NOVEL MECHANISMS FOR ALLOCATION OF HETEROGENEOUS ITEMS IN STRATEGIC SETTINGS

A Thesis
Submitted in Partial Fulfilment of the
Requirements for the Degree of
Doctor of Philosophy
in the Faculty of Engineering

by
GUJAR SUJIT PRAKASH

Under the Supervision of
PROF. Y. NARAHARI

COMPUTER SCIENCE AND AUTOMATION
INDIAN INSTITUTE OF SCIENCE
BANGALORE, INDIA
Dedicated

to

My Parents
Acknowledgements

It is a pleasure to thank those who made this thesis possible. I owe my deepest gratitude to my research supervisor, Prof Dr. Y Narahari. Without his enthusiasm, his inspiration, and his ability to explain things in a clear and precise manner, this thesis would not have been possible. I feel extremely fortunate to have him as my advisor. I am further in debt to him for guiding me in various kinds of academic and professional activities. He has supported me in infinite number of ways. He is not just a research supervisor, but a Guru, whose support extends beyond the research.

I wish to express my deep acknowledgments to Prof. David Parkes from Harvard University, for supervising my research on “dynamic mechanisms without money” and very insightful discussions. It had been a great experience to work with him. I would take this opportunity to thank James Zou with whom I had many brainstorming discussions and whose company made my stay at Harvard a memorable one. My special thanks to Akash Das Sarma for extremely useful discussions.

I am thankful to Infosys Technologies Limited for awarding me Infosys Fellowship and travel grants for attending various conferences/workshops.

No words can describe my gratitude to my Parents, Ma and Baba, who have been constantly supportive and source of inspiration. I dedicate this thesis to them. I am indebted to all those, notably sister Rajashri and brother in law, Sachin Gujar who have rendered lot of support to my family during my absence.

It is an honor for me to thank Naresh Manwani, Radhika Arava, N Rama Suri, Shailesh Patil, Sunil Shelke, and Vivek Gupta for useful and meaningful discussions on various topics over a cup of coffee or across lunches. They have proof-read my papers on several occasions and their company made my stay at IISc an unforgettable experience. I wish to thank my lab mates Moon Chetry, Pankaj Dayama, Rohith Vallam, and Swaprava Nath for reading various chapters of the thesis and suggesting useful changes and also creating a friendly ambiance at e-commerce lab. I want to express my gratitude to my friends Balakrishna Dawari, Deepak Hinge, Dinesh Garg, Hari Gupta, Hrishikesh Gadgil, Madhuri Daware, Megha Mohabey, Milind Kulkarni, Monali Bhatre, Neeraj Kumbharkarna, Neelesh Pimprikar, Pankaj Bhambhani, Pradeep Murugessan, Raghav Gautam, Rakesh Sharan, Ramkrishnan Kannan, Sachin Shinde, Sameer
Pawar, Shivaji Gholap, Swati Aggrawal, Vinayak Kulkarni, Vinod Mandalia, and Viral Solanki who have been a great friend circle and due to which I had a wonderful stay in IISc. I would like to take this opportunity to express special thanks to my fiancée Shradha and also to my mother-in-law and father-in-law.

I owe a lot to all the teachers, faculty from IISc, who have helped me and inspired me along the way. I am grateful to all the staff at the Department of Computer Science and Automation for their support and timely help. Last but not the least, I thank the anonymous reviewers of my papers whose valuable feedback have helped a lot in my research.
Abstract

Allocation of objects or resources to competing agents is a ubiquitous problem in the real world. For example, a federal government may wish to allocate different types of spectrum licenses to telecom service providers; a search engine has to assign different sponsored slots to the ads of advertisers; etc. The agents involved in such situations have private preferences over the allocations. The agents, being strategic, may manipulate the allocation procedure to get a favourable allocation. If the objects to be allocated are heterogeneous (rather than homogeneous), the problem becomes quite complex. The allocation problem becomes even more formidable in the presence of a dynamic supply and/or demand. This doctoral work is motivated by such problems involving strategic agents, heterogeneous objects, and dynamic supply and/or demand. In this thesis, we model such problems in a standard game theoretic setting and use mechanism design to propose novel solutions to the problems. We extend the current state-of-the-art in a non-trivial way by solving the following problems:

- Optimal combinatorial auctions with single minded bidders, generalizing the existing methods to take into account multiple units of heterogeneous objects
- Multi-armed bandit mechanisms for sponsored search auctions with multiple slots, generalizing the current methods that only consider a single slot.
- Strategyproof redistribution mechanisms for heterogeneous objects, expanding the scope of the current state of practice beyond homogeneous objects
- Online allocation mechanisms without money for one-sided and two-sided matching markets, extending the existing methods for static settings.
Research Papers Based on the Thesis Work

Journal Publications


Book Chapter


Conference Publications


James Zou, Sujit Gujar and David Parkes, “Tolerable Manipulability in Dynamic Assignment without Money”. In the Proceedings 24th AAAI Conference on Artificial Intelligence (AAAI '10), 2010.


Papers Submitted to Journals


Sujit Gujar and Y. Narahari, “Optimal Multi-Unit Combinatorial Auctions”.

viii
Acronyms and Notation

Chapter 2
Mechanism Design Theory: An Overview

MD  Mechanism Design
SCF  Social Choice Function
IC  Incentive Compatibility (Compatible)
DSIC  Dominant Strategy Incentive Compatibility (Compatible)
BIC  Bayesian Incentive Compatibility (Compatible)
AE  Allocative Efficiency (Allocatively Efficient)
BB  Budget Balance
N  Set of n agents

Θi  Type set of Agent i
Θ  = Θ1 × ... × Θn
Θ−i  = Θ1 × ... × Θi−1 × Θi+1 × ... × Θn

θi  Actual type of agent i, θi ∈ Θi
θ  = (θ1, ..., θn)
θ−i  = (θ1, ..., θi−1, θi+1, ..., θn)

ˆθi  Reported type of agent i, ˆθi ∈ Θi
ˆθ  = (ˆθ1, ..., ˆθn)
ˆθ−i  = (ˆθ1, ..., ˆθi−1, ˆθi+1, ..., ˆθn)

Φi(.)  A CDF on Θi
ϕi(.)  A PDF on Θi
Φ(.)  A CDF on Θ
ϕ(.)  A PDF on Θ

X  Outcome Set
x  A particular outcome, x ∈ X

ui(.)  Utility function of agent i

f(.)  A social choice function

M  An indirect mechanism
A direct revelation mechanism

\( g(\cdot) \) Outcome rule of an indirect mechanism

\( C_i \) Set of actions available to agent \( i \) in an indirect mechanism

\( C = C_1 \times \ldots \times C_n \)

\( b_i \) Bid of agent \( i \)

\( b = (b_1, \ldots, b_n) \)

\( b_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n) \)

\( b^{(k)} \) \( k^{th} \) highest element in \( (b_1, \ldots, b_n) \)

\( (b_{-i})^{(k)} \) \( k^{th} \) highest element in \( (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n) \)

\( s_i(\cdot) \) Strategy of agent \( i \)

\( s(\cdot) = (s_1(\cdot), \ldots, s_n(\cdot)) \)

\( K \) Set of project choices

\( k \) A particular project choice, \( k \in K \)

\( t_i \) Monetary transfer to agent \( i \)

\( v_i(\cdot) \) Valuation function of agent \( i \)

\( U_i(\cdot) \) Expected utility function of agent \( i \)

\( \mu \) Matching

\( M \) Set of Men

\( W \) Set of Women

DA Male-proposal Deferred Acceptance

TTCA Top Trading Cycle Algorithm

Chapter 3

Optimal Multi-Unit Combinatorial Auctions

\( I \) Set of items to be procured

\( D_j \) Demand for item \( j \)

\( N \) Set of sellers

\( c_i \) True cost of production of one unit of bundle of interest to the seller \( i \), \( c_i \in [c_i^l, c_i^u] \)

\( q_i \) True capacity for bundle which seller \( i \) can supply, \( q_i \in [q_i^l, q_i^u] \)

\( \hat{c}_i \) Reported cost by the seller \( i \)

\( \hat{q}_i \) Reported capacity by the seller \( i \)

\( \theta_i \) True type i.e. cost and capacity of the seller \( i \), \( \theta_i = (c_i, q_i) \)

\( b_i \) Bid of the seller \( i \), \( b_i = (\hat{c}_i, \hat{q}_i) \)

\( b \) Bid vector, \( (b_1, b_2, \ldots, b_n) \)

\( b_{-i} \) Bid vector without the seller \( i \), i.e. \( (b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n) \)
\( t_i(b) \)  
Payment to the seller \( i \) when submitted bid vector is \( b \)

\( T_i(b_i) \)  
Expected payment to the seller \( i \) when he submits bid \( b_i \).  
Expectation is taken over all possible values of \( b_{-i} \)

\( x_i = x_i(b) \)  
Quantity of the bundle to be procured from the seller \( i \) when the bid vector is \( b \)

\( X_i(b_i) \)  
Expected quantity of the bundle to be procured from the seller \( i \) when he submits bid \( b_i \).  
Expectation is taken over all possible values of \( b_{-i} \)

\( f_i(c_i, q_i) \)  
Joint probability density function of \( (c_i, q_i) \)

\( F_i(c_i, q_i) \)  
Cumulative distribution function of \( f_i(c_i, q_i) \)

\( f_i(c_i | q_i) \)  
Conditional probability density function of production cost when it is given that the capacity of the seller \( i \) is \( q_i \)

\( F_i(c_i | q_i) \)  
Cumulative distribution function of \( f_i(c_i | q_i) \)

\( H_i(c_i, q_i) \)  
Virtual cost function for seller \( i \),
\[ H_i(c_i, q_i) = c_i + \frac{F_i(c_i | q_i)}{f_i(c_i | q_i)} \]

\( \rho_i(b_i) \)  
Expected offered surplus to seller \( i \), when his bid is \( b_i \)

\( u_i(b, \theta_i) \)  
Utility to seller \( i \), when bid vector is \( b \) and his type is \( \theta_i \)

\( U_i(b_i, \theta_i) \)  
Expected utility to the seller \( i \), when he submits bid \( b_i \) and his type is \( \theta_i \).  
Expectation is taken over all possible values of \( b_{-i} \)

OCAS  
Optimal Combinatorial Auction with Single minded bidders

VD-OCAS  
Optimal Combinatorial Auction with Single minded bidders offering Volume Discounts

**Chapter 4**

**Truthful Multi-Armed Bandit Mechanisms for Multi-Slot Sponsored Search Auctions**

\( K = \{1, 2, \ldots, k\} \), Set of agents

\( M = \{1, 2, \ldots, m\} \) Set of slots

\( i \)  
Index of an agent, \( i = 1, 2, \ldots, k \)

\( j \)  
Index of a slot, \( j = 1, 2, \ldots, m \)

\( T \)  
Total number of rounds

\( t \)  
A particular round \( t \in \{1, 2, \ldots, T\} \)

\( \rho_{ij}(t) = 1 \) if agent \( i \) gets a click in slot \( j \) in round \( t \)

\( \rho(t) = 0 \) otherwise

\( \rho(t) = (\rho_{ij}(t))_{i \in K, j \in M} \)
\[ \rho = (\rho(1), \rho(2), \ldots, \rho(T)) \]

\[ v_i \quad \text{Agent } i \text{'s valuation of a click to her ad} \]

\[ b_i \quad \text{Bid by agent } i \]

\[ b_{-i} \quad \text{Bid vector of bids of all the agents except } i \]

\[ b \quad \text{Bid vector, indicating bids of all the agents} \]

\[ = (b_i, b_{-i}) = (b_1, b_2, \ldots, b_k) \]

\[ A_{ij}(b, \rho, t) = 1 \text{ If an agent } i \text{ is allocated slot } j \text{ in round } t \]

\[ = 0 \text{ otherwise} \]

\[ A(b, \rho, t) = \left(A_{ij}(b, \rho, t)\right)_{i \in K, j \in M} \]

\[ A(b, \rho) = (A(b, \rho, 1), A(b, \rho, 2), \ldots, A(b, \rho, T)), \text{ Allocation rule} \]

\[ C_i(b, \rho) \quad \text{Total number of clicks obtained by an agent } i \text{ in } T \text{ rounds} \]

\[ P_i(b, \rho) \quad \text{Payment made by agent } i \]

\[ P(b, \rho) = (P_1(\cdot), P_2(\cdot), \ldots, P_k(\cdot)), \text{ Payment rule} \]

\[ U_i(v_i, b, \rho) \quad \text{Utility of an agent } i \text{ in } T \text{ rounds} \]

\[ = v_i C_i(b, \rho) - P_i(b, \rho) \]

\[ b_i^+ \quad \text{A real number } > b_i \]

\[ \alpha_i \quad \text{Click probability associated with agent } i \]

\[ \beta_j \quad \text{Click probability associated with slot } j \]

\[ \mu_{ij} \quad \text{The probability that an ad of an agent } i \text{ receives click when the agent} \]

\[ \text{is allotted slot } j. \]

\[ N(b, \rho, i, t) \quad \text{Set of slot agent pairs in round } t \text{ that influence agent} \]

\[ i \text{ in some future rounds} \]

\[ \text{CTR} \quad \text{Click Through Rate (Click Probability)} \]

\[ \text{DSIC} \quad \text{Dominant Strategy Incentive Compatible} \]

---

**Chapter 5**

**Redistribution Mechanisms for Assignment of Heterogeneous Objects**

\[ n \quad \text{Number of agents} \]

\[ N \quad \text{Set of the agents } = \{1, 2, \ldots, n\} \]

\[ p \quad \text{Number of objects} \]

\[ i \quad \text{Index for an agent, } i = 1, 2, \ldots, n \]

\[ j \quad \text{Index for object, } j = 1, 2, \ldots, p \]

\[ \mathbb{R}_+ \quad \text{Set of positive real numbers} \]

\[ \Theta_i \quad \text{The space of valuations of agent } i, \Theta_i = \mathbb{R}_+^p \]

\[ b_i \quad \text{Bid submitted by agent } i, = (b_{i1}, b_{i2}, \ldots, b_{ip}) \in \Theta_i \]
\( b \) \((b_1, b_2, \ldots, b_n)\), the bid vector

\( K \) The set of all allocations of \( p \) objects to \( n \) agents, each getting at most one object

\( k(b) \) An allocation, \( k(\cdot) \in K \), corresponding to the bid profile \( b \)

\( k^*(b) \) An allocatively efficient allocation when the bid profile is \( b \)

\( k^*_{-i}(b) \) An allocatively efficient allocation when the bid profile is \( b \) and agent \( i \) is excluded from the system

\( v_i(k(b)) \) Valuation of the allocation \( k \) to the agent \( i \), when \( b \) is the bid profile

\( v \) \( v : K \rightarrow \mathbb{R} \), the valuation function, \( v(k(b)) = \sum_{i \in N} v_i(k(b)) \)

\( t_i(b) \) Payment made by agent \( i \) in the Clarke pivotal mechanism, when the bid profile is \( b \), \( t_i(b) = v_i(k^*(b)) - (v(k^*(b)) - v(k^*_{-i}(b))) \)

\( t(b) \) The Clarke payment, that is, the total payment received from all the agents, \( t(b) = \sum_{i \in N} t_i(b) \)

\( t^{-i} \) The Clarke payment received in the absence of the agent \( i \)

\( r_i(b) \) Rebate to agent \( i \) when bid profile is \( b \)

\( e \) The redistribution index of the mechanism,

---

**Chapter 6**

**Dynamic Stable Matching in Two-Sided Markets**

\( \mu \) Matching

\( M \) \( \{m_1, m_2, \ldots, m_n\} \), Set of Men

\( W \) \( \{w_1, w_2, \ldots, w_n\} \), Set of Women

\( T \) Number of rounds in matching game

DA Male-proposal Deferred Acceptance

GSODAS General period Stable Online Deferred Acceptance with Substitutes

ROMA Randomized Online Matching Algorithm

---

**Chapter 7**

**Dynamic House Allocation**

\( H \) \( \{h_1, h_2, \ldots, h_n\} \), Set of distinct houses

\( N \) \( \{1, 2, \ldots, n\} \), Set of agents

\( \succ_i \) Strict preference of agent \( A_i \) on set of house

\( \succ_{-i} \) Preference profile of agents except \( A_i \)

\( Sched_N \) Set of all possible arrival and departure schedules
A particular arrival-departure schedule of the agents, \( \rho \in Sched_N \)

\( \succ \) \((\succ_1, \succ_2, \ldots, \succ_n) = (\succ_i, \succ_{-i})\), Preference profile

\( U \) Set of all possible strict preference profiles, \( \succ \in U \)

\( x \) \( x : N \to H \), A house allocation

\( x(i) \) House allocated to agent \( i \)

\( X \) Set of all possible feasible house allocations

\( f \) \( f : U \times Sched_N \to X \), House allocation mechanism

**TTCA** Top Trading Cycle Algorithm

**DO-TTCA** Departing agents Online TTCA

**T-TTCA** Threshold TTCA

**SO-TTCA** Stochastic Optimization TTCA

**Chapter 8**

**Dynamic Allocation Mechanisms for Assignment of Heterogeneous Objects**

\( A \) \( \{1, 2, \ldots, n\} \), Set of agents

\( I \) \( \{I_1, \ldots, I_m\} \) Set of indivisible, distinct objects

\( T \) \( \{1, 2, \ldots\} \) Set of discrete time periods

\( \alpha_k \) Arrival period of agent \( k \), \( \alpha_k \in T \)

\( \beta_k \) Departure period of agent \( k \), \( \beta_k \in T \)

\( \phi_i \) Strict preference of agent \( i \) over \( I \)

\( \phi_{-i} \) Preferences of all the agents except \( i \)

\( \phi \) \((\phi_i, \phi_{-i})\) Preference profile of all the agents

\( \theta_k \) \((\alpha_k, \beta_k, \phi_k)\), Type of agent \( k \)

\( r(k, j) \) Index of \( j^{th} \) ranked item for agent \( k \)

\( x \) \( x : A \to I \), an allocation of the goods

\( x(i) \) denotes an object that agent \( i \) receives in allocation \( x \)

\( X \) Set of all feasible allocations

\( f \) \( \Theta^n \to X \), Allocation mechanism

**APSD** Arrival Priority Serial Dictatorship

**SR** Scoring Rule
Contents

Acknowledgements iii

Abstract v

Research Papers Based on the Thesis Work vii

Acronyms and Notation ix

1 Introduction 1

1.1 Motivation and Background 1

1.1.1 Mechanism Design 3

1.2 The Problems Addressed in The Thesis 4

1.2.1 Problems Involving Monetary Transfers 4

1.2.2 Problems without Monetary Transfers 7

1.2.3 Observations 10

1.3 Contributions and Thesis Outline 12

2 Mechanism Design Theory: An Overview 17

2.1 The Mechanism Design Environment 17

2.2 Examples of Social Choice Functions 21

2.3 Implementation of Social Choice Functions 24

2.3.1 Implementation Through Direct Mechanisms 24

2.3.2 Implementation Through Indirect Mechanisms 26

2.3.3 Bayesian Game Induced by a Mechanism 29

2.3.4 Implementation of a Social Choice Function by a Mechanism 31

2.4 Incentive Compatibility and the Revelation Theorem 32

2.4.1 Incentive Compatibility (IC) 33

2.4.2 The Revelation Principle for Dominant Strategy Equilibrium 35

2.4.3 The Revelation Principle for Bayesian Nash Equilibrium 37

2.5 Properties of Social Choice Functions 39

2.5.1 Ex-Post Efficiency 39

2.5.2 Dictatorship in SCFs 39

2.5.3 Individual Rationality 39

2.5.4 Efficiency 41

2.6 The Gibbard–Satterthwaite Impossibility Theorem 42

2.6.1 The G–S Theorem 43

2.6.2 Implications of the G–S Theorem 46
2.7 The Quasilinear Environment

2.8 Groves Mechanisms
  2.8.1 VCG Mechanisms
  2.8.2 The Groves Theorem
  2.8.3 Groves Mechanisms and Budget Balance

2.9 Clarke (Pivotal) Mechanisms
  2.9.1 Clarke Mechanisms and Weak Budget Balance
  2.9.2 Clarke Mechanisms and Individual Rationality

2.10 Mechanism Design Space in Quasilinear Environment

2.11 DSIC Mechanisms without Money

2.12 Two-Sided Matchings
  2.12.1 Some Important Definitions

2.13 Allocation Mechanisms
  2.13.1 Notation and Definitions

2.14 House Allocation
  2.14.1 Notation and Important Definitions
  2.14.2 Extensions to the House Allocation Problem

2.15 A Summary of Mechanisms without Money

3 Optimal Multi-Unit Combinatorial Auctions

3.1 Introduction
  3.1.1 Motivation and Background
  3.1.2 Contributions and Outline

3.2 The Model
  3.2.1 Some Preliminaries

3.3 Optimal Multi-Unit Combinatorial Procurement Auction
  3.3.1 Necessary and Sufficient Conditions for BIC and IR
  3.3.2 Allocation and Payment Rules
  3.3.3 Optimal Auction under Regularity Assumption

