p_i.$$

This assumption is called the **quasi-linear utility** assumption.

The crucial element of the quasi-linearity assumption is that we can separate the value from the alternative and the utility from the payment. Moreover, now we cannot have

unrestricted ranking of outcomes - for instance if $p_i > p_i'$, we must always have (a, p_i') preferred to (a, p_i). The separability of value from alternative and the utility from payment allows us to formulate incentive compatibility constraints in a more lucid manner.

12.1 A General Model

The set of agents is denoted by $N = \{1, \ldots, n\}$. The set of potential social decisions or outcomes or alternatives is denoted by the set A, which can be finite or infinite. For our purposes, we will assume A to be finite. Every agent has a private information, called his **type**. The type of agent $i \in N$ is denoted by t_i which lies in some set T_i, called the type space. Type t_i can be a multi-dimensional vector in $\mathbb{R}^K$, where K is some positive integer. We denote a profile of types as $t = (t_1, \ldots, t_n)$ and the product of type spaces of all agents as $T^n = \times_{i \in N} T_i$. The type space T_i reflects the information the mechanism designer has about agent i.

Agents have preferences over alternatives which depends on their respective types. This is captured using a utility function. The utility function of agent $i \in N$ is $v_i : A \times T_i \to \mathbb{R}$. Thus, $v_i(a, t_i)$ denotes the utility of agent $i \in N$ for decision $a \in A$ when his type is $t_i \in T_i$. Note that the mechanism designer knows T_i and the utility function v_i. Of course, he does not know the *realizations* of each agent's type.

We will restrict attention to this setting, called the **private values** setting, where the utility function of an agent is independent of the types of other agents, and is completely known to him.

12.2 Allocation Rules

A **decision rule or an allocation rule** f is a mapping $f : T^n \to A$. Hence, an allocation rule gives a decision as a function of the types of the agents. From every type profile matrix, we construct a valuation matrix with n rows (one row for every agent) and $|A|$ columns. An entry in this matrix corresponding to type profile t, agent i, and $a \in A$ has value $v_i(a, t_i)$. We show one valuation matrix for $N = \{1, 2\}$ and $A = \{a, b, c\}$ below.

$$\begin{bmatrix} v_1(a, t_1) & v_1(b, t_1) & v_1(c, t_1) \\ v_2(a, t_2) & v_2(b, t_2) & v_2(c, t_2) \end{bmatrix}$$

Here, we give some examples of allocation rules.

- **Constant allocation:** The constant allocation rule f^c allocates some $a \in A$ for every $t \in T^n$. In particular, there exists $a \in A$ such that for every $t \in T$ we have

$$f^c(t) = a.$$

- **Dictator allocation:** The dictator allocation rule f^d allocates the *best* decision of some **dictator** agent $i \in N$. In partcular, let $i \in N$ be the dictator agent. Then, for every $t_i \in T_i$ and every $t_{-i} \in T_{-i}$,

$$f^d(t_i, t_{-i}) \in \arg\max_{a \in A} v_i(a, t_i).$$

  It picks a dictator i and always chooses the column in the valuation matrix for which the i row has the maximum value in the valuation matrix.

- **Efficient allocation:** The efficient allocation rule f^e is the one which maximizes the sum of values of agents. In particular, for every $t \in T^n$,

$$f^e(t) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, t_i).$$

  This rule first sums the entries in each of the columns in the valuation matrix and picks a column which has the maximum sum.

  Hence, efficiency implies that the total value of agents is maximized in all states of the world (i.e., for all possible type profiles of agents). We will discuss why this is *Pareto efficient* later.

  Consider an example where a seller needs to sell an object to a set of buyers. In any allocation, one buyer gets the object and the others get nothing. The buyer who gets the object realizes his value for the object, while others realize no utility. Clearly, to maximize the total value of the buyers, we need to maximize this realized value, which is done by allocating the object to the buyer with the highest value.

  This particular allocation rule is also referred to as the **utilitarianism** allocation rule.

- **Anti-efficient allocation:** The anti-efficient allocation rule f^a is the one which minimizes the sum of values of agents. In particular, for every $t \in T^n$

$$f^a(t) \in \arg\min_{a \in A} \sum_{i \in N} v_i(a, t_i).$$

- **Weighted efficient/utilitarianism allocation:** The weighted efficient allocation rule f^w is the one which maximizes the weighted sum of values of agents. In particular, there exists $\lambda \in \mathbb{R}^n_+ \setminus \{0\}$ such that for every $t \in T^n$,

$$f^w(t) \in \arg\max_{a \in A} \sum_{i \in N} \lambda_i v_i(a, t_i).$$

  This rule first does a weighted sum of the entries in each of the columns in the valuation matrix and picks a column which has the maximum weighted sum.

- **Affine maximizer allocation:** The affine maximizer allocation rule f^a is the one which maximizes the weighted sum of values of agents and a term for every allocation. In particular, there exists $\lambda \in \mathbb{R}^n_+ \setminus \{0\}$ and $\kappa : A \to \mathbb{R}$ such that for every $t \in T^n$,

$$f^a(t) \in \arg\max_{a \in A} \big[\sum_{i \in N} \lambda_i v_i(a, t_i) - \kappa(a) \big].$$

  This rule first does a weighted sum of the entries in each of the columns in the valuation matrix and subtracts κ term corresponding to this column, and picks the column which has this sum highest.

- **Max-min (Rawls) allocation:** The max-min (Rawls) allocation rule f^r picks the allocation which maximizes the minimum value of agents. In particular for every $t \in T^n$,

$$f^r(t) \in \arg\max_{a \in A} \min_{i \in N} v_i(a, t_i).$$

  This rule finds the minimum entry in each column of the valuation matrix and picks the column which has the maximum such minimum entry.

12.3 Payment Functions

We will now introduce the notion of payment function. A payment function of agent i is a mapping $p_i : T^n \to \mathbb{R}$, where $p_i(t)$ represents the payment of agent i when type profile is $t \in T^n$. Note that $p_i(\cdot)$ can be negative or positive or zero. A positive $p_i(\cdot)$ indicates that the agent is paying money.

In many situations, we want the total payment of agents to be either non-negative (i.e., decision maker does not incur a loss) or to be zero. A payment rule $p = (p_1, \ldots, p_n)$ is **feasible** if $\sum_{i \in N} p_i(t) \geq 0$ for all $t \in T^n$. Similarly, a payment rule $p = (p_1, \ldots, p_n)$ is **balanced** if $\sum_{i \in N} p_i(t) = 0$ for all $t \in T^n$.

12.4 Incentive Compatibility

A **social choice function** is a pair $F = (f, p = (p_1, \ldots, p_n))$, where f is an allocation rule and $p_1, \ldots, p_n$ are payment functions of agents. Hence, the input to a social choice function is the types of the agents. The output is a decision and payments given the reported types. Under a social choice function $F = (f, p)$ the utility of agent $i \in N$ with type t_i when all agents "report" $\hat{t}$ as their types is given by

$$u_i(\hat{t}, t_i, F = (f, p)) = v_i(f(\hat{t}), t_i) - p_i(\hat{t}).$$

This is the quasi-linear utility function, where net utility of the agent is linear in his payment.

The mechanism, as before, is a complicated object. But applying revelation principle, we will focus on direct mechanisms. A direct mechanism is a social choice function $F = (f, p = (p_1, \ldots, p_n))$. A direct mechanism (or associated social choice function) is **strategy-proof or dominant strategy incentive compatible (DSIC)** if for every agent $i \in N$, every $t_{-i} \in T_{-i}$, and every $s_i, t_i \in T_i$, we have

$$v_i(f(t_i, t_{-i}), t_i) - p_i(t_i, t_{-i}) \geq v_i(f(s_i, t_{-i}), t_i) - p_i(s_i, t_{-i}),$$

i.e., truth-telling is a dominant strategy. In this case, we will say that the payment functions $(p_1, \ldots, p_n)$ **implement** the allocation rule f (in dominant strategies) or, simply, f is implementable. Sometimes, we will also say that $(p_1, \ldots, p_n)$ makes f DSIC.

The underlying idea is that if the mechanism designer had perfect information about the types of agents, then he would have liked to implement f (this is sometimes referred to as the *first-best* decision). However, since he does not have the type information, he will like to implement f using payment functions.

12.5 An Example

Consider an example with two agents $N = \{1, 2\}$ and two possible types for each agent $T_1 = T_2 = \{t^H, t^L\}$. Let $f : T_1 \times T_2 \to A$ be an allocation rule, where A is the set of alternatives. In order that f is implementable, we must find payment functions p_1 and p_2 such that the following conditions hold. For every type $t_2 \in T_2$ of agent 2, agent 1 must satisfy

$$\begin{aligned} v_1(f(t^H, t_2), t^H) - p_1(t^H, t_2) &\geq v_1(f(t^L, t^2), t^H) - p_1(t^L, t_2), \\ v_1(f(t^L, t_2), t^L) - p_1(t^L, t_2) &\geq v_1(f(t^H, t^2), t^L) - p_1(t^H, t_2). \end{aligned}$$

Similarly, for every type $t_1 \in T_2$ of agent 1, agent 2 must satisfy

$$v_2(f(t_1, t^H), t^H) - p_2(t_1, t^H) \geq v_2(f(t^1, t^L), t^H) - p_2(t_1, t^L),$$
$$v_2(f(t_1, t^L), t^L) - p_2(t_1, t^L) \geq v_2(f(t^1, t^H), t^L) - p_2(t_1, t^H).$$

Here, we can treat p_1 and p_2 as variables. The existence of a solution to these linear inequalities guarantee f to be implementable.

12.6 Two Properties of Payments

Suppose f is an implementable allocation rule. Then, there exists payment functions $p \equiv (p_1, \ldots, p_n)$ such that $(f, p \equiv (p_1, \ldots, p_n))$ is strategy-proof. This means for every agent $i \in N$ and every t_{-i}, we must have

$$v_i(f(t_i, t_{-i}), t_i) - p_i(t_i, t_{-i}) \geq v_i(f(s_i, t_{-i}), t_i) - p_i(s_i, t_{-i}) \qquad \forall\ s_i, t_i \in T_i.$$

Using p, we define another set of payment functions. For every agent $i \in N$, we choose an arbitrary function $h_i : T_{-i} \to \mathbb{R}$. So, $h_i(t_{-i})$ assigns a real number to every type profile t_{-i} of other agents. Now, define the new payment function q_i of agent i as

$$q_i(t_i, t_{-i}) = p_i(t_i, t_{-i}) + h_i(t_{-i}). \tag{3}$$

We will argue the following.

Lemma **8** *If $(f, p \equiv (p_1, \ldots, p_n))$ is strategy-proof, then $(f, q \equiv (q_1, \ldots, q_n))$ is strategy-proof, where q is defined as in Equation 3.*

Proof: Fix agent i and type profile of other agents at t_{-i}. To show (f, q) is strategy-proof, note that for any pair of types $t_i, s_i \in T_i$, we have

$$\begin{aligned} v_i(f(t_i, t_{-i}), t_i) - q_i(t_i, t_{-i}) &= v_i(f(t_i, t_{-i}), t_i) - p_i(t_i, t_{-i}) - h_i(t_{-i}) \\ &\geq v_i(f(s_i, t_{-i}), t_i) - p_i(s_i, t_{-i}) - h_i(t_{-i}) \\ &= v_i(f(s_i, t_{-i}), t_i) - q_i(s_i, t_{-i}), \end{aligned}$$

where the inequality followed from the fact that (f, p) is strategy-proof. ■

This shows that if we find one set of payment functions which makes f DSIC, then we can find an infinite set of payment functions which makes f DSIC. Moreover, these payments

differ by a constant for every $i \in N$ and for every t_{-i}. In particular, the payments p and q defined above satisfy the property that for every $i \in N$ and for every t_{-i},

$$p_i(t_i, t_{-i}) - q_i(t_i, t_{-i}) = p_i(s_i, t_{-i}) - q_i(s_i, t_{-i}) = h_i(t_{-i}) \qquad \forall\ s_i, t_i \in T_i.$$

We can ask the converse question. When is it that any two payments which make f DISC differ by a constant? We will answer this question later.

The other property that we discuss of payments is the fact that they depend only on allocations. Let (f, p) be strategy-proof. Consider an agent $i \in N$ and a type profile t_{-i}. Let s_i and t_i be two types of agent i such that $f(s_i, t_{-i}) = f(t_i, t_{-i}) = a$. Then, the incentive constraints give us the following.

$$v_i(a, t_i) - p_i(t_i, t_{-i}) \geq v_i(a, t_i) - p_i(s_i, t_{-i})$$
$$v_i(a, s_i) - p_i(s_i, t_{-i}) \geq v_i(a, s_i) - p_i(t_i, t_{-i}).$$

This shows that $p_i(s_i, t_{-i}) = p_i(t_i, t_{-i})$. Hence, for any pair of types $s_i, t_i \in T_i$, $f(s_i, t_{-i}) = f(t_i, t_{-i})$ implies that $p_i(s_i, t_{-i}) = p_i(t_i, t_{-i})$. So, payment is a function of types of other agents and the allocation chosen.

12.7 Efficient Allocation Rule is Implementable

We discussed the efficient allocation rule earlier. Here, we show that there is a large class of payment functions that can implement the efficient allocation rule. First, we show that the efficient allocation rule is Pareto efficient.

Definition **19** *An allocation rule f is* **Pareto efficient** *at a type profile t if for every payment vector $(p_1, \ldots, p_n)$, there exists no alternative $b \neq f(t)$ and payment vector $(p'_1, \ldots, p'_n)$ with $\sum_{i \in N} p'_i = \sum_{i \in N} p_i$, such that $v_i(f(t), t_i) - p_i \leq v_i(b, t_i) - p'_i$ for all $i \in N$ with strict inequality holding for at least one $i \in N$. An allocation rule f is Pareto efficient if it Pareto efficient at every type profile t.*

We argue that the efficient allocation rule is Pareto efficient. [10]

Lemma **9** *An allocation rule is Pareto efficient if and only if it is efficient.*

[10]The notion of Pareto efficiency that we use here is a cardinal notion. Earlier, we used Pareto efficiency in the models without transfers, and that was an ordinal notion of Pareto efficiency.

Proof: Consider a profile t and let the outcome according to the efficient allocation rule be a. Suppose alternative b is such that

$$\sum_{i \in N} v_i(a, t_i) > \sum_{i \in N} v_i(b, t_i).$$

We will show that choosing b along with payment vector $\hat{p}_1, \ldots, \hat{p}_n)$ is not Pareto optimal. So, the utility of agent i from this allocation is $v_i(b, t_i) - \hat{p}_i$.

Let

$$\delta = \frac{1}{n}\big[\sum_{i \in N} v_i(a, t_i) - \sum_{i \in N} v_i(b, t_i)\big].$$

Note that $\delta > 0$. Define a new payment of agent i as

$$q_i = v_i(a, t_i) - v_i(b, t_i) + \hat{p}_i - \delta.$$

Notice that $\sum_{i \in N} q_i = \sum_{i \in N} \hat{p}_i$ and $v_i(a, t_i) - q_i = v_i(b, t_i) - \hat{p}_i + \delta > v_i(b, t_i) - \hat{p}_i$. Hence, with the same total payment, we can choose a as the outcome and strictly improve the utility of every agent. So, choosing b is not Pareto optimal. This shows that every Pareto efficient allocation rule must be efficient.

We now show that choosing a is Pareto optimal. Suppose the corresponding payment vector is $(p_1, \ldots, p_n)$. Suppose choosing b at the same total payment Pareto dominates choosing a. Then, it must be that there is some payment $(q_1, \ldots, q_n)$ with $\sum_{i \in N} q_i = \sum_{i \in N} p_i$ and $v_i(b, t_i) - q_i \geq v_i(a, t_i) - p_i$ for all $i \in N$ with strict inequality holding for at least one agent $i \in N$. Then, adding it over all $i \in N$, gives $\sum_{i \in N} v_i(b, t_i) - \sum_{i \in N} q_i > \sum_{i \in N} v_i(a, t_i) - \sum_{i \in N} p_i$. Using the fact that $\sum_{i \in N} q_i = \sum_{i \in N} p_i$, we get $\sum_{i \in N} v_i(b, t_i) > \sum_{i \in N} v_i(a, t_i)$. This contradicts the definition of a. ■

We will now show that the efficient allocation rule is implementable. We know that in case of sale of a single object efficient allocation rule can be implemented by the second-price payment function. A fundamental result in mechanism design is that the efficient allocation rule is always implementable (under private values and quasi-linear utility functions). For this, a family of payment rules are known which makes the efficient allocation rule implementable. This family of payment rules is known as the *Groves* payment rules, and the corresponding direct mechanisms are known as the **Groves mechanisms** (Groves, 1973).

For agent $i \in N$, for every $t_{-i} \in T_{-i}$, the payment in the Groves mechanism is:

$$p_i^g(t_i, t_{-i}) = h_i(t_{-i}) - \sum_{j \neq i} v_j(f^e(t_i, t_{-i}), t_j),$$

where h_i is any function $h_i : T_{-i} \to \mathbb{R}$ and f^e is the efficient allocation rule.

We give an example in the case of single object auction. Let $h_i(t_{-i}) = 0$ for all i and for all t_{-i}. Let there be four buyers with values (types): 10,8,6,4. Then, efficiency requires us to give the object to the first buyer. Now, the total value of buyers other than buyer 1 in the efficient allocation is zero. Hence, the payment of buyer 1 is zero. The total value of buyers other than buyer 2 (or buyer 3 or buyer 4) is the value of the first buyer (10). Hence, all the other buyers are rewarded 10. Thus, this particular choice of h_i functions led to the auction: the highest bidder wins but pays nothing and those who do not win are awarded an amount equal to the highest bid.

THEOREM **13** *Groves mechanisms are strategy-proof.*

Proof: Consider an agent $i \in N$, $s_i, t_i \in T_i$, and $t_{-i} \in T_{-i}$. Then, we have

$$\begin{aligned} v_i(f^e(t_i, t_{-i}), t_i) - p_i^g(t_i, t_{-i}) &= \sum_{j \in N} v_j(f^e(t_i, t_{-i}), t_j) - h_i(t_{-i}) \\ &\geq \sum_{j \in N} v_j(f^e(s_i, t_{-i}), t_j) - h_i(t_{-i}) \\ &= v_i(f^e(s_i, t_{-i}), t_i) - \left[h_i(t_{-i}) - \sum_{j \neq i} v_j(f^e(s_i, t_{-i}), t_j)\right] \\ &= v_i(f^e(s_i, t_{-i}), t_i) - p^g(s_i, t_{-i}), \end{aligned}$$

where the inequality comes from efficiency. Hence, Groves mechanisms are strategy-proof. ■

An implication of this is that efficient allocation rule is implementable using the Groves payment rules. The natural question to ask is whether there are payment rules besides the Groves payment rules which make the efficient allocation rule DSIC. We will study this question formally later. A quick answer is that it depends on the type spaces of agents and the value function. For many reasonable type spaces and value functions, the Groves payment rules are the only payment rules which make the efficient allocation rule DSIC.

13 The Vickrey-Clarke-Groves Mechanism

A particular mechanism in the class of Groves mechanism is intuitive and has many nice properties. It is commonly known as the **pivotal mechanism** or the Vickrey-Clarke-Groves (VCG) mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973). The VCG mechanism is characterized by a unique $h_i(\cdot)$ function. In particular, for every agent $i \in N$ and every

$t_{-i} \in T_{-i}$,

$$h_i(t_{-i}) = \max_{a \in A} \sum_{j \neq i} v_j(a, t_j).$$

This gives the following payment function. For every $i \in N$ and for every $t \in T$, the payment in the VCG mechanism is

$$p_i^{vcg}(t) = \max_{a \in A} \sum_{j \neq i} v_j(a, t_j) - \sum_{j \neq i} v_j(f^e(t), t_j). \tag{4}$$

Note that $p_i^{vcg}(t) \geq 0$ for all $i \in N$ and for all $t \in T^n$. Hence, the payment function in the VCG mechanism is a feasible payment function.