3.4 Volume Discounts
  3.4.1 Optimal Combinatorial Auction with Volume Discounts
  3.4.2 VD-OCAS Under Regularity Assumption

3.5 An Optimal Auction when Bidders are XOR-Minded
  3.5.1 Notation
  3.5.2 Optimal Auctions when Bidders are XOR Minded

3.6 Conclusion

4 Truthful Multi-Armed Bandit Mechanisms for Multi-Slot Sponsored Search Auctions

4.1 Introduction
  4.1.1 Sponsored Search Auctions
  4.1.2 Multi-Armed Bandit Mechanisms
  4.1.3 Related Work
  4.1.4 Our Contributions

4.2 System Setup and Notation
  4.2.1 Important Notions and Definitions

4.3 Characterization of Truthful MAB Mechanisms

xvi
4.3.1 Unknown and Unconstrained CTRs ........................................ 113
4.3.2 Higher Slot Click Precedence ........................................ 116
4.3.3 When CTR Pre-estimates are Available ......................... 118
4.3.4 When CTR is Separable ........................................ 122
4.4 Experimental Analysis ......................................................... 123
4.5 Conclusion ........................................................................... 124

5 Redistribution Mechanisms for Assignment of Heterogeneous Objects 127
5.1 Introduction ....................................................................... 127
5.1.1 Relevant Work ................................................................. 129
5.1.2 Contributions and Outline .............................................. 129
5.2 Preliminaries and Notation .................................................. 131
5.2.1 The Model and Notation ................................................ 131
5.2.2 Important Definitions .................................................... 131
5.2.3 Optimal Worst Case Redistribution when Objects are Identical ........................................ 131
5.3 Impossibility of Linear Rebate Function with Non-Zero Redistribution Index ........................................ 133
5.4 A Redistribution Mechanism for Heterogeneous Objects when Valuations have a Scaling Based Relationship ........................................ 136
5.4.1 The Proposed Mechanism .............................................. 137
5.5 Non-linear Redistribution Mechanisms for the Heterogeneous Setting ........................................ 141
5.5.1 BAILEY-CAVALLO Mechanism ........................................ 141
5.5.2 A Redistribution Mechanism for the Heterogeneous Setting - HETERO ........................................ 143
5.6 Experimental Analysis ......................................................... 146
5.6.1 Empirical Evidence for Individual Rationality of HETERO ........................................ 146
5.6.2 BAILEY-CAVALLO vs HETERO ........................................ 146
5.7 Conclusion ........................................................................... 148
5.7.1 Properties of the Ranking System .................................... 149
5.7.2 Ranking among the Agents ............................................. 149
5.7.3 Proof of Lemma 1 ............................................................. 150
5.7.4 Proof of Lemma 2 ............................................................. 151

6 Dynamic Stable Matching in Two-Sided Markets 153
6.1 Introduction ....................................................................... 153
6.1.1 Contributions ................................................................. 154
6.1.2 Related Work ................................................................. 155
6.2 Preliminaries ...................................................................... 156
6.3 Introducing a Fall-Back Option ............................................ 158
6.3.1 GSODAS ................................................................. 160
6.3.2 Stability ................................................................. 161
6.3.3 Randomized Online Matchings .................................... 163
6.3.4 Comparison with a Stochastic Optimization based Approach ........................................ 164
6.4 Experimental Results ......................................................... 165
6.5 Conclusions ...................................................................... 168
# Dynamic House Allocation

## 7.1 Introduction

- **7.1.1 Contributions**
- **7.1.2 Related Work**

## 7.2 The Model

## 7.3 Dynamic Top Trading Cycle Mechanisms

- **7.3.1 The Static TTCA**
- **7.3.2 Online TTCA**
- **7.3.3 Precluding Multiple Trades**

## 7.4 Partition Mechanisms

- **7.4.1 Simple Partition Mechanisms**
- **7.4.2 Stochastic Optimization**
- **7.4.3 Partition mechanisms are simple**

## 7.5 Simulation Results

## 7.6 Conclusions

---

# Dynamic Allocation Mechanisms for Assignment of Heterogeneous Objects

## 8.1 Introduction

- **8.1.1 Contributions**
- **8.1.2 Related Work**

## 8.2 The Model

## 8.3 Strategyproof Mechanisms

## 8.4 Heuristic Allocation Methods

- **8.4.1 Rank Efficiency Analysis**
- **8.4.2 What is The Scoring Rule Doing Right?**

## 8.5 Tolerable Manipulability

- **8.5.1 Experimental Results**

## 8.6 Conclusions

---

# Summary and Future Work

## 9.1 PART 1: MECHANISMS WITH MONEY AND STATIC AGENTS

## 9.2 PART 2: DYNAMIC MECHANISMS WITHOUT MONEY

---

# Bibliography
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Social choice functions and properties satisfied by them</td>
<td>43</td>
</tr>
<tr>
<td>2.2</td>
<td>Notation: matching</td>
<td>62</td>
</tr>
<tr>
<td>2.3</td>
<td>An example of a matching problem</td>
<td>63</td>
</tr>
<tr>
<td>2.4</td>
<td>An example to illustrate DA is not strategyproof</td>
<td>65</td>
</tr>
<tr>
<td>2.5</td>
<td>Notation: allocation mechanisms</td>
<td>66</td>
</tr>
<tr>
<td>2.6</td>
<td>Notation: house allocation</td>
<td>68</td>
</tr>
<tr>
<td>2.7</td>
<td>Summary: mechanisms without money</td>
<td>73</td>
</tr>
<tr>
<td>3.1</td>
<td>Notation: optimal combinatorial auctions in the presence of single minded bidders</td>
<td>80</td>
</tr>
<tr>
<td>3.2</td>
<td>Notation: XOR minded bidders</td>
<td>96</td>
</tr>
<tr>
<td>4.1</td>
<td>MAB Results</td>
<td>107</td>
</tr>
<tr>
<td>4.2</td>
<td>Notation: Multi-Armed Bandit mechanisms</td>
<td>110</td>
</tr>
<tr>
<td>5.1</td>
<td>Notation: redistribution mechanisms</td>
<td>132</td>
</tr>
<tr>
<td>5.1</td>
<td>Construction of agent preferences used for worst-case substitutes requirement in online matching mechanisms</td>
<td>160</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 A typical combinatorial procurement auction .......................................................... 5  
1.2 Sponsored slots in a search page .............................................................................. 6  
1.3 Dynamic matching scenario ..................................................................................... 8  
1.4 Dynamic house allocation ....................................................................................... 9  
1.5 Dynamic allocation mechanisms ............................................................................. 10  
1.6 Mechanism design problem space classification ....................................................... 11  
2.1 Mechanism design environment .............................................................................. 20  
2.2 Procurement with two suppliers .............................................................................. 21  
2.3 The idea behind implementation by an indirect mechanism ................................... 30  
2.4 Revelation principle for dominant strategy equilibrium ........................................... 36  
2.5 Revelation principle for Bayesian Nash equilibrium ................................................ 37  
2.6 Combined view of revelation theorems for dominant strategy equilibrium and Bayesian Nash equilibrium ................................................................. 38  
2.7 An illustration of the Gibbard–Satterthwaite Theorem ............................................ 46  
2.8 Escape routes for G–S Theorem .............................................................................. 47  
2.9 Space of DSIC social choice functions in quasilinear environment ....................... 55  
2.10 Mechanism design space in quasilinear environment ............................................. 59  
2.11 House allocation problem and TTCA .................................................................... 70  
2.12 Execution of TTCA: step 2 ................................................................................... 70  
2.13 Execution of TTCA: step 3 ................................................................................... 71  
3.1 A typical combinatorial procurement auction .......................................................... 76  
3.2 Volume discount: cost-demand curve ...................................................................... 91  
3.3 XOR bidding ........................................................................................................... 98  
4.1 Sponsored slots in a search page .............................................................................. 103  
4.2 A sponsored search auction scenario .................................................................... 108  
4.3 Average case regret and worst case regret in a logarithmic scale ......................... 124  
5.1 Redistribution index vs number of objects when number of agents = 10 ............. 147  
5.2 The number of substitutes required for men in GSODAS as n increases, fixing 
    T = 2 .................................................................................................................. 165  
5.3 The number of substitutes required for men in GSODAS as T increases, fixing 
    n = 20 ................................................................................................................ 166  
5.4 The rank-efficiency (x-axis) vs. the number of unstable men (y-axis) for n = 10 
    and T = 2 ........................................................................................................... 167
6.4 The rank-efficiency (x-axis) vs. the number of unstable men (y-axis) for $n = 20$
and $T = 4$ ........................................ 167

7.1 Rank-efficiency against the number of agents in environment $[E1]$ for patient
agents with $\lambda = n/8$ and $\mu = 0.01\lambda$ ........................................ 182

7.2 Rank-efficiency against the number of agents in environment $[E2]$ for less patient
agents with $\lambda = n/8$ and $\mu = 0.1\lambda$ ........................................ 183

7.3 Rank-efficiency against the number of agents in environment $[E3]$ with a uniform
arrival-departure model ......................................................... 183

7.4 Rank-efficiency against the number of agents in environment $[E4]$ with preferences
that are correlated across agents and depend on the popularity index of a house. 184

8.1 Rank-efficiency under truthful agents as the similarity in item popularity is ad-
justed, for 10 items and 10 agents ........................................ 195

8.2 Rank-efficiency under truthful agents as the population increases, for 10 items
10 agents and similarity of 0.3 ........................................ 196

8.3 Rank-efficiency of SR as the fraction of strategic agents varies, compared to SR
with entirely truthful agents and to Greedy. 10 agents, 10 items, similarity=0.3 198

8.4 Rank-efficiency of SR as the fraction of strategic agents varies and with errors in
agent beliefs. 10 agents, 10 items, similarity =0.3 ............................ 200

9.1 Mechanism design problem space classification .................................. 201
Chapter 1

Introduction

In the second half of the twentieth century, game theory and mechanism design have found widespread use in a gamut of applications in engineering. More recently, they have emerged as important tools to model, analyze, and solve issues in decentralized design problems, or in complex decision systems involving multiple autonomous agents that interact strategically in a rational and intelligent way. Typically, these agents have some private information and the system’s decision or the social outcome depends upon how the agents report the information they have. Whenever verification about the agents actions/reports is not possible, the rational and intelligent agents act in such a way as to maximize their own benefits from the system. Game theory and mechanism design theory play an important role in such situations.

Allocation of heterogeneous objects/resources is a common prevalent task in network economics. For example, a seller may be interested in allocating various goods and make profit, a government may wish to allocate various spectrum bands, a search engine is required to assign sponsored slots to the ads of competing agents, etc. It may be the case of assigning tasks on a collaborative web site like wiki among volunteers. The agents involved in such allocations have private preferences over the allocations. The agents act strategically to receive best possible allocation. Our thesis is motivated by such allocation of heterogeneous objects to strategic agents.

1.1 Motivation and Background

Consider the following allocation problems which commonly occur in the real world.

Scenario 1. Consider a situation in which a company wishes to procure a large number of various items. The company is naturally interested in doing this at minimum possible cost while
the various suppliers for these items are interested in making maximum profits out of such deals. The items that the company wishes to procure may be multiple units of distinct items.

**Scenario 2.** Typically whenever a user searches on a search engine, the engine displays the search results and along with the search results, the engine also displays advertisements related to the search query. Generally there are more agents competing to display their ads than the number of slots available for display of such ads. Every agent has different valuations for their advertisements in different slots. The search engine faces the problem of allocating these slots among the competing agents.

**Scenario 3.** Suppose a government agency wants to allot various land properties among its different subdivisions. The allocating body has limited number of sanctioned land properties to allot among its subdivisions. Now each subdivision might have different utilities to the different properties. In such a setting, how should the government allocate the properties among its subdivisions? The central authority may charge the sub-divisions for allocating them their favorable property. However the goal of the government is not to make money out of the allocation.

**Scenario 4.** Consider campus recruitments where there are various positions available for fresh graduates in various companies. These companies have preferences over these students and the students have preferences over these companies. There is a need to design a matching procedure among the students and the job openings for freshers.

**Scenario 5.** In the era of the Internet numerous collaborative websites like wiki have evolved. Such sites need regular maintenance tasks to be performed. There are volunteers who do these jobs various tasks. However they have preferences over these jobs. How are the volunteers allocated to the jobs?

**Scenario 6.** Consider the problem of reallocating office spaces amongst the employees. The employees have been assigned office spaces. However they are interested in swapping their spaces amongst themselves. How to facilitate such reallocations?
One can notice that in all the above problems, there is allocation of distinct objects. For example, in scenario 2, the company is assigning which objects to be procured from which supplier or in scenario 4, the allocation of various job positions amongst students. The participating agents in these are rational and intelligent. Their preferences or valuations for the objects are private to the agents. However the allocations need to be done with these informations. The agents may try to manipulate the allocation mechanisms to have better allocations for themselves which may result in loss of utilities for the other agents. The theory of mechanism design plays important role here.

1.1.1 Mechanism Design

The importance of mechanism design in the current context can be seen by the fact that the Nobel Prize in Economic Sciences for the year 2007 was jointly awarded to three economists, Leonid Hurwicz, Eric Maskin, and Roger Myerson for having laid the foundations of mechanism design theory [1]. Earlier, in 1996, William Vickrey, the inventor of the famous Vickrey auction had been awarded the Nobel Prize in Economic Sciences.

The agents are rational in the game theoretic sense of making decisions consistently in pursuit of their own individual objectives. Each agent’s objective is to maximize the expected value of his/her own payoff measured in some utility scale. Selfishness or self-interest is an important implication of rationality. Each agent is intelligent in the game theoretic sense of knowing everything about the underlying game that a game theorist knows and each agent can make any inferences about the game that a game theorist can make. In particular, each agent is strategic, that is, takes into account his/her knowledge or expectation of behavior of other agents and is capable of doing the required computations.

The theory of mechanism design is concerned with settings where a policy maker (or social planner) faces the problem of aggregating the announced preferences of multiple agents into a collective (or social) decision when the actual preferences are not publicly known. Mechanism design theory uses the framework of non-cooperative games with incomplete information and seeks to study how the privately held preference information can be elicited and the extent to which the information elicitation problem constrains the way in which social decisions can respond to individual preferences. In fact, mechanism design can be viewed as reverse engineering of games or equivalently as the art of designing the rules of a game to achieve a specific desired outcome. The main focus of mechanism design is to design institutions or protocols that satisfy certain desired objectives, assuming that the individual agents, interacting through the institution, will act strategically and may hold private information that is relevant to the decision at hand.

Mechanism design is concerned with how to implement system-wide solutions to problems
that involve *multiple self-interested agents*, each with *private information* about their preferences. A mechanism could be viewed as an institution or a framework of protocols that would prescribe particular ways of interaction among the agents so as to ensure a socially desirable outcome from this interaction. Without the mechanism, the interaction among the agents may lead to an outcome that is far from socially optimal. One can view mechanism design as an approach to solving a well-formulated but *incompletely specified optimization problem* where *some of the inputs to the problem are held by the individual agents*. So in order to solve the problem, the *social planner* needs to elicit these private values from the individual agents.

If the mechanism is designed in such a way that, it is a best response for each agent, to report his/her private values/information truthfully to the mechanism, we say the mechanism is *incentive compatible*. Incentive compatibility is defined in various forms. Dominant strategy incentive compatibility (DSIC) is the strongest one and the most desirable. However, in general settings, it is very difficult to achieve. Bayesian incentive compatibility (BIC) is much weaker and easier to achieve. There are other game theoretic properties as well, a mechanism should have. *Individual rationality* means, the agents are not worse off by participating in the mechanism. *Allocative efficiency* implies the objects are assigned to those who value them the most. Chapter 2 provides formal definitions of all the above terms. It is impossible to satisfy many of game theoretic properties simultaneously. Chapter 2 describes these impossibility theorems and the possibilities. Thus, given a problem, first we have to identify the system wide goal and the properties that a mechanism should satisfy. In short, mechanism design is the art and science of designing a game in the presence of multiple self interested, intelligent and rational agents with private information. In the next section, we describe the mechanism design problems addressed in this thesis.

### 1.2 The Problems Addressed in The Thesis

As seen above, there is a plethora of problems wherein mechanism design theory plays a central role to solve them. Our work is motivated by the following allocation problems in the network economics.

#### 1.2.1 Problems Involving Monetary Transfers

**Problem 1: Combinatorial Auctions**

Consider an organization or a company wishing to procure multiple, distinct items in large volumes. There are different vendors who can supply various items and their combinations. It has been shown that, rather than auctioning for each of the items individually, it is better to allow suppliers to bid on various combinations of the items for better social welfare as well as to
reduce the cost of procurement. The auctions in which the bidders can submit bids on different combinations of the items are referred to as *Combinatorial Auctions*. Also, the suppliers are capcaciated, in the sense that they can only supply limited quantity of the items which they can provide. Typically, the costs of production and the capacities are private information to the vendors. The goal of the company is to procure these items at minimum cost. Figure 1.1 depicts the settings. The social planner’s goal is to design a combinatorial procurement auction that minimizes cost to the company and is Bayesian incentive compatible as well as individually rational for the suppliers. Such auctions are referred to as *optimal auctions* [2].

![Figure 1.1: A typical combinatorial procurement auction](image)

**Problem 2: Multi-Armed Bandit Mechanisms**

The most popular search engines, viz Google, Yahoo!, Bing and many more offer the search free for the end user. Billions of people access these sites. Whenever a user searches any set of keywords on a search engine, along with the search results, called as *organic results*, the search engine displays advertisements related to those keywords on the right side of the organic results or at the top of the organic results. The search engine charges the advertiser for displaying her ad if her ad receives a click. The decision regarding which ads are to be displayed and their respective order is based on the bids submitted by the advertisers indicating the maximum amount they are willing to pay per click. So for each keyword, there is an auction that takes place in the background. These auctions are called *sponsored search auctions* or *pay-per click auctions*. Figure 1.2 captures the scenario. To perform any optimizations, such as maximizing social welfare or maximizing revenue to the search engine, the true per-click valuations of the advertisers are needed. Being rational, the advertisers may actually manipulate their bids and therefore a primary goal of the search engine is to design an auction for which it is in the best
interest of each advertiser to bid truthfully irrespective of the bids of the other advertisers. Such an auction is said to be Dominant Strategy Incentive Compatible (DSIC), or truthful.

The click on a displayed ad by the end user is a random event. The probability of the ad getting clicked depends upon the advertiser as well as where the ad is displayed. These click probabilities or clickthrough rates (CTRs), play a crucial role in these auctions. If the search engine knows the CTRs, then its problem is only to design a truthful auction. However, the search engine may not know the CTRs beforehand. Thus the problem of the search engine is two fold: (1) learn the CTR values, (2) design a truthful auction.

Typically, the same set of agents compete for the given set of keywords. The search engine can exploit this fact to learn the CTRs by initially displaying ads by various advertisers. Such problems are referred to as Multi-Armed Bandit (MAB) problems [3]. Since the agents may not report their true values, the problem of the search engine can be described as one of designing an incentive compatible MAB mechanism. The problem that we address is, what are all the truthful MAB mechanisms for multi-slot sponsored search auctions.

![Figure 1.2: Sponsored slots in a search page](image)

**Problem 3: Redistribution Mechanisms**

A government agency procure $p$ distinct (heterogeneous) land properties for assigning them among $n > p$ of its sub-divisions. Each of the sub-divisions needs at most one of these properties
and it has different valuations for different properties. Now, the government wishes the allocation of these properties to be fair, meaning, those who value these properties the most, should receive them. The sub-divisions being rational and intelligent need not report their values truthfully. The government may charge the sub-divisions for truthful elicitation of their values. Note that, here the government is not interested in making money out of such assignment. It can simply redistribute the money collected, back among its sub-divisions. However, this may violate certain game theoretic properties. Thus, there is need for designing an appropriate mechanism. Such mechanisms are referred to as redistribution mechanisms.

1.2.2 Problems without Monetary Transfers

Many of the economic institutions work with numeraire, that is, money. However, in some of the market places like assigning university dorms to students, matching jobs to workers, admitting students to colleges, we cannot use money for assignment. These markets are either one-sided or two-sided. Dorm assignment to students involves preferences only from the students and hence is called as one-sided. However, in the case of job assignments, the firms offering jobs have preferences over the workers and vice-versa. These are two-sided markets. The strategic agents in such markets, who have preferences over the assignments, may try to manipulate the assignment procedures.

Also, in real life, all the agents may not be available simultaneously. So, some of the decisions have to be taken online, before the agents’ departure.

Problem 4: Matchings

Consider the campus recruitment job market. Companies visit colleges in various time slots during the year, while students are seeking a position throughout the year. In our terms, this is a two-sided matching problem in which the companies are “dynamic” with arrival and departure times while the students are “static” and always present in the market, although perhaps already matched.

Further suppose that students may seek to obtain a better match by strategic misreporting of their preferences over the companies, while the companies report true preference rankings on the students. We can assume this because, it is generally known what skill sets companies want (e.g., which grades, in which kinds of classes, etc.). Student preferences on companies may be predetermined or determined dynamically as companies arrive, as long as preference orderings over earlier companies are not changed by subsequent arrivals. For companies, it is probably easiest to think that their preferences over students are determined upon arrival. What is a reasonable student-companies matching procedure, so that students do not manipulate it?
This problem is depicted in Figure 1.3. There are three companies and three candidates applying for a position in these companies. The preferences of the candidates are shown using color code for the companies. Companies 1 and 2 arrive in the system in period 1. Company 1 leaves the system at the end of period 1 while Company 3 arrives in period 2. How to match the companies with the students in such dynamic settings?

![Dynamic matching scenario](image)

Figure 1.3: Dynamic matching scenario

**Problem 5: House Allocation**

In Figure 1.4, we show the dynamic house allocation problem. There are four agents in the market who wish to reallocate their goods among themselves. However they are available in the market on various days. That is, Agent 1 is available on Monday and Tuesday while Agent 2 is available for trading only on Monday and Agent 3 on Tuesday, Wednesday. Agent 4 is available only on Wednesday. Their preferences over the objects are as shown in the figure. The problem is how to reallocate the goods in dynamic settings? We can consider the objects owned by the agents as houses and can consider this problem as dynamic house allocation.

The classic house allocation problem, (with static agents) is one of an exchange of distinct objects by self-interested agents, each with an initial ownership of a single object and strict preferences on objects. There are no monetary transfers. Suppose, a university is interested
in facilitating trades in student housing, in which each student can choose to opt in each year into a trading opportunity, which runs for an interval of time of, for example the last month of the current lease, and so is different for each student because it depends on when they entered into a lease. We can also consider a problem of reallocating office space amongst employees. So, the problem which we address is dynamic in nature. In our dynamic model, we have agent arrivals and departures, with agents able to trade amongst other agents present in the market. Here each agent owns an object when it enters into the market, which we refer to as house. By facilitating such trade, each agent has to be ensured of a house at least as good as the house it owns when it enters the market. What is a good online mechanism that is strategyproof, individually rational?

**Problem 6: Allocation Mechanisms**

Consider the allocation of tasks to agents that arrive and depart and have preferences on tasks and a time window within which they can be allocated a task. The problem can also be one of resource allocation, where agents arrive and demand access to a resource before departure. Motivating domains include those of car pooling in which the agents are commuters and the
resources are seats in shared cars, or science collaborators in which the agents are people looking to perform useful work for the community. We consider problems in which it is undesirable to use money; e.g., because of community norms, legal constraints, or inconvenience. If at all we are allocating any task/resource to any agent, it has to be done before his departure. How these tasks/resources should be allocated amongst dynamically arriving agents?

Figure 1.5 depicts this problem where there are three objects \( O_1, O_2 \) and \( O_3 \) to be allocated to three dynamically arriving agents. Agent 1 is available in period 1 and 2. Agent 2 is available in period 2, 3 and 4 while agent 3 is available in period 3, 4 and 5. The agents’ preferences are shown as in Figure 1.5. How the objects to be allocated to these three agents?

![Diagram of allocation mechanisms](image)

**Figure 1.5: Dynamic allocation mechanisms**

### 1.2.3 Observations

Note, in all the above examples, the nature of the problem is such that

- the participating agents are strategic
- the problems are with incomplete information, that is, the decision maker/social planner does not have true information about the agents preferences/utilities.
- the problems involves decision making regarding allocation of heterogeneous objects


in some situations monetary transfers are possible and in some cases, monetary transfers are not allowed

all the problems are game theoretic in nature and naturally fit into a mechanism design framework

Note that, in Problems 1, 2, and 3 all the agents are simultaneously available for taking a system wide decision. We say such agents are static or the settings is offline. In Problems 4, 5, and 6 the agents are dynamic and the decision pertaining to each agent has to be taken before he or she departs the system. We say such agents are online or say online settings. Also, there is one more notable difference in the first three examples and the latter three examples. In the first three examples, the agents are making a payment to the system to have a preferable decision. However, in the latter three examples, the monetary transfers are not allowed or may not even be legal. Thus, for each mechanism design problem, we can assign attributes (i) whether the agents are static or online in nature and (ii) whether the mechanism involves money or is without money. A classification of mechanism design problems based on these two attributes is shown in Figure 1.6.

The state of the art addresses various game-theoretic issues in the markets without money in the presence of the static agents (Region C in Figure 1.6). That is, all the agents are present together while the assignments are done. In real life, all the agents may not be available simultaneously. So, some of the decisions have to be taken online, before the agents’ depart. We study incentive issues, efficiency, and stability in such dynamic environments, without monetary transfers.
We describe our contributions in the next section. We also concurrently provide an outline of the thesis.

1.3 Contributions and Thesis Outline

We design creative solutions to the problems stated above. In particular, we study mechanisms for the problems described in Problems 1-6. These problems either come under region $A$ or $D$ in Figure 1.6. To the best of our knowledge, we are the first one to address the problems in region $D$ in mechanism design framework.

Thesis is organized as follows. The thesis is divided into two parts. In the first part, (Part-1), we address the problems in the presence of static agents and monetary transfers play an important role, that is, the mechanism design problems which fall in the region $A$ in Figure 1.6. These include, the problems 1, 2, and 3. In the second part, (Part-2), we address the problems in the presence of online agents and where the monetary transfers are not allowed, that is, the mechanism design problems which fall in the region $D$ in Figure 1.6. These include the problems 4, 5, and 6.

First we review the important results in mechanism design theory in Chapter 2. Each of the chapters 3-7 is organized as an independent paper. The state of the art related with each of the above problem is reviewed in the corresponding chapter. Then each chapter proceeds with the model and presents our results. The thesis is concluded with summary and directions for future work in Chapter 8.

This dissertation is a good blend of theoretical results with simulations based observations. In particular, our contributions are as follows:

PART 1: MECHANISMS WITH MONEY AND STATIC AGENTS

Chapter 3: Optimal Multi-Unit Combinatorial Auctions

The current art in optimal combinatorial auctions is limited to handling the case of a single unit of multiple items, with each bidder bidding on exactly one bundle (single minded bidders). We extend the current art by proposing an optimal auction for procuring multiple units of multiple items when the bidders are single minded. We develop a procurement auction that minimizes the cost of procurement while satisfying Bayesian incentive compatibility and interim individual rationality which we refer to as $OCAS$, (Optimal Combinatorial Auction with Single Minded Bidder). Under appropriate regularity conditions, the $OCAS$ also satisfies dominant

\footnote{In Part-1, the buyers are generally treated as feminine gender and the sellers are masculine gender. In Part-2 all the agents are treated as masculine gender.}
strategy incentive compatibility. We also design an auction when the bidders offer volume discounts for the case of two items multi-unit procurement auction, namely VD-OCAS, (Optimal Combinatorial Auction with Single Minded Bidder offering Volume Discounts). We conjecture that the VD-OCAS is an optimal auction for the settings it has been designed for. Then, we introduce the notion of XOR minded bidder, who is submitting an XOR bid on two disjoint subsets of bundles. We discuss how the existing theory may be extended to characterize optimal combinatorial auction in the presence of XOR minded bidders.

Chapter 4: Truthful Multi-Armed Bandit Mechanisms for Multi-Slot Sponsored Search Auctions

In sponsored search auctions, to maximize the social welfare, the search engine needs to know the click probabilities of the agents. Typically, as the same set of agents is competing for the same keywords repeatedly, the search engine can learn these probabilities over the course of repeated auctions. So, the mechanism designer has to incentivize agents to report the value that they derive from a click on the ad truthfully as well as learn click-probabilities. These mechanisms are referred to as Multi-Armed-Bandit (MAB) Mechanisms. Recently, Babaioff, Sharma, and Slivkins have derived a characterization of truthful MAB mechanisms in pay-per-click sponsored search auctions if there is a single slot for each keyword. We essentially extend the results of Babaioff, Sharma, and Slivkins \cite{5} and Devanur and Kakade \cite{6} to the non-trivial general case of two or more sponsored slots. The precise question we address is: What are all the MAB mechanisms for multi-slot pay-per-click sponsored search auctions that are dominant strategy incentive compatible (or truthful)? Our findings show that, when the CTRs across agent-slot pair are un-restricted, then any truthful mechanism has to satisfy stringent conditions which lead to $\Theta(T)$ regret, where $T$ is the number of rounds played. Then we restrict our attention to various real settings where CTRs are not unrestricted and characterize necessary as well as sufficient conditions for an MAB mechanism to be truthful.

Chapter 5: Redistribution Mechanisms for Assignment of Heterogeneous Objects

Due to Green-Laffont impossibility theorem \cite{7}, in general, no mechanism that satisfies dominant strategy incentive compatibility and allocative efficiency is budget balanced. We can first collect the payments by using well known Vickrey-Clarke-Groves, VCG mechanism and then we can minimize the worst case budget imbalance by redistributing this surplus in the system among the participating agents in the form of rebate. Such mechanisms are referred to as redistribution mechanisms. Designing a redistribution mechanism calls for designing an appropriate rebate function. To measure a performance of a redistribution mechanism, we introduce the
notion of redistribution index, which captures the worst case fraction of the VCG surplus that gets redistributed. We address heterogeneous object assignments among competing agents and provide an impossibility theorem: No linear rebate function guarantees non-zero redistribution index. When the objects are homogeneous, Moulin [8] and Guo and Contizer [9] have proposed a mechanism that is worst case optimal. We extend Moulin’s mechanism to the heterogeneous objects case and conjecture it to be individually rational and has worst case optimal redistribution index. Our simulations provide a strong empirical evidence for the conjecture. To the best of our knowledge, this is the first attempt to design a redistribution mechanism for the assignment of heterogeneous objects. We also design a worst-case optimal redistribution mechanism when agents’ valuations have scaling based relation.

PART 2: DYNAMIC MECHANISMS WITHOUT MONEY

Chapter 6: Dynamic Stable Matching in Two-Sided Markets

In static settings, the deferred acceptance algorithm by Shapley-Gale [10], produces a stable matching. For online settings, in enabling stability properties, so that no pair of agents can usefully deviate from the match, we consider the use of a fall-back option where the dynamic agents can be matched, if needed, with a limited number of agents from a separate “reserve” pool. We refer to such re-match as substitute. We introduce a novel mechanism called GSODAS (General Period, Stable Online Deferred Acceptance with Substitutes), which is truthful for agents on the static side of the market and is stable. We also study two random mechanisms that do not use any substitutes and compare rank-analysis with GSODAS by simulations and show GSODAS dominate these algorithms, which already have stability issues. We also prove that any substitute avoided in GSODAS leads to unstability on worst case analysis.

Chapter 7: Dynamic House Allocation

This chapter addresses Problem 5, that is, allocation of heterogeneous objects with initial endowments. For static agents case, Top Trading Cycle Algorithm (TTCA) is the unique algorithm that has certain game theoretic properties, namely strategyproof, individually rational and belongs to the core. So, we use TTCA as building block for the dynamic house allocation problem. Our contributions are as follows. We show that, no online mechanism is Pareto efficient as well as individually rational, while any one of these properties is always possible to achieve. We prove that, under reasonable assumptions, in any strategyproof mechanism, agents cannot change the period in which he/she trades. We then propose a class of mechanisms, namely, Partition Mechanisms which are strategyproof and individually rational. We propose ex-ante rank-efficiency as
a performance measure for comparison of various partition mechanisms. We propose three partition mechanisms namely, DO-TTCA, T-TTCA and a stochastic optimization based mechanism, SO-TTCA. Our experimental results show that, SO-TTCA and T-TTCA outperform DO-TTCA when the agents have uniform arrival time and patience, that is the duration for which they stay in the system. For environments in which agents tend to be impatient, T-TTCA has competitive performance but is less successful than SO-TTCA in identifying partitions in which each agent tends to have a large number of trading partners.

Chapter 8: Dynamic Allocation Mechanisms for Assignment of Heterogeneous Objects

This chapter addresses Problem 6, that is, allocation of heterogeneous objects without initial endowments. We consider one-sided markets involving the assignment of the objects to dynamically arriving agents. In this context, we characterize all truthful mechanisms with certain game theoretic properties when agents cannot lie about arrival-departure schedules. In the settings where agents can lie about arrival-departures, Arrival Priority Serial Dictatorship, APSD, is the only strategyproof mechanisms. We show that APSD has poor performance on competitive analysis based on the rank of the object that the agent receives. This motivates our heuristic based mechanism, namely Scoring Rule, (SR). Though it does not satisfy strategyproofness, it performs quite well on rank-efficiency (5%-10% improvement), if all the agents are truthful. We also study stochastic optimization based mechanism CONSENSUS and by simulation show that SR out performs CONSENSUS.
Chapter 2

Mechanism Design Theory: An Overview

It is hard to believe that a man is telling the truth when you know that you would lie if you were in his place.

— H. L. Mencken

Mechanism design is the art and science of designing a game in the presence of multiple self interested intelligent agents with private information in such a way that truth telling is a best response for the agents at an equilibrium. In this chapter we provide an overview of mechanism design. We review important results from mechanism design theory which are crucial for this thesis. This material is called out from [11, 12, 13]. A comprehensive survey of mechanism design theory is available in [11, 12].

2.1 The Mechanism Design Environment

The following provides a general setting for formulating, analyzing, and solving mechanism design problems.

• There are \( n \) agents, 1, 2, \ldots, \( n \), with \( N = \{1, 2, \ldots, n\} \). The agents are rational and intelligent.

• \( X \) is a set of alternatives or outcomes. The agents are required to make a collective choice from the set \( X \).

• Prior to making the collective choice, each agent privately observes his preferences over the alternatives in \( X \). This is modeled by supposing that agent \( i \) privately observes a parameter or signal \( \theta_i \) that determines his preferences. The value of \( \theta_i \) is known to agent \( i \) and is not known to the other agents. \( \theta_i \) is called a private value or type of agent \( i \).
We denote by $\Theta_i$ the set of private values of agent $i$, $i = 1, 2, \ldots, n$. The set of all type profiles is given by $\Theta = \Theta_1 \times \ldots \times \Theta_n$. A typical type profile is represented as $\theta = (\theta_1, \ldots, \theta_n)$.

It is assumed that there is a common prior distribution $\Phi \in \Delta(\Theta)$. To maintain consistency of beliefs, individual belief functions $p_i$ that describe the beliefs that player $i$ has about the type profiles of the rest of the players can all be derived from the common prior.

Individual agents have preferences over outcomes that are represented by a utility function $u_i : X \times \Theta_i \rightarrow \mathbb{R}$. Given $x \in X$ and $\theta_i \in \Theta_i$, the value $u_i(x, \theta_i)$ denotes the payoff that agent $i$ having type $\theta_i \in \Theta_i$ receives from a decision $x \in X$. In the more general case, $u_i$ depends not only on the outcome and the type of player $i$, but could depend on the types of the other players also, so $u_i : X \times \Theta \rightarrow \mathbb{R}$. We restrict our attention to the former case in this thesis since most real-world situations fall into the former category.

The set of outcomes $X$, the set of players $N$, the type sets $\Theta_i$ ($i = 1, \ldots, n$), the common prior distribution $\Phi \in \Delta(\Theta)$, and the payoff functions $u_i$ ($i = 1, \ldots, n$) are assumed to be common knowledge among all the players. The specific value $\theta_i$ observed by agent $i$ is private information of agent $i$.

Social Choice Functions

Since the agents' preferences depend on the realization of their types $\theta = (\theta_1, \ldots, \theta_n)$, it is natural to make the collective decision to depend on $\theta$. This leads to the definition of a social choice function.

**Definition 2.1 (Social Choice Function).** Given a set of agents $N = \{1, 2, \ldots, n\}$, their type sets $\Theta_1, \Theta_2, \ldots, \Theta_n$, and a set of outcomes $X$, a social choice function is a mapping

$$f : \Theta_1 \times \cdots \times \Theta_n \rightarrow X$$

that assigns to each possible type profile $(\theta_1, \theta_2, \ldots, \theta_n)$ a collective choice from the set of alternatives.

**Example 2.1 (Shortest Path Problem with Incomplete Information).** Consider a connected directed graph with a source vertex and destination vertex identified. Let the graph have $n$ edges, each owned by a rational and intelligent agent. Let the set of agents be denoted by $N = \{1, 2, \ldots, n\}$. Assume that the cost of the edge is private information of the agent owning the edge and let $\theta_i$ be this private information for agent $i$ ($i = 1, 2, \ldots, n$). Let us say that a
social planner is interested in finding a shortest path from the source vertex to the destination vertex. The social planner knows everything about the graph except the costs of the edges. So, the social planner first needs to extract this information from each agent and then find a shortest path from the source vertex to the destination vertex. Thus there are two problems facing the social planner, which are described below.

---

**Preference Elicitation Problem**

Consider a social choice function $f : \Theta_1 \times \ldots \times \Theta_n \rightarrow X$. The types $\theta_1, \ldots, \theta_n$ of the individual agents are private information of the agents. Hence for the social choice $f(\theta_1, \ldots, \theta_n)$ to be chosen when the individual types are $\theta_1, \ldots, \theta_n$, each agent must disclose its true type to the social planner. However, given a social choice function $f$, a given agent may not find it in its best interest to reveal this information truthfully. This is called the *preference elicitation* problem or the *information revelation* problem. In the shortest path problem with incomplete information, the preference elicitation problem is to elicit the true values of the costs of the edges from the respective edge owners.

**Preference Aggregation Problem**

Once all the agents report their types, the profile of reported types has to be transformed to an outcome, based on the social choice function. Let $\theta_i$ be the true type and $\hat{\theta}_i$ the reported type of agent $i$ ($i = 1, \ldots, n$). The process of computing $f(\hat{\theta}_1, \ldots, \hat{\theta}_n)$ is called the *preference aggregation* problem. In the shortest path problem with incomplete information, the preference aggregation problem is to compute a shortest path from the source vertex to the destination vertex, given the structure of the graph and the (reported) costs of the edges. The preference aggregation problem is usually an optimization problem. Figure 2.1 provides a pictorial representation of all the elements making up the mechanism design environment.

**Direct and Indirect Mechanisms**

One can view mechanism design as the process of solving an incompletely specified optimization problem where the specification is first elicited and then the underlying optimization problem is solved. Specification elicitation is basically the preference elicitation or type elicitation problem. To elicit the type information from the agents in a truthful way, there are broadly two kinds of mechanisms, which are aptly called *indirect mechanisms* and *direct mechanisms*. We define these below. In these definitions, we assume that the set of agents $N$, the set of outcomes $X$, the
sets of types $\Theta_1, \ldots, \Theta_n$, a common prior $\Phi \in \Delta(\Theta)$, and the utility functions $u_i : X \times \Theta_i \to \mathbb{R}$ are given and are common knowledge.

**Definition 2.2 (Direct Mechanism).** Given a social choice function $f : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n \to X$, a direct (revelation) mechanism consists of the tuple $(\Theta_1, \Theta_2, \ldots, \Theta_n, f(.))$.

The idea of a direct mechanism is to directly seek the type information from the agents by asking them to reveal their true types.

**Definition 2.3 (Indirect Mechanism).** An indirect (revelation) mechanism consists of a tuple $(S_1, S_2, \ldots, S_n, g(.))$ where $S_i$ is a set of possible actions for agent $i \ (i = 1, 2, \ldots, n)$ and $g : S_1 \times S_2 \times \ldots \times S_n \to X$ is a function that maps each action profile to an outcome.

The idea of an indirect mechanism is to provide a choice of actions to each agent and specify an outcome for each action profile. This induces a game among the players and the strategies played by the agents in an equilibrium of this game will indirectly reflect their original types.

In the next four sections of this chapter, we will understand the process of mechanism design in the following way. First, we provide an array of examples to understand social choice functions and to appreciate the need for mechanisms. Next, we understand the process of implementing social choice functions through mechanisms. Following this, we will introduce the important notion of incentive compatibility and present a fundamental result in mechanism design, the
Then we will look into different properties that we would like a social choice function to satisfy.

### 2.2 Examples of Social Choice Functions

**Example 2.2** (Procurement of a Single Indivisible Resource). Procurement is a ubiquitous activity in any organization. Every organization procures a variety of direct and indirect materials. For example, a factory procures raw material or sub-assemblies from a pool of suppliers. In a computational grid, a grid user procures computational or storage resources from the grid. A network user procures network resources. A dynamic supply chain planner procures supply chain service providers. Every organization procures indirect materials such as office supplies and services.