A careful look at Equation 4 shows that the second term on the right hand side is the sum of values of agents other than i in the efficient decision. The first term on the right hand side is the maximum sum of values of agents other than i (note that this corresponds to an efficient decision when agent i is excluded from the economy). Hence, the payment of agent i in Equation 4 is the *externality* agent i inflicts on other agents because of his presence, and this is the amount he *pays*. Thus, every agent pays his externality to other agents in the VCG mechanism.

The payoff of an agent in the VCG mechanism has a nice interpretation too. Denote the payoff of agent i in the VCG mechanism when his true type is t_i and other agents report t_{-i} as $\pi_i^{vcg}(t_i, t_{-i})$. By definition, we have

$$\begin{aligned}
\pi_i^{vcg}(t_i, t_{-i}) &= v_i(f^e(t_i, t_{-i}), t_i) - p_i^{vcg}(t_i, t_{-i}) \\
&= v_i(f^e(t_i, t_{-i}), t_i) - \max_{a \in A} \sum_{j \neq i} v_j(a, t_j) + \sum_{j \neq i} v_j(f^e(t_i, t_{-i}), t_j) \\
&= \max_{a \in A} \sum_{j \in N} v_j(a, t_j) - \max_{a \in A} \sum_{j \neq i} v_j(a, t_j),
\end{aligned}$$

where the last equality comes from the definition of efficiency. The first term is the total value of *all* agents in an efficient allocation rule. The second term is the total value of *all agents except agent* i in an efficient allocation rule of the economy in which agent i is absent. Hence, payoff of agent i in the VCG mechanism is his **marginal contribution** to the economy.

13.1 Illustration of the VCG (Pivotal) Mechanism

Consider the sale of a single object using the VCG mechanism. Fix an agent $i \in N$. Efficiency says that the object must go to the bidder with the highest value. Consider the two possible

	$\emptyset$	$\{1\}$	$\{2\}$	$\{1,2\}$
$v_1(\cdot)$	0	8	6	12
$v_2(\cdot)$	0	9	4	14

Table 16: An Example of VCG Mechanism with Multiple Objects

cases. In one case, bidder i has the highest value. So, when bidder i is present, the sum of values of other bidders is zero (since no other bidder wins the object). But when bidder i is absent, the maximum sum of value of other bidders is the second highest value (this is achieved when the second highest value bidder is awarded the object). Hence, the externality of bidder i is the second-higest value. In the case where bidder $i \in N$ does not have the highest value, his externality is zero. Hence, for the single object case, the VCG mechanism is simple: award the object to the bidder with the highest (bid) value and the winner pays the amount equal to the second highest (bid) value but other bidders pay nothing. This is the well-known second-price auction or the Vickrey auction. By Theorem 13, it is strategy-proof.

Consider the case of choosing a public project. There are three possible projects - an opera house, a park, and a museum. Denote the set of projects as $A = \{a, b, c\}$. The citizens have to choose one of the projects. Suppose there are three citizens, and the values of citizens are given as follows (row vectors are values of citizens and columns have three alternatives, a first, b next, and c last column):

$$\begin{bmatrix} 5 & 7 & 3 \\ 10 & 4 & 6 \\ 3 & 8 & 8 \end{bmatrix}$$

It is clear that it is efficient to choose alternative b. To find the payment of agent 1 according to the VCG mechanism, we find its externality on other agents. Without agent 1, agents 2 and 3 can get a maximum total value of 14 (on project c). When agent 1 is included, their total value is 12. So, the externality of agent 1 is 2, and hence, its VCG payment is 2. Similarly, the VCG payments of agents 2 and 3 are respectively 0 and 4.

We illustrate the VCG mechanism for the sale of multiple objects by an example. Consider the sale of two objects, with values of two agents on bundles of goods given in Table 16. The efficient allocation in this example is to give bidder 1 object 2 and bidder 2 object 1 (this generates a total value of $6 + 9 = 15$, which is higher than any other allocation). Let us calculate the externality of bidder 1. The total value of bidders other than bidder 1, i.e. bidder 2, in the efficient allocation is 9. When bidder 1 is removed, bidder 2 can get a maximum value of 14 (when he gets both the objects). Hence, externality of bidder 1 is $14 - 9 = 5$. Similarly, we can compute the externality of bidder 2 as $12 - 6 = 6$. Hence, the

	$\emptyset$	$\{1\}$	$\{2\}$
$v_1(\cdot)$	0	5	3
$v_2(\cdot)$	0	3	4
$v_3(\cdot)$	0	2	2

Table 17: An Example of VCG Mechanism with Multiple Objects

payments of bidders 1 and 2 are 5 and 6 respectively.

Another simpler combinatorial auction setting is when agents or bidders are interested (or can be allocated) in at most one object - this is the case in job markets or housing markets. Then, every bidder has a value for every object but wants at most one object. Consider an example with three agents and two objects. The valuations are given in Table 17. The total value of agents in the efficient allocation is $5+4=9$ (agent 1 gets object 1 and agent 2 gets object 2, but agent 3 gets nothing). Agents 2 and 3 get a total value of $4+0=4$ in this efficient allocation. When we maximize over agents 2 and 3 only, the maximum total value of agents 2 and 3 is $6=4+2$ (agent 2 gets object 2 and agent 3 gets object 1). Hence, externality of agent 1 on others is $6-4=2$. Hence, VCG payment of agent 1 is 2. Similarly, one can compute the VCG payment of agent 2 to be 2.

13.2 The VCG Mechanism in the Combinatorial Auctions

We have already shown that the VCG mechanism has several interesting properties: (a) it is dominant strategy incentive compatible, (b) the allocation rule is efficient, and (c) payments are non-negative, and hence, feasible. We discuss below a specific model and show that stronger properties than these are also true in this model.

The particular model we discuss is the combinatorial auction problem. We now describe the formal model. There is a set of objects $M=\{1,\ldots,m\}$. The set of *bundles* is denoted by $\Omega=\{S:S\subseteq M\}$. The type of an agent $i\in N$ is a vector $t_i\in\mathbb{R}_+^{|\Omega|}$. Hence, $T_1=\ldots=T_n=\mathbb{R}_+^{|\Omega|}$. Here, $t_i(S)$ denotes the value of agent (bidder) i on bundle S. An allocation in this case is a partitioning of the set of objects: $X=(X_0,X_1,\ldots,X_n)$, where $X_i\cap X_j=\emptyset$ and $\cup_{i=0}^n X_i=M$. Here, X_0 is the unallocated set of objects and X_i $(i\neq 0)$ is the bundle allocated to agent i, where X_i can be empty set also. It is natural to assume $t_i(\emptyset)=0$ for all t_i and for all i.

Let f^e be the efficient allocation rule. Another crucial feature of the combinatorial auction setting is it is *externality free*. Suppose $f^e(t)=X$. Then $v_i(X,t_i)=t_i(X_i)$, i.e., utility of agent i depends on the bundle allocated to agent i only, but not on the bundles allocated to

other agents.

The first property of the VCG mechanism we note in this setting is that the *losers* pay zero amount. Suppose i is a *loser* (i.e., gets empty bundle in efficient allocation) when the type profile is $t = (t_1, \ldots, t_n)$. Let $f^e(t) = X$. By assumption, $v_i(X_i, t_i) = t_i(\emptyset) = 0$. Let $Y \in \arg\max_a \sum_{j \neq i} v_j(a, t_j)$. We need to show that $p_i^{vcg}(t_i, t_{-i}) = 0$. Since the VCG mechanism is feasible, we know that $p_i^{vcg}(t_i, t_{-i}) \geq 0$. Now,

$$\begin{aligned} p_i^{vcg}(t_i, t_{-i}) &= \max_{a \in A} \sum_{j \neq i} v_j(a, t_j) - \sum_{j \neq i} v_j(f^e(t_i, t_{-i}), t_j) \\ &= \sum_{j \neq i} t_j(Y_j) - \sum_{j \neq i} t_j(X_j) \\ &\leq \sum_{j \in N} t_j(Y_j) - \sum_{j \in N} t_j(X_j) \\ &\leq 0, \end{aligned}$$

where the first inequality followed from the facts that $t_i(Y_i) \geq 0$ and $t_i(X_i) = 0$, and the second inequality followed from the efficiency of X. Hence, $p_i^{vcg}(t_i, t_{-i}) = 0$.

An important property of a mechanism is **individual rationality** or **voluntary participation**. Suppose by not participating in a mechanism an agent gets zero payoff. Then the mechanism must give non-negative payoff to the agent in every state of the world (i.e., in every type profile of agents). The VCG mechanism in the combinatorial auction setting satisfies individual rationality. Consider a type profile $t = (t_1, \ldots, t_n)$ and an agent $i \in N$. Let $Y \in \arg\max_a \sum_{j \neq i} v_j(a, t_j)$ and $X \in \arg\max_a \sum_{j \in N} v_j(a, t_j)$. Now,

$$\begin{aligned} \pi_i^{vcg}(t) &= \max_a \sum_{j \in N} v_j(a, t_j) - \max_a \sum_{j \neq i} v_j(a, t_j) \\ &= \sum_{j \in N} t_j(X_j) - \sum_{j \neq i} t_j(Y_j) \\ &\geq \sum_{j \in N} t_j(X_j) - \sum_{j \in N} t_j(Y_j) \\ &\geq 0, \end{aligned}$$

where the first inequality followed from the fact that $t_j(Y_j) \geq 0$ and the second inequality followed from efficiency of X. Hence, $\pi_i^{vcg}(t) \geq 0$, i.e., the VCG mechanism is individual rational.

13.3 The Sponsored Search Auctions

Google sells advertisement slots to advertisers via auctions. The auctions are run for **every** search phrase. Fix a particular search phrase, say, "hotels in New Delhi". Once this phrase is searched on Google, bidders (computer programmed agents of different companies) participate in this auction. An advertisement that can appear along side a search page is called a **slot**. For every search phrase, there is a fixed number of slots available and fixed number of bidders interested. Suppose there are m slots and n bidders for the phrase "hotels in New Delhi". Assume $n > m$. The type of each bidder is a single number - θ_i for bidder i. Type of an agent represents the value that agent derives when his advertisement is **clicked**. Every slot has a probability of getting clicked. This is called the **clickthrough rate (CTR)**. The CTR of slot i is α_i. The CTR vector $\alpha = (\alpha_1, \ldots, \alpha_m)$ is known to everyone. The slots are naturally ordered top to bottom, and assume that, let $\alpha_1 > \alpha_2 > \ldots > \alpha_m$.

An alternative in this model represents an assignment of agents to slots (with some agents not receiving any slot). Let A be the set of all alternatives. An alternative $a \in A$ can be described by a n dimensional vector integers in $\{0, 1, \ldots, m\}$, where a_i indicates the slot to which agent i is assigned, and $a_i = 0$ means agent i is not assigned to any slot. The value function of agent i is his expected value $v_i(a, \theta_i) = \theta_i \alpha_{a_i}$, where $\alpha_0 = 0$.

Suppose $n = 4$ and $m = 3$. Let $\theta_1 = 10, \theta_2 = 8, \theta_3 = 6, \theta_4 = 5$. Let $\alpha_1 = 0.8, \alpha_2 = 0.6, \alpha_3 = 0.5$. In efficiency, the slots should go to agents with top 3 θ-values, who are agents $1, 2$, and 3.

The total value obtained in the efficient allocation is $10(0.8) + 8(0.6) + 6(0.5) = 15.8$. So, agents other than agent 1 get a total value of $8(0.6) + 6(0.5) = 7.8$. If agent 1 was not there, then the total value obtained in the efficient allocation is $8(0.8) + 6(0.6) + 5(0.5) = 12.5$. Hence, his externality is $12.5 - 7.8 = 4.7$, and his VCG payment is thus 4.7. Similarly, VCG payments of agents 2 and 3 are 3.1 and 2.5 respectively.

Generally, agents with top m θ values get the top m slots with ith ($i \leq m$) highest θ value agent getting the ith slot. Without loss of generality, assume that $\theta_1 \geq \theta_2 \geq \ldots \theta_n$. In efficiency, agents 1 to m get a slot. In particular, agent j ($j \leq m$) gets slot j with clickthrough rate α_j. Any agent j pays zero if he is not allocated a slot, i.e., $j > m$. For any agent $j \leq m$, we need to compute his externality. Note that the total value of agents other than agent j in an efficient allocation is

$$\sum_{i=1}^{j-1} \theta_i \alpha_i + \sum_{i=j+1}^{m} \theta_i \alpha_i.$$

If agent j is removed, then the total value of agents other than agent j in an efficient

allocation is

$$\sum_{i=1}^{j-1} \theta_i \alpha_i + \sum_{i=j+1}^{m+1} \theta_i \alpha_{i-1}.$$

So, the externality of agent j is

$$\theta_{m+1}\alpha_m + \sum_{i=j+1}^{m} \theta_i(\alpha_{i-1} - \alpha_i),$$

where we assume that the summation term for $j = m$ is zero.

Google uses something called a **Generalized Second Price (GSP)** auction: (a) agents with top m θ_i values are given the slots with highest agent getting the top slot (i.e., slot with highest CTR), second highest agent getting the next top slot, and so on, (b) if an agent wins slot k with CTR α_k, he pays $\theta_{m+1}\alpha_k$ (where θ_{m+1} is the highest losing type).

In the previous example, agent 1 will pay $5(0.8) = 4$ in the GSP. This is clearly different from what he should pay in the VCG mechanism. In the example above, fix the bids of agents other than agent 2 at (agent 1: 10, agent 3: 6, agent 4: 5). Now, let agent 2 not bid truthfully, and bid $10 + \epsilon$ ($\epsilon > 0$) to become the highest bidder. So, he gets the top slot with clickthrough rate 0.8. So, his value is now $8 \times 0.8 = 6.4$ (remember, his true type is $\theta_2 = 8$). He pays $5 \times 0.8 = 4$. So, his net utility is 2.4. If he is truthful he pays $5 \times 0.6 = 3$, and gets a value of $8 \times 0.6 = 4.8$. So, his net utility of being truthful is 1.8. So, deviation is profitable, and truth-telling is not a dominant strategy.

14 Affine Maximizer Allocation Rules are Implementable

As discussed earlier, an affine maximizer allocation rule is characterized by a vector of non-negative weights $\lambda \equiv (\lambda_1, \ldots, \lambda_n)$, not all equal to zero, for agents and a mapping $\kappa : A \to \mathbb{R}$. If $\lambda_i = \lambda_j$ for all $i, j \in N$ and $\kappa(a) = 0$ for all $a \in A$, we recover the efficient allocation rule. When $\lambda_i = 1$ for some $i \in N$ and $\lambda_j = 0$ for all $j \neq i$, and $\kappa(a) = 0$ for all $a \in A$, we get the dictatorial allocation rule. Thus, the affine maximizer is a general class of allocation rules. We show that there exists payment rules which implements the affine maximizer allocation rule. For this we only consider a particular class of affine maximizers.

Definition **20** *An affine maximizer allocation rule f^a with weights $\lambda_1, \ldots, \lambda_n$ and $\kappa : A \to \mathbb{R}$ satisfies* **independence of irrelevant agents (IIA)** *if for all $i \in N$ with $\lambda_i = 0$, we have that for all t_{-i} and for all s_i, t_i, $f(s_i, t_{-i}) = f(t_i, t_{-i})$.*

The IIA property is a consistent tie-breaking requirement. For instance, consider the dictatorship allocation rule with two agents $\{1,2\}$. Suppose agent 1 is the dictator: $\lambda_1 = 1, \lambda_2 = 0$ and suppose there are three alternatives $\{a,b,c\}$. Since the allocation rule is a dictatorship, $\kappa(a) = \kappa(b) = \kappa(c) = 0$. The type of each agent is a vector in $\mathbb{R}^3$ describing the value for each alternative. For instance $t_1 = (5,5,3)$ means, agent 1 has value 5 for alternatives a and b and value 3 for alternative c. Since values on alternatives can be the same, we can break the ties in this dictatorship by considering values of agent 2. In particular, if there are more than one alternatives that maximize the value for agent 1, then we choose an alternative that is the worst for agent 2. For instance, if $t_1 = (5,5,3)$ and $t_2 = (4,3,2)$, then $f(t_1,t_2) = b$ (since $t_2(b) = 3 < t_2(a) = 4$). But then, consider $t_2' = (3,4,2)$ and note that $f(t_1,t_2') = a$. This is a violation of IIA.

Allocation rules violating IIA may not be implementable (i.e., there may not exist payment rules that make the resulting mechanism strategy-proof). However, we show that every IIA affine maximizer is implementable. Fix an IIA affine maximizer allocation rule f^a, characterized by λ and κ. We generalize Groves payments for this allocation rule.

For agent $i \in N$, for every $t_{-i} \in T_{-i}$, the payment in the *generalized* Groves mechanism is:

$$p_i^{gg}(t_i,t_{-i}) = \begin{cases} h_i(t_{-i}) - \frac{1}{\lambda_i}\big[\sum_{j\neq i}\lambda_j v_j(f^a(t_i,t_{-i}),t_j) + \kappa(f^a(t_i,t_{-i}))\big] & \text{if } \lambda_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

where h_i is any function $h_i : T_{-i} \to \mathbb{R}$ and f^a is the IIA affine maximizer allocation rule.

THEOREM **14** *An IIA affine maximizer allocation rule is implementable using the generalized Groves mechanism.*

Proof: Consider an agent $i \in N$, $s_i, t_i \in T_i$, and $t_{-i} \in T_{-i}$. Suppose $\lambda_i > 0$. Then, we have

$$\begin{aligned} v_i(f^a(t_i,t_{-i}),t_i) - p_i^{gg}(t_i,t_{-i}) &= \frac{1}{\lambda_i}\big[\sum_{j\in N}\lambda_j v_j(f^a(t_i,t_{-i}),t_j) - \kappa(f^a(t_i,t_{-i}))\big] - h_i(t_{-i}) \\ &\geq \frac{1}{\lambda_i}\big[\sum_{j\in N}\lambda_j v_j(f^a(s_i,t_{-i}),t_j) - \kappa(f^a(s_i,t_{-i}))\big] - h_i(t_{-i}) \\ &= v_i(f^a(s_i,t_{-i}),t_i) - h_i(t_{-i}) + \frac{1}{\lambda_i}\big[\sum_{j\neq i}\lambda_j v_j(f^a(s_i,t_{-i}),t_j) + \kappa(f^a(s_i,t_{-i}))\big] \\ &= v_i(f^a(s_i,t_{-i}),t_i) - p^{gg}(s_i,t_{-i}), \end{aligned}$$

where the inequality comes from the definition of affine maximization. If $\lambda_i = 0$, then $f^a(t_i,t_{-i}) = f^a(s_i,t_{-i})$ for all $s_i, t_i \in T_i$ (by IIA). Also $p_i^{gg}(t_i,t_{-i}) = p_i^{gg}(s_i,t_{-i}) = 0$ for all

$s_i, t_i \in T_i$. Hence, $v_i(f^a(t_i, t_{-i}), t_i) - p_i^{gg}(t_i, t_{-i}) = v_i(f^a(s_i, t^{-i}), t_i) - p_i^{gg}(s_i, t_{-i})$. So, the generalized Groves payment rule implements the affine maximizer allocation rule. ■

14.1 Public Good Provision

The public good provision problem is a classic problem. There are two alternatives: a_1 is the alternative to provide the public good and a_0 is the alternative of not providing the public good. The value from a_0 is zero to all the agents. Agents derive value from a_1 which is private information. Denote the value of agent i for a_1 as θ_i. There is a cost of C providing the public good.