The generic procurement situation involves a buyer, a pool of suppliers, and items to be procured. We consider a simple abstraction of the problem by considering a buying agent (call the agent 0) and two selling agents (call them 1 and 2), so we have \( N = \{0, 1, 2\} \). See Figure 2.2. An indivisible item or resource is to be procured from one of the sellers in return for a monetary consideration. An outcome here can be represented by \( x = (y_0, y_1, y_2, t_0, t_1, t_2) \). For \( i = 0 \), we have

\[
\begin{align*}
y_0 &= 0 \quad \text{if the buyer buys the good} \\
    &= 1 \quad \text{otherwise} \\
t_0 &= \text{monetary transfer received by the buyer.}
\end{align*}
\]
For $i = 1, 2$, we have

\begin{align*}
y_i &= 1 \quad \text{if agent } i \text{ supplies the good to the buyer} \\
      &= 0 \quad \text{if agent } i \text{ does not supply the good} \\
t_i &= \text{monetary transfer received by the agent } i.
\end{align*}

The set $X$ of all feasible outcomes is given by

\[ X = \{ (y_0, y_1, y_2, t_0, t_1, t_2) : y_i \in \{0, 1\}, \sum_{i=0}^{2} y_i = 1, t_i \in \mathbb{R}, \sum_{i=0}^{2} t_i \leq 0 \}. \]

The constraint $\sum_{i} t_i \leq 0$ implies that the total money received by all the agents is less than or equal to zero. That is, total money paid by all the agents is greater than or equal to zero (that is, buyer pays at least as much as the sellers receive. The excess between the payment and receipts is the surplus). For $x = (y_0, y_1, y_2, t_0, t_1, t_2)$, define the utilities to be of the form:

\[ u_i(x, \theta_i) = u_i((y_0, y_1, y_2, t_0, t_1, t_2), \theta_i) = -y_i \theta_i + t_i ; \quad i = 1, 2 \]

where $\theta_i \in \mathbb{R}$ can be viewed as seller $i$’s valuation of the good. Such utility functions are said to be of *quasilinear* form (because it is linear in some of the variables and possibly non-linear in the other variables). We will be studying such utility forms quite extensively in this chapter.

We make the following assumptions regarding valuations.

- The buyer has a *known* value $\theta_0$ for the good. This valuation does not depend on the choice of the seller from whom the item is purchased.

- Let $\Theta_i$ be the real interval $[\theta_2, \bar{\theta}_i]$. The types $\theta_1$ and $\theta_2$ of the sellers are drawn independently from the interval $[\theta_2, \bar{\theta}_i]$ and this fact is *common knowledge* among all the players. The type of a seller is to be viewed as the *willingness to sell* (minimum price below which the seller is not interested in selling the item).

Consider the following social choice function.

- The buyer buys the good from the seller with the lowest willingness to sell. If both the sellers have the same type, the buyer will buy the object from seller 1.

- The buyer pays the selected selling agent his willingness to sell.

The above social choice function $f(\theta) = (y_0(\theta), y_1(\theta), y_2(\theta), t_0(\theta), t_1(\theta), t_2(\theta))$ can be precisely written as

\[ y_0(\theta) = 0 \quad \forall \theta \]
We will refer to the above SCF as SCF-PROC1 in the sequel.

Suppose we consider another social choice function, which has the same allocation rule as the one we have just studied but has a different payment rule: The buyer now pays the winning seller a payment equal to the second lowest willingness to sell (as usual, the losing seller does not receive any payment). The new social choice function, which we will call SCF-PROC2, will be the following.

Let us define one more SCF, which we call SCF-PROC3, in the following way. SCF-PROC3 has the allocation rule as SCF-PROC1 and SCF-PROC2, but the payments are defined as:

Let the reason for defining the payment rule in the above way will become clear in the next section, where we will discuss the implementability of SCF-PROC1, SCF-PROC2, and SCF-PROC3.

\[
\begin{align*}
y_1(\theta) &= 1 \text{ if } \theta_1 \leq \theta_2 \\
&= 0 \text{ if } \theta_1 > \theta_2 \\
y_2(\theta) &= 1 \text{ if } \theta_1 > \theta_2 \\
&= 0 \text{ if } \theta_1 \leq \theta_2 \\
t_1(\theta) &= \theta_1 y_1(\theta) \\
t_2(\theta) &= \theta_2 y_2(\theta) \\
t_0(\theta) &= -(t_1(\theta) + t_2(\theta)).
\end{align*}
\]
2.3 Implementation of Social Choice Functions

In the preceding section, we have seen a series of examples of social choice functions. In this section, we motivate, through illustrative examples, the concept of implementation of social choice functions. We then formally define the notion of implementation through direct mechanisms and indirect mechanisms.

2.3.1 Implementation Through Direct Mechanisms

We first provide examples to motivate implementation by direct mechanisms.

Example 2.3 (Implementability of SCF-PROC1). Recall the social choice function SCF-PROC1 that we introduced in the context of procurement of a single indivisible resource (Example 2.2). Recall the definition of SCF-PROC1:

\[
\begin{align*}
  y_0(\theta) &= 0 \quad \forall \theta \\
  y_1(\theta) &= 1 \quad \text{if } \theta_1 \leq \theta_2 \\
  &\quad = 0 \quad \text{if } \theta_1 > \theta_2 \\
  y_2(\theta) &= 1 \quad \text{if } \theta_1 > \theta_2 \\
  &\quad = 0 \quad \text{if } \theta_1 \leq \theta_2 \\
  t_1(\theta) &= \theta_1 y_1(\theta) \\
  t_2(\theta) &= \theta_2 y_2(\theta) \\
  t_0(\theta) &= -(t_1(\theta) + t_2(\theta)).
\end{align*}
\]

We note that the social choice function is very attractive to the buyer since the buyer will capture all of the consumption benefits that are generated by the good. We assume that $\theta_1$ and $\theta_2$ are drawn independently from a uniform distribution over $[0,1]$. Now we ask the question: Can we implement this social choice function? The answer for this question is no. The following analysis shows why.

Let us say seller 2 announces his true value $\theta_2$. Suppose the valuation of seller 1 is $\hat{\theta}_1$, and he announces $\hat{\theta}_1$. If $\theta_2 \geq \hat{\theta}_1$, then seller 1 is the winner, and his utility will be $\hat{\theta}_1 - \theta_1$. If $\theta_2 < \hat{\theta}_1$, then seller 2 is the winner, and seller 1’s utility is zero. Since seller 1 wishes to maximize his expected utility he solves the problem

\[
\max_{\hat{\theta}_1}(\hat{\theta}_1 - \theta_1) P\{\theta_2 \geq \hat{\theta}_1\}.
\]

Since $\theta_2$ is uniformly distributed on $[0,1]$,

\[
P\{\theta_2 \geq \hat{\theta}_1\} = 1 - P\{\theta_2 < \hat{\theta}_1\} = 1 - \hat{\theta}_1.
\]
Thus seller 1 tries to solve the problem:

$$\max_{\hat{\theta}_1} (\hat{\theta}_1 - \theta_1)(1 - \hat{\theta}_1).$$

This problem has the solution

$$\hat{\theta}_1 = \frac{1 + \theta_1}{2}.$$

Thus if seller 2 announces his true valuation, then the best response for seller 1 is to announce $\frac{1 + \theta_1}{2}$.

Similarly if seller 1 announces his true valuation $\theta_1$, then the best response of seller 2 is to announce $\frac{1 + \theta_1}{2}$. Thus there is no incentive for the sellers to announce their true valuations. So, a social planner who wishes to realize the above social choice function finds the rational players will not report their true private values. Thus the social choice function cannot be implemented through a direct mechanism.

**Example 2.4** (Implementability of SCF-PROC2). Recall the social choice function SCF-PROC2, again in the context of procurement of a single indivisible resource (Example 2.2):

\[
\begin{align*}
  y_0(\theta) &= 0 \quad \forall \theta \\
  y_1(\theta) &= 1 \quad \text{if } \theta_1 \leq \theta_2 \\
               &= 0 \quad \text{if } \theta_1 > \theta_2 \\
  y_2(\theta) &= 1 \quad \text{if } \theta_1 > \theta_2 \\
               &= 0 \quad \text{if } \theta_1 \leq \theta_2 \\
  t_1(\theta) &= \theta_2 y_1(\theta) \\
  t_2(\theta) &= \theta_1 y_2(\theta) \\
  t_0(\theta) &= -(t_1(\theta) + t_2(\theta)).
\end{align*}
\]

We now show that the function SCF-PROC2 can be implemented. Let us say seller 2 announces his valuation as $\hat{\theta}_2$. There are two cases.

1. $\theta_1 \leq \hat{\theta}_2$
2. $\theta_1 > \hat{\theta}_2$.

**Case 1: $\theta_1 \leq \hat{\theta}_2$**

Let $\hat{\theta}_1$ be the announcement of seller 1. Here there are two cases.

- If $\hat{\theta}_1 \leq \hat{\theta}_2$, then the payoff for seller 1 is $\hat{\theta}_2 - \theta_1 \geq 0$.
- If $\hat{\theta}_1 > \hat{\theta}_2$, then the payoff for seller 1 is 0.
Thus in this case, the maximum payoff possible is $\hat{\theta}_2 - \theta_1 \geq 0$.

If $\hat{\theta}_1 = \theta_1$ (that is, seller 1 announces his true valuation), then payoff for seller 1 is $\hat{\theta}_2 - \theta_1$, which happens to be the maximum possible payoff as shown above. Thus announcing $\theta_1$ is a best response to seller 1 whatever the announcement of seller 2.

**Case 2: $\theta_1 > \hat{\theta}_2$**

Here again there are two cases: $\hat{\theta}_1 \leq \hat{\theta}_2$ and $\hat{\theta}_1 > \hat{\theta}_2$.

- If $\hat{\theta}_1 \leq \hat{\theta}_2$, then the payoff for seller 1 is $\hat{\theta}_2 - \theta_1$, which is negative.
- If $\hat{\theta}_1 > \hat{\theta}_2$, then seller 1 does not win, and payoff for him is zero.

Thus in this case, the maximum payoff possible is 0.

If $\hat{\theta}_1 = \theta_1$, payoff for seller 1 is 0. By announcing $\hat{\theta}_1 = \theta_1$, his true valuation, seller 1 gets zero payoff, which in this case is a best response.

We can now make the following observations about this example.

- Revealing his true valuation is optimal for seller 1 regardless of what seller 2 announces.
- Similarly, announcing his true valuation is optimal for seller 2 whatever the announcement of seller 1.
- More formally, truth revelation is a weakly dominant strategy for each player.
- Thus this social choice function can be implemented even though the valuations are private values. We simply ask each seller to report his type and then we choose $f(\theta)$.

### 2.3.2 Implementation Through Indirect Mechanisms

The examples above have shown us a possible way in which to try to implement a social choice function. The protocol we followed for implementing the social choice functions was:

- Ask each agent to reveal his or her types $\theta_i$;
- Given the announcements $(\hat{\theta}_1, \ldots, \hat{\theta}_n)$, choose the outcome $x = f(\hat{\theta}_1, \ldots, \hat{\theta}_n) \in X$.

Such a method of trying to implement an SCF is referred to as a direct revelation mechanism. Another approach to implementing a social choice function is the indirect way. Here the mechanism makes the agents interact through an institutional framework in which there are rules governing the actions the agents would be allowed to play and in which there is a way of transforming these actions into a social outcome. The actions the agents choose will depend on their
private values and become the strategies of the players. Auctions provide an example of indirect mechanisms. We provide an example right away.

**Example 2.5** (First Price Procurement Auction). Consider an auctioneer or a buyer and two potential sellers as before. Here each seller submits a sealed bid, \( b_i \geq 0 \) \((i = 1, 2)\). The sealed bids are examined and the seller with the lower bid is declared the winner. If there is a tie, seller 1 is declared the winner. The winning seller receives an amount equal to his bid from the buyer. The losing seller does not receive anything.

Note that there is a subtle difference between the situations in Example 2.3 and Example 2.5. In Example 2.3 (direct mechanism), each seller is asked to announce his type, whereas in Example 2.5 (indirect mechanism), each seller is asked to submit a bid. The bid submitted may (and will) of course depend on the type. Based on the type, the seller has a strategy for bidding. So it becomes a game.

Let us make the following assumptions:

1. \( \theta_1, \theta_2 \) are independently drawn from the uniform distribution on \([0, 1]\).

2. The sealed bid of seller \( i \) takes the form \( b_i(\theta_i) = \alpha_i \theta_i + \beta_i \), where \( \alpha_i \in [0, 1], \beta_i \in [0, 1 - \alpha_i] \).

He has to make sure that \( b_i \in [0, 1] \). The term \( \beta_i \) is like a fixed cost whereas \( \alpha_i \theta_i \) indicates a fraction of the true cost. 

Seller 1’s problem is now to bid in a way to maximize his payoff:

\[
\max_{1 \geq b_1 \geq 0} (b_1 - \theta_1) P\{b_2(\theta_2) \geq b_1\}
\]

\[
P\{b_2(\theta_2) \geq b_1\} = 1 - P\{b_2(\theta_2) < b_1\}
= 1 - P\{\alpha_2 \theta_2 + \beta_2 < b_1\}
= 1 - \frac{b_1 - \beta_2}{\alpha_2} \quad \text{if} \quad b_1 \geq \beta_2
\]

since \( \theta_2 \) is uniform over \([0, 1]\)

\[
= 1 \quad \text{if} \quad b_1 < \beta_2. \quad \tag{2.1}
\]

Thus seller 1’s problem is:

\[
\max_{b_1 \geq \beta_2} (b_1 - \theta_1)(1 - \frac{b_1 - \beta_2}{\alpha_2}).
\]

The solution to this problem is

\[
b_1(\theta_1) = \frac{\alpha_2 + \beta_2}{2} + \frac{\theta_1}{2}. \quad \tag{2.3}
\]
We can show on similar lines that
\[ b_2(\theta_2) = \frac{\alpha_1 + \beta_1}{2} + \frac{\theta_2}{2}. \] (2.4)

As the bid of seller \( i \) takes the form
\[ b_i(\theta_i) = \alpha_i \theta_i + \beta_i, \]
where \( \alpha_i \in [0, 1], \beta_i \in [0, 1 - \alpha_i] \), from the equations (2.3) and (2.4), we obtain \( \alpha_1 = \alpha_2 = \frac{1}{2} \). As the goal of each seller is to maximize the profit and \( \beta_i \in [0, 1 - \alpha_i] \), \( \beta_1 = \beta_2 = \frac{1}{2} \). Then we get
\[ b_1(\theta_1) = \frac{1 + \theta_1}{2} \quad \forall \ \theta_1 \in \Theta_1 = [0, 1] \]
\[ b_2(\theta_2) = \frac{1 + \theta_2}{2} \quad \forall \ \theta_2 \in \Theta_2 = [0, 1]. \]

Note that if \( b_2(\theta_2) = \frac{1 + \theta_2}{2} \), the best response of seller 1 is \( b_1(\theta_1) = \frac{1 + \theta_1}{2} \) and vice-versa. Hence the profile \( \left( \frac{1 + \theta_1}{2}, \frac{1 + \theta_2}{2} \right) \) is a Bayesian Nash equilibrium of an underlying Bayesian game. In other words, there is a Bayesian Nash equilibrium of an underlying game (induced by the indirect mechanism called the first price procurement auction) that (indirectly) yields the outcome
\[ f(\theta) = (y_0(\theta), y_1(\theta), y_2(\theta), t_0(\theta), t_1(\theta), t_2(\theta)) \]
such that
\[
\begin{align*}
y_0(\theta) &= 0 \quad \forall \ \theta \in \Theta \\
y_1(\theta) &= 1 \quad \text{if } \theta_1 \leq \theta_2 \\
&= 0 \quad \text{else} \\
y_2(\theta) &= 1 \quad \text{if } \theta_1 > \theta_2 \\
&= 0 \quad \text{else} \\
t_1(\theta) &= \frac{1 + \theta_1}{2} y_1(\theta) \\
t_2(\theta) &= \frac{1 + \theta_2}{2} y_2(\theta) \\
t_0(\theta) &= -(t_1(\theta) + t_2(\theta)).
\end{align*}
\]

The above SCF is precisely SCF-PROC3 that we had introduced in Example 2.2.

---

**Example 2.6 (Second Price Procurement Auction).** Here, each seller is asked to submit a sealed bid \( b_i \geq 0 \). The bids are examined, and the seller with the lower bid is declared the winner. In case there is a tie, seller 1 is declared the winner. The winning seller receives as payment from the buyer an amount equal to the second lowest bid. The losing bidder does not receive
anything. In this case, we can show that $b_i(\theta_i) = \theta_i$ for $i = 1, 2$ constitutes a weakly dominant strategy for each player. The arguments are identical to those in Example 2.4.

Thus the game induced by the indirect mechanism second price procurement auction has a weakly dominant strategy in which the social choice function SCF-PROC2 is implemented.

We can summarize the findings of the current section so far in the following way.

- The function SCF-PROC1 cannot be implemented.

- The function SCF-PROC2 can be implemented in dominant strategies by a direct mechanism. Also, the indirect mechanism, namely second price procurement auction, implements SCF-PROC2 in dominant strategies.

- The function SCF-PROC3 is implemented in Bayesian Nash equilibrium by an indirect mechanism, the first price procurement auction.

### 2.3.3 Bayesian Game Induced by a Mechanism

Recall that a mechanism is an institution or a framework with a set of rules that prescribe the actions available to players and specify how these action profiles are transformed into outcomes. A mechanism specifies an action set for each player. The outcome function gives the rule for obtaining outcomes from action profiles. Given:

1. a set of agents $\{1, 2, \ldots, n\}$,

2. type sets $\Theta_1, \ldots, \Theta_n$,

3. a common prior $\phi \in \Delta(\Theta)$,

4. a set of outcomes $X$,

5. utility functions $u_1, \ldots, u_n$, with $u_i : X \times \Theta_i \rightarrow \mathbb{R}$,

a mechanism $M = (S_1, \ldots, S_n, g(.))$ induces a Bayesian game

$$(N, (\Theta_i), (S_i), (p_i), (U_i))$$

among the players where

$$U_i(\theta_1, \ldots, \theta_n, s_1, \ldots, s_n) = u_i(g(s_1, \ldots, s_n), \theta_i).$$
Strategies in the Induced Bayesian Game

A strategy \( s_i \) for an agent \( i \) in the induced Bayesian game is a function \( s_i : \Theta_i \rightarrow S_i \). Thus, given a private value \( \theta_i \in \Theta_i \), \( s_i(\theta_i) \) will give the action of player \( i \). The strategy \( s_i(\cdot) \) will specify actions corresponding to private values. In the auction scenario, the bid \( b_i \) of player \( i \) is a function of his valuation \( \theta_i \). For example, \( b_i(\theta_i) = \alpha_i \theta_i + \beta_i \) is a particular strategy for player \( i \).

Figure 2.3 captures the idea behind an indirect mechanism and the Bayesian game that is induced by an indirect mechanism.

Figure 2.3: The idea behind implementation by an indirect mechanism

Example 2.7 (Bayesian Game Induced by First Price Procurement Auction). First, note that \( N = \{0, 1, 2\} \). The type sets are \( \Theta_0, \Theta_1, \Theta_2 \), and the common prior is \( \phi \in \Delta(\Theta) \). The set of outcomes is

\[
X = \{(y_0, y_1, y_2, t_0, t_1, t_2) : y_i \in \{0, 1\}, y_0 + y_1 + y_2 = 1, t_i \in \mathbb{R}, t_0 + t_1 + t_2 \leq 0\}
\]

\[
u_i((y_0, y_1, y_2, t_0, t_1, t_2), \theta_i) = -\theta_i y_i + t_i \quad i = 1, 2
\]

\[
u_0((y_0, y_1, y_2, t_0, t_1, t_2), \theta_0) = \theta_0 y_0 + t_0
\]

\[
S_1 = \mathbb{R}_+; \quad S_2 = \mathbb{R}_+
\]
\[ g(b_0, b_1, b_2) = (y_0(b_0, b_1, b_2), y_1(b_0, b_1, b_2), y_2(b_0, b_1, b_2),
\]
\[ t_0(b_0, b_1, b_2), t_1(b_0, b_1, b_2), t_2(b_0, b_1, b_2)) \]

such that

\[ y_0(b_0, b_1, b_2) = 0 \quad \forall \ b_0, b_1, b_2 \]
\[ y_1(b_0, b_1, b_2) = 1 \quad \text{if} \ \ b_1 \leq b_2 \]
\[ = 0 \quad \text{if} \ \ b_1 > b_2 \]
\[ y_2(b_0, b_1, b_2) = 1 \quad \text{if} \ \ b_1 > b_2 \]
\[ = 0 \quad \text{if} \ \ b_1 \leq b_2 \]
\[ t_1(b_0, b_1, b_2) = b_1 y_1(b_0, b_1, b_2) \]
\[ t_2(b_0, b_1, b_2) = b_2 y_2(b_0, b_1, b_2) \]
\[ t_0(b_0, b_1, b_2) = -(t_1(b_0, b_1, b_2) + t_2(b_0, b_1, b_2)) \]

The game induced by the second price procurement auction will be similar except for appropriate changes in \( t_1 \) and \( t_2 \).