The "first-best" allocation rule in this case is to provide the public good when the sum of values of agents is greater than or equal to C. This can be written as an affine maximizer rule. Choose $\kappa(a_0) = 0, \kappa(a_1) = -C$ and $\lambda_i = 1$ for all $i \in N$, where N is the set of agents.

The pivotal mechanism corresponding to this allocation rule is the first one that Clarke called the pivotal mechanism. An agent i is **pivotal** if his inclusion in the decision process changes the decision for the other $N \setminus \{i\}$ agents. In particular, if agents in $N \setminus \{i\}$ chose not to be provided the public good using the first-best rule, and when agent i was added, agents in N chose to get the public good using the first-best rule. Here, agent i is pivotal. Note that if agents in $N \setminus \{i\}$ chose to get the public good using the first-best rule, and when agent i is added, agents in N will always choose to get the public good using the first-best rule. Hence, agent i cannot be pivotal here.

The pivotal mechanism in this problem states that an agent i pays zero if he is not pivotal and pays an amount equal to his externality if he is pivotal. The externality can be computed easily. Note that at a type profile $\theta \equiv (\theta_1, \ldots, \theta_n)$, if the public good is not provided, then it will not be provided without any agent. Hence, no agent is pivotal and payment of all the agents are zero. But if the public good is provided at θ and agent i is pivotal, then removing agent i changes the decision to not provide the public good. This implies that $\sum_{j \neq i} \theta_j < C$. Hence, without agent i, the total utility to all the agents in $N \setminus \{i\}$ is zero. Once, agent i arrives, their total utility is $\sum_{j \neq i} \theta_j - C$. Hence, his payment is $C - \sum_{j \neq i} \theta_j$.

Now, it is easy to verify that this corresponds to the payment we described in the previous section, where we take $h_i(\theta_{-i})$ to be the maximum sum of values without agent i in the first-best allocation rule.

14.2 Restricted and Unrestricted Type Spaces

Consider a simple model where $t_i \in \mathbb{R}^{|A|}$, where A is finite and $v_i(a, t_i) = t_i(a)$ for all $i \in N$. So, the type space of agent i is now $T_i \subseteq \mathbb{R}^{|A|}$. We say type space T_i of agent i is **unrestricted** if $T_i = \mathbb{R}^{|A|}$. So, all possible vectors in $\mathbb{R}^{|A|}$ is likely to be the type of agent i if its type space is unrestricted. Notice that it is an extremely restrictive assumption. We give two examples where unrestricted type space assumption is **not** natural.

- Choosing a public project. Suppose we are given a set of public projects to choose from. Each of the possible public projects (alternatives) is a "good" and not a "bad". In that case, it is natural to assume that the value of an agent for any alternative is non-negative. Further, it is reasonable to assume that the value is bounded. Hence, $T_i \subseteq \mathbb{R}_+^{|A|}$ for every agent $i \in N$. So, unrestricted type space is not a natural assumption here.

- Auction settings. Consider the sale of a single object. The alternatives in this case are $A = \{a_0, a_1, \ldots, a_n\}$, where a_0 denote the alternative that the object is not sold to any agent and a_i with $i > 0$ denotes the alternative that the object is sold to agent i. Notice here that agent i has **zero** value for all the alternatives except alternative a_i. Hence, the unrestricted type space assumption is not valid here.

Are there problems where the unrestricted type space assumption is natural? Suppose the alternatives are such that it can be a "good" or "bad" for the agents, and any possible value is plausible. If we accept the assumption of unrestricted type spaces, then the following is an important theorem. We skip the long proof.

Theorem **15 (Roberts' theorem)** *Suppose A is finite and $|A| \geq 3$. Further, type space of every agent is unrestricted. Then, if an onto allocation rule is implementable, then it is an affine maximizer.*

We have already shown that IIA affine maximizers are implementable by constructing generalized Groves payments which make them DSIC. Roberts' theorem shows that these are almost the entire class. The assumptions in the theorem are crucial. If we relax unrestricted type spaces or let $|A| = 2$ or allow randomization, then the set of DSIC allocation rules are larger.

It is natural to ask why restricted type spaces allow for larger class of allocation rules to be DSIC. The answer is very intuitive. Remember that the type space is something that the mechanism designer knows (about the range of private types of agents). If the type space is

restricted then the mechanism designer has more precise information about the types of the agents. So, there is *less opportunity* for an agent to lie. Given an allocation rule f if we have two type spaces T and $\bar{T}$ with $T \subsetneq \bar{T}$, then it is possible that f is DSIC in T but not in $\bar{T}$ since $\bar{T}$ allows an agent a larger set of type vectors where it can deviate. In other words, the set of constraints in the DSIC definition is larger for $\bar{T}$ then for T. So, finding payments to make f DSIC is difficult for larger type spaces but easier for smaller type spaces. Hence, the set of DSIC allocation rules becomes larger as we shrink the type space of agents.

15 Single Object Auction

In the single object auction case, the type set of an agent is one dimensional, i.e., $T_i \subseteq \mathbb{R}^1$ for all $i \in N$. This reflects the value of an agent if he wins the object. An allocation gives a probability of winning the object. Let A denote the set of all **deterministic** allocations (i.e., allocations in which the object either goes to a single agent or is unallocated). Let ΔA denote the set of all probability distributions over A. An allocation rule is now a mapping $f : T^n \to \Delta A$.

Given an allocation, $a \in \Delta A$, we denote by a_i the allocation probability of agent i. It is standard to have $v_i(a, s_i) = a_i \times s_i$ for all $a \in \Delta A$ and $s_i \in T_i$ for all $i \in N$. Such a form of v_i is called a **product form**.

For an allocation rule f, we denote $f_i(t_i, t_{-i})$ as the probability of winning the object of agent i when he reports t_i and others report t_{-i}.

15.1 The Vickrey Auction

Before analyzing a single object sale mechanism, we first take a look at the Vickrey auction. Consider the Vickrey auction and an agent i. Denote the highest valuation among agents in $N \setminus \{i\}$ as $v^{(2)}$. Suppose the valuation of agent i is v_i. Then, according to the rules of the Vickrey auction, agent i does not win the object if $v_i < v^{(2)}$ and wins the object if $v_i > v^{(2)}$. Further, his net utility from the Vickrey auction is zero if he does not win the object. If he wins the object, then his net utility is $v_i - v^{(2)}$, i.e., increases linearly with v_i.

If we draw the net utility as a function of v_i, it will look something like in Figure 10. Notice that this function is *convex* and its derivative is zero if $v_i < v^{(2)}$ and 1 if $v_i > v^{(2)}$. This function is not differentiable at $v_i = v^{(2)}$. Hence, the derivative of the net utility function (wherever it exists) coincides with the probability of winning the object - see Figure 10. Since a convex function is differentiable *almost everywhere*, this fact is true almost everywhere.

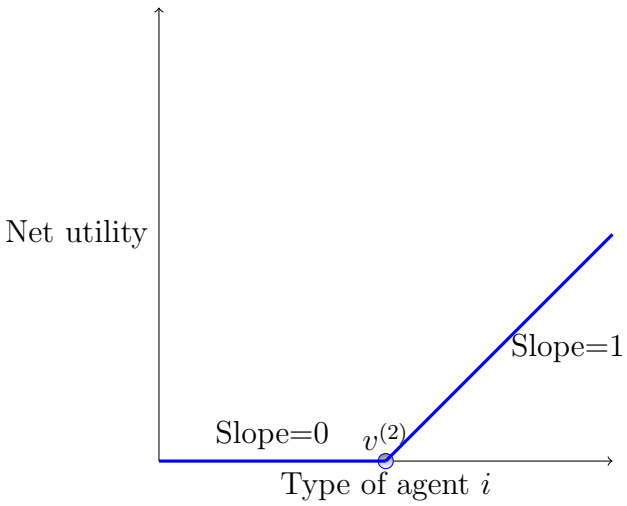

Figure 10: Net utility as a function of type of agent i

These observations hold in general, and it is true for *any* dominant strategy incentive compatible mechanism. To show this, we first record some elementary facts from convex analysis.

15.2 Facts from Convex Analysis

We will state some basic facts about convex functions. We will only be interested in functions of the form $g : I \to \mathbb{R}$, where $I \subseteq \mathbb{R}$ is an interval.

Definition **21** *A function $g : I \to \mathbb{R}$ is* **convex** *if for every $x, y \in I$ and for every $\lambda \in (0, 1)$, we have*

$$\lambda g(x) + (1 - \lambda)g(y) \geq g(\lambda x + (1 - \lambda)y).$$

Convex functions are continuous in the interior of its domain. So, if $g : I \to \mathbb{R}$ is convex, then g is continuous in the interior of I. Further, g is differentiable *almost everywhere* in I. More formally, there is a subset of $I' \subseteq I$ such that I' is dense in I, $I \setminus I'$ has measure zero [11], and g is differentiable at every point in I'. If g is differentiable at $x \in I$, we denote the derivative of g at x as $g'(x)$. The following notion extends the idea of a derivative.

Definition **22** *For any $x \in I$, x^* is a* **subgradient** *of the function $g : I \to \mathbb{R}$ if*

$$g(z) \geq g(x) + x^*(z - x) \qquad \forall\ z \in I.$$

[11] This means that $I \setminus I'$ is countable.

LEMMA **10** *Suppose $g : I \to \mathbb{R}$ is a convex function. Suppose x is in the interior of I and g is differentiable at x, then $g'(x)$ is the unique subgradient of g at x.*

Proof: Consider any $x \in I$ in the interior of I such that the convex function $g : I \to \mathbb{R}_+$ is differentiable at x. Now, pick any $z \in I$. Assume that $z > x$ (a similar proof works if $z < x$). For any $(z - x) \geq h > 0$, we note that $x + h = \frac{h}{(z-x)}z + (1 - \frac{h}{(z-x)})x$. As a result, convexity of g ensures that

$$\frac{h}{(z-x)}g(z) + (1 - \frac{h}{(z-x)})g(x) \geq g(x+h).$$

Simplifying, we get

$$\frac{g(z) - g(x)}{(z-x)} \geq \frac{g(x+h) - g(x)}{h}.$$

Since this is true for any $h > 0$, it is also true that

$$\frac{g(z) - g(x)}{(z-x)} \geq \lim_{h \to 0} \frac{g(x+h) - g(x)}{h} = g'(x).$$

Hence, $g'(x)$ is a subgradient of g at x. This also shows that there is at least one subgradient of g at x.

To show uniqueness, suppose there is another subgradient $x^* \neq g'(x)$ at x. Suppose $x^* > g'(x)$. Then, for all $h > 0$, we know that

$$\frac{g(x+h) - g(x)}{h} \geq x^* > g'(x).$$

But since this is true for all $h > 0$, we have that

$$g'(x) = \lim_{h \to 0} \frac{g(x+h) - g(x)}{h} \geq x^* > g'(x),$$

which is a contradiction.

Suppose $x^* < g'(x)$. Then, for all $h > 0$, we know that

$$g(x-h) \geq g(x) - x^* h.$$

Equivalently,

$$\frac{g(x) - g(x-h)}{h} \leq x^*.$$

Since this is true for all $h > 0$, we have that

$$g'(x) = \lim_{h \to 0} \frac{g(x) - g(x-h)}{h} \leq x^*.$$

This is a contradiction. ■

Lemma 10 extends in the following natural way.

LEMMA **11** *Suppose $g : I \to \mathbb{R}$ is a convex function. Then for every $x \in I$, the subgradient of g at x exists.*

We skip the proof of Lemma 11. Lemma 10 showed that if g is differentiable at x and x is in the interior, then $g'(x)$ is the unique subgradient. For all other points in x (which is a set of measure zero), the set of subgradients can be shown to be a convex set. In particular, if x is an interior point of I where g is not differentiable, then we can define $g'_+(x) = \lim_{z \to x : z \in I', z > x} g'(z)$ and $g'_-(x) = \lim_{z \to x : z \in I', z < x} g'(x-h)$, where I' is the set of points where g is differentiable. These limits exist since the set of points where g is differentiable is dense in I. One can easily show that $g'_+(x) \geq g'_-(x)$. We can then show that the set of subgradients of g at x is $[g'_-(x), g'_+(x)]$.

The set of subgradients of g at a point $x \in I$ will be denoted by $\partial g(x)$. By Lemma 10, $\partial g(x)$ is equal to $\{g'(x)\}$ if $x \in I'$ and by Lemma 11, it is non-empty otherwise The following lemma is crucial.

LEMMA **12** *Suppose $g : I \to \mathbb{R}$ is a convex function. Let $\phi : I \to \mathbb{R}$ such that $\phi(z) \in \partial g(z)$ for all $z \in I$. Then, for all $x, y \in I$ such that $x > y$, we have $\phi(x) \geq \phi(y)$.*

Proof: By definition, $g(x) \geq g(y) + \phi(y)(x - y)$ and $g(y) \geq g(x) + \phi(x)(y - x)$. Adding these two inequalities, we get $(x - y)(\phi(x) - \phi(y)) \geq 0$. Since $x > y$, we get $\phi(x) \geq \phi(y)$. ■

As a corollary to Lemma 12, we get that if g is differentiable at x and y and $x > y$, then we have $g'(x) \geq g'(y)$. This also shows that for any $x \in I$, $g'_+(x) \geq g'_-(x)$.

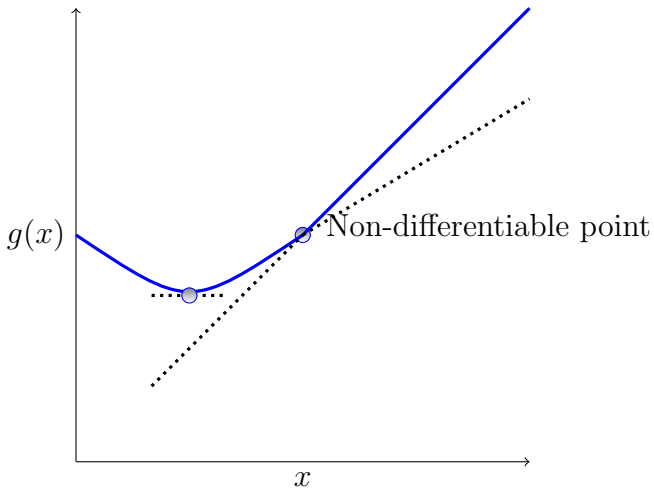

Figure 11: A convex function and its subgradients

Figure 11 illustrates the idea. It shows a convex function g and two points in its domain. The left one is a point where g is differentiable and its unique subgradient is shown in Figure 11. On the other hand, the right one is a point where g is not differentiable. Figure 11 shows the least subgradient and the maximum subgradient at that point. Any selection from that cone will be a suitable subgradient at that point.

If g is differentiable everywhere, then g can be written as the definite integral of its derivative. In particular, if $x, y \in I$, then $g(x) = g(y) + \int_y^x g'(z)dz$. However, this can be extended easily to convex functions since a convex function is differentiable almost everywhere. The following lemma establishes that. We skip its proof.

LEMMA **13** *Let $g : I \to \mathbb{R}$ be a convex function. Then, for any $x, y \in I$,*

$$g(x) = g(y) + \int_y^x \phi(z)dz,$$

where $\phi : I \to \mathbb{R}$ is a map satisfying $\phi(z) \in \partial g(z)$ for all $z \in I$.

15.3 Monotonicity and Revenue Equivalence

We now use the facts from convex analysis to establish a fundamental theorem in single object auction analysis. A crucial property that we will use is the following monotonicity property of allocation rules.

DEFINITION **23** *An allocation rule f is called* **non-decreasing** *if for every agent $i \in N$ and every $t_{-i} \in T_{-i}$ we have $f_i(t_i, t_{-i}) \geq f_i(s_i, t_{-i})$ for all $s_i, t_i \in T_i$ with $s_i < t_i$.*

A non-decreasing allocation rule satisfies a simple property. For every agent and for every report of other agents, the probability of winning the object does not decrease with increase in type of this agent. Figure 12 shows a non-decreasing allocation rule.

This property characterizes the set of implementable allocation rules in this case.

THEOREM **16** *Suppose T_i is an interval $[0, b_i]$ for all $i \in N$ and v is in product form. An allocation rule $f : T^n \to \Delta A$ and a payment rule $(p_1, \ldots, p_n)$ is DSIC if and only if f is non-decreasing and for all $i \in N$, for all $t_{-i} \in T^{n-1}$, and for all $t_i \in T_i$*

$$p_i(t_i, t_{-i}) = p_i(0, t_{-i}) + t_i f_i(t_i, t_{-i}) - \int_0^{t_i} f_i(x_i, t_{-i})dx_i.$$

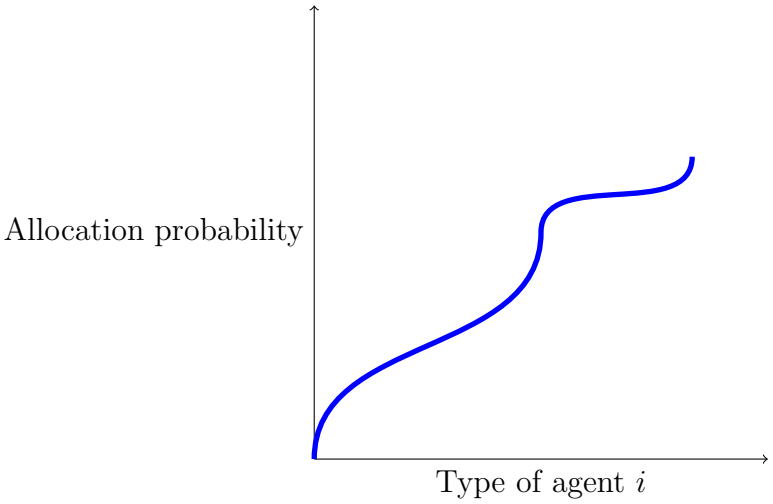

Figure 12: Non-decreasing allocation rule

Proof: Given a mechanism $M \equiv (f, p_1, \ldots, p_n)$, the indirect utility function of agent i from the mechanism M when other agents report t_{-i} is defined as

$$\mathcal{U}_i^M(t_i, t_{-i}) = t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) \quad \forall\ t_i \in T_i.$$

The indirect utility is the net utility of agent i by reporting his true type (given the reports of other agents). Using $\mathcal{U}^M$, we can rewrite the incentive constraints as follows. Mechanism M is dominant strategy incentive compatible if and only if for all $i \in N$ and for all $t_{-i} \in T^{n-1}$, we have

$$\mathcal{U}_i^M(t_i, t_{-i}) \geq \mathcal{U}_i^M(s_i, t_{-i}) + f_i(s_i, t_{-i})(t_i - s_i) \qquad \forall\ s_i, t_i \in T_i.$$

Now, fix an agent $i \in N$ and t_{-i}. Suppose mechanism $M \equiv (f, p_1, \ldots, p_n)$ is DSIC. We do the proof in some steps.

Step 1 - Subgradient. Define $g(t_i) = \mathcal{U}_i^M(t_i, t_{-i})$ for all $t_i \in T_i$ and $\phi(t_i) = f_i(t_i, t_{-i})$. Then, DSIC implies that for all $s_i, t_i \in T_i$, we have

$$g(t_i) \geq g(s_i) + \phi(t_i)(t_i - s_i).$$

Hence, $\phi(t_i)$ is a subgradient of g at t_i.

Step 2 - Convexity of $\mathcal{U}_i^M$. Next, we show that g is convex. To see this, pick $x_i, z_i \in T_i$ and consider $y_i = \lambda x_i + (1-\lambda) z_i$ for some $\lambda \in (0,1)$. Due to DSIC, we know that

$$g(x_i) \geq g(y_i) + (x_i - y_i)\phi(y_i)$$
$$g(z_i) \geq g(y_i) + (z_i - y_i)\phi(y_i)$$

Multiplying the first inequality by λ and the second by $(1-\lambda)$ and adding them together gives

$$\lambda g(x_i) + (1-\lambda) g(z_i) \geq g(y_i).$$

Step 3 - Apply Lemmas 12 and 13. By Lemma 12, g is non-decreasing. By Lemma 13, for any $t_i \in T_i$,

$$g(t_i) = g(0) + \int_0^{t_i} \phi(x_i) dx_i.$$

Substituting for g, we get

$$\mathcal{U}_i^M(t_i, t_{-i}) = \mathcal{U}_i^M(0, t_{-i}) + \int_0^{t_i} f_i(x_i, t_{-i}) dx_i.$$

Substituting for $\mathcal{U}^M$, we get

$$p_i(t_i, t_{-i}) = p_i(0, t_{-i}) + t_i f_i(t_i, t_{-i}) - \int_0^{t_i} f_i(x_i, t_{-i}) dx_i.$$

This proves one direction.

Now, for the converse. If f is non-decreasing and p_i for all i is of the form described, then we have to show that the mechanism is DSIC. To show this, fix, $i \in N$, t_{-i}, and consider s_i, t_i. Now, substituting for p_i, we get

$$\begin{aligned}\big[t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i})\big] - \big[t_i f_i(s_i, t_{-i}) - p_i(s_i, t_{-i})\big] &= (s_i - t_i) f_i(s_i, t_{-i}) - \int_{t_i}^{s_i} f_i(x_i, t_{-i}) dx_i \\ &\geq 0,\end{aligned}$$

where the inequality followed from the fact that f is non-decreasing. ■

An implication of this result is the following. Take two payment functions p and q that make f DSIC. Then, for every $i \in N$ and every t_{-i}, we know that for every $s_i, t_i \in T_i$,

$$p_i(s_i, t_{-i}) - p_i(t_i, t_{-i}) = \big[s_i f_i(s_i, t_{-i}) - \int_0^{s_i} f_i(x_i, t_{-i}) dx_i\big] - \big[t_i f_i(t_i, t_{-i}) - \int_0^{t_i} f_i(x_i, t_{-i}) dx_i\big]$$

and

$$q_i(s_i, t_{-i}) - q_i(t_i, t_{-i}) = \left[s_i f_i(s_i, t_{-i}) - \int_0^{s_i} f_i(x_i, t_{-i})dx_i\right] - \left[t_i f_i(t_i, t_{-i}) - \int_0^{t_i} f_i(x_i, t_{-i})dx_i\right]$$

Hence,

$$p_i(s_i, t_{-i}) - p_i(t_i, t_{-i}) = q_i(s_i, t_{-i}) - q_i(t_i, t_{-i}),$$
$$\text{or } p_i(s_i, t_{-i}) - q_i(s_i, t_{-i}) = p_i(t_i, t_{-i}) - q_i(t_i, t_{-i}).$$

This result is also known as the revenue equivalence result in single object auction.

One important difference between the characterization in Theorem 16 and the characterization of Roberts' theorem or Gibbard-Satterthwaite theorem is worth pointing out. The latter characterizations are very specific about the parameters to be used in the mechanism - Gibbard-Satterthwaite theorem points to dictatorship, which identifies a one parameter (the dictator) mechanism; similarly, Roberts' theorem asks us to design mechanisms by identifying weights for agents and alternatives and then doing a maximization of weighted values. However, the characterization in Theorem 16 is implicit. It only identifies *properties* of a mechanism that is necessary and sufficient for DSIC. It is still useful for verifying if a given mechanism is DSIC or not.

An immediate corollary of Theorem 16 is the following.

COROLLARY **2** *An allocation rule is implementable if and only if it is non-decreasing.*

Proof: Suppose f is an implementable allocation rule. Then, there exists $(p_1, \ldots, p_n)$ such that $(f, p_1, \ldots, p_n)$ is DSIC. One direction of Theorem 16 showed that f must be non-decreasing. For the converse, Theorem 16 identified payment rules that make a non-decreasing allocation rule implementable. ■

The fact that any non-decreasing allocation rule can be implemented in the single object auction rule is insightful. Many allocation rules can be verified if they are DSIC or not by checking if they are non-decreasing. The constant allocation rule is clearly non-decreasing (it is constant in fact). The dicatorial allocation rule is also non-decreasing. The efficient allocation rule is non-decreasing because if you are winning the object by reporting some type, efficiency guarantees that you will continue to win it by reporting a higher type (remember that efficient allocation rule in the single object case awards the object to an agent with the highest type).

Efficient allocation rule with a *reserve price* is the following allocation rule. If types of all agents are below a threshold level r, then the object is not sold, else all agents whose

type is above r are considered and sold to one of these agents who has the highest type. It is clear that this allocation rule is also DSIC since it is non-decreasing. We will encounter this allocation rule again when we study optimal auction design.

Consider an agent $i \in N$ and fix the types of other agents at t_{-i}. Figure 12 shows how agent i's probability of winning the object can change in a DSIC allocation rule. If we restrict attention to DSIC allocation rules which either do not give the object to an agent or gives it to an agent with probability 1, then the shape of the curve depicting probability of winning the object will be a step function. We call such allocation rules **deterministic** allocation rules.

15.4 The Efficient Allocation Rule and the Vickrey Auction

We start off by deriving a deterministic mechanism using Theorem 16. The mechanism we focus is the Vickrey auction that uses the efficient allocation rule. Though the efficient allocation rule may break ties using randomization, we assume that ties are broken deterministically, i.e., each agent gets the object either with probability 1 or 0.

Suppose f is the efficient allocation. We know that the class of Groves payments make f DSIC. Suppose we impose the restriction that $p_i(0, t_{-i}) = 0$ for all $i \in N$ and for all t_{-i}. Note that if t_i is not the highest type in the profile, then $f_i(x_i, t_{-i}) = 0$ for all $x_i \leq t_i$. Hence, by Theorem 16, $p_i(t_i, t_{-i}) = 0$. If t_i is the highest type and t_j is the second highest type in the profile, then $f_i(x_i, t_{-i}) = 0$ for all $x_i \leq t_j$ and $f_i(x_i, t_{-i}) = 1$ for all $t_i \geq x_i > t_j$. So, using Theorem 16, $p_i(t_i, t_{-i}) = t_i - [t_i - t_j] = t_j$. This is indeed the Vickrey auction. The revenue equivalence result says that any other DSIC auction must have payments which differ from the Vickrey auction by the amount a bidder pays at type 0, i.e., $p_i(0, t_{-i})$.

15.5 Deterministic Allocations Rules

Call an allocation rule f **deterministic** (in single object setting) if for all $i \in N$ and every type profile t, we have $f_i(t) \in \{0, 1\}$. The aim of this section is to show the simple nature of payment rules for a deterministic allocation rule to be DSIC. We assume that set of types of agent i is $T_i = [0, b_i]$. Suppose f is a deterministic allocation rule which is DSIC. Hence, it is non-decreasing. For every $i \in N$ and every t_{-i}, the shape of $f_i(\cdot, t_{-i})$ is a step function (as in Figure 13). Now, define,

$$\kappa_i^f(t_{-i}) = \begin{cases} \inf\{t_i \in T_i : f_i(t_i, t_{-i}) = 1\} & \text{if } f_i(t_i, t_{-i}) = 1 \text{ for some } t_i \in T_i \\ 0 & \text{otherwise} \end{cases}$$

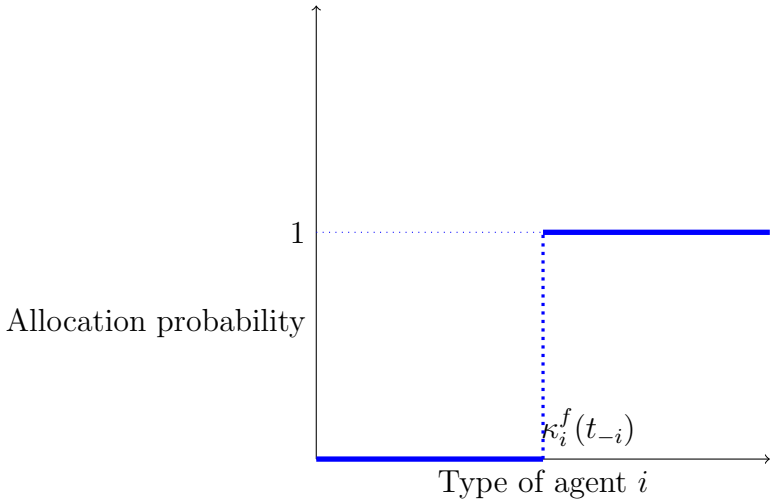

Figure 13: A deterministic implementable allocation rule

If f is DSIC, then it is non-decreasing, which implies that for all $t_i > \kappa_i^f(t_{-i})$, i gets the object and for all $t_i < \kappa_i^f(t_{-i})$, i does not get the object.

Consider a type $t_i \in T_i$. If $f_i(t_i, t_{-i}) = 0$, then using revenue equivalence, we can compute any payment which makes f DSIC as $p_i(t_i, t_{-i}) = p_i(0, t_{-i})$. If $f_i(t_i, t_{-i}) = 1$, then $p_i(t_i, t_{-i}) = p_i(0, t_{-i}) + t_i - [t_i - \kappa_i^f(t_{-i})] = p_i(0, t_{-i}) + \kappa_i^f(t_{-i})$. Hence, if p makes f DSIC, then for all $i \in N$ and for all t

$$p_i(t) = p_i(0, t_{-i}) + \kappa_i^f(t_{-i}).$$

The payments when $p_i(0, t_{-i}) = 0$ has special interpretation. If $f_i(t) = 0$, then agent i pays nothing (losers pay zero). If $f_i(t) = 1$, then agent i pays the minimum amount required to win the object when types of other agents are t_{-i}. If f is the efficient allocation rule, this reduces to the second-price Vickrey auction.

We can also apply this to other allocation rules. Suppose $N = \{1, 2\}$ and the allocations are $A = \{a_0, a_1, a_2\}$, where a_0 is the allocation where the seller keeps the object, a_i $(i \neq 0)$ is the allocation where agent i keeps the object. Given a type profile $t = (t_1, t_2)$, the seller computes, $U(t) = \max(2, t_1^2, t_2^3)$, and allocation is a_0 if $U(t) = 2$, it is a_1 if $U(t) = t_1^2$, and a_2 if $U(t) = t_2^3$. Here, 2 serves as a (pseudo) *reserve price* below which the object is unsold. It is easy to verify that this allocation rule is non-decreasing, and hence DSIC. Now, consider a type profile $t = (t_1, t_2)$. For agent 1, the minimum he needs to bid to win against t_2 is $\sqrt{\max\{2, t_2^3\}}$. Similarly, for agent 2, the minimum he needs to bid to win against t_1 is $(\max\{2, t_1^2\})^{\frac{1}{3}}$. Hence, the following is a payment scheme which makes this allocation rule DSIC. At any type profile $t = (t_1, t_2)$, if none of the agents win the object, they do not pay

anything. If agent 1 wins the object, then he pays $\sqrt{\max\{2, t_2^3\}}$, and if agent 2 wins the object, then he pays $(\max\{2, t_1^2\})^{\frac{1}{3}}$.

15.6 Individual Rationality

We can find out conditions under which a mechanism is individually rational. Notice that the form of individual rationality we use is *ex post* individual rationality.

Lemma **14** *Suppose a mechanism (f, p) is strategy-proof. The mechanism (f, p) is individually rational if and only if for all $i \in N$ and for all t_{-i},*

$$p_i(0, t_{-i}) \leq 0.$$

Further a mechanism (f, p) is individually rational and $p_i(t_i, t_{-i}) \geq 0$ for all $i \in N$ and for all t_{-i} if and only if for all $i \in N$ and for all t_{-i},

$$p_i(0, t_{-i}) = 0.$$

Proof: Suppose (f, p) is individually rational. Then $0 - p_i(0, t_{-i}) \geq 0$ for all $i \in N$ and for all t_{-i}. For the converse, suppose $p_i(0, t_{-i}) \leq 0$ for all $i \in N$ and for all t_{-i}. In that case, $t_i - p_i(t_i, t_{-i}) = t_i - p_i(0, t_{-i}) - t_i f_i(t_i, t_{-i}) + \int_0^{t_i} f_i(x_i, t_{-i}) dx_i \geq 0$.

Individual rationality says $p_i(0, t_{-i}) \leq 0$ and the requirement $p_i(0, t_{-i}) \geq 0$ ensures $p_i(0, t_{-i}) = 0$. For the converse, $p_i(0, t_{-i}) = 0$ ensures individual rationality. ■

Hence, individual rationality along with the requirement that payments are always non-negative pins down $p_i(0, t_{-i}) = 0$ for all $i \in N$ and for all t_{-i}.

16 Optimal Auction Design

This section will describe the design of optimal auction for selling a single indivisible object to a set of bidders (buyers) who have quasi-linear utility functions. The seminal paper in this area is (Myerson, 1981). We present a detailed analysis of this work. Before I describe the formal model, let me describe some popular auction forms used in practice.

16.1 Auctions for a Single Indivisible Object

A single indivisible object is for sale. Let us consider four bidders (agents or buyers) who are interested in buying the object. Let the valuations of the bidders be $10, 8, 6$, and 4

respectively. Let us discuss some commonly used auction formats using this example. As before, let us assume agents/bidders have quasi-linear utility functions and private values.

- **Posted price:** The seller announces a price at which he will sell the object. The first buyer to express demand at this price wins the object. It is a very common form of selling. Since the seller does not elicit any information from the buyers, this makes sense if the seller has good information about the values of buyers to set his price.

- **First-price auction:** In the first-price auction, every bidder is asked to report a bid, which indicates his value. The highest bidder wins the auction and pays the price he bid. Of course, the bid amount need not equal the value. But if the bidders bid their value, then the first bidder will win the object and pay an amount of 10.

- **Second-price auction:** In the second-price auction, like the first-price auction, each bidder is asked to report a bid. The highest bidder wins the auction and pays the price of the second highest bid. This is the Vickrey auction we have already discussed. As we saw, a dominant strategy in this auction is that bidders will bid their values. Hence, the first bidder will win the object but pay a price equal to 8, the second highest value.

- **Dutch auction:** The Dutch auction, popular for selling flowers in the Netherlands, falls into a class of auctions called the *open-cry* auctions. The Dutch auction starts at a high price and the price of the object is lowered by a small amount (called the *bid decrement*) in iterations. In every iteration, bidders can express their interest to buy the object. The price of the object is lowered only if no bidder shows interest. The auction stops as soon as any bidder shows interest. The first bidder to show interest wins the object at the current price.

  In the example above, suppose the Dutch auction is started at price 12 and let the bid decrement be 1. At price 12, no bidder should express interest since valuation of all bidders are less than 12. After price 10, the first bidder may choose to express interest since he starts getting non-negative utility from the object for any price less than or equal to 10. If he chooses to express interest, then the auction would stop and he will win the object. Clearly, it is not an equilibrium for the bidder to express interest at 10 since he can potentially get more payoff by waiting for the price to fall. Indeed, in equilibrium (under some conditions), the bidder will show interest at a price just below his valuation.

- **English auction:** The English auction is also an open-cry auction. The seller starts the auction at a low price and raises it by a small amount (called the *bid increment*)

in iterations. In every iteration, like in the Dutch auction, the bidders are asked if they are interested in buying the object. The price is raised only if more than one bidder shows interest. The auction stops as soon as one or less number of bidders show interest. The last bidder to show interest wins the auction at the price he last showed interest.

In the example above, suppose the English auction is started at price 0 and let the bid increment be 1. Then, at price 4 the bidder with value 4 will stop showing interest (since he starts getting non-positive payoff from that price onwards). Similarly, at prices 6, bidder with value 6 will drop out. Finally, bidder with value 8 will drop out at price 8. At this price, only bidder with value 10 will show interest. Hence, the auction will stop at price 8, and the bidder with value 10 will win the object at price 8. Notice that the outcome of the auction is the same as the second-price auction. This is no coincidence. It can be argued easily that it is an equilibrium (under private values model) for bidders to show interest (bid) till the price reaches their value in the English auction. Hence, the outcome of the English auction is the same as the second-price auction.

One can think of many more auction formats - though they may not be used in practice. Having learnt and thought about these auction formats, some natural questions arise. Is there an equilibrium strategy for the bidder in each of these auctions? What kind of auctions are incentive compatible? What is the ranking of these auctions in terms of expected revenue? Which auction gives the maximum expected revenue to the seller over all possible auctions?

Myerson (1981) answers many of these questions. First, using the revelation principle (for Bayesian incentive compatibility), he concludes that for every auction (sealed-bid or open-cry or others) there exists a direct mechanism with the same social choice function, and thus giving the same expected revenue to the seller. So, he focuses on direct mechanisms without loss of generality. Second, he characterizes direct mechanisms which are Bayesian incentive compatible. Third, he shows that all Bayesian incentive compatible mechanisms which have the same allocation rule, differ in revenue by a constant amount. Using these results, he is able to give a precise description of an auction which gives the maximum expected revenue. He calls such an auction an *optimal auction*. Under some conditions on the valuation distribution of bidders, the optimal auction is a modified second-price auction. Next, we describe these results formally.

16.2 The Model

There is a single indivisible object for sale, whose value for the seller is zero. The set of bidders is denoted by $N = \{1, \ldots, n\}$. Every bidder has a value (this is his *type*) for the object. The value of bidder $i \in N$ is drawn from $[0, b_i]$ using a distribution with density function g_i and cumulative density G_i. We assume that each bidder draws his value independently and this value is completely determined by this draw (i.e., knowledge of other information such as value of other bidders does not influence his value). This model of valuation is referred to as the **private independent value model**. We let the joint density function of values of all the bidders as g and the joint density function of values of all the bidders except bidder i as g_{-i}. Due to the independence assumption, for every profile of values $t = (t_1, \ldots, t_n)$

$$g(t_1, \ldots, t_n) = g_1(t_1) \times \ldots \times g_n(t_n)$$
$$g_{-i}(t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) = g_1(t_1) \times \ldots \times g_{i-1}(t_{i-1}) \times g_{i+1}(t_{i+1}) \times \ldots g_n(t_n).$$

Let $T_i = [0, b_i]$ and $T^n = [0, b_1] \times \ldots \times [0, b_n]$. Similarly, let $T_{-i} = \times_{j \in N \setminus \{i\}} T_j$. A typical valuation of bidder i will be denoted as $t_i \in T_i$, a valuation profile of bidders will be denoted as $t \in T^n$, and a valuation profile of bidders in $N \setminus \{i\}$ will be denoted as $t_{-i} \in T_{-i}$. The valuation profile $t = (t_1, \ldots, t_i, \ldots, t_n)$ will sometimes be denoted as (t_i, t_{-i}). We assume that $g_i(t_i) > 0$ for all $i \in N$ and for all $t_i \in T_i$.