### 2.3.4 Implementation of a Social Choice Function by a Mechanism

We now formalize the notion of implementation of a social choice function by a mechanism.

**Definition 2.4** (Implementation of an SCF). *We say that a mechanism \( \mathcal{M} = ((S_i)_{i \in N}, g(\cdot)) \) implements the social choice function \( f(\cdot) \) if there is a pure strategy equilibrium \( s^*(\cdot) = (s^*_1(\cdot), \ldots, s^*_n(\cdot)) \) of the Bayesian game \( \Gamma^b \) induced by \( \mathcal{M} \) such that \( \forall (\theta_1, \ldots, \theta_n) \in \Theta, \)

\[ g(s^*_1(\theta_1), \ldots, s^*_n(\theta_n)) = f(\theta_1, \ldots, \theta_n) \]

Figure \ref{fig:mechanism} explains the idea behind a mechanism implementing a social choice function. Depending on the nature of the underlying equilibrium, two ways of implementing an SCF \( f(\cdot) \) are standard in the literature.

**Definition 2.5** (Implementation in Dominant Strategies). *We say that a mechanism \( \mathcal{M} = ((S_i)_{i \in N}, g(\cdot)) \) implements the social choice function \( f(\cdot) \) in dominant strategy equilibrium if there is a weakly dominant strategy equilibrium \( s^*(\cdot) = (s^*_1(\cdot), \ldots, s^*_n(\cdot)) \) of the game \( \Gamma^b \) induced by \( \mathcal{M} \) such that

\[ g(s^*_1(\theta_1), \ldots, s^*_n(\theta_n)) = f(\theta_1, \ldots, \theta_n) \ \forall \ (\theta_1, \ldots, \theta_n) \in \Theta. \]
Note 2.1. Since a strongly dominant strategy equilibrium is automatically a weakly dominant strategy equilibrium, the above definition applies to the strongly dominant case also. In the latter case, we could say the implementation is in strongly dominant strategy equilibrium. It is worth recalling that there could exist multiple weakly dominant strategy equilibria whereas a strongly dominant strategy equilibrium, if it exists, is unique.

Definition 2.6 (Implementation in Bayesian Nash Equilibrium). We say that a mechanism \( \mathcal{M} = ((S_i)_{i \in N}, g(\cdot)) \) implements the social choice function \( f(\cdot) \) in Bayesian Nash equilibrium if there is a pure strategy Bayesian Nash equilibrium \( s^*(\cdot) = (s^*_1(\cdot), \ldots, s^*_n(\cdot)) \) of the game \( \Gamma^b \) induced by \( \mathcal{M} \) such that

\[
g(s^*_1(\theta_1), \ldots, s^*_n(\theta_n)) = f(\theta_1, \ldots, \theta_n) \quad \forall (\theta_1, \ldots, \theta_n) \in \Theta.
\]

Note 2.2. In the definition, what is implicitly implied is a pure strategy Bayesian Nash equilibrium. Such an equilibrium may or may not exist, but we implicitly assume that such an equilibrium exists.

Note 2.3. The game \( \Gamma^b \) induced by the mechanism \( \mathcal{M} \) may have more than one equilibrium, but the above definition requires only that one of them induces outcomes in accordance with the SCF \( f(\cdot) \). Implicitly, then, the above definition assumes that, if multiple equilibria exist, the agents will play the equilibrium that the mechanism designer (social planner) wants. This is an extremely important problem that is addressed by a theory called implementation theory. A brief idea about implementation theory will be provided in Section 2.21.

Note 2.4. Another implicit assumption of the above definition is that the game induced by the mechanism is a simultaneous move game, that is all the agents, after learning their types, choose their actions simultaneously.

2.4 Incentive Compatibility and the Revelation Theorem

The notion of incentive compatibility is perhaps the most fundamental concept in mechanism design, and the revelation theorem is perhaps the most fundamental result in mechanism design. We have already seen that mechanism design involves the preference revelation (or elicitation) problem and the preference aggregation problem. The preference revelation problem involves eliciting truthful information from the agents about their types. In order to elicit truthful information, there is a need to somehow make truth revelation a best response for the agents, consistent with rationality and intelligence assumptions. Offering incentives is a way of doing this; incentive compatibility essentially refers to offering the right amount of incentive to induce truth revelation by the agents. There are broadly two types of incentive compatibility: (1) Truth
revelation is a best response for each agent irrespective of what is reported by the other agents; (2) Truth revelation is a best response for each agent whenever the other agents also reveal their true types. The first one is called dominant strategy incentive compatibility (DSIC), and the second one is called Bayesian Nash incentive compatibility (BIC). Since truth revelation is always with respect to types, only direct revelation mechanisms are relevant when formalizing the notion of incentive compatibility. The notion of incentive compatibility was first introduced by Leonid Hurwicz [14].

2.4.1 Incentive Compatibility (IC)

Definition 2.7 (Incentive Compatibility). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is said to be incentive compatible (or truthfully implementable) if the Bayesian game induced by the direct revelation mechanism $\mathcal{D} = (((\Theta_i)_{i \in N}, f(\cdot)))$ has a pure strategy equilibrium $s^*(\cdot) = (s^*_1(\cdot), \ldots, s^*_n(\cdot))$ in which $s^*_i(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, \forall i \in N$.

That is, truth revelation by each agent constitutes an equilibrium of the game induced by $\mathcal{D}$. It is easy to infer that if an SCF $f(\cdot)$ is incentive compatible then the direct revelation mechanism $\mathcal{D} = (((\Theta_i)_{i \in N}, f(\cdot)))$ can implement it. That is, directly asking the agents to report their types and using this information in $f(\cdot)$ to get the social outcome will solve both the problems, namely, preference elicitation and preference aggregation.

Based on the type of equilibrium concept used, two types of incentive compatibility are defined.

Definition 2.8 (Dominant Strategy Incentive Compatibility (DSIC)). A social choice function $f : \Theta_1 \times \ldots \times \Theta_n \to X$ is said to be dominant strategy incentive compatible (or truthfully implementable in dominant strategies) if the direct revelation mechanism $\mathcal{D} = (((\Theta_i)_{i \in N}, f(\cdot)))$ has a weakly dominant strategy equilibrium $s^*(\cdot) = (s^*_1(\cdot), \ldots, s^*_n(\cdot))$ in which $s^*_i(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, \forall i \in N$.

That is, truth revelation by each agent constitutes a dominant strategy equilibrium of the game induced by $\mathcal{D}$. Strategyproof, cheat-proof, straightforward are the alternative phrases used for this property.

Example 2.8 (Dominant Strategy Incentive Compatibility of Second Price Procurement Auction). It is easy to see that the social choice function implemented by the second price auction is dominant strategy incentive compatible.
Using the definition of a dominant strategy equilibrium in Bayesian games, the following necessary and sufficient condition for an SCF \( f(\cdot) \) to be dominant strategy incentive compatible can be easily derived:

\[
 u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq u_i(f(\hat{\theta}_i, \theta_{-i}), \theta_i), \forall i \in N, \forall \theta_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}, \forall \hat{\theta}_i \in \Theta_i. \tag{2.5}
\]

The above condition says that if the SCF \( f(\cdot) \) is DSIC, then, irrespective of what the other agents report, it is always a best response for agent \( i \) to report his true type \( \theta_i \).

**Definition 2.9** (Bayesian Incentive Compatibility (BIC)). A social choice function \( f : \Theta_1 \times \ldots \times \Theta_n \rightarrow X \) is said to be Bayesian incentive compatible (or truthfully implementable in Bayesian Nash equilibrium) if the direct revelation mechanism \( D = ((\Theta_i)_{i \in N}, f(\cdot)) \) has a Bayesian Nash equilibrium \( s^*(\cdot) = (s^*_1(\cdot), \ldots, s^*_n(\cdot)) \) in which \( s^*_i(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, \forall i \in N \).

That is, truth revelation by each agent constitutes a Bayesian Nash equilibrium of the game induced by \( D \).

**Example 2.9** (Bayesian Incentive Compatibility of First Price Procurement Auction). We have seen that the first price procurement auction for a single indivisible item implements the following social choice function:

\[
 f(\theta) = (y_0(\theta), y_1(\theta), y_2(\theta), t_0(\theta), t_1(\theta), t_2(\theta))
\]

with

\[
 y_0(\theta) = 0 \quad \forall \theta \in \Theta \\
 y_1(\theta) = 1 \quad \text{if } \theta_1 \leq \theta_2 \\
 \quad = 0 \quad \text{otherwise} \\
 y_2(\theta) = 1 \quad \text{if } \theta_1 > \theta_2 \\
 \quad = 0 \quad \text{otherwise} \\
 t_1(\theta) = \frac{1 + \theta_1}{2} y_1(\theta) \\
 t_2(\theta) = \frac{1 + \theta_2}{2} y_2(\theta) \\
 t_0(\theta) = -(t_1(\theta) + t_2(\theta)).
\]

If seller 1 has type \( \theta_1 \), then his optimal bid \( \hat{\theta}_1 \) is obtained by solving

\[
 \max_{\hat{\theta}_1} \left( \frac{1 + \hat{\theta}_1}{2} - \theta_1 \right) \cdot P\{\theta_2 \geq \hat{\theta}_1\}.
\]
This is the same as
\[ \max_{\theta_1} \left( \frac{1 + \hat{\theta}_1}{2} - \theta_1 \right) (1 - \hat{\theta}_1). \]

This yields \( \hat{\theta}_1 = \theta_1 \). Thus it is optimal for seller 1 to reveal his true private value if seller 2 reveals his true value. The same situation applies to seller 2. This implies that the social choice function is Bayesian Nash incentive compatible (since the equilibrium involved is a Bayesian Nash equilibrium).

Using the definition of a Bayesian Nash equilibrium in Bayesian games, the following necessary and sufficient condition for an SCF \( f(\cdot) \) to be Bayesian incentive compatible can be easily derived:
\[
E_{\theta_{-i}} [u_i ( f(\theta_i, \theta_{-i}), \theta_i) | \theta_i] \geq E_{\theta_{-i}} [u_i (f(\hat{\theta}_i, \theta_{-i}), \theta_i)| \theta_i], \forall i \in N, \forall \theta_i \in \Theta_i, \forall \hat{\theta}_i \in \Theta_i
\]
where the expectation is taken over the type profiles of agents other than agent \( i \).

Note 2.5. If a social choice function \( f(\cdot) \) is dominant strategy incentive compatible then it is also Bayesian incentive compatible. The proof of this follows trivially from the fact that a weakly dominant strategy equilibrium is necessarily a Bayesian Nash equilibrium.

### 2.4.2 The Revelation Principle for Dominant Strategy Equilibrium

The revelation principle basically illustrates the relationship between an indirect mechanism \( \mathcal{M} \) and a direct revelation mechanism \( \mathcal{D} \) with respect to a given SCF \( f(\cdot) \). This result enables us to restrict our inquiry about truthful implementation of an SCF to the class of direct revelation mechanisms only.

**Theorem 2.1.** Suppose that there exists a mechanism \( \mathcal{M} = (S_1, \ldots, S_n, g(\cdot)) \) that implements the social choice function \( f(\cdot) \) in dominant strategy equilibrium. Then \( f(\cdot) \) is dominant strategy incentive compatible.

**Proof:**

If \( \mathcal{M} = (S_1, \ldots, S_n, g(\cdot)) \) implements \( f(\cdot) \) in dominant strategies, then there exists a profile of strategies \( s^*(\cdot) = (s_1^*(\cdot), \ldots, s_n^*(\cdot)) \) such that
\[
g(s_1^*(\theta_1), \ldots, s_n^*(\theta_n)) = f(\theta_1, \ldots, \theta_n) \quad \forall (\theta_1, \ldots, \theta_n) \in \Theta
\]
and
\[
u_i(g(s_i^*(\theta_i), s_{-i}(\theta_{-i})), \theta_i) \geq u_i(g(s_i'(\theta_i), s_{-i}(\theta_{-i})), \theta_i)
\]
\[\forall i \in N, \forall \theta_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}, \forall s_i'(\cdot) \in S_i, \forall s_{-i}(\cdot) \in S_{-i} \tag{2.8}\]
Condition (2.8) implies, in particular, that
\[ u_i(g(s_i^*(\theta_i), s_{-i}^*(\theta_{-i})), \theta_i) \geq u_i(g(\hat{s}_i^*(\hat{\theta}_i), s_{-i}^*(\theta_{-i})), \theta_i) \]
\[ \forall i \in N, \forall \theta_i \in \Theta_i, \forall \hat{\theta}_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}. \]  
(2.9)

Conditions (2.7) and (2.9) together imply that
\[ u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq u_i(f(\hat{\theta}_i, \theta_{-i}), \theta_i), \forall i \in N, \forall \theta_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}, \forall \hat{\theta}_i \in \Theta_i. \]

But this is precisely condition (2.5), the condition for \( f(\cdot) \) to be truthfully implementable in dominant strategies.

The idea behind the revelation principle can be understood with the help of Figure 2.4. In this picture, DSI represents the set of all social choice functions that are implementable in dominant strategies and DSIC is the set of all social choice functions that are dominant strategy incentive compatible. The picture depicts the obvious fact that DSIC is a subset of DSI and illustrates the revelation theorem by showing that the set difference between these two sets is the empty set, thus implying that DSIC is precisely the same as DSI.
2.4.3 The Revelation Principle for Bayesian Nash Equilibrium

**Theorem 2.2.** Suppose that there exists a mechanism \( \mathcal{M} = (S_1, \ldots, S_n, g(\cdot)) \) that implements the social choice function \( f(\cdot) \) in Bayesian Nash equilibrium. Then \( f(\cdot) \) is truthfully implementable in Bayesian Nash equilibrium (Bayesian incentive compatible).

![Diagram](image)

**Figure 2.5:** Revelation principle for Bayesian Nash equilibrium

**Proof:**

If \( \mathcal{M} = (S_1, \ldots, S_n, g(\cdot)) \) implements \( f(\cdot) \) in Bayesian Nash equilibrium, then there exists a profile of strategies \( s^*(\cdot) = (s^*_1(\cdot), \ldots, s^*_n(\cdot)) \) such that

\[
g(s^*_1(\theta_1), \ldots, s^*_n(\theta_n)) = f(\theta_1, \ldots, \theta_n) \forall (\theta_1, \ldots, \theta_n) \in \Theta
\]

(2.10)

and

\[
E_{\theta_{-i}} \left[ u_i(g(s^*_1(\theta_i), s^*_i(\theta_{-i})), \theta_i)|\theta_1 \right] \geq E_{\theta_{-i}} \left[ u_i(g(s'_{i}(\theta_i), s^*_i(\theta_{-i})), \theta_i)|\theta_1 \right]
\forall i \in N, \forall \theta_i \in \Theta_i, \forall s'_i(\cdot) \in S_i.
\]

(2.11)

Condition (2.11) implies, in particular, that

\[
E_{\theta_{-i}} \left[ u_i(g(s^*_1(\theta_i), s^*_i(\theta_{-i})), \theta_i)|\theta_1 \right] \geq E_{\theta_{-i}} \left[ u_i(g(s^*_1(\tilde{\theta}_i), s^*_i(\theta_{-i})), \theta_i)|\theta_1 \right]
\forall i \in N, \forall \tilde{\theta}_i \in \Theta_i, \forall \theta_i \in \Theta_i.
\]

(2.12)
Conditions (2.11) and (2.12) together imply that

\[ E_{\theta_{-i}} [u_i (f(\theta_i, \theta_{-i}), \theta_i)] \geq E_{\hat{\theta}_i} [u_i (f(\hat{\theta}_i, \theta_{-i}), \theta_i)], \forall i \in N, \forall \theta_i \in \Theta_i, \forall \hat{\theta}_i \in \Theta_i. \]

But this is precisely condition (2.6), the condition for \( f(\cdot) \) to be truthfully implementable in Bayesian Nash equilibrium.

\[\square\]

In a way similar to the revelation principle for dominant strategy equilibrium, the revelation principle for Bayesian Nash equilibrium can be explained with the help of Figure 2.5. In this picture, BNI represents the set of all social choice functions which are implementable in Bayesian Nash equilibrium and BIC is the set of all social choice functions which are Bayesian incentive compatible. The picture depicts the fact that BIC is a subset of BNI and illustrates the revelation theorem by showing that the set difference between these two sets is the empty set, thus implying that BIC is precisely the same as BNI.

Figure 2.6 provides a combined view of both the revelation theorems that we have seen in this section.

![Figure 2.6: Combined view of revelation theorems for dominant strategy equilibrium and Bayesian Nash equilibrium](image-url)
2.5 Properties of Social Choice Functions

We have seen that a mechanism provides a solution to both the preference elicitation problem and preference aggregation problem, if the mechanism can implement the desired social choice function $f(\cdot)$. It is obvious that some SCFs are implementable and some are not. Before we look into the question of characterizing the space of implementable social choice functions, it is important to know which social choice function ideally a social planner would wish to implement. In this section, we highlight a few properties of an SCF that ideally a social planner would wish the SCF to have.

2.5.1 Ex-Post Efficiency

**Definition 2.10 (Ex-Post Efficiency).** The SCF $f : \Theta \rightarrow X$ is said to be ex-post efficient (or Paretoian) if for every profile of agents’ types, $\theta \in \Theta$, the outcome $f(\theta)$ is a Pareto optimal outcome. The outcome $f(\theta_1, \ldots, \theta_n)$ is Pareto optimal if there does not exist any $x \in X$ such that:

$$u_i(x, \theta_i) \geq u_i(f(\theta), \theta_i) \ \forall \ i \in N \text{ and } u_i(x, \theta_i) > u_i(f(\theta), \theta_i) \text{ for some } i \in N.$$ 

**Example 2.10 (Procurement of a Single Indivisible Item).** We have looked at three social choice functions, SCF-PROC1, SCF-PROC2, SCF-PROC3, in the previous section. One can show that all these SCFs are ex-post efficient.

2.5.2 Dictatorship in SCFs

We define this through a dictatorial social choice function.

**Definition 2.11 (Dictatorship).** A social choice function $f : \Theta \rightarrow X$ is said to be dictatorial if there exists an agent $d$ (called dictator) who satisfies the following property:

$$\forall \theta \in \Theta, \ f(\theta) \text{ is such that } u_d(f(\theta), \theta_d) \geq u_d(x, \theta_d) \ \forall x \in X.$$ 

A social choice function that is not dictatorial is said to be nondictatorial.

In a dictatorial SCF, every outcome that is picked by the SCF is such that it is a most favored outcome for the dictator.

2.5.3 Individual Rationality

Individual rationality is also often referred to as voluntary participation property. Individual rationality of a social choice function essentially means that each agent gains a nonnegative
utility by participating in a mechanism that implements the social choice function. There are three stages at which individual rationality constraints (also called participation constraints) may be relevant in a mechanism design situation.

**Ex-Post Individual Rationality**

These constraints become relevant when any agent $i$ is given a choice to withdraw from the mechanism at the ex-post stage, that is, after all the agents have announced their types and an outcome in $X$ has been chosen. Let $\pi_i(\theta_i)$ be the utility that agent $i$ receives by withdrawing from the mechanism when his type is $\theta_i$. Then, to ensure agent $i$’s participation, we must satisfy the following *ex-post participation (or individual rationality) constraints*

$$u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq \pi_i(\theta_i) \quad \forall (\theta_i, \theta_{-i}) \in \Theta.$$

**Interim Individual Rationality**

Let the agent $i$ be allowed to withdraw from the mechanism only at an interim stage that arises after the agents have learned their type but before they have chosen their actions in the mechanism. In such a situation, the agent $i$ will participate in the mechanism only if his interim expected utility $U_i(\theta_i|f) = E_{\theta_{-i}}[u_i(f(\theta_i, \theta_{-i}), \theta_i)|\theta_i]$ from social choice function $f(\cdot)$, when his type is $\theta_i$, is greater than $\pi_i(\theta_i)$. Thus, *interim participation (or individual rationality) constraints* for agent $i$ require that

$$U_i(\theta_i|f) = E_{\theta_{-i}}[u_i(f(\theta_i, \theta_{-i}), \theta_i)|\theta_i] \geq \pi_i(\theta_i) \quad \forall \theta_i \in \Theta_i.$$

**Ex-Ante Individual Rationality**

Let agent $i$ be allowed to refuse to participate in a mechanism only at ex-ante stage, that is, before the agents learn their type. In such a situation, the agent $i$ will participate in the mechanism only if his ex-ante expected utility $U_i(f) = E_{\theta}[u_i(f(\theta_i, \theta_{-i}), \theta_i)]$ from social choice function $f(\cdot)$ is at least $E_{\theta_i}[\pi_i(\theta_i)]$. Thus, *ex-ante participation (or individual rationality) constraints* for agent $i$ require that

$$U_i(f) = E_{\theta}[u_i(f(\theta_i, \theta_{-i}), \theta_i)] \geq E_{\theta_i}[\pi_i(\theta_i)].$$

The following proposition establishes a relationship among the three different participation constraints discussed above. The proof is left as an exercise.
**Proposition 2.1.** For any social choice function $f(\cdot)$, we have

$$f(\cdot) \text{ is ex-post IR } \Rightarrow f(\cdot) \text{ is interim IR } \Rightarrow f(\cdot) \text{ is ex-ante IR}.$$ 

### 2.5.4 Efficiency

We have seen the notion of ex-post efficiency already. Depending on the epoch at which we look into the game, we have three notions of efficiency, on the lines of individual rationality. These notions were introduced by Holmstorm and Myerson [15]. Let $F$ be any collection of social choice functions that are of interest.

**Definition 2.12 (Ex-Ante Efficiency).** For any given set of social choice functions $F$, and any member $f(\cdot) \in F$, we say that $f(\cdot)$ is ex-ante efficient in $F$ if there is no other $\hat{f}(\cdot) \in F$ having the following two properties:

\[
E_{\theta}[u_i(\hat{f}(\theta), \theta_i)] \geq E_{\theta}[u_i(f(\theta), \theta_i)] \quad \forall \, i = 1, \ldots, n,
\]
\[
E_{\theta}[u_i(\hat{f}(\theta), \theta_i)] > E_{\theta}[u_i(f(\theta), \theta_i)] \text{ for some } i.
\]

**Definition 2.13 (Interim Efficiency).** For any given set of social choice functions $F$, and any member $f(\cdot) \in F$, we say that $f(\cdot)$ is interim efficient in $F$ if there is no other $\hat{f}(\cdot) \in F$ having the following two properties:

\[
E_{\theta_i}[u_i(\hat{f}(\theta), \theta_i)|\theta_i] \geq E_{\theta_i}[u_i(f(\theta), \theta_i)|\theta_i] \quad \forall \, i = 1, \ldots, n, \forall \, \theta_i \in \Theta_i,
\]
\[
E_{\theta_i}[u_i(\hat{f}(\theta), \theta_i)|\theta_i] > E_{\theta_i}[u_i(f(\theta), \theta_i)|\theta_i] \text{ for some } i \text{ and some } \theta_i \in \Theta_i.
\]

**Definition 2.14 (Ex-Post Efficiency).** For any given set of social choice functions $F$, and any member $f(\cdot) \in F$, we say that $f(\cdot)$ is ex-post efficient in $F$ if there is no other $\hat{f}(\cdot) \in F$ having the following two properties:

\[
 u_i(\hat{f}(\theta), \theta_i) \geq u_i(f(\theta), \theta_i) \quad \forall \, i = 1, \ldots, n, \forall \, \theta \in \Theta,
\]
\[
 u_i(\hat{f}(\theta), \theta_i) > u_i(f(\theta), \theta_i) \text{ for some } i \text{ and some } \theta \in \Theta.
\]

Using the above definition of ex-post efficiency, we can say that a social choice function $f(\cdot)$ is ex-post efficient in the sense of Definition 2.14 if and only if it is ex-post efficient in the sense of Definition 2.11 when we take $F = \{ f : f \text{ is a mapping from } \Theta \text{ to } X \}$.

The following proposition establishes a relationship among these three different notions of efficiency.

41
Proposition 2.2. Given any set of feasible social choice functions $F$ and $f(\cdot) \in F$, we have

\[ f(\cdot) \text{ is ex-ante efficient} \Rightarrow f(\cdot) \text{ is interim efficient} \Rightarrow f(\cdot) \text{ is ex-post efficient}. \]

For a proof of the above proposition, refer to Proposition 23.F.1 of [16]. Also, compare the above proposition with the Proposition 2.1.

2.6 The Gibbard–Satterthwaite Impossibility Theorem

We have seen in the last section that dominant strategy incentive compatibility is an extremely desirable property of social choice functions. However the DSIC property, being a strong one, precludes certain other desirable properties to be satisfied. In this section, we discuss the Gibbard–Satterthwaite impossibility theorem (G–S theorem, for short), which shows that the DSIC property will force an SCF to be dictatorial if the utility environment is an unrestricted one. In fact, in the process, even ex-post efficiency will have to be sacrificed. One can say that the G–S theorem has shaped the course of research in mechanism design during the 1970s and beyond, and is therefore a landmark result in mechanism design theory. The G–S theorem is credited independently to Gibbard in 1973 [17] and Satterthwaite in 1975 [18]. The G–S theorem is a brilliant reinterpretation of the famous Arrow’s impossibility theorem (which we discuss in the next section). We start our discussion of the G–S theorem with a motivating example.

Example 2.11 (Supplier Selection Problem). We have $N = \{1, 2\}$, $X = \{x, y, z\}$, $\Theta_1 = \{a_1\}$, and $\Theta_2 = \{a_2, b_2\}$. Consider the following utility functions:

\[
\begin{align*}
u_1(x, a_1) &= 100; \quad & u_1(y, a_1) &= 50; \quad & u_1(z, a_1) &= 0 \\
u_2(x, a_2) &= 0; \quad & u_2(y, a_2) &= 50; \quad & u_2(z, a_2) &= 100 \\
u_2(x, b_2) &= 30; \quad & u_2(y, b_2) &= 60; \quad & u_2(z, b_2) &= 20.
\end{align*}
\]

We observe for this example that the DSIC and BIC notions are identical since the type of player 1 is common knowledge and hence player 1 always reports the true type (since the type set is a singleton). Consider the social choice function $f$ given by $f(a_1, a_2) = x$; $f(a_1, b_2) = x$. It can be seen that this SCF is ex-post efficient.

To investigate DSIC, suppose the type of player 2 is $a_2$. If player 2 reports his true type, then the outcome is $x$. If he misreports his type as $b_2$, then also the outcome is $x$. Hence there is no incentive for player 2 to misreport. A similar situation presents itself when the type of player 2 is $b_2$. Thus $f$ is DSIC.
In both the type profiles, the outcome happens to be the most favorable one for player 1, that is, \( x \). Therefore, player 1 is a dictator and \( f \) is dictatorial. Thus the above function is ex-post efficient and DSIC but dictatorial.

Now, let us consider a different SCF \( h \) defined by \( h(a_1, a_2) = y; h(a_1, b_2) = x \). Following similar arguments as above, \( h \) can be shown to be ex-post efficient and nondictatorial but not DSIC. Table 2.1 lists all the nine possible social choice functions in this scenario and the combination of properties each function satisfies.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( f_1(a_1, a_2) )</th>
<th>( f_1(a_1, b_2) )</th>
<th>EPE</th>
<th>DSIC</th>
<th>NON-DICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x )</td>
<td>( x )</td>
<td>√</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>2</td>
<td>( x )</td>
<td>( y )</td>
<td>√</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>3</td>
<td>( x )</td>
<td>( z )</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>4</td>
<td>( y )</td>
<td>( x )</td>
<td>√</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>5</td>
<td>( y )</td>
<td>( y )</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>6</td>
<td>( y )</td>
<td>( z )</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>7</td>
<td>( z )</td>
<td>( x )</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>8</td>
<td>( z )</td>
<td>( y )</td>
<td>√</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>9</td>
<td>( z )</td>
<td>( z )</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Table 2.1: Social choice functions and properties satisfied by them

Note that the situation is quite desirable with the following SCFs.

\[
\begin{align*}
f_5(a_1, a_2) &= y; \quad f_5(a_1, b_2) = y \\
f_7(a_1, a_2) &= z; \quad f_7(a_1, b_2) = x.
\end{align*}
\]

The reason is these functions are ex-post efficient, DSIC, and also nondictatorial. Unfortunately however, such desirable situations do not occur in general. In the present case, the desirable situations do occur because of certain reasons that will become clear soon. In a general setting, ex-post efficiency, DSIC, and nondictatorial properties can never be satisfied simultaneously. In fact, even DSIC and nondictatorial properties cannot coexist. This is the implication of the powerful Gibbard–Satterthwaite Theorem, (G–S Theorem).

### 2.6.1 The G–S Theorem

We will build up some notation before presenting the theorem. We have already seen that the preference of an agent \( i \), over the outcome set \( X \), when its type is \( \theta_i \) can be described by means of
utility function \( u_i(\cdot, \theta_i) : X \to \mathbb{R} \), which assigns a real number to each element in \( X \). A utility function \( u_i(\cdot, \theta_i) \) always induces a unique preference relation \( \succsim \) on \( X \) which can be described in the following manner
\[
x \succsim y \iff u_i(x, \theta_i) \geq u_i(y, \theta_i).
\]

The above preference relation is often called a rational preference relation and it is formally defined as follows.

**Definition 2.15 (Rational Preference Relation).** We say that a relation \( \succsim \) on the set \( X \) is called a rational preference relation if it possesses the following three properties:

1. **Reflexivity:** \( \forall x \in X \), we have \( x \succsim x \).
2. **Completeness:** \( \forall x, y \in X \), we have that \( x \succsim y \) or \( y \succsim x \) (or both).
3. **Transitivity:** \( \forall x, y, z \in X \), if \( x \succsim y \) and \( y \succsim z \), then \( x \succsim z \).

The following proposition establishes the relationship between these two ways of expressing the preferences of an agent \( i \) over the set \( X \).

**Proposition 2.3.**

1. If a preference relation \( \succsim \) on \( X \) is induced by some utility function \( u_i(\cdot, \theta_i) \), then it will be a rational preference relation.
2. For every preference relation \( \succsim \) on \( X \), there may not exist a utility function that induces it. However, when the set \( X \) is finite, given any preference relation, there will exist a utility function that induces it.
3. For a given preference relation \( \succsim \) on \( X \), there might be several utility functions that induce it. Indeed, if the utility function \( u_i(\cdot, \theta_i) \) induces \( \succsim \), then \( u'_i(x, \theta_i) = f(u_i(x, \theta_i)) \) is another utility function that will also induce \( \succsim \), where \( f : \mathbb{R} \to \mathbb{R} \) is a strictly increasing function.

**Strict Total Preference Relations**

We now define a special class of rational preference relations that satisfy the antisymmetry property also.

**Definition 2.16 (Strict-total Preference Relation).** We say that a rational preference relation \( \succsim \) is strict-total if it possesses the antisymmetry property, in addition to reflexivity, completeness, and transitivity. By antisymmetry, we mean that, for any \( x, y \in X \) such that \( x \neq y \), we have either \( x \succsim y \) or \( y \succsim x \), but not both.
The strict-total preference relation is also known as a *linear order relation* because it satisfies the properties of the usual *greater than or equal to* relationship on the real line. Let us denote the set of all rational preference relations and strict-total preference relations on the set $X$ by $\mathcal{R}$ and $\mathcal{P}$, respectively. It is easy to see that $\mathcal{P} \subset \mathcal{R}$.

**Ordinal Preference Relations**

In a mechanism design problem, for agent $i$, the preference over the set $X$ is described in the form of a utility function $u_i : X \times \Theta_i \to \mathbb{R}$. That is, for every possible type $\theta_i \in \Theta_i$ of agent $i$, we can define a utility function $u_i(\cdot, \theta_i)$ over the set $X$. Let this utility function induce a rational preference relation $\succeq_i(\theta_i)$ over $X$. The set $\mathcal{R}_i = \{ \succeq : \succeq = \succeq_i(\theta_i) \text{ for some } \theta_i \in \Theta_i \}$ is known as the set of ordinal preference relations for agent $i$. It is easy to see that $\mathcal{R}_i \subset \mathcal{R} \quad \forall \ i \in N$.

With all the above notions in place, we are now in a position to state the G–S theorem.

**Theorem 2.3** (Gibbard–Satterthwaite Impossibility Theorem). Consider a social choice function $f : \Theta \to X$. Suppose that

1. The outcome set $X$ is finite and contains at least three elements,
2. $\mathcal{R}_i = \mathcal{P} \quad \forall \ i \in N,$
3. $f(\cdot)$ is an onto mapping, that is, the image of SCF $f(\cdot)$ is the set $X$.

Then the social choice function $f(\cdot)$ is dominant strategy incentive compatible iff it is dictatorial.

For a proof of this theorem, the reader is referred to Proposition 23.C.3 of the book by Mas-Colell, Whinston, and Green [16]. We only provide a brief outline of the proof. To prove the necessity, we assume that the social choice function $f(\cdot)$ is dictatorial and it is shown that $f(\cdot)$ is DSIC. This can be shown in a fairly straightforward way. The proof of the sufficiency part of the theorem starts with the assumption that $f(\cdot)$ is DSIC and proceeds in three steps:

1. It is shown using the second condition of the theorem ($\mathcal{R}_i = \mathcal{P} \quad \forall \ i \in N$) that $f(\cdot)$ is monotonic.
2. Next using conditions (2) and (3) of the theorem, it is shown that monotonicity implies ex-post efficiency.
3. Finally, it is shown that a SCF $f(\cdot)$ that is monotonic and ex-post efficient is necessarily dictatorial.

Figure 2.7 shows a pictorial representation of the G–S theorem. The figure depicts two classes $F_1$ and $F_2$ of social choice functions. The class $F_1$ is the set of all SCFs that satisfy conditions
(1) and (2) of the theorem while the class $F_2$ is the set of all SCFs that satisfy conditions (1) and (3) of the theorem. The class $GS$ is the set of all SCFs in the intersection of $F_1$ and $F_2$ which are DSIC. The functions in the class GS have to be necessarily dictatorial.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{gibbard-satterthwaite-theorem.png}
\caption{An illustration of the Gibbard–Satterthwaite Theorem}
\end{figure}

\subsection*{2.6.2 Implications of the G–S Theorem}

One way to get around the impossible situation described by the G–S Theorem is to hope that at least one of the conditions (1), (2), and (3) of the theorem does not hold. We discuss each one of these below.

- Condition (1) asserts that $|X| \geq 3$. This condition is violated only if $|X| = 1$ or $|X| = 2$. The case $|X| = 1$ corresponds to a trivial situation and is not of interest. The case $|X| = 2$ is more interesting but is of only limited interest. A public project problem where only a go or no-go decision is involved and no payments by agents are involved corresponds to this situation.

- Condition (2) asserts that $R_i = \mathcal{P} \forall i \in N$. This means that the preferences of each agent
cover the entire space of strict total preference relations on $X$. That is, each agent has an extremely rich set of preferences. If we are able to somehow restrict the preferences, we can hope to violate this condition. One can immediately note that this condition was violated in the motivating example (Example 2.11, the supplier selection problem). The celebrated class of VCG mechanisms has been derived by restricting the preferences to the quasilinear domain. This will be discussed in good detail in a later section.

- Condition (3) asserts that $f$ is an onto function. Note that this condition also was violated in Example 2.11. This provides one more route for getting around the G–S Theorem.

Another way of escaping from the jaws of the G–S Theorem is to settle for a weaker form of incentive compatibility than DSIC. We have already discussed Bayesian incentive compatibility (BIC) which only guarantees that reporting true types is a best response for each agent whenever all other agents also report their true types. Following this route leads us to Bayesian incentive compatible mechanisms. These are discussed in good detail in a future section.

The G–S Theorem is an influential result that defined the course of mechanism design research in the 1970s and 1980s.

2.7 The Quasilinear Environment

This is the most extensively studied special class of environments where the Gibbard–Satterthwaite theorem does not hold. In the quasilinear environment, an alternative $x \in X$ is a vector of the form $x = (k, t_1, \ldots, t_n)$, where $k$ is an element of a set $K$, which is called the set of project choices or set of allocations. The set $K$ is usually assumed to be finite. The term $t_i \in \mathbb{R}$ represents the monetary transfer to agent $i$. If $t_i > 0$ then agent $i$ will receive the money and if $t_i < 0$ then agent $i$ will pay the money. We assume that we are dealing with a system in which the $n$ agents have no external source of funding, i.e., $\sum_{i=1}^{n} t_i \leq 0$. This condition is known as the weak budget.
balance condition. The set of alternatives \( X \) is therefore

\[
X = \left\{ (k, t_1, \ldots, t_n) : k \in K; \ t_i \in \mathbb{R} \ \forall \ i \in N; \ \sum_i t_i \leq 0 \right\}.
\]

A social choice function in this quasilinear environment takes the form \( f(\theta) = (k(\theta), t_1(\theta), \ldots, t_n(\theta)) \) where, for every \( \theta \in \Theta \), we have \( k(\theta) \in K \) and \( \sum_i t_i(\theta) \leq 0 \). Note that here we are using the symbol \( k \) both as an element of the set \( K \) and as a function going from \( \Theta \) to \( K \). It should be clear from the context as to which of these two we are referring. For a direct revelation mechanism \( D = ((\Theta_i)_{i \in N}, f(\cdot)) \) in this environment, the agent \( i \)'s utility function takes the quasilinear form

\[
u_i(x, \theta_i) = u_i((k, t_1, \ldots, t_n), \theta_i) = v_i(k, \theta_i) + m_i + t_i
\]

where \( m_i \) is agent \( i \)'s initial endowment of the money and the function \( v_i(\cdot) \) is known as agent \( i \)'s valuation function. Recall from our discussion of mechanism design environment (Section 2.1) that the utility functions \( u_i(\cdot) \) are common knowledge. In the context of a quasilinear environment, this implies that for any given type \( \theta_i \) of any agent \( i \), the social planner and every other agent \( j \) have a way to know the function \( v_i(\cdot, \theta_i) \). In many cases, the set \( \Theta_i \) of the direct revelation mechanism \( D = ((\Theta_i)_{i \in N}, f(\cdot)) \) is actually the set of all feasible valuation functions \( v_i \) of agent \( i \). That is, each possible function represents the possible types of agent \( i \). Therefore, in such settings, reporting a type is the same as reporting a valuation function.

Immediate examples of quasilinear environment include many of the previously discussed examples, such as the first price and second price auctions (Example 2.2) etc. In the quasilinear environment, we can define two important properties of a social choice function, namely, allocative efficiency and budget balance.

**Definition 2.17 (Allocative Efficiency (AE)).** We say that a social choice function \( f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot)) \) is allocatively efficient if for each \( \theta \in \Theta \), \( k(\theta) \) satisfies the following condition

\[
k(\theta) \in \arg\max_{k \in K} \sum_{i=1}^n v_i(k, \theta_i).
\]  

Equivalently,

\[
\sum_{i=1}^n v_i(k(\theta), \theta_i) = \max_{k \in K} \sum_{i=1}^n v_i(k, \theta_i).
\]

The above definition implies that for every \( \theta \in \Theta \), the allocation \( k(\theta) \) will maximize the sum of the values of the players. In other words, every allocation is a value maximizing allocation,
or the objects are allocated to the players who value the objects most. This is an extremely
desirable property to have for any social choice function. The above definition implicitly assumes
that for any given $\theta$, the function $\sum_{i=1}^{n} v_i(., \theta_i) : K \rightarrow \mathbb{R}$ attains a maximum over the set $K$.

Definition 2.18 (Budget Balance (BB)). We say that a social choice function $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ is budget balanced if for each $\theta \in \Theta$, $t_1(\theta), \ldots, t_n(\theta)$ satisfy the following condition:

$$\sum_{i=1}^{n} t_i(\theta) = 0.$$ (2.14)

Many authors prefer to call this property strong budget balance, and they refer to the property of having $\sum_{i=1}^{n} t_i(\theta) \leq 0$ as weak budget balance. In this thesis, we will use the term budget balance to refer to strong budget balance.

Budget balance ensures that the total receipts are equal to total payments. This means that the system is a closed one, with no surplus and no deficit. The weak budget balance property means that the total payments are greater than or equal to total receipts.

The following lemma establishes an important relationship of these two properties of an SCF with the ex-post efficiency of the SCF.

Lemma 2.1. A social choice function $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ is ex-post efficient in quasi-linear environment if and only if it is allocatively efficient and budget balanced.