16.3 The Direct Mechanisms

Though a mechanism can be very complicated, a direct mechanism is simpler to describe. By virtue of the revelation principle (Proposition 2), we can restrict attention to direct mechanisms only. Henceforth, I will refer to a direct mechanism as simply a mechanism.

Let A be the set of all deterministic alternatives, i.e., $A = \{a_0, a_1, \ldots, a_n\}$, where a_0 is the allocation where the seller keeps the object and a_i for $1 \leq i \leq n$ denotes the allocation where agent i gets the object. Let $\mathcal{L}(A)$ be the set of all probability distributions over A. A direct mechanism M in this context is $M = (f, p_1, \ldots, p_n)$, where $f : T^n \to \Delta A$ is the **allocation rule** and for every $i \in N$, $p_i : T^n \to \mathbb{R}$ is the **payment rule** of agent i. Given a mechanism $M = (f, p_1, \ldots, p_n)$, a bidder $i \in N$ with (true) value $t_i \in T_i$ gets the following utility when all the buyers report values $s = (s_1, \ldots, s_i, \ldots, s_n)$

$$u_i^M(s; t_i) = f_i(s)t_i - p_i(s),$$

where $f_i(s)$ is the probability that agent i gets the object at type profile s and $p_i(s)$ is the payment of agent i at type profile s.

Every mechanism $(f, p_1, \ldots, p_n)$ induces an expected allocation rule and an expected payment rule (α, π), defined as follows. The expected allocation of agent i when he reports $s_i \in T_i$ in allocation rule f is

$$\alpha_i(s_i) = \int_{T_{-i}} f_i(s_i, s_{-i}) g_{-i}(s_{-i}) ds_{-i}.$$

Similarly, the expected payment of bidder i when he reports $s_i \in T_i$ in payment rule p_i is

$$\pi_i(s_i) = \int_{T_{-i}} p_i(s_i, s_{-i}) g_{-i}(s_{-i}) ds_{-i}.$$

So, the expected utility from a mechanism $M \equiv (f, p_1, \ldots, p_n)$ to an agent i with true value t_i by reporting a value s_i is $\alpha_i(s_i)t_i - \pi_i(s_i)$.

Definition **24** *A mechanism $(f, p_1, \ldots, p_n)$ is* **Bayesian incentive compatible (BIC)** *if for every agent $i \in N$ and for every possible values $s_i, t_i \in T_i$ we have*

$$\alpha_i(t_i)t_i - \pi_i(t_i) \geq \alpha_i(s_i)t_i - \pi_i(s_i). \qquad \textbf{(BIC)}$$

Equation **BIC** says that a bidder maximizes his expected utility by reporting true value. Given that other bidders report truthfully, when bidder i has value t_i, he gets more expected utility by reporting t_i than by reporting any other value $s_i \in T_i$.

16.4 Bayesian Incentive Compatible Mechanisms

We say an allocation rule f is **Bayes-Nash implementable** if there exists payment rules $(p_1, \ldots, p_n)$ such that $(f, p_1, \ldots, p_n)$ is a Bayesian incentive compatible mechanism.

We say that an allocation rule f is **non-decreasing in expectation (NDE)** if for all $i \in N$ and for all $s_i, t_i \in T_i$ with $s_i < t_i$ we have $\alpha_i(s_i) \leq \alpha_i(t_i)$. Similar to the characterization in the dominant strategy case, we have a characterization in the Bayesian incentive compatible mechanisms.

Theorem **17** *A mechanism $(f, p_1, \ldots, p_n)$ is Bayesian incentive compatible if and only if f is NDE and for every $i \in N$, p_i satisfies*

$$\pi_i(t_i) = \pi_i(0) + t_i\alpha_i(t_i) - \int_0^{t_i} \alpha_i(s_i) ds_i \qquad \forall\ t_i \in [0, b_i].$$

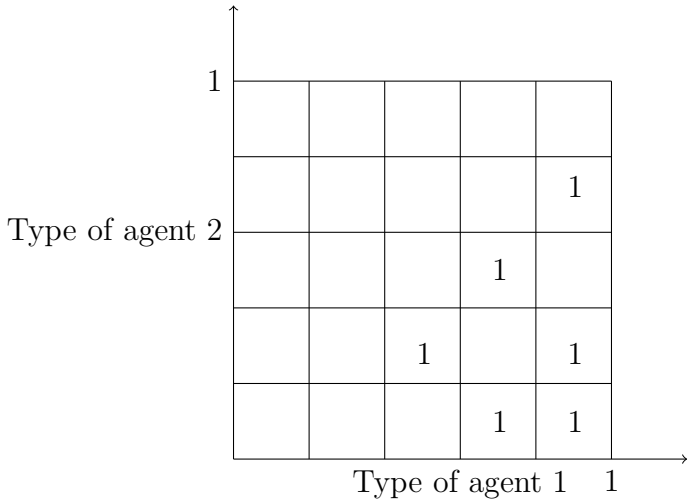

Figure 14: A BIC allocation rule which is not DSIC

The proof is a replication of the arguments we did for dominant strategy case in Theorem 16. We skip the proof (but you are encouraged to reconstruct the arguments).

A BIC allocation rule need not be DSIC. We give an example to illustrate this. Consider a setting with two agents $N = \{1, 2\}$. Suppose the values of both the agents are drawn uniformly from $[0, 1]$. Figure 14 shows an allocation rule f.

The type profiles are divided into cells of equal size (25 of them in total). Some of the cells are assigned some numbers - this is the probability with which agent 1 gets the object in f. The cells in which no number is written, the probability of agent 1 getting the object at those profiles is zero. For our purpose, the probability of agent 2 getting the object is irrelevant - for simplicity, we can assume it to be zero (hence, in all other cells the seller keeps the object).

An easy calculation reveals that the expected probability of agent 1 winning the object is non-decreasing: it is zero if $t_1 \leq \frac{2}{5}$, it is $\frac{1}{5}$ if $t_1 \in (\frac{2}{5}, \frac{3}{5}]$, it is $\frac{2}{5}$ if $t_1 \in (\frac{3}{5}, \frac{4}{5}]$, and it is $\frac{3}{5}$ if $t_1 > \frac{4}{5}$. Hence, the allocation rule a is BIC but not DSIC.

Theorem 17 says that the (expected) payment of a bidder in a mechanism is uniquely determined by the allocation rule once we fix the expected payment of a bidder with the lowest type. Hence, a mechanism is uniquely determined by its allocation rule and the payment of a bidder with the lowest type.

It is instructive to examine the payment function when $\pi_i(0) = 0$. Then payment of agent i at type t_i becomes

$$\pi_i(t_i) = \alpha_i(t_i)t_i - \int_0^{t_i} \alpha_i(x_i)dx_i.$$

Because of non-decreasing $\alpha_i(\cdot)$ this is always greater than or equal to zero - it is the difference between area of the rectangle with sides $\alpha_i(x_i)$ and x_i and the area under the curve $\alpha_i(\cdot)$ from 0 to x_i.

We next impose a the analogue of individual rationality in the Bayesian set up.

DEFINITION **25** *A mechanism* $(f, p_1, \ldots, p_n)$ *is* **interim individually rational (IIR)** *if for every bidder* $i \in N$ *we have*

$$\alpha_i(t_i)t_i - \pi_i(t_i) \geq 0 \qquad \forall\ t_i \in T_i.$$

IIR is weaker than the (ex post) individual rationality we had discussed earlier since IIR only requires *interim* expected utility from truthtelling to be non-negative. The set of BIC and IIR mechanisms can now be characterized as follows.

THEOREM **18** *A mechanism* $(f, p_1, \ldots, p_n)$ *is BIC and IIR if and only if*

(1) f *is NDE.*

(2) For all $i \in N$*,*

$$\pi_i(t_i) = \pi_i(0) + t_i\alpha_i(t_i) - \int_0^{t_i} \alpha_i(s_i)ds_i \qquad \forall\ t_i \in [0, b_i].$$

(3) For all $i \in N$*,* $\pi_i(0) \leq 0$*.*

Proof: Suppose $(f, p_1, \ldots, p_n)$ is BIC. By Theorem 17, (1) and (2) follows. Applying IIR at $t_i = 0$, we get $\pi_i(0) \leq 0$, which is (3).

Now, suppose (1),(2), and (3) holds for a mechanism $(f, p_1, \ldots, p_n)$. By Theorem 17, the mechanism is BIC. At any type t_i,

$$\begin{aligned} t_i\alpha_i(t_i) - \pi_i(t_i) &= \int_0^{t_i} \alpha_i(s_i)ds_i - \pi_i(0) \\ &\geq \int_0^{t_i} \alpha_i(s_i)ds_i \\ &\geq 0, \end{aligned}$$

where the first inequality follows from (3). Hence, the mechanism satisfies IIR. ■

16.5 Optimal Mechanisms

The optimal mechanism is a mechanism in the class of mechanisms identified in Theorem 18 that maximizes the expected revenue of the seller over all mechanisms identified in Theorem 18. To compute expected revenue from a mechanism $(f, p \equiv (p_1, \ldots, p_n))$, we note that the expected payment of agent i with type t_i is $\pi_i(t_i)$. Hence, (ex-ante) expected payment of agent i to this mechanism is

$$\int_0^{b_i} \pi_i(t_i) g_i(t_i) dt_i.$$

Hence, the expected revenue from the mechanism $(f, p \equiv (p_1, \ldots, p_n))$ is

$$\Pi(f, p) = \sum_{i \in N} \int_0^{b_i} \pi_i(t_i) g_i(t_i) dt_i.$$

We say a mechanism (f, p) is an **optimal mechanism** if

- (f, p) is Bayesian incentive compatible and individually rational,
- and $\Pi(f, p) \geq \Pi(f', p')$ for any other Bayesian incentive compatible and individually rational mechanism (f', p').

Theorem 18 will play a crucial role since it has identified the entire class of BIC and IIR mechanisms, over which we are optimizing.

Fix a mechanism (f, p) which is Bayesian incentive compatible and individually rational. For any bidder $i \in N$, the expected payment of bidder $i \in N$ is given by

$$\int_0^{b_i} \pi_i(t_i) g_i(t_i) dx_i = \pi_i(0) + \int_0^{b_i} \alpha_i(t_i) t_i g_i(t_i) dt_i - \int_0^{b_i} \int_0^{t_i} \big(\alpha_i(s_i) ds_i\big) g_i(s_i) ds_i,$$

where the last equality comes by using revenue equivalence (Theorem 17). By interchanging the order of integration in the last term, we get

$$\begin{aligned} \int_0^{b_i} \int_0^{t_i} \big(\alpha_i(s_i) ds_i\big) g_i(t_i) dt_i &= \int_0^{b_i} \big(\int_{t_i}^{b_i} g_i(s_i) ds_i\big) \alpha_i(t_i) dt_i \\ &= \int_0^{b_i} (1 - G_i(t_i)) \alpha_i(t_i) dt_i. \end{aligned}$$

Hence, we can write

$$\Pi(a, p) = \sum_{i \in N} \pi_i(0) + \sum_{i \in N} \int_0^{b_i} \big(t_i - \frac{1 - G_i(t_i)}{g_i(t_i)}\big) \alpha_i(t_i) g_i(t_i) dt_i.$$

We now define the **virtual valuation** of bidder $i \in N$ with valuation $t_i \in T_i$ as

$$w_i(t_i) = t_i - \frac{1 - G_i(t_i)}{g_i(t_i)}.$$

Note that since $g_i(t_i) > 0$ for all $i \in N$ and for all $t_i \in T_i$, the virtual valuation $w_i(t_i)$ is well defined. Also, virtual valuations can be negative. Using this and the definition of $\alpha_i(\cdot)$, we can write

$$\begin{aligned}
\Pi(f,p) &= \sum_{i\in N} \pi_i(0) + \sum_{i\in N} \int_0^{b_i} w_i(t_i)\alpha_i(t_i)g_i(t_i)dt_i \\
&= \sum_{i\in N} \pi_i(0) + \sum_{i\in N} \int_0^{b_i} \Big(\int_{T_{-i}} f_i(t_i, t_{-i})g_{-i}(t_{-i})dt_{-i}\Big) w_i(t_i)g_i(t_i)dt_i \\
&= \sum_{i\in N} \pi_i(0) + \sum_{i\in N} \int_{T^n} w_i(t_i)f_i(t)g(t)dt \\
&= \sum_{i\in N} \pi_i(0) + \int_{T^n} \left[\sum_{i\in N} w_i(t_i)f_i(t)\right] g(t)dt.
\end{aligned}$$

Since IIR requires $\pi_i(0) \leq 0$ for all $i \in N$, if we want to maximize $\Pi(f,p)$, we must set $\pi_i(0) = 0$ for all $i \in N$. As a result, the optimization problem only involves finding the allocation rule, and the payment rule can be computed using Theorem 17 and setting $\pi_i(0) = 0$ for all $i \in N$. So, we can succinctly write down the optimal mechanism optimization problem.

$$\max_f \int_{T^n} \left[\sum_{i\in N} w_i(t_i)f_i(t)\right] g(t)dt$$
$$\text{subject to} \quad f \text{ is NDE.}$$

The term in the objective function is exactly the **total expected virtual valuation** from an allocation rule. This is because, the term $\sum_{i\in N} w_i(t_i)f_i(t)$ is the total *realized* virtual valuation of all bidders at type profile t from allocation rule f. This observation leads to the following important result.

THEOREM **19** *The allocation rule in an optimal mechanism maximizes the total expected virtual valuation among all Bayes-Nash implementable allocation rules.*

Without the constraint that f has to be NDE, we can maximize our objective function by doing a *point-wise* maximization. In particular, at every type profile t, we assign $f_i(t) = 0$ for all $i \in N$ if $w_i(t_i) < 0$ for all $i \in N$; else we assign $f_i(t) = 1$ for some $i \in N$ such

that $w_i(t_i) \geq w_j(t_j)$ for all $j \neq i$. In other words, the highest virtual valuation agent wins the object if he has non-negative virtual valuation, else the object is unsold. Clearly, this maximizes the objective function without the NDE constraint. Now, it may so happen that the optimal solution obtained may not satisfy the NDE constraint. Below, we impose conditions on the distributions of agents that ensure that the unconstrained optimal solution satisfies the constraints, and hence, a constrained optimal solution.

DEFINITION **26** *A virtual valuation w_i of agent i is* **regular** *if for all $s_i, t_i \in T_i$ with $s_i > t_i$, we have $w_i(s_i) > w_i(t_i)$.*

Regularity requires that the virtual valuation functions are strictly increasing. The following condition on distributions ensures that regularity holds. The hazard rate of a distribution g_i is defined as $\lambda_i(t_i) = \frac{g_i(t_i)}{1-G_i(t_i)}$ for all $i \in N$.

LEMMA **15** *If the hazard rate λ_i is non-decreasing, then the virtual valuation w_i is regular.*

Proof: Consider $s_i, t_i \in T_i$ such that $s_i > t_i$. Then,

$$w_i(s_i) = s_i - \frac{1}{\lambda_i(s_i)} > t_i - \frac{1}{\lambda_i(t_i)} = w_i(t_i).$$

■

The uniform distribution satisfies the non-decreasing hazard rate condition. Because $\frac{1-G_i(t_i)}{g_i(t_i)} = b_i - t_i$, which is non-increasing in t_i. For the exponential distribution, $g_i(t_i) = \mu e^{-\mu t_i}$ and $G_i(t_i) = 1 - e^{-\mu t_i}$. Hence, $\frac{1-G_i(t_i)}{g_i(t_i)} = \frac{1}{\mu}$, which is a constant. So, the exponential distribution also satisfies the non-decreasing hazard rate condition.

This leads to our main observation.

LEMMA **16** *Suppose regularity holds for the virtual valuation of each agent. Then, the allocation rule in the optimal mechanism solves the following unconstrained optimization problem.*

$$\max_f \int_{T^n} \left[\sum_{i \in N} w_i(t_i) f_i(t) \right] g(t) dt.$$

Proof: We have already seen that the optimal solution to the unconstrained optimization problem is done as follows: for every type profile t, $f_i(t) = 0$ for all $i \in N$ if $w_i(t_i) < 0$ for all $i \in N$ and $f_i(t) = 1$ for some $i \in N$ if $w_i(t_i) \geq 0$ and $w_i(t_i) \geq w_j(t_j)$ for all $j \in N$. If the regularity condition holds, then f is NDE. To see this, consider a bidder $i \in N$ and $s_i, t_i \in T_i$ with $s_i > t_i$. Regularity gives us $w_i(s_i) > w_i(t_i)$. By the definition of the allocation rule,

for all $t_{-i} \in T_{-i}$, we have $f_i(s_i, t_{-i}) \geq f_i(t_i, t_{-i})$. Hence, f is non-decreasing, and hence, it is NDE. ■

Our discussions to the main theorem of this section.

THEOREM **20** *Suppose the regularity holds for each agent. Consider the following allocation rule f^*. For every type profile $t \in T^n$, $f_i^*(t) = 0$ if $w_i(t_i) < 0$ for all $i \in N$ and else, $f_i^*(t) = 1$ for some $i \in N$ such that $w_i(t_i) \geq 0$, $w_i(t_i) \geq w_j(t_j)\ \forall\ j \in N$. There exists payments $(p_1, \ldots, p_n)$ such that $(f^*, p_1, \ldots, p_n)$ is an optimal mechanism.*

We now come back to the payments. To remind, we need to ensure that payments satisfy the revenue equivalence and $\pi_i(0) = 0$ for all $i \in N$. Since f^* can be implemented in dominant strategies and it is a deterministic allocation rule, we can ensure this by satisfying the revenue equivalence formulae for the dominant strategy case (which simplifies if the allocation rule is deterministic) and setting $p_i(0, t_{-i}) = 0$ for all i and for all t_{-i}. From our earlier analysis, the payment then is uniquely determined as the following (from Theorem 16).

For every $i \in N$ and for every t_{-i}, let $\kappa_i^{f^*}(t_{-i}) = \inf\{t_i : f_i^*(t_i, t_{-i}) = 1\}$. If $f^*(t_i, t_{-i}) = 0$ for all $t_i \in T_i$, then set $\kappa_i^{f^*}(t_{-i}) = 0$.

THEOREM **21** *Suppose the regularity holds for each agent. Consider the following allocation rule f^*. For every type profile $t \in T^n$, For every type profile $t \in T^n$, $f_i^*(t) = 0$ if $w_i(t_i) < 0$ for all $i \in N$ and else, $f_i^*(t) = 1$ for some $i \in N$ such that $w_i(t_i) \geq 0$, $w_i(t_i) \geq w_j(t_j)\ \forall\ j \in N$. For every agent $i \in N$, consider the following payment rule. For every $(t_i, t_{-i}) \in T^n$,*

$$p_i^*(t_i, t_{-i}) = \begin{cases} 0 & \text{if } f_i^*(t_i, t_{-i}) = 0 \\ \kappa_i^{f^*}(t_{-i}) & \text{if } f_i^*(t_i, t_{-i}) = 1 \end{cases}$$

The mechanism $(f^*, p_1^*, \ldots, p_n^*)$ is an optimal mechanism.

Proof: By Theorem 20, there is an optimal mechanism involving f^*. Under regularity, f^* is non-decreasing, and hence, dominant strategy implementable. For the mechanism to be optimal, we only need to show that $(p_1^*, \ldots, p_n^*)$ satisfy the revenue equivalence formulae in Theorem 17 with $\pi_i^*(0) = 0$ for all $i \in N$.

The payments $(p_1^*, \ldots, p_n^*)$ satisfy the revenue equivalence formula in Theorem 16. Hence, by Theorem 16, $(f^*, p_1^*, \ldots, p_n^*)$ is dominant strategy incentive compatible, and hence, BIC. So, they satisfy the revenue equivalence formula in Theorem 17. Since $p_i^*(0, t_{-i}) = 0$ for all $i \in N$ and for all t_{-i}, we have $\pi_i^*(t_i) = 0$ for all $i \in N$ and for all $t_i \in T_i$. This shows that $(f^*, p_1^*, \ldots, p_n^*)$ is an optimal mechanism. ■

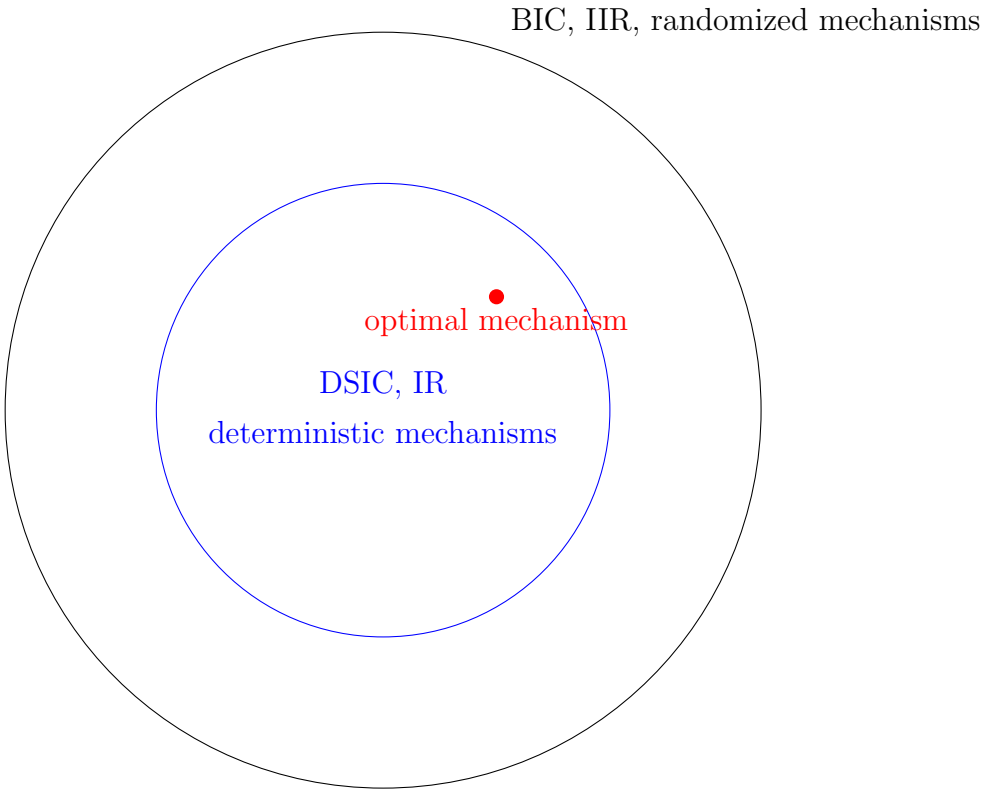

Figure 15: Optimal mechanism is DSIC, IR, and deterministic

Figure 15 highlights the fact that we started out searching for an optimal mechanism in a large family of BIC, IIR, and randomized mechanisms. But the optimal mechanism turned out to be DSIC, IR, and deterministic.

If the regularity condition does not hold, the optimal mechanism is more complicated, and you can refer to Myerson's paper for a complete treatment.

16.5.1 Symmetric Bidders

Finally, we look at the special case where the buyers are **symmetric**, i.e., they draw the valuations using the same distribution - $g_i = g$ and $T_1 = T_2 = \ldots = T_n$ for all $i \in N$. So, virtual valuations are the same: $w_i = w$ for all $i \in N$. In this case $w(t_i) > w(t_j)$ if and only if $t_i > t_j$ by regularity. Hence, maximum virtual valuation corresponds to the maximum valuation.

Thus, $\kappa_i(t_{-i}) = \max\{w^{-1}(0), \max_{j \neq i} t_j\}$. This is exactly, the second-price auction with the reserve price of $w^{-1}(0)$. Hence, when the buyers are symmetric, then the second-price auction with a reserve price equal to $w^{-1}(0)$ is optimal.

16.5.2 An Example

Consider a setting with two buyers whose values are distributed uniformly in the intervals $T_1 = [0, 12]$ (buyer 1) and $T_2 = [0, 18]$ (buyer 2). Virtual valuation functions of buyer 1 and buyer 2 are given as:

$$w_1(t_1) = t_1 - \frac{1 - G_1(t_1)}{g_1(t_1)} = t_1 - (12 - t_1) = 2t_1 - 12$$
$$w_2(t_2) = t_2 - \frac{1 - G_2(t_2)}{g_2(t_2)} = t_2 - (18 - t_2) = 2t_2 - 18.$$

Hence, the reserve prices for both the bidders are respectively $r_1 = 6$ and $r_2 = 9$. The optimal auction outcomes are shown for some instances in Table 18.

Valuations	Allocation (who gets object)	Payment of Buyer 1	Payment of Buyer 2
$(t_1 = 4, t_2 = 8)$	Object not sold	0	0
$(t_1 = 2, t_2 = 12)$	Buyer 2	0	9
$(t_1 = 6, t_2 = 6)$	Buyer 1	6	0
$(t_1 = 9, t_2 = 9)$	Buyer 1	6	0
$(t_1 = 8, t_2 = 15)$	Buyer 2	0	11

Table 18: Description of Optimal Mechanism

16.5.3 Efficiency and Optimality

One of the conclusions that we can draw from the previous analysis is that the optimal mechanism is not efficient. We illustrate this with an example. Suppose there are two agents $N = \{1, 2\}$. Suppose $T_1 = [0, 10]$ and $T_2 = [0, 6]$. To compute the optimal mechanism, we need to compute the virtual valuation functions. For agent 1, for every $t_1 \in T_1$, we have

$$w_1(t_1) = 2t_1 - 10.$$

For agent 2, for every $t_2 \in T_2$, we have

$$w_2(t_2) = 2t_2 - 6.$$

The optimal mechanism is shown in Figure 16. Notice that the object is unsold if $t_1 < 5$ and $t_2 < 3$. This is inefficient. This inefficiency occurs because of the reserve prices in the optimal mechanism. There is another source of inefficiency. Efficiency requires that agent

1 wins the object if $t_1 > t_2$. However, the optimal mechanism requires that $2t_1 - 10 \geq 0$ and $2t_1 - 10 \geq 2t_2 - 6$. This means, agent 2 wins the object in some cases where agent 1 should have won - this is shown in Figure 16. For instance, at the type profile $(5, 4)$, we have $2t_2 - 6 = 2 > 0 = 2t_1 - 5$. Hence, agent 2 wins the object, but efficiency requires agent 1 must win the object here. This inefficiency occurs because the virtual valuation function of both the agents is not the same, which happens because the distribution of values is asymmetric. When bidders are symmetric, this source of inefficiency disappears. So, with symmetric bidders, whenever the object is allocated, it is allocated efficiently.

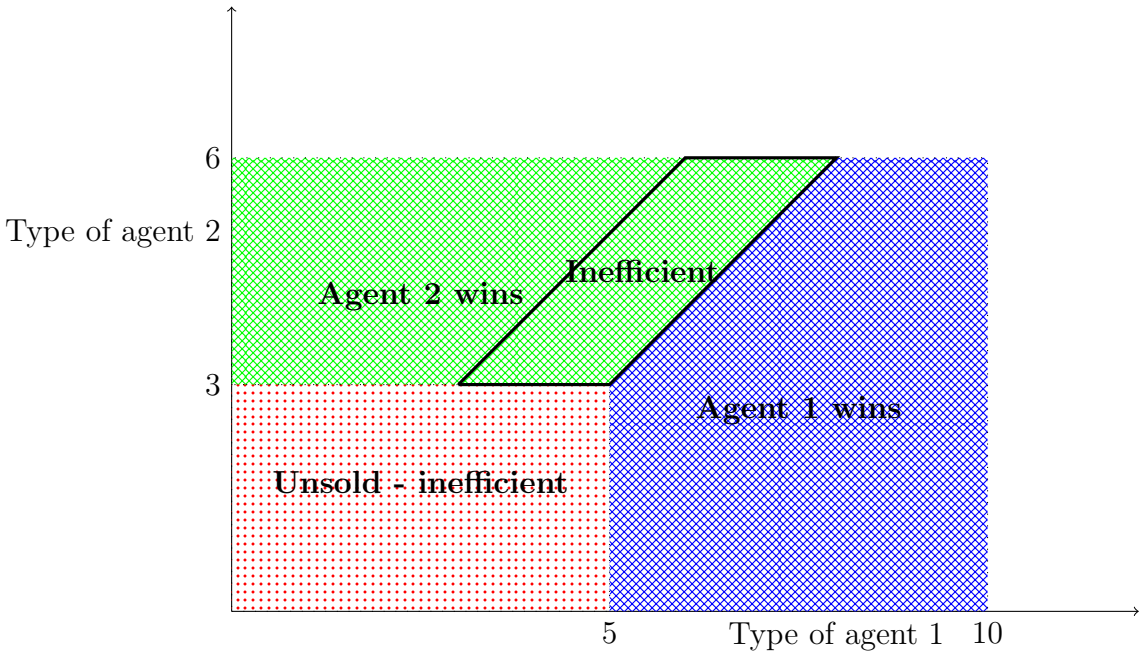

Figure 16: Inefficiency of optimal mechanism

16.5.4 Surplus Extraction

Note that in the optimal mechanism buyers whose values are positive will walk away with some positive expected utility. This is because the optimal mechanism satisfies individual rationality and the payment of bidder i at valuation profile x, $\kappa_i(x_{-i})$, is usually smaller than x_i. The expected utility of a bidder is sometimes referred to as his **informational rent**. This informational rent is accrued by bidder i because of the fact that he has complete knowledge of his value (private value).

One way to think of the single object auction problem is that there is a maximum achievable surplus equal to maximum value of the object. The seller and the bidders divide this

surplus amongst themselves by the auctioning procedure. Since the seller does not have information about bidders' values and bidders are perfectly informed about their individual values, the seller is unable to extract full surplus extraction [12].

17 Impossibility of Efficiency and Budget-balance

We investigate the important question of achieving budget-balance in mechanisms. The reason budget-balance is an important objective in mechanism is the following. The efficient allocation rule maximizes the sum of values to agents. But the eventual net utility to the agents is their values from the alternative minus the transfer amount. Hence, maximizing sum of values need not maximize the sum of net utilities to the agents. Indeed, both the maximizations are equivalent if and only if the sum of transfers is equal to zero. Hence, achieving budget-balance allows us to achieve efficiency in the "real" way - this is often called the "first-best" efficiency.

The main result from this section is that Bayesian incentive compatibility, efficiency, and interim individual rationality are often incompatible. We illustrate this in two models: (a) bilateral trading and (b) public good provision. These are two simple models to study. In the bilateral trading model, there are two agents: a seller and a buyer. The seller has an object that it can trade to the buyer. If trade takes place, then the seller incurs a cost (his type) and the buyer realizes a value (his type). If no trade takes place, then both the agents get zero utility. Here, budget-balance is an important aspect of the mechanism since whatever the buyer pays must be received by the seller.

In the public good provision problem, agents are deciding whether to choose a public project or not. If the project is not selected, then they incur zero cost and zero value. But if the project is selected, then they incur individual value (type) from the project. The project also has a cost. The objective is to choose the project efficiently, i.e., choose the project if and only if when sum of values is larger than the cost. However, the payments of the agents must also cover the cost of the project. We will show that there is no efficient BIC mechanism that is interim individually rational can cover the cost of the project.

[12] In advanced auction theory lectures, you will learn that it is possible for seller to extract entire surplus if there is some degree of correlation between values of bidders.

17.1 A General Model and Characterization of Budget-Balance

In this section, we will consider a very general model that covers the bilateral trading model as a special case. We will derive some necessary and sufficient conditions in this model for a BIC, efficient, and IIR mechanism to exist.

In our model, the set of agents is $N = \{1, \ldots, n\}$ and the set of alternatives is A (a finite set). Each agent $i \in N$, has a non-empty subset of alternatives $A_i \subseteq A$ from which it gets a value t_i. The value of the agent is his type (private information). For any alternative $a \notin A_i$, agent i gets zero value. The subset A_i for each i, is publicly known. For instance, in the bilateral trading model, $A = \{a_0, a_1\}$, where a_0 is the no-trade alternative and a_1 is the trade alternative. For both the buyer and the seller, $A_i = \{a_1\}$ since trade induces a value for the buyer and cost (-ve of value) for the seller. Similarly, in the public good provision problem, A_i consists of the alternative where the project is chosen.

Let $T_i = [\ell_i, b_i]$ be the type space of agent i. We do not assume $\ell_i = 0$ since type of an agent may be negative (for instance, cost of the seller). Let $T^n = T_1 \times \ldots \times T_n$. An allocation rule $f : T^n \to \mathcal{L}(A)$ chooses a lottery over A at every type profile, where $\mathcal{L}(A)$ is the set of all probability distributions over A. We write $f_a(t)$ as the probability of alternative a being chosen. For any agent $i \in N$, the probability that agent i gets a non-zero value at type profile t is denoted by $f_i(t)$, and is given by

$$f_i(t) = \sum_{a \in A_i} f_a(t).$$

We will focus on Bayesian incentive compatible mechanisms. For any mechanism $M \equiv (f, p)$, where $p \equiv (p_1, \ldots, p_n)$ denotes the payment rules of the agents, let $\alpha_i(t_i)$ and $\pi_i(t_i)$ denote the expected probability of agent i getting non-zero value and his expected payment respectively when his type is t_i. Further, let $U_i^M(t_i)$ denote the expected net utility of agent i when his type is t_i, i.e.,

$$U_i^M(t_i) = t_i \alpha_i(t_i) - \pi_i(t_i).$$

Using the standard techniques we used in the earlier section, we can arrive at the following result.

Theorem **22** *A mechanism (f, p) is Bayesian incentive compatible if and only if*

1. *$\alpha_i(\cdot)$ is non-decreasing for every $i \in N$ and*

2. For all $i \in N$,

$$U_i^M(t_i) = U_i^M(\ell_i) + \int_{\ell_i}^{t_i} \alpha_i(s_i) ds_i \qquad \forall\ t_i \in T_i.$$

You are encouraged to go through the earlier proofs, and derive this again. An outcome of Theorem 22 is that if there are two mechanisms $M = (f, p)$ and $M' = (f, p')$ implementing the same allocation rule f, then for all $i \in N$ and for all $t_i \in T_i$, we have

$$U_i^M(t_i) - U_i^{M'}(t_i) = U_i^M(\ell_i) - U_i^{M'}(\ell_i). \tag{5}$$

17.2 The Modified Pivotal Mechanism

We now investigate a modified pivotal mechanism. A mechanism (f, p) is budget-balanced if for all type profiles t,

$$\sum_{i \in N} p_i(t) = 0.$$

The main question that we address in this section is if there exists a Bayesian incentive compatible mechanism implementing the efficient allocation rule, which is budget-balanced and interim individually rational. To remind, a mechanism M is interim individually rational (IIR) if $U_i^M(t_i) \geq 0$ for all $i \in N$ and for all $t_i \in T_i$.

To answer this question, we go back to a slight variant of the pivotal mechanism. We denote this mechanism as $M^* \equiv (f^*, p^*)$, where f^* is the efficient allocation rule and p^* is defined as follows. To remind, f^* is defined as an allocation rule that satisfies for every type profile t,

$$f^*(t) \in \arg\max_{a \in A} \sum_{i \in N} t_i \mathbf{1}_i^a,$$

where $\mathbf{1}_i^a = 1$ if $a \in A_i$ and zero otherwise for every $i \in N$ and for every $a \in A$.

For every type profile t, denote by $W(t)$ the total value of agents in the efficient allocation, i.e.,

$$W(t) = \sum_{i \in N} f_i(t) t_i.$$

Denote by $W_{-i}(t)$ the sum $\sum_{j \neq i} f_j(t) t_j$. Then the payment of agent i at type profile t in the modified pivotal mechanism M^* is given by

$$p_i^*(t) = W(\ell_i, t_{-i}) - W_{-i}(t).$$

This is a slight modification of the pivotal mechanism we had defined earlier - the $h_i(\cdot)$ function here is defined slightly differently. Since M^* is a Groves mechanism, it is dominant strategy incentive compatible.

Note that for every $i \in N$, we have $U_i^{M^*}(\ell_i) = 0$. This means that $U_i^{M^*}(t_i) \geq 0$ for all $i \in N$ and for all $t_i \in T_i$ (try to show this formally). Hence, M^* is individually rational. Now, consider any other mechanism $M \equiv (f^*, p)$ which is Bayesian incentive compatible and individually rational. By Equation 5, we know that for every $i \in N$ and for every $t_i \in T_i$,

$$U_i^M(t_i) - U_i^M(\ell_i) = U_i^{M^*}(t_i) - U_i^{M^*}(\ell_i) = U_i^{M^*}(t_i),$$

where we used $U_i^{M^*}(\ell_i) = 0$ for the last equality. Since M is individually rational, $U_i^M(\ell_i) \geq 0$. Hence, $U_i^M(t_i) \geq U_i^{M^*}(t_i)$. Since M and M^* have the same allocation rule f^*, we get that

$$t_i \alpha_i^*(t_i) - \pi_i^*(t_i) \geq t_i \alpha_i^*(t_i) - \pi_i(t_i),$$

where α_i^* is the expected allocation probability in f^* for agent i. Hence, we have for every $i \in N$, $\pi_i^*(t_i) \geq \pi_i(t_i)$ for all $t_i \in T_i$. This observation is summarized in the following proposition.

PROPOSITION **13** *Among all Bayesian incentive compatible and IIR mechanisms which implement the efficient allocation rule, the modified pivotal mechanism maximizes the expected payment of every agent.*

But the modified pivotal mechanism will usually not balance the budgets. Here is an example in the bilateral trading model with one buyer and one seller. Suppose the buyer has a value of 10 and the seller has a value of -5 (cost of 5). Suppose buyer's values are from $[0, 10]$ and seller's values are from $[-10, -5]$. Then, efficiency tells us that the object is allocated to the buyer at this type profile. His payment is $0 + 5 = 5$ and the payment of the seller is $0 - 10 = -10$. So, there is a net deficit of 5.

The modified pivotal mechanism, though not budget-balanced, will play an important role in identifying necessary and sufficient conditions on when a BIC, efficient, and IIR mechanism can be budget-balanced.

17.3 The AGV Mechanism

We start off by defining a new mechanism which is Bayesian incentive compatible, efficient, and balances budget. It is called the Arrow-d'Aspremont-Gerard-Varet (AGV) mechanism

(also called the expected externality mechanism). As we will see, the AGV mechanism is not dominant strategy incentive compatible and fails IIR.

The AGV mechanism $M^A \equiv (f^*, p^A)$ is defined as follows. The payment in M^A is defined as follows. For every agent i define the **expected welfare or remainder utility** of agents other than agent i, when agent i has type t_i as

$$r_i(t_i) = E_{t_{-i}}[W_{-i}(t_i, t_{-i})] = \int_{T_{-i}} [\sum_{k \neq i} f_k^*(t_i, t_{-i}) t_k] g_{-i}(t_{-i}) dt_{-i},$$

where g_{-i} is the distribution of types of agents other than agent i. Note that $r_i(t_i)$ is the expected welfare of agents other than i when agent i reports t_i. Then, the payment of agent i at type profile t is defined as,

$$p_i^A(t) = \frac{1}{n-1} \sum_{j \neq i} r_j(t_j) - r_i(t_i).$$

This mechanism is clearly budget-balanced since summing the payments of all the agents cancel out terms. The interpretation of this mechanism is the following. We can interpret $r_i(x_i)$ as the expected utility left for others when agent i he reports t_i. So, $\frac{1}{n-1} \sum_{j \neq i} r_j(x_j)$ is the average remainder utility of other agents. The term $r_i(x_i)$ is his own remainder utility. This difference is the payment.