Proof:

Let us assume that $f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ is allocatively efficient and budget balanced. This implies that for any $\theta \in \Theta$, we have

$$\sum_{i=1}^{n} u_i(f(\theta), \theta_i) = \sum_{i=1}^{n} v_i(k(\theta), \theta_i) + \sum_{i=1}^{n} t_i(\theta)$$

$$= \sum_{i=1}^{n} v_i(k(\theta), \theta_i) + 0$$

$$\geq \sum_{i=1}^{n} v_i(k, \theta_i) + \sum_{i=1}^{n} t_i; \forall x = (k, t_1, \ldots, t_n)$$

$$= \sum_{i=1}^{n} u_i(x, \theta_i); \forall (k, t_1, \ldots, t_n) \in X.$$ 

That is if the SCF is allocatively efficient and budget balanced then for any type profile $\theta$ of the agent, the outcome chosen by the social choice function will be such that it maximizes the total utility derived by all the agents. This will automatically imply that the SCF is ex-post efficient.

To prove the other part, we will first show that if $f(\cdot)$ is not allocatively efficient, then, it cannot be ex-post efficient and next we will show that if $f(\cdot)$ is not budget balanced then it
cannot be ex-post efficient. These two facts together will imply that if \( f(\cdot) \) is ex-post efficient then it will have to be allocatively efficient and budget balanced, thus completing the proof of the lemma.

To start with, let us assume that \( f(\cdot) \) is not allocatively efficient. This means that \( \exists \theta \in \Theta \), and \( k \in K \) such that
\[
\sum_{i=1}^{n} v_i(k, \theta_i) > \sum_{i=1}^{n} v_i(k(\theta), \theta_i).
\]
This implies that there exists at least one agent \( j \) for whom \( v_j(k, \theta_i) > v_j(k(\theta), \theta_i) \). Now consider the following alternative \( x \)
\[
x = \left( k, (t_i = t_i(\theta) + v_i(k(\theta), \theta_i) - v_i(k, \theta_i))_{i \neq j}, t_j = t_j(\theta) \right).
\]
It is easy to verify that \( u_i(x, \theta_i) = u_i(f(\theta), \theta_i) \forall i \neq j \) and \( u_j(x, \theta_i) > u_j(f(\theta), \theta_i) \), implying that \( f(\cdot) \) is not ex-post efficient.

Next, we assume that \( f(\cdot) \) is not budget balanced. This means that there exists at least one agent \( j \) for whom \( t_j(\theta) < 0 \). Let us consider the following alternative \( x \)
\[
x = \left( k, (t_i = t_i(\theta) + v_i(k(\theta), \theta_i) - v_i(k, \theta_i))_{i \neq j}, t_j = t_j(\theta) \right).
\]
It is easy to verify that for the above alternative \( x \), we have \( u_i(x, \theta_i) = u_i(f(\theta), \theta_i) \forall i \neq j \) and \( u_j(x, \theta_i) > u_j(f(\theta), \theta_i) \) implying that \( f(\cdot) \) is not ex-post efficient.

The next lemma summarizes another fact about social choice functions in quasilinear environment.

**Lemma 2.2.** All social choice functions in quasilinear environments are nondictatorial.

**Proof:**
If possible, assume that a social choice function, \( f(\cdot) \), is dictatorial in the quasilinear environment. This means that there exists an agent called the dictator, say \( d \in N \), such that for each \( \theta \in \Theta \), we have
\[
u_d(f(\theta), \theta_d) \geq u_d(x, \theta_d) \forall x \in X.
\]
However, because of the environment being quasilinear, we have \( u_d(f(\theta), \theta_d) = v_d(k(\theta), \theta_d) + t_d(\theta) \). Now consider the following alternative \( x \in X : \)
\[
x = \begin{cases} 
(k(\theta), (t_i = t_i(\theta))_{i \neq d}, t_d = t_d(\theta) - \sum_{i=1}^{n} t_i(\theta)) & : \sum_{i=1}^{n} t_i(\theta) < 0 \\
(k(\theta), (t_i = t_i(\theta))_{i \neq d, j}, t_d = t_d(\theta) + \epsilon, t_j = t_j(\theta) - \epsilon) & : \sum_{i=1}^{n} t_i(\theta) = 0
\end{cases}
\]
where $\epsilon > 0$ is any arbitrary number, and $j$ is any agent other than $d$. It is easy to verify, for the above outcome $x$, that we have $u_d(x, \theta_d) > u_d(f(\theta), \theta_d)$, which contradicts the fact that $d$ is a dictator.

In view of Lemma 2.2, the social planner need not have to worry about the nondictatorial property of the social choice function in quasilinear environments and he can simply look for whether there exists any SCF that is both ex-post efficient and dominant strategy incentive compatible. Furthermore, in the light of Lemma 2.1, we can say that the social planner can look for an SCF that is allocatively efficient, budget balanced, and dominant strategy incentive compatible. Once again the question arises whether there could exist social choice functions which satisfy all these three properties — AE, BB, and DSIC.

### 2.8 Groves Mechanisms

An important possibility result in mechanism design is that in the quasilinear environment, there exist social choice functions that are both allocatively efficient and dominant strategy incentive compatible. These are in general called the VCG (Vickrey–Clarke–Groves) mechanisms.

#### 2.8.1 VCG Mechanisms

The VCG mechanisms are named after their famous inventors William Vickrey, Edward Clarke, and Theodore Groves. It was Vickrey who introduced the famous Vickrey auction (second price sealed bid auction) in 1961 [19]. To this day, the Vickrey auction continues to enjoy a special place in the annals of mechanism design. Clarke [20] and Groves [21] came up with a generalization of the Vickrey mechanisms and helped define a broad class of dominant strategy incentive compatible mechanisms in the quasilinear environment. VCG mechanisms are by far the most extensively used among quasilinear mechanisms. They derive their popularity from their mathematical elegance and the strong properties they satisfy.

#### 2.8.2 The Groves Theorem

The following theorem provides a sufficient condition for an allocatively efficient social function in quasilinear environment to be dominant strategy incentive compatible.

**Theorem 2.4** (Groves Theorem). Let the SCF $f(\cdot) = (k^*(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ be allocatively efficient. Then $f(\cdot)$ is dominant strategy incentive compatible if it satisfies the following payment
structure (popularly known as the Groves payment (incentive) scheme):

$$t_i(\theta) = \left[ \sum_{j \neq i} v_j(k^*(\theta), \theta_j) \right] + h_i(\theta_{-i}) \quad \forall \ i = 1, \ldots, n \quad (2.15)$$

where $h_i : \Theta_{-i} \to \mathbb{R}$ is any arbitrary function that honors the feasibility condition $\sum_i t_i(\theta) \leq 0 \ \forall \ \theta \in \Theta$.

**Proof:**
The proof is by contradiction. Suppose $f(\cdot)$ satisfies both allocative efficiency and the Groves payment structure but is not DSIC. This implies that $f(\cdot)$ does not satisfy the following necessary and sufficient condition for DSIC: $\forall i \in N \ \forall \theta \in \Theta,$

$$u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq u_i(f(\theta'_i, \theta_{-i}), \theta_i) \ \forall \theta'_i \in \Theta_i \ \forall \theta_{-i} \in \Theta_{-i}.$$

This implies that there exists at least one agent $i$ for which the above is false. Let $i$ be one such agent. That is, for agent $i$,

$$u_i(f(\theta'_i, \theta_{-i}), \theta_i) > u_i(f(\theta_i, \theta_{-i}), \theta_i)$$

for some $\theta_i \in \Theta_i$, for some $\theta_{-i} \in \Theta_{-i}$, and for some $\theta'_i \in \Theta_i$. Thus, for agent $i$, there would exist $\theta_i \in \Theta_i, \theta'_i \in \Theta_i, \theta_{-i} \in \Theta_{-i}$ such that

$$v_i(k^*(\theta'_i, \theta_{-i}), \theta_i) + t_i(\theta'_i, \theta_{-i}) + m_i > v_i(k^*(\theta_i, \theta_{-i}), \theta_i) + t_i(\theta_i, \theta_{-i}) + m_i.$$

Recall that

$$t_i(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) + \sum_{j \neq i} (k^*(\theta_i, \theta_{-i}), \theta_j)$$

$$t_i(\theta'_i, \theta_{-i}) = h_i(\theta_{-i}) + \sum_{j \neq i} (k^*(\theta'_i, \theta_{-i}), \theta_j).$$

Substituting these, we get

$$v_i(k^*(\theta'_i, \theta_{-i}), \theta_i) + \sum_{j \neq i} v_i(k^*(\theta'_j, \theta_{-i}), \theta_j) > v_i(k^*(\theta_i, \theta_{-i}), \theta_i) + \sum_{j \neq i} v_i(k^*(\theta_j, \theta_{-i}), \theta_j),$$

which implies

$$\sum_{i=1}^{n} v_i(k^*(\theta'_i, \theta_{-i}), \theta_i) > \sum_{i=1}^{n} v_i(k^*(\theta_i, \theta_{-i}), \theta_i).$$

The above contradicts the fact that $f(\cdot)$ is allocatively efficient. This completes the proof. \[\blacksquare\]
The following are a few interesting implications of the above theorem.

1. Given the announcements \( \theta_{-i} \) of agents \( j \neq i \), the monetary transfer to agent \( i \) depends on his announced type only through effect of the announcement of agent \( i \) on the project choice \( k^*(\theta) \).

2. The change in the monetary transfer of agent \( i \) when his type changes from \( \theta_i \) to \( \hat{\theta}_i \) is equal to the effect that the corresponding change in project choice has on total value of the rest of the agents. That is,

\[
t_i(\theta_i, \theta_{-i}) - t_i(\hat{\theta}_i, \theta_{-i}) = \sum_{j \neq i} \left[ v_j(k^*(\theta_i, \theta_{-i}), \theta_j) - v_j(k^*(\hat{\theta}_i, \theta_{-i}), \theta_j) \right].
\]

Another way of describing this is to say that the change in monetary transfer to agent \( i \) reflects exactly the externality he is imposing on the other agents.

After the famous result of Groves, a direct revelation mechanism in which the implemented SCF is allocatively efficient and satisfies the Groves payment scheme is called a Groves Mechanism.

**Definition 2.19 (Groves Mechanisms).** A direct mechanism, \( \mathcal{D} = ((\Theta_i)_{i \in N}, f(\cdot)) \) in which \( f(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot)) \) satisfies allocative efficiency (2.13) and Groves payment rule (2.15) is known as a Groves mechanism.

In mechanism design parlance, Groves mechanisms are popularly known as Vickrey–Clarke–Groves (VCG) mechanisms because the Clarke mechanism is a special case of Groves mechanism, and the Vickrey mechanism is a special case of Clarke mechanism. We will discuss this relationship later in this chapter.

The Groves theorem provides a sufficiency condition under which an allocatively efficient (AE) SCF will be DSIC. The following theorem due to Green and Laffont [7] provides a set of conditions under which the condition of Groves Theorem also becomes a necessary condition for an AE SCF to be DSIC. In this theorem, we let \( \mathcal{F} \) denote the set of all possible functions \( f : K \to \mathbb{R} \).

**Theorem 2.5 (First Characterization Theorem of Green–Laffont).** Suppose for each agent \( i \in N \) that \( \{v_i(\cdot, \theta_i) : \theta_i \in \Theta_i\} = \mathcal{F} \), that is, every possible valuation function from \( K \) to \( \mathbb{R} \) arises for some \( \theta_i \in \Theta_i \). Then any allocatively efficient social choice function \( f(\cdot) \) will be dominant strategy incentive compatible if and only if it satisfies the Groves payment scheme given by Equation (2.15).

Note that in the above theorem, every possible valuation function from \( K \) to \( \mathbb{R} \) arises for any \( \theta_i \in \Theta_i \). In the following characterization theorem, again due to Green and Laffont [7], \( \mathcal{F} \) is replaced with with \( \mathcal{F}_c \) where \( \mathcal{F}_c \) denotes the set of all possible continuous functions \( f : K \to \mathbb{R} \).
Theorem 2.6 (Second Characterization Theorem of Green–Laffont). Suppose for each agent \(i \in N\) that \(\{v_i(\cdot, \theta_i) : \theta_i \in \Theta_i\} = \mathcal{F}_c\), that is, every possible continuous valuation function from \(K\) to \(\mathbb{R}\) arises for some \(\theta_i \in \Theta_i\). Then any allocatively efficient social choice function \(f(\cdot)\) will be dominant strategy incentive compatible if and only if it satisfies the Groves payment scheme given by Equation (2.15).

### 2.8.3 Groves Mechanisms and Budget Balance

Note that a Groves mechanism always satisfies the properties of AE and DSIC. Therefore, if a Groves mechanism is budget balanced, then it will solve the problem of the social planner because it will then be ex-post efficient and dominant strategy incentive compatible. By looking at the definition of the Groves mechanism, one can conclude that it is the functions \(h_i(\cdot)\) that decide whether or not the Groves mechanism is budget balanced. The natural question that arises now is whether there exists a way of defining functions \(h_i(\cdot)\) such that the Groves mechanism is budget balanced. In what follows, we present one possibility result and one impossibility result in this regard.

#### Possibility and Impossibility Results for Quasilinear Environments

Green and Laffont [7] showed that in a quasilinear environment, if the set of possible types for each agent is sufficiently rich then ex-post efficiency and DSIC cannot be achieved together. The precise statement is given in the form of the following theorem.

Theorem 2.7 (Green–Laffont Impossibility Theorem). Suppose for each agent \(i \in N\) that \(\mathcal{F} = \{v_i(\cdot, \theta_i) : \theta_i \in \Theta_i\}\), that is, every possible valuation function from \(K\) to \(\mathbb{R}\) arises for some \(\theta_i \in \Theta_i\). Then there is no social choice function that is ex-post efficient and DSIC.

Thus, the above theorem says that if the set of possible types for each agent is sufficiently rich then there is no hope of finding a way to define the functions \(h_i(\cdot)\) in Groves payment scheme so that we have \(\sum_{i=1}^n t_i(\theta) = 0\). However, one special case in which a positive result arises is summarized in the form of following possibility result.

Theorem 2.8 (A Possibility Result for Budget Balance of Groves Mechanisms). If there is at least one agent whose preferences are known (that is, the type set is a singleton set) then it is possible to choose the functions \(h_i(\cdot)\) so that \(\sum_{i=1}^n t_i(\theta) = 0\).

**Proof:**

Let agent \(i\) be such that his preferences are known, that is \(\Theta_i = \{\theta_i\}\). In view of this condition, it is easy to see that for an allocatively efficient social choice function \(f(\cdot) = (k^*(\cdot), t_1(\cdot), \ldots, t_n(\cdot))\),
the allocation $k^*(\cdot)$ depends only on the types of the agents other than $i$. That is, the allocation $k^*(\cdot)$ is a mapping from $\Theta_{-i}$ to $K$. Let us define the functions $h_j(\cdot)$ in the following manner:

$$h_j(\theta_{-j}) = \begin{cases} h_j(\theta_{-j}) : j \neq i \\ -\sum_{r \neq i} h_r(\theta_{-r}) - (n-1) \sum_{r=1}^{n} v_r(k^*(\theta), \theta_r) : j = i. \end{cases}$$

It is easy to see that under the above definition of the functions $h_i(\cdot)$, we will have $t_i(\theta) = -\sum_{j \neq i} t_j(\theta)$.

Figure 2.9 captures the main results of this section by showing what the space of social choice functions looks like in the quasilinear environment. The exhibit brings out various possibilities and impossibilities in the quasilinear environment, based on the results that we have discussed so far.

Figure 2.9: Space of DSIC social choice functions in quasilinear environment

2.9 Clarke (Pivotal) Mechanisms

A special case of Groves mechanism was developed independently by Clarke in 1971 [20] and is known as the Clarke, or the pivotal mechanism. It is a special case of Groves mechanisms in the sense of using a natural special form for the function $h_i(\cdot)$. In the Clarke mechanism, the
function $h_i(\cdot)$ is given by the following relation:

$$h_i(\theta_{-i}) = -\sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \quad \forall \theta_{-i} \in \Theta_{-i}, \forall i = 1, \ldots, n$$

(2.16)

where $k^*_{-i}(\theta_{-i}) \in K_{-i}$ is the choice of a project that is allocatively efficient if there were only the $n - 1$ agents $j \neq i$. Formally, $k^*_{-i}(\theta_{-i})$ must satisfy the following condition.

$$\sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \geq \sum_{j \neq i} v_j(k, \theta_j) \quad \forall k \in K_{-i}$$

(2.17)

where the set $K_{-i}$ is the set of project choices available when agent $i$ is absent. Substituting the value of $h_i(\cdot)$ from Equation (2.16) in Equation (2.15), we get the following expression for agent $i$’s transfer in the Clarke mechanism:

$$t_i(\theta) = \left[ \sum_{j \neq i} v_j(k^*(\theta), \theta_j) \right] - \left[ \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \right].$$

(2.18)

The above payment rule has an appealing interpretation: Given a type profile $\theta = (\theta_1, \ldots, \theta_n)$, the monetary transfer to agent $i$ is given by the total value of all agents other than $i$ under an efficient allocation when agent $i$ is present in the system minus the total value of all agents other than $i$ under an efficient allocation when agent $i$ is absent in the system.

2.9.1 Clarke Mechanisms and Weak Budget Balance

Recall from the definition of Groves mechanisms that, for weak budget balance, we should choose the functions $h_i(\theta_{-i})$ in a way that the weak budget balance condition $\sum_{i=1}^n t_i(\theta) \leq 0$ is satisfied. In this sense, the Clarke mechanism is a useful special case because it achieves weak budget balance under fairly general settings. In order to understand these general sufficiency conditions, we define following quantities

$$B^*(\theta) = \left\{ k \in K : k \in \arg \max_{k \in K} \sum_{j=1}^n v_j(k, \theta_j) \right\}$$

$$B^*(\theta_{-i}) = \left\{ k \in K_{-i} : k \in \arg \max_{k \in K_{-i}} \sum_{j \neq i} v_j(k, \theta_j) \right\}$$

where $B^*(\theta)$ is the set of project choices that are allocatively efficient when all the agents are present in the system. Similarly, $B^*(\theta_{-i})$ is the set of project choices that are allocatively efficient if all agents except agent $i$ were present in the system. It is obvious that $k^*(\theta) \in B^*(\theta)$
and \( k_{-i}^*(\theta_{-i}) \in B^*(\theta_{-i}) \).

Using the above quantities, we define the following properties of a direct revelation mechanism in quasilinear environment.

**Definition 2.20 (No Single Agent Effect).** We say that mechanism \( \mathcal{M} \) has no single agent effect if for each agent \( i \), for each \( \theta \in \Theta \), and for each \( k^*(\theta) \in B^*(\theta) \), we have a \( k \in K_{-i} \) such that

\[
\sum_{j \neq i} v_j(k, \theta_j) \geq \sum_{j \neq i} v_j(k^*(\theta), \theta_j).
\]

In view of the above properties, we have the following proposition that gives a sufficiency condition for Clarke mechanism to be weak budget balanced.

**Proposition 2.4.** If the Clarke mechanism has no single agent effect, then the monetary transfer to each agent would be non-positive, that is, \( t_i(\theta_i) \leq 0 \ \forall \ \theta \in \Theta; \ \forall \ i = 1, \ldots, n \). In such a situation, the Clarke mechanism would satisfy the weak budget balance property.

**Proof:**

Note that by virtue of no single agent effect, for each agent \( i \), each \( \theta \in \Theta \), and each \( k^*(\theta) \in B^*(\theta) \), there exists a \( k \in K_{-i} \) such that

\[
\sum_{j \neq i} v_j(k, \theta_j) \geq \sum_{j \neq i} v_j(k^*(\theta), \theta_j).
\]

However, by definition of \( k^*_{-i}(\theta_{-i}) \), given by Equation (2.17), we have

\[
\sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \geq \sum_{j \neq i} v_j(k, \theta_j) \ \forall \ k \in K_{-i}.
\]

Combining the above two facts, we get

\[
\sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \geq \sum_{j \neq i} v_j(k^*(\theta), \theta_j) \\
\Rightarrow 0 \geq t_i(\theta) \\
\Rightarrow 0 \geq \sum_{i=1}^{n} t_i(\theta).
\]

In what follows, we present an interesting corollary of the above proposition.

**Corollary 2.1.**
1. \( t_i(\theta) = 0 \) iff \( k^*(\theta) \in B^*(\theta_{-i}) \). That is, agent \( i \)'s monetary transfer is zero iff his announcement does not change the project decision relative to what would be allocatively efficient for agents \( j \neq i \) in isolation.

2. \( t_i(\theta) < 0 \) iff \( k^*(\theta) \not\in B^*(\theta_{-i}) \). That is, agent \( i \)'s monetary transfer is negative iff his announcement changes the project decision relative to what would be allocatively efficient for agents \( j \neq i \) in isolation. In such a situation, the agent \( i \) is known to be “pivotal” to the efficient project choice, and he pays a tax equal to his effect on the other agents.

2.9.2 Clarke Mechanisms and Individual Rationality

We have studied individual rationality (also called voluntary participation) property in Section 2.5.3. The following proposition investigates the individual rationality of the Clarke mechanism. First, we provide two definitions.