Of course, this payment does not correspond to a Groves payment, and hence, this is not a dominant strategy incentive compatible mechanism. However, we show below that it is BIC.

THEOREM **23** *The AGV mechanism is BIC, efficient, and budget-balanced.*

Proof: The AGV mechanism is efficient by definition, and we have seen that it is budget-balanced. To see that it is Bayesian incentive compatible, consider agent i and suppose all other agents report truthfully and report t_{-i}. Consider $s_i, t_i \in T_i$. Now,

$$\begin{aligned}
\alpha_i^*(t_i) t_i - \pi_i^A(t_i) - \alpha_i^*(s_i) t_i + \pi_i^A(s_i) &= E_{t_{-i}}\big[t_i(f_i^*(t_i, t_{-i}) - f_i^*(s_i, t_{-i}) + r_i(t_i) - r_i(s_i)\big] \\
&= E_{t_{-i}}\big[t_i(f_i^*(t_i, t_{-i}) - f_i^*(s_i, t_{-i}) + W_{-i}(t_i, t_{-i}) - W_{-i}(s_i, t_{-i})\big] \\
&= E_{t_{-i}}\big[\sum_{j \in N} t_j(f_i^*(t_i, t_{-i}) - \sum_{j \in N} t_j f_j^*(s_i, t_{-i})\big] \\
&\geq 0,
\end{aligned}$$

where the last inequality followed from efficiency. Hence, the AGV is BIC. ■

We explain the AGV mechanism in the bilateral trading model. There are two agents - a buyer (denoted by b) and a seller (denoted by s). The buyer's values are drawn uniformly from $[0, 10]$ and the seller's values are drawn uniformly from $[-10, -5]$. Consider the type profile where the buyer has a value of 8 and the seller has a value of -7. By efficiency the object must be allocated to the buyer (since $8 - 7 > 0$). We now compute the remainder utility of every agent. For the buyer, the remainder utility at value 8 is

$$r_b(8) = \int_{-10}^{-5} a_s^*(8, x_s) x_s g_s(x_s) dx_s = \int_{-8}^{-5} x_s \frac{1}{5} dx_s = \frac{-39}{10}.$$

For the seller, the remainder utility at value -7 is

$$r_s(-7) = \int_{0}^{10} a_b^*(x_b, -7) x_b g_b(x_b) dx_b = \int_{7}^{10} x_b \frac{1}{10} dx_b = \frac{51}{20}.$$

Hence, the payments of the buyer and the seller is given as

$$p_b^A(8, -7) = r_s(-7) - r_b(8) = \frac{149}{20}$$
$$p_s^A(8, -7) = r_b(8) - r_s(-7) = \frac{-149}{20}.$$

Although in this instance, the AGV is also individually rational, in general, it is not IIR - we will prove a general theorem regarding this. On one hand, the pivotal mechanism is efficient, Bayesian incentive compatible, and individually rational but not budget-balanced. On the other hand, the AGV mechanism is efficient, Bayesian incentive compatible, and budget-balanced but may not be IIR.

We say a mechanism $M \equiv (f, p)$ runs an expected surplus if

$$E_t\big[\sum_{i \in N} p_i(t)\big] \geq 0.$$

Note that

$$\begin{aligned} E_t\big[\sum_{i \in N} p_i(t)\big] &= \int_T \big(\sum_{i \in N} p_i(t)\big) g(t) dx \\ &= \sum_{i \in N} \int_{T_i} \big(\int_{T_{-i}} p_i(t_i, t_{-i}) g_{-i}(t_{-i}) dt_{-i} \big) g_i(t_i) dx_i \\ &= \sum_{i \in N} \int_{T_i} \pi_i(t_i) g_i(t_i) dx_i \end{aligned}$$

Hence, an equivalent way of saying that a mechanism $M \equiv (f, p)$ runs an expected surplus is

$$\sum_{i \in N} E_{t_i}\big[\pi_i(t_i)\big] \geq 0.$$

THEOREM **24** *There exists an efficient, Bayesian incentive compatible, and interim individually rational mechanism which balances budget if and only if the modified pivotal mechanism runs an expected surplus.*

Proof: Suppose there exists a Bayesian incentive compatible mechanism $M = (f^*, p_1, \ldots, p_n)$ that is budget-balanced, efficient, and IIR. Assume for contradiction that the modified pivotal mechanism M^* does not run an expected surplus. Then $\sum_{i \in N} E_{t_i}\left[\pi_i^{M^*}(t_i)\right] < 0$. By Proposition 13, $\pi_i^{M^*}(t_i) \geq \pi_i^M(t_i)$ for all i and for all t_i. Hence,

$$\sum_{i \in N} E_{t_i}\left[\pi_i^M(t_i)\right] \leq \sum_{i \in N} E_{t_i}\left[\pi_i^{M^*}(t_i)\right] < 0.$$

This implies that $\sum_{i \in N} E_{t_i}\left[\pi_i^M(t_i)\right] < 0$. But

$$\sum_{i \in N} E_{t_i}\left[\pi_i^M(t_i)\right] = E_t\Big[\sum_{i \in N} p_i(t)\Big] = 0$$

since M is budget-balanced. This is a contradiction.

Now, suppose the pivotal mechanism runs an expected surplus. So, $\sum_{i \in N} E_{t_i}\left[\pi_i^M(t_i)\right] \geq 0$. Then, we will construct a mechanism which is efficient, Bayesian incentive compatible, individually rational, and budget-balanced. Define for every agent $i \in N$,

$$U_i^{M^*}(\ell_i) - U_i^{M^A}(\ell_i) = d_i,$$

where M^A is the AGV mechanism. Note that by Equation 5 for all $i \in N$ and for all $t_i \in T_i$, we have

$$U_i^{M^*}(t_i) - U_i^{M^A}(t_i) = d_i.$$

This means, for all $i \in N$ and for all $t_i \in T_i$, we have

$$\pi_i^A(t_i) - \pi_i^{M^*}(t_i) = d_i,$$

where π^A is the expected payment in the AGV mechanism. [13] This implies that

$$E_{t_i}\left[\pi_i^A(t_i) - \pi_i^{M^*}(t_i)\right] = E_{t_i} d_i = d_i$$

Then, we have

$$\sum_{i \in N} E_{t_i}\left[\pi_i^A(t_i)\right] - \sum_{i \in N} E_{t_i}\left[\pi_i^{M^*}(t_i)\right] = \sum_{i \in N} d_i.$$

[13] Note that Proposition 13 cannot be applied here. This is because the AGV mechanism is not IIR, and Proposition 13 applies only to BIC, efficient, and IIR mechanisms. Hence, we *cannot* conclude that $d_i \leq 0$ for all $i \in N$.

This means

$$E_t\big[\sum_{i\in N} p^A(t)\big] - E_t\big[\sum_{i\in N} p_i^{M^*}(t)\big] = \sum_{i\in N} d_i.$$

Using the fact that the AGV mechanism is budget-balanced and the pivotal mechanism runs an expected surplus, we get that $\sum_{i\in N} d_i \leq 0$.

Now, we define another mechanism $\bar{M} = (f^*, \bar{p})$ as follows. For every $i \in N$ with $i \neq 1$, and for every type profile x,

$$\bar{p}_i(t) = p_i^A(t) - d_i.$$

For agent 1, at every type profile t, his payment is

$$\bar{p}_1(t) = p_1^A(t) + \sum_{j\neq 1} d_j.$$

Note that $\bar{M}$ is produced from the AGV mechanism M^A. We have only added constants to the payments of agents of M^A, and the allocation rule has not changed from M^A. Hence, by revenue equivalence, $\bar{M}$ is also Bayesian incentive compatible. Also, since M^A is budget-balanced, by definition of $\bar{M}$, it is also budget-balanced.

We will show that $\bar{M}$ is individually rational. To show this, consider $i \neq 1$ and a type $t_i \in T_i$. Then,

$$U_i^{\bar{M}}(t_i) = U_i^{M^A}(t_i) + d_i = U_i^{M^*}(t_i) \geq 0,$$

where the inequality follows from the fact that the pivotal mechanism is individually rational. For agent 1, consider any type $t_1 \in T_1$. Then,

$$U_1^{\bar{M}}(t_1) = U_1^{M^A}(t_1) - \sum_{j\neq 1} d_j \geq U_1^{M^A}(t_1) + d_1 = U_1^{M^*}(t_1) \geq 0,$$

where the first inequality comes from the fact that $\sum_{j\in N} d_j \leq 0$ and the second inequality follows from the fact that the pivotal mechanism is individually rational. ■

17.4 Impossibility in Bilateral Trading

We will now show the impossibility of efficient Bayes-Nash implementation, IIR, and budget-balancedness in a model of bilateral trading. In this model there are two agents: a buyer, denoted by b, and a seller, denoted by s. Seller s has a privately known cost $c \in [c_l, c_u]$ and

buyer b has a privately known value $v \in [v_l, v_u]$. Suppose that $v_l < c_u$ and $v_u \geq c_l$ - this is to allow for trade in some type profiles and no trade in some type profiles. [14] If the object is sold and the price paid by the buyer is p_b and the price received by the seller is p_s, then the net payoff to the seller is $p_s - c$ and that to the buyer is $v - p_b$. Efficiency here boils down making trade whenever $v > c$. If $v > c$, the seller must produce the object at cost c and sell it to the buyer. Budget-balance requires that $p_b = p_s$.

The following theorem is attributed to Myerson and Satterthwaite, and is called the Myerson-Satterthwaite impossibility in bilateral trading.

THEOREM **25** *In the bilateral trading problem, there is no mechanism that is efficient, Bayesian incentive compatible, IIR, and budget-balanced.*

Proof: By Theorem 24, it is enough to show that the modified pivotal mechanism runs an expected deficit. For this, note that when the type profile is (v, c), the modified pivotal mechanism works as follows.

- If $v \leq c$, then there is no trade and no payments are made.
- If $v > c$, there is trade and the buyer pays $\max\{c, v_l\}$ and the seller receives $\min\{v, c_u\}$.

Consider a type profile (v, c) such that there is trade. This implies that $v > c$. Let p_s be the payment *received* by the seller and p_b be the payment *given* by the buyer. By definition, $p_s = \min\{v, c_u\}$ and $p_b = \max\{c, v_l\}$. Note that if $\min\{v, c_u\} = v$, then we know that $v > c$ (since trade is taking place) and $v \geq v_l$ (by definition). So, we have

$$p_s = v \geq \max\{c, v_l\} = p_b.$$

Similarly, if $\min\{v, c_u\} = c_u$, then we know that $c_u \geq c$ (by definition) and $c_u > v_l$ (by our assumption). Hence, again,

$$p_s = c_u \geq \max\{c, v_l\} = p_b.$$

Hence, the total payment at all the profiles where trade takes place is $p_b - p_s \leq 0$. Since there is a positive measure of profiles where $p_s > p_b$ (this happens whenever $\min\{v, c_u\} = v$), the expected payment from the modified pivotal mechanism is negative. By Theorem 24, there is no mechanism that is efficient, Bayesian incentive compatible, individually rational, and budget-balanced. ■

[14] If this condition is violated, then efficiency will require either trade in all type profiles or no trade in all type profiles.

17.5 Impossibility in Choosing a Public Project

We now apply our earlier result to the problem of choosing a public project. There are two choices available $A = \{0, 1\}$, where 0 indicates not choosing the public project and 1 indicates choosing the public project. There is a cost incurred if the public project is chosen, and it is denoted by c. There are n agents, denoted by the set $N = \{1, \ldots, n\}$. The value of each agent for the project is denoted by v_i. The set of possible values of agent i is denoted by $V_i = [0, b_i]$.

An allocation rule f gives a number $f(v) \in [0, 1]$ at every valuation profile v. The interpretation of $f(v)$ is the probability with which the public project is chosen. Let $\alpha(v_i)$ be the expected probability with which the public project is chosen if agent i reports v_i.

It is then easy to extend the previous analysis and show that $M \equiv (f, p)$ is Bayesian incentive compatible if and only if $\alpha(\cdot)$ is non-decreasing and the expected net utility of agent i at type v_i satisfies

$$U_i^M(v_i) = U_i^M(0) + \int_0^{v_i} \alpha(x_i)dx_i.$$

We say an allocation rule f^* is **cost-efficient** if at every type profile v, $f^*(v) = 1$ if $\sum_{i \in N} v_i \geq c$ and $f^*(v) = 0$ if $\sum_{i \in N} v_i < c$. We can now define the modified pivotal mechanism for f^*. Denote the total welfare of agents at a valuation profile v by

$$W(v) = [\sum_{i \in N} v_i - c]f^*(v)$$

Now, the payment in the modified pivotal mechanism is computed as follows. At valuation profile v, the payment of agent i is,

$$\begin{aligned} p_i^*(v) &= W(0, v_{-i}) - W_{-i}(v) = [\sum_{j \neq i} v_j - c]f^*(0, v_{-i}) - [\sum_{j \neq i} v_j - c]f^*(v) \\ &= [c - \sum_{j \neq i} v_j][f^*(v) - f^*(0, v_{-i})]. \end{aligned}$$

Now, fix a valuation profile v and an agent i. Note that $f^*(v) \geq f^*(0, v_{-i})$ for all v. If $f^*(0, v_{-i}) = 0$ and $f^*(v) = 1$, then $p_i^*(v) = c - \sum_{j \neq i} v_j$. But $f^*(0, v_{-i}) = 0$ implies that $c > \sum_{j \neq i} v_j$. Hence, $p_i^*(v) > 0$. Hence, $p_i^*(v) > 0$ if and only if $f^*(v) = 1$ but $f^*(0, v_{-i}) = 0$ - such an agent i is called a "pivotal agent". In all other cases, we see that $p_i^*(v) = 0$. Note that when $p_i^*(v) > 0$, we have $p_i^*(v) = c - \sum_{j \neq i} v_j \leq v_i$. Hence, the modified pivotal mechanism is individually rational. To see, why it has dominant strategy incentive compatibility, fix an agent i and a profile (v_i, v_{-i}). If the public project is not chosen then his net utility is zero.

Suppose he reports v_i', and the public project is chosen, then he pays $c - \sum_{j \neq i} v_j > v_i$, by definition (since the public project is not chosen at v, we must have this). Hence, his net utility in that case is $v_i - [c - \sum_{j \neq i} v_j] < 0$. So, this is not a profitable deviation. If the public project is chosen, he gets a non-negative utility, but reporting v_i' does not change his payment if the project is still chosen. If by reporting v_i', the project is not chosen, then his utility is zero. Hence, this is not a profitable deviation either.

Also, note that when agent i reports $v_i = 0$, then cost-efficiency implies that his net utility is zero in the modified pivotal mechanism - he pays zero irrespective of whether the project is chosen or not.

Using this, we can write that at any valuation profile v where the public project is chosen (i.e., $\sum_{i \in N} v_i \geq c$), the total payments of agents as follows. Let P be the set of pivotal agents at valuation profile v. Note that only pivotal agents make payments at any valuation profile.

$$\begin{aligned}
\sum_{i \in N} p_i^*(v) &= \sum_{i \in P} p_i^*(v) \\
&= \sum_{i \in P} [c - \sum_{j \neq i} v_j] \\
&= |P|c - \sum_{i \in P} \sum_{j \neq i} v_j \\
&= |P|c - (|P| - 1) \sum_{i \in P} v_i - |P| \sum_{i \notin P} v_i \\
&\leq |P|c - (|P| - 1) \sum_{i \in N} v_i \\
&\leq c,
\end{aligned}$$

where the equalities come from algebraic manipulation and the last inequality comes from the fact that $c \leq \sum_{i \in N} v_i$. Note that the last inequality is strict whenever $\sum_{i \in N} v_i > c$. Of course, if $P = \emptyset$, then $\sum_{i \in N} p_i^*(v) = 0$. Hence, the total payments in the modified pivotal mechanism is **always** less than or equal to c. Moreover, if there is a positive probability with which choosing the public project strictly better than not choosing (i.e., $\sum_{i \in N} v_i > c$), then the total payment in the modified pivotal mechanism is strictly less than c.

This means the total expected payment is also less than c. To see this, note that at any type profile v $\sum_{i \in N} \pi_i^*(v_i) = \sum_{i \in N} \int_{V_{-i}} p_i^*(v_i, x_{-i}) g_{-i}(x_{-i}) dx_{-i} < c$, where the strict inequality follows from the fact that for some type profiles (where public project is chosen), we have shown that the total payment in the modified pivotal mechanism is strictly less than c.

Now, consider any other cost-efficient, Bayesian incentive compatible, and individually

rational mechanism M. By revenue equivalence, the expected payment of agent i at value v_i of M and the modified pivotal mechanism M^* is related as follows:

$$U_i^M(v_i) - U_i^M(0) = U_i^{M^*}(v_i) - U_i^{M^*}(0) = U_i^{M^*}(v_i).$$

Using the fact that $U_i^M(0) \geq 0$, we get that $U_i^M(v_i) \geq U_i^{M^*}(v_i)$. Hence, like Proposition 13, the expected payments of each agent in M is no greater than the expected payment in the modified pivotal mechanism. This means for every type profile v, we have $\sum_{i \in N} \pi_i^M(v) \leq \sum_{i \in N} \pi_i^*(v) < c$. Then, there is some type profile v, at which the total payments of all the agents in M is less than c. This leads to the following result.

THEOREM **26** *Suppose that with positive probability, it is strictly better to choose the public project than not. Then, there is no cost-efficient, Bayesian incentive compatible, IIR mechanism which covers the cost of the public project.*

18 Multidimensional Mechanism Design

The analysis of optimal mechanism design and efficient and budget-balanced mechanism design was possible because of the one-dimensional type space assumed. The problem of finding similar results when the type of each agent is multidimensional is a significantly challenging problem. However, some of the results that we discussed can still be generalized to the multidimensional environment. We discuss this next.

For simplicity of exposition, we assume that there is only one agent. In this case, the solution concept will not matter - dominant strategy and Bayesian reduce to the same thing. However, if you want to extend this result to a multiple agent framework, you need to add for all t_{-i} in the dominant strategy implementation and integrate out over T_{-i} in the Bayesian implementation.

The notation will be as before. Let A be some finite set of alternatives and $\mathcal{L}(A)$ be the set of all lotteries over A. There is a single agent. The type of the agent is $t \in \mathbb{R}^{|A|}$. Here, we will use $t(a)$ to denote the value of the agent for alternative a. The type space of the agent is some set $T \subseteq \mathbb{R}^{|A|}$. Some examples are useful to see the applicability of this setting.

- **Multi-object auction with unit demand.** A seller is selling a set of objects to a buyer who can be assigned at most one object. The value for the buyer for each object is his type. The set of alternatives is the set of objects (and the alternative $\emptyset$ indicating not being assigned to any object).

- **Combinatorial auction.** This is the same model as the previous one but now the buyer can buy multiple objects. Hence, the set of alternatives is the set of all subsets of objects. The value for each subset is the type of the agent.

- **Public project selection.** A planner needs to choose a project from multiple projects. The value of the agent for each project is his type.

Like in voting problems, it is expected that not all vectors in $\mathbb{R}^{|A|}$ are allowed to be types. Hence, the type space can be a strict subset of $\mathbb{R}^{|A|}$ with some restrictions. For instance, in the combinatorial auction problem, we may require that for any pair of object a and b, at any type t, $t(\{a,b\}) = t(a) + t(b)$. This puts restrictions on how the type space looks. In the public project problem, type vector may be single peaked with respect to some exogenous ordering of the projects.

We will assume that all these restrictions are embedded in T. As in the analysis of the single object auction, we will first give a characterization of all incentive compatible mechanisms.

18.1 Incentive Compatible Mechanisms

A mechanism consists of an allocation rule $f : T \to \mathcal{L}(A)$ and a payment rule $p : T \to \mathbb{R}$. If type t is reported to the mechanism, then $f(t)$ is a probability distribution over alternatives at that type, where we denote by $f_a(t)$ the probability associated with alternative a. Hence, an agent with type s who reports type t to the mechanism (f, p) gets a net utility of

$$s \cdot f(t) - p(t),$$

where $s \cdot f(t) = \sum_{a \in A} s(a) f_a(t)$.

As before, we associate with a mechanism $M \equiv (f, p)$, a net utility function $\mathcal{U}^M : T \to \mathbb{R}$, defined as

$$\mathcal{U}^M(t) := t \cdot f(t) - p(t) \quad \forall\, t \in T,$$

which is the truth-telling net utility from the mechanism.

Definition **27** *A mechanism $M \equiv (f, p)$ is* **incentive compatible** *if for every $s, t \in T$, we have*

$$t \cdot f(t) - p(t) \geq t \cdot f(s) - p(s),$$

or equivalently,

$$\mathcal{U}^M(t) \geq \mathcal{U}^M(s) + (t - s) \cdot f(s).$$

An allocation rule f is **implementable** *if there exist a payment rule p such that (f,p) is incentive compatible.*

Our first step is to generalize the characterization of mechanisms in Theorem 16 to this environment. For this, we first need to define an appropriate notion of monotonicity of allocation rule in this type space. Since, the type is multidimensional, it is not clear how this can be defined. But the following is a well-known form of monotonicity in multidimensional environment.

DEFINITION **28** *An allocation rule f is* **monotone** *if for every $s,t \in T$, we have*

$$(t-s)\cdot\big(f(t)-f(s)\big)\geq 0.$$

This condition is often referred to as the **2-cycle monotonicity** condition. We will discuss the reasons below.

Our extension of Theorem 16 uses monotonicity.

THEOREM **27** *Suppose $T \subseteq \mathbb{R}^{|A|}$ is convex. A mechanism $M \equiv (f,p)$ is incentive compatible if and only if*

(a) f is monotone,

(b) for every $s,t \in T$,

$$\mathcal{U}^M(t) = \mathcal{U}^M(s) + \int_0^1 \psi^{s,t}(z)dz,$$

where $\psi^{s,t}(z) = (t-s)\cdot f(s+z(t-s))$ for all $z \in [0,1]$.

Proof: Suppose $M \equiv (f,p)$ is such that (a) and (b) hold. We will show that M is incentive compatible. Choose any $s,t \in T$.

STEP 1. We first show that for every $z,z' \in [0,1]$ with $z > z'$, we have $\psi^{s,t}(z) \geq \psi^{s,t}(z')$. Pick $z,z' \in [0,1]$ with $z > z'$. Since f is monotone, we have

$$\big[(s+z(t-s))-(s+z'(t-s))\big]\cdot\big[f(s+z(t-s))-f(s+z'(t-s))\big]\geq 0.$$

Simplifying, we get

$$(z-z')(t-s)\cdot\big[f(s+z(t-s))-f(s+z'(t-s))\big]\geq 0.$$

But $z > z'$ implies $(t-s)\cdot\big[f(s+z(t-s))-f(s+z'(t-s))\big] \geq 0$, which implies $\psi^{s,t}(z)-\psi^{s,t}(z') \geq 0$.

Step 2. Now, we can write

$$\begin{aligned}\mathcal{U}^M(t) - \mathcal{U}^M(s) - (t-s)\cdot f(s) &= \int_0^1 \psi^{s,t}(z)dz - (t-s)\cdot f(s)\\ &\geq \psi^{s,t}(0) - (t-s)\cdot f(s)\\ &= 0,\end{aligned}$$

where the first equality follows from (b), the second inequality from Step 1 (non-decreasingness of $\psi^{s,t}$), and the last equality follows from the fact that $\psi^{s,t}(0) = (t-s)\cdot f(s)$. This shows that M is incentive compatible.

Now, for the other direction, we assume that $M \equiv (f,p)$ is incentive compatible. We show (a) and (b) some steps. Consider any $s,t \in T$.

Step A. Since M is incentive compatible, we get

$$\begin{aligned}t\cdot f(t) - p(t) &\geq t\cdot f(s) - p(s)\\ s\cdot f(s) - p(s) &\geq s\cdot f(t) - p(t).\end{aligned}$$

Adding these two inequalities, we get $(t-s)\cdot(f(t)-f(s)) \geq 0$. Hence, f is monotone.

Step B. We define for every $z \in [0,1]$,

$$\phi(z) := \mathcal{U}^M(s + z(t-s)).$$

We now show that ϕ is a convex function. To see this, pick $\bar{z}, \hat{z} \in [0,1]$ and $\lambda \in (0,1)$. Let $\tilde{z} = \lambda\bar{z} + (1-\lambda)\hat{z}$. Since M is incentive compatible, we get

$$\begin{aligned}\phi(\bar{z}) = \mathcal{U}^M(s+\bar{z}(t-s)) &\geq \mathcal{U}^M(s+\tilde{z}(t-s)) + (\bar{z}-\tilde{z})(t-s)\cdot f(s+\tilde{z}(t-s))\\ &= \phi(\tilde{z}) + (\bar{z}-\tilde{z})(t-s)\cdot f(s+\tilde{z}(t-s)).\end{aligned}$$

Similarly, we have

$$\phi(\hat{z}) \geq \phi(\tilde{z}) + (\hat{z}-\tilde{z})(t-s)\cdot f(s+\tilde{z}(t-s)).$$

Multiplying the first inequality by λ and the second one by $(1-\lambda)$ and summing them, we get

$$\lambda\phi(\bar{z}) + (1-\lambda)\phi(\hat{z}) \geq \phi(\tilde{z}).$$

This show that ϕ is convex.

Next, incentive compatibility of M implies that for every $z, z' \in [0,1]$, we have

$$\mathcal{U}^M(s + z(t-s)) \geq \mathcal{U}^M(s + z'(t-s)) + (z - z')(t-s) \cdot f(s + z'(t-s)).$$

This implies that for every $z, z' \in [0,1]$, we have

$$\phi(z) \geq \phi(z') + (z - z')\psi^{s,t}(z').$$

Hence, $\psi(z')$ is the subgradient of the convex function ϕ at z'. By Lemma 13, we get that for every $z' \in [0,1]$,

$$\phi(z') = \phi(0) + \int_0^{z'} \psi^{s,t}(z)dz.$$

Hence,

$$\phi(1) = \phi(0) + \int_0^1 \psi^{s,t}(z)dz.$$

Substituting, we get

$$\mathcal{U}^M(t) = \mathcal{U}^M(s) + \int_0^1 \psi^{s,t}(z)dz.$$

■

Revenue/payoff equivalence. Theorem 27 immediately implies a payoff equivalence result. Consider two incentive compatible mechanisms $M = (f, p)$ and $M' = (f, p')$ using the same allocation rule f. Fix some type $t^0 \in T$. By Theorem 27, for every $t \in T$,

$$\mathcal{U}^M(t) - \mathcal{U}^M(t^0) = \mathcal{U}^{M'}(t) - \mathcal{U}^{M'}(t^0).$$

Hence, mechanisms M and M' assign different net utilities to the agent if and only if $\mathcal{U}^M(t^0)$ and $\mathcal{U}^{M'}(t^0)$ are different. In other words, if two incentive compatible mechanisms use the same allocation rule and assign the same net utility to the agent at *some* type, then they must assigning the same net utility to the agent at *all* types. This is known as the **payoff equivalence** result.

One-dimensional problems. We remark that monotonicity reduces to "non-decreasingness" discussed in Theorem 16 for one-dimensional problem. We say a type space T is **one-dimensional** if there exists an alternative $a^* \in A$ such that $t(a) = 0$ for all $a \neq a^*$ and for all $t \in T$. In the single object auction setting a^* is the alternative where the agent wins the object. Note that if T is one-dimensional, then for any $s, t \in T$, $(t - s)$ is a vector

whose components corresponding to any alternative $a \neq a^*$ are zero. Hence, for any $s, t \in T$, $(t-s)(f(t)-f(s))$, can be written as

$$\big(t(a^*) - s(a^*)\big)\big(f_a(t) - f_a(s)\big).$$

Monotonicity requires that the above term is non-negative. This is equivalent to saying that if $t(a^*) > s(a^*)$, then $f_a(t) \geq f_a(s)$.

In one-dimensional problem statement (b) in Theorem 27 also simplifies - compare it with the analogue statement in Theorem 16. Suppose the value for alternative a^* is lies in $[\ell, H]$. For any $x \in [\ell, H]$, we write the unique type t with $t(a^*) = x$ as t^x. Now, fix a mechanism $M \equiv (f, p)$. Then, statement (b) is equivalent to requiring that for any $x, y \in [\ell, H]$, we must have

$$\begin{aligned}\mathcal{U}^M(t^x) &= \mathcal{U}^M(t^y) + \int_0^1 (t^x - t^y) \cdot f(t^y + z(t^x - t^y))dz \\ &= \mathcal{U}^M(t^y) + \int_0^1 (x-y) f_{a^*}(t^y + z(t^x - t^y))dz\end{aligned}$$

Define $\phi(x') := f_{a^*}(x')$ for all $x' \in [\ell, H]$. So, the above equation reduced to

$$\mathcal{U}^M(t^x) = \mathcal{U}^M(t^y) + \int_0^1 (x-y)\phi(y + z(x-y))dz \qquad = \mathcal{U}^M(t^y) + \int_y^x \phi(x')dx'$$

Now, if we only require for every $x \in [\ell, H]$,

$$\mathcal{U}^M(t^x) = \mathcal{U}^M(t^\ell) + \int_\ell^x \phi(x')dx',$$

then this will imply that for any $x, y \in [\ell, H]$

$$\begin{aligned}\mathcal{U}^M(t^x) - \mathcal{U}^M(t^y) &= \int_\ell^x \phi(x')dx' - \int_\ell^y \phi(x')dx' \\ &= \int_y^x \phi(x')dx',\end{aligned}$$

as desired in (b). This explains the weaker analogue of (b) in the one-dimensional version in Theorem 16. However, in the multidimensional case we need the stronger version as stated in (b) of Theorem 27. In other words, when type space is multidimensional, requiring (b) in Theorem 27 to hold for every $t \in T$ with respect to some "base" type s_0 does not imply (b) to hold for every pair of types $s, t \in T$.

18.2 The Implementation Problem

We now turn to the implementation problem, i.e., identifying conditions on an allocation rule that characterizes implementability. Corollary 2 achieves this in the one-dimensional type space. It shows that non-decreasingness of an allocation rule characterizes implementability in the one-dimensional type space. Since monotonicity is the natural generalization (as we showed above) of non-decreasingness for multidimensional type space, a natural conjecture is then that monotonicity is equivalent to implementability. This conjecture is false. The reason for this is the same reason why (b) in Theorem 27 is stronger than the analogue statement in Theorem 16. In one-dimensional type space if an allocation rule is non-decreasing, then fixing the $\mathcal{U}^M$ value for the "lowest" type uniquely fixes the value of $\mathcal{U}^M$ for all other types using (b), and this automatically ensures the statement (b). However, in multidimensional type space, fixing $\mathcal{U}^M$ for some "base" type and then using (b) to fix the value of $\mathcal{U}^M$ for all other types does not ensure (b) to hold for all pairs of types.

To extend Corollary 2, we need a stronger version of monotonicity. Consider an implementable allocation rule f and two types $s, t \in T$. Since f is implementable there exist a payment rule p such that the mechanism $M \equiv (f, p)$ is incentive compatible. Then,

$$t \cdot f(t) - p(t) \geq t \cdot f(s) - p(s)$$

$$s \cdot f(s) - p(s) \geq s \cdot f(t) - p(t).$$

Adding these two constraints, we get that $(t - s) \cdot \big(f(t) - f(s)\big) \geq 0$, i.e., f is monotone. We can do this exercise for a longer sequence of types too. For instance, take three types $s, t, x \in T$ and consider the incentive constraints

$$t \cdot f(t) - p(t) \geq t \cdot f(s) - p(s)$$

$$s \cdot f(s) - p(s) \geq s \cdot f(x) - p(x)$$

$$x \cdot f(x) - p(x) \geq x \cdot f(t) - p(t).$$

Again, adding these constraints will cancel the payment terms and we will be left with only a condition on allocation rules.

To define this longer sequence condition, we define some notation. Let $\ell^f(s, t) = t \cdot \big(f(t) - f(s)\big)$ for every $s, t \in T$. Note that incentive constraint from true type t to type s is: $p(t) - p(s) \leq \ell^f(s, t)$. A good way to interpret this is that we create a directed graph G^f with set of nodes T (possibly infinite nodes). For every pair of nodes $s, t \in T$, we put an edge from s to t and another from t to s. So, G^f is a complete directed graph. We assign a weight

of $\ell^f(s,t)$ to the edge from s to t. Monotonicity requires that the for every $s,t \in T$, we must have $\ell^f(s,t) + \ell^f(t,s) \geq 0$, i.e., 2-cycles (cycles involving pairs of nodes) have non-negative length. The longer sequence condition requires cycles of arbitrary number of nodes must have non-negative length.

DEFINITION **29** *An allocation rule satisfies* **cycle monotonicity** *if for any finite sequence of types $(s^1, \ldots, s_k)$ each belonging to T, we have*

$$\sum_{j=1}^{k} \ell^f(s^j, s^{j+1}) \geq 0,$$

where $s^{k+1} \equiv s^1$.

Using ideas explained above, it is routine to verify that every implementable allocation rule satisfies cycle monotonicity. The following theorem shows that the converse also holds - the theorem does not require any assumption on type spaces (Theorem 27 required the type space to be convex).

THEOREM **28** *An allocation rule is implementable if and only if it is cyclically monotone.*

Proof: Suppose f is an implementable allocation rule. Consider a finite and distinct sequence of types $(t^1, t^2, \ldots, t^k)$ with $k \geq 2$. Since f is implementable, there exists a payment rule p such that

$$\begin{aligned}
p(t^2) - p(t^1) &\leq \ell^f(t^1, t^2) \\
p(t^3) - p(t^2) &\leq \ell^f(t^2, t^3) \\
\ldots &\leq \ldots \\
\ldots &\leq \ldots \\
p(t^k) - p(t^{k-1}) &\leq \ell^f(t^{k-1}, t^k) \\
p(t^1) - p(t^k) &\leq \ell^f(t^k, t^1).
\end{aligned}$$

Adding these inequalities, we obtain that $\ell^f(t^1,t^2)+\ell^f(t^2,t^3)+\ldots+\ell^f(t^{k-1},t^k)+\ell^f(t^k,t^1) \geq 0$.

Now, suppose f satisfies cycle monotonicity. For any two types $s,t \in T$, let $P(s,t)$ denote the set of all (finite) paths from s to t in G^f. The set $P(s,t)$ is non-empty because the direct edge from s to t in G^f always exists. Define the **shortest path** length from s to t ($s \neq t$) as follows.

$$dist(s,t) = \inf_{P \in P(s,t)} \ell^f(P),$$

where $\ell^f(P)$ is the length of path P. Let $dist(s,s) = 0$ for all $s \in T$. First, we show that $dist(s,t)$ is finite. Consider any path $P \in P(s,t)$. By cycle monotonicity, $\ell^f(P) \geq -\ell^f(t,s)$. Hence, $dist(s,t) \geq -\ell^f(t,s)$. Since $\ell^f(t,s)$ is bounded, $dist(s,t)$ is finite.

Now, fix a type $r \in T$. Consider two types $s,t \in T$. We first prove a useful lemma.

LEMMA **17** *Suppose f satisfies cycle monotonicity. For any $r,s,t \in T$ with $s \neq t$, we have $dist(r,t) \leq dist(r,s) + \ell^f(s,t)$.*

Proof: If $r = t$, $dist(r,t) = dist(r,r) = 0$. By cycle monotonicity, $dist(t,s) \geq -l(s,t)$ or $dist(r,s) + \ell^f(s,t) = dist(t,s) + \ell^f(s,t) \geq 0 = dist(r,r) = dist(r,t)$. If $r = s$, then $dist(r,t) \leq \ell^f(r,t) = dist(r,s) + \ell^f(s,t)$. If $r \neq s \neq t$, consider any path P from r to s. We distinguish between two possible cases.

CASE 1: Path P contains t. In that case, let Q_1 be the path from r to t in P and Q_2 be the path from t to s. Hence, $\ell^f(P) = \ell^f(Q_1) + \ell^f(Q_2)$. Adding $\ell^f(s,t)$ on both sides, we get $\ell^f(P) + \ell^f(s,t) = \ell^f(Q_1) + \ell^f(Q_2) + \ell^f(s,t)$. Using cycle monotonicity, we get $\ell^f(Q_2) + \ell^f(s,t) \geq 0$, and hence, $\ell^f(P) + \ell^f(s,t) \geq \ell^f(Q_1) \geq dist(r,t)$. Hence, $\ell^f(P) + \ell^f(s,t) \geq dist(r,t)$.

CASE 2: Path P does not contain t. In that case, by definition $dist(r,t) \leq \ell^f(P) + \ell^f(s,t)$, i.e., $\ell^f(P) + \ell^f(s,t) \geq dist(r,t)$.

Hence, in both cases, we see $\ell^f(P) + \ell^f(s,t) \geq dist(r,t)$. Since this holds for every path from r to s, we have $dist(r,s) + \ell^f(s,t) \geq dist(r,t)$. ■

Now, define the following payment rule: let $p(s) = dist(r,s)$ for all $s \in T$.

Take any $s,t \in T$. We have $p(t) - p(s) = dist(r,t) - dist(r,s) \leq \ell^f(s,t)$, where the inequality follows from Lemma 17. Hence, f is implementable. ■

As noted cycle monotonicity is a significantly stronger condition than monotonicity. We say f is **deterministic** if for all $t \in T$ and for all $a \in A$, $f_a(t) \in \{0,1\}$. Monotonicity has been shown to imply cycle monotonicity if type space is convex and allocation rule is deterministic. We state this as a theorem below without giving a proof.

THEOREM **29** *Suppose T is convex and let $f : T \to \mathcal{L}(A)$ be a deterministic allocation rule. Then, f is implementable if and only if it is monotone.*

18.3 Optimal Multi-Object Auction

Our discussions so far have shown how many of the results for one-dimensional mechanism design can be extended when the type space is multidimensional. Though, it gives an expression for payment (via (b) of Theorem 27), this expression is not as easy to handle because the expectation over the type space is now complicated. As a result, the Myersonian technique that we employed for the one-dimensional type space does not yield any useful result. It is still an open question on how to derive optimal multi-object auction even for the two object case.

References

E. Clarke. Multipart pricing of public goods. *Public Choice*, 8:19–33, 1971.

T. Groves. Incentives in teams. *Econometrica*, 41:617–663, 1973.

J. C. Harsanyi. Games of incomplete information played by 'bayesian' players. *Management Science*, 14:159–189, 320–334, 486–502, 1967-68.

Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.