**Definition 2.21** (Choice Set Monotonicity). We say that a mechanism \( \mathcal{M} \) is choice set monotone if the set of feasible outcomes \( X \) (weakly) increases as additional agents are introduced into the system. An implication of this property is \( K_{-i} \subset K \ \forall \ i = 1, \ldots, n \).

**Definition 2.22** (No Negative Externality). Consider a choice set monotone mechanism \( \mathcal{M} \). We say that the mechanism \( \mathcal{M} \) has no negative externality if for each agent \( i \), each \( \theta \in \Theta \), and each \( k^*_{-i}(\theta_{-i}) \in B^*(\theta_{-i}) \), we have

\[
v_i(k^*_{-i}(\theta_{-i}), \theta_i) \geq 0.
\]

We now state and prove a proposition which provides a sufficient condition for the ex-post individual rationality of the Clarke mechanism. Recall from Section 2.5.3 the notation \( \overline{w}_i(\theta_i) \), which represents the utility that agent \( i \) receives by withdrawing from the mechanism.

**Proposition 2.5** (Ex-Post Individual Rationality of Clarke Mechanism ). Let us consider a Clarke mechanism in which

1. \( \overline{w}_i(\theta_i) = 0 \ \forall \theta_i \in \Theta_i; \ \forall \ i = 1, \ldots, n \),

2. The mechanism satisfies choice set monotonicity property,

3. The mechanism satisfies no negative externality property.

Then the Clarke mechanism is ex-post individual rational.

**Proof:**
Recall that utility $u_i(f(\theta), \theta_i)$ of an agent $i$ in Clarke mechanism is given by

$$u_i(f(\theta), \theta_i) = v_i(k^*(\theta), \theta_i) + \left[ \sum_{j \neq i} v_j(k^*(\theta), \theta_j) \right] - \left[ \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \right]$$

$$= \left[ \sum_{j} v_j(k^*(\theta), \theta_j) \right] - \left[ \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \right].$$

By virtue of choice set monotonicity, we know that $k^*_{-i}(\theta_{-i}) \in K$. Therefore, we have

$$u_i(f(\theta), \theta_i) \geq \left[ \sum_{j} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \right] - \left[ \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j) \right]$$

$$= v_i(k^*_{-i}(\theta_{-i}), \theta_i)$$

$$\geq 0 = \pi_i(\theta_i).$$

The last step follows due to the fact that the mechanism has no negative externality.

Figure 2.10: Mechanism design space in quasilinear environment
2.10 Mechanism Design Space in Quasilinear Environment

We have seen G–S Theorem (Theorem 2.3) in Section 2.6. As shown in Figure 2.8, we have seen two approaches to escape the theorem. First we restrict the space of preferences by introducing money into system and restricting our attention to quasi-linear settings. We also, seen weaker solution concept of Bayesian implementation. We described many possibilities and impossibilities in quasi-linear settings. Figure 2.10 shows the space of mechanisms taking into account all the results we have studied so far. A careful look at the diagram suggests why designing a mechanism that satisfies a certain combination of properties is quite intricate.

In next section we talk DSIC mechanisms without involving money and still getting around G–S Theorem 2.3.

2.11 DSIC Mechanisms without Money

Recall G–S Theorem (Theorem 2.3). When there are at least three outcomes, and for each agent, the set of allowable preferences is the set of all strict preferences on the outcome set, then any strategyproof mechanism is necessarily a dictatorial one. However, this theorem is often compared with Procrustean bed. As shown in Figure 2.8, we can always restrict the set of allowable preferences to get around this impossibility theorem. We can restrict the set of preferences by introducing money into the system and using quasi-linear environments as seen in Section 2.7. This is suitable for a large number of economic institutions. However, it should be noted that, as explained in Problems 4, 5, and 6 in Chapter 1, in some settings, it is undesirable or infeasible to use money; e.g., because of community norms, legal constraints, or inconvenience. How do we obtain strategyproof mechanism in such cases? If is possible to achieve by restricting the set of allowable preferences. In particular, we consider the following markets without monetary transfers.

Market 1: There are job openings in companies and final year students are willing to opt for these positions. There is a need to match the students with these positions. Each student is indifferent among the matches wherein he/she gets the same position. In this problem the companies may have preferences over the students. Such a market is called a two-sided market. This problem is generally abstracted as a marriage problem where men represent one side of the market and women another. The goal of the social planner is to aggregate the preferences of men over women and that of women over men and produce a stable matching.

\[2\text{We define stable matching formally in Section 2.12.}\]
Market 2: There are \( n \) objects to be assigned among \( n \) competing agents. The agents have preferences over such assignments. How do we assign these objects among the competing agents? Such markets are called as one-sided markets as only the agents have preferences and not the objects.

Market 3: There are \( n \) objects owned by \( n \) agents. The agents have preferences over these objects and each agent would like to get an object at least as good as the object he/she is currently owning. This is also a one-sided market. These markets are referred to as housing markets.

In the above markets, the set of outcomes (\( X \)) is the set of feasible object matchings or allocations. Let \( x(i) \) denote the match/object that agent \( i \) receives. The agents are indifferent in the matchings or allocations in which they receive same match/object. That is, for each agent \( i \), \( x, y \in X \) are indifferent if \( x(i) = y(i) \). Thus, the set of allowable preferences is a strict subset of all possible preferences over \( X \). Hence, the G–S Theorem (Theorem 2.3) does not apply here. The preferences of an agent are captured by a preference ordering of the match or objects. For agent \( i \) we denote its ordinal preference over the object/match it receives by \( \succ_i \). If agent \( i \) receives higher utility by receiving object \( j \) than object \( k \), we say \( j \succ_i k \). and all assignments in which it receives \( j \) are preferable to all assignments in which it receives \( k \). Also, we use ordinal utilities in this section rather than cardinal utilities. That is, we use,

\[
 u_i(x, \succ_i) > u_i(y, \succ_i) \iff x(i) \succ_i y(i), \forall x, y \in X 
\]

The preference ordering \( \succ_i \) is private information to agent \( i \). That is, \( \theta_i = \succ_i \). Matching procedure or object allocation procedure takes reported \( \succ \) as input and produces a matching or an allocation. So, mechanism design comes into picture. Instead of \( \theta_i \) and \( u_i(\cdot) \), we use only \( \succ_i \)s for describing mechanisms and the strategy-proofness properties. We explain the relevant state-of-the-art for each one of the three markets in the following subsections. For more on these problems, interested readers are referred to a survey by Sönmez and Ünver [22].

In Section 2.12, we describe the results related to the matching problem. In Section 2.13, we discuss heterogeneous object assignment. We describe the results on house allocation markets in Section 2.14. We briefly summarize the results of the three problems in Section 2.15.

### 2.12 Two-Sided Matchings

Consider the college admission problem, where colleges seek students and they have preferences over the students. The students need an admission in a college and they have preferences over
the colleges. So there is need of a procedure to match the students with the colleges, taking into account preferences of both the sides. In 1962, Gale and Shapley [3], abstracted such two-sided matching problems as a marriage problem in which one side is considered as men and the other as women. The men may represent students aspiring for college admissions and women as colleges. It may be interns-hospitals, job aspirants-employers, etc. In all these, there is a need to match one side of the market with the other.

2.12.1 Some Important Definitions

For simplicity, assume that there are equal numbers of men and women and nobody prefers to remain single. Everybody needs to be matched with an agent from the other side of the market. The agents have strict preferences over the other side of the market and these preferences are private to the agents. Table 2.2 summarizes the notation used in this section.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>${1, 2, \ldots, n}$, Set of men</td>
</tr>
<tr>
<td>$W$</td>
<td>${1, 2, \ldots, n}$, Set of women</td>
</tr>
<tr>
<td>$m$</td>
<td>A man $\in M$</td>
</tr>
<tr>
<td>$w$</td>
<td>A woman $\in W$</td>
</tr>
<tr>
<td>$\succ_i$</td>
<td>The strict preference of agent $i$ over the set of opposite sex agents</td>
</tr>
<tr>
<td>$\succ_{-i}$</td>
<td>Preferences of all the agents except agent $i$</td>
</tr>
<tr>
<td>$\succ$</td>
<td>$(\succ_i, \succ_{-i})$, Preference profile of all the agents</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Matching</td>
</tr>
<tr>
<td>$X$</td>
<td>Set of all matchings</td>
</tr>
<tr>
<td>DA</td>
<td>Male-proposal Deferred Acceptance</td>
</tr>
</tbody>
</table>

Table 2.2: Notation: matching

**Definition 2.23** (Matching). *Matching is a function, $\mu : M \cup W \rightarrow M \cup W$ such that, $\mu(m) \in W \forall m \in M$, $\mu(w) \in M \forall w \in W$ and $\mu(\mu(i)) = i \forall i \in M \cup W$.*

**Definition 2.24** (Blocking pair). *A pair $(m, w)$ is said to be a blocking pair for a matching $\mu$, if they prefer to match with each other, than their match in $\mu$. That is, $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$.*

**Definition 2.25** (Stable Matching). *A matching $\mu$ is said to be stable if there is no $(m, w)$ pair that blocks $\mu$.*

A stable matching is preferable because, if matching is not stable, the blocking pair would always prefer not to follow the matching suggested by $\mu$ and match with each other. Now the questions are, given preferences of the agents, does there exist a stable matching? If yes,
how to select a stable matching? In 1962, Gale and Shapley proposed an algorithm, namely Deferred Acceptance (DA). It has two variants depending upon who is proposing, namely male-proposal deferred acceptance and female-proposal deferred acceptance. In this chapter we talk about male-proposal deferred acceptance which we simply refer to as DA in short. The same algorithm and similar results hold true if the roles of male and females are interchanged.

Definition 2.26 (Male-proposal Deferred Acceptance, (DA)). In the first round, each man proposes to his most preferred woman. Each woman keeps the best match and rejects other men. All rejected men then propose in the next round to their next preferred woman. The procedure continues till there are no more rejections.

We explain DA with an example. Consider a matching problem with four men and four women. Let their preferences be as in Table 2.3. For the above preferences, the DA algorithm is executed as follows:

- In the first iteration, $m_1, m_2$ propose to $w_2$ and $m_3, m_4$ propose to $w_1$. $w_2$ accepts proposal from $m_2$ and rejects $m_1$. Similarly $w_1$ accepts $m_4$. Thus at the end of the first iteration, the matches are $m_2 - w_2$ and $m_4 - w_1$.

- In the next iteration, $m_1$ proposes to $w_1$ and $m_3$ to $w_3$. $w_1$ accepts $m_1$ by rejecting $m_4$. Thus at the end of the second iteration, the matches are $m_1 - w_1, m_2 - w_2$, and $m_3 - w_3$.

- In the third iteration, $m_4$ proposes to $w_2$ who in turn declines $m_2$ and accepts $m_4$. At the end of this iteration, the matches are $m_1 - w_1, m_3 - w_3, m_4 - w_2$.

- In the fourth iteration, $m_2$ and $w_3$ are matched.

- In the last iteration, $m_3$ and $w_4$ are matched. Thus the final match is $(m_1 - w_1, m_2 - w_3, m_3 - w_4, m_4 - w_2)$.

Note that each man has a finite number of women to propose to. Thus DA terminates in finite number of steps. DA always produces a stable matching as shown in the following theorem.

Theorem 2.9. DA produces a stable matching.
Proof:
Suppose there exists a blocking pair \((m, w)\) in a matching \(\mu\) produced by DA. That is, \(w \succ_m \mu(m)\) and \(m \succ_w \mu(w)\). \(w \succ_m \mu(m)\) means \(m\) proposed to \(w\) before \(\mu(m)\) in DA and he got rejected by \(w\). \(w\) rejected \(m\) means she received a better match than \(m\) or was already matched with a better match. That is, \(\mu(w) \succ_w m\). This contradicts \(m \succ_w \mu(w)\).

The above theorem also shows that there exists at least one stable matching for all type profiles. There may be multiple stable matchings at each type profile. So which one to select? Agents may prefer one stable matching over another. The following definition formalizes the notion of an optimal matching.

**Definition 2.27 (Male Optimal Stable Matching).** A matching \(\mu\) is said to be a Male Optimal stable matching if there does not exist any other matching \(\nu\) which is at least as preferable as \(\mu\) for all men. That is, \(\nu(m) \succ_m \mu(m)\) \(\forall m \in M\) and \(\exists\) at least one \(m'\) such that, \(\nu(m') \succ_m \mu(m')\).

Similarly, one can define female optimal stable matching. It can be shown that DA always chooses male optimal stable matching and the female proposing version selects female optimal stable matching. We state this result without proof. For the proof, refer to [23].

**Theorem 2.10.** DA always selects male optimal stable matching.

Let us now turn to the question of strategyproofness. First, let us define what is exactly strategyproofness in the context of matching. If \(f\) is a matching procedure which takes preference profile, \(\succ\), as input and outputs a matching, we define strategyproofness for \(f\) as,

**Definition 2.28.** Let \(\mu = f(\succ_i, \succ_{-i})\) and \(\mu' = f(\succ'_i, \succ_{-i})\). We say \(f\) is strategyproof if for each agent \(i \in M \cup W\), \(\forall \succ_i, \) and \(\forall \succ'_i\)

\[
\mu(i) \succ_i \mu'(i) \text{ or } \mu(i) = \mu'(i)
\]

That is, it is a best response for each agent to report his/her preferences truthfully to the matching procedure.

Is DA strategyproof (DSIC) when considered as a matching mechanism? The answer is no. Consider matching problem with three men and three women with preferences as in Table 2.4. At this preference profile, DA matches agents as, \(m_1 - w_3, m_2 - w_1, m_3 - w_2\). However, \(w_1\) can report her preference as \(m_3 \succ_{w_1} m_1 \succ_{w_1} m_2\), in which case she is matched with \(m_3\), her best possible match. Thus, \(w_1\) is able to manipulate DA and hence is not strategyproof.

Consider the following mechanism, namely man-proposal serial dictatorship mechanism. Assign priorities to the men which are independent of their preferences. The man with the highest priority is matched with the best possible match for him. The man with the second highest
priority is matched with the best possible match among unmatched women and the procedure continues. The man-proposal serial dictatorship is strategyproof. However, this procedure does not yield stable matching. Is there any mechanism for matching which always yields stable matching and is strategyproof? The answer is negative. We state an important result by Roth [24] without proof. For the proof, the readers are referred to [24].

Theorem 2.11. There does not exist a matching procedure which always produces stable matching and is strategyproof for all the agents.

Though DA is not strategyproof, Roth [23] shows that it is strategyproof for men. If we can assume that the women are reporting their preferences truthfully, then we can use DA which produces stable matching and is strategyproof for men. However, it selects male optimal stable matching which is a worst stable matching for the females. Also in real life, agents may prefer to remain single rather than matching with some of the agents. Agents’ preferences need not be strict. There is a rich body of literature available on matching. For further details on this topic, interested readers are referred to [25, 22].

In marriage problems, there are two sides of the market and both sides have preferences. There are markets in which there is agents on one side have preferences. Such markets are called one-sided markets. The next two subsections discuss one-sided markets.

### 2.13 Allocation Mechanisms

Consider the allocation of indivisible objects among competing agents. The objects are not owned by anybody and every agent needs only one of them. Each agent has preferences over the objects. We can assume that each agent is indifferent among allocations in which it receives the same object. Thus the G–S Theorem (Theorem 2.3) does not apply here. Svenssons [26] characterized the strategyproof mechanisms in this context under fairly general settings. Before getting into this result, we need some notation and definitions.
\[ N \{1, 2, \ldots, n\}, \text{Set of agents} \]
\[ M \{1, 2, \ldots, m\} \text{ Set of indivisible, distinct objects} \]
\[ \succ_i \text{ Strict preference of agent } i \text{ over } M \]
\[ \succ_{-i} \text{ Preferences of all the agents except } i \]
\[ \succ (\succ_i, \succ_{-i}) \text{ Preference profile of all the agents} \]
\[ \mathcal{U} \text{ Set of all possible preference profiles, } \succ \in \mathcal{U} \]
\[ x : N \to M \text{, an allocation of the goods} \]
\[ x(i) \text{ denotes an object that agent } i \text{ receives in allocation } x \]
\[ X \text{ Set of feasible allocations of goods in which} \]
\[ \text{each agent receives at most one object and no object is allocated to two distinct agents} \]
\[ f : \mathcal{U} \to X, \text{ Allocation mechanism} \]

**Table 2.5: Notation: allocation mechanisms**

2.13.1 Notation and Definitions

**Definition 2.29.** We say an allocation mechanism \( f \) is strategyproof if \( \forall i \in N, \forall \succ_{-i} \text{ and } \forall \succ_i' \)

\[
f(\succ_i, \succ_{-i}) \succ_i f(\succ_i', \succ_{-i}) \text{ or } f(\succ_i, \succ_{-i}) = f(\succ_i', \succ_{-i})
\]

**Definition 2.30 (Non Bossy).** Let \( x = f(\succ_i, \succ_{-i}) \text{ and } x' = f(\succ_i', \succ_{-i}) \). We say an allocation mechanism \( f \) is non-bossy if for each agent \( i \), if \( x(i) = x'(i) \), then \( x = x' \). That is, if \( i \) receives the same object at preferences \( (\succ_i, \succ_{-i}) \) and \( (\succ_i', \succ_{-i}) \), then everybody receives the same object at these two preferences.

Non-bossy property means nobody can change outcome for others by reporting different preference without affecting its own outcome. Another desirable property of a mechanism is that the labels of the objects should not matter. This is called as neutrality. More formally,

**Definition 2.31 (Neutral).** An allocation mechanism \( f \) is said to be neutral if for each agent \( i \), and each permutation \( \pi \) of objects, \( \pi^{-1}f(\pi \succ)(i) = f(\succ)(i) \), where \( \pi \succ \) is a preference profile in which each agent \( i \) changes its preference from \( \succ_i \) according to \( \pi \).

According to the G–S Theorem (Theorem 2.3), under reasonable assumptions, any strategyproof mechanism is dictatorial. In the case of allocation mechanisms, a notion similar to that of dictatorship exists, namely serial dictatorship.

**Definition 2.32 (Serial Dictatorship).** A serial dictatorship has a priority ranking of the agents, \( h : N \to N \), such that agents are ordered \( h(1), h(2), \ldots, h(n) \). \( h(1) \) is assigned the most preferred items. \( h(2) \) is assigned its most preferred item still available and so on.
Now we state an important result by Svensson’s [26].

**Theorem 2.12.** Any non-bossy and neutral mechanism is strategyproof if and only if it is a serial dictatorship.

Thus, the space of allocation mechanisms is severely restricted. However, Papai [27] relaxes the requirement of neutrality and shows that a much richer class of mechanisms is strategyproof.

In this section, we assumed that the objects are not owned by agents. What if each agent has ownership for one object? Then the problem of reallocation of the objects is called the house allocation problem. We study house allocation problem in the next Section.

### 2.14 House Allocation

Consider a scenario in which university dorms have been assigned to the students and the students wish to reassign dorms among themselves. Or there are agents with property rights over objects and they wish to trade these objects among themselves. However, it may be the case that monetary transfers are not allowed or not legal. For example, consider a kidney exchange. There are patient-donor pairs in which the donor’s kidney is incompatible with the patient. However, it may be the case that the donor’s kidney is compatible with a patient in another such pair, and the donor’s kidney in the second pair is compatible with the patient in the first pair. In this case, they can exchange kidneys. However, it is not legal to charge money for kidney donations or exchanges. These problems can be abstracted as house allocation. In the house allocation problem, each of a set of self-interested agents owns a distinct object (a house) and has strict preferences on houses [3]. The problem is to find a reallocation of objects amongst agents that is robust against misreports of preferences by agents while identifying beneficial trades and without using money. Such markets are commonly referred as housing markets or Shapley-Scarf economy. In 1974, Shapley and Scarf proposed an algorithm, namely Top Trading Cycle Algorithm (TTCA), which is strategyproof and selects reallocation from the core. The core is a set of reallocations which no subset of agents can improve upon by trading amongst themselves.

#### 2.14.1 Notation and Important Definitions

**Definition 2.33.** An assignment of houses, \( x \in X \), Pareto dominates \( y \in X \), if \( x \succeq_i y \) for all \( A_i \) and \( x \succ_j y \) for some \( A_j \). An allocation \( y \in X \) is Pareto efficient if there is no allocation \( x \in X \) that Pareto dominates \( y \).

**Definition 2.34 (Pareto efficient).** A mechanism \( f \) is Pareto efficient if allocation \( x = f(\succ) \) is Pareto efficient for all preference profiles \( \succ \).
\[ H = \{h_1, h_2, \ldots, h_n\}, \text{Set of distinct houses} \]
\[ N = \{A_1, A_2, \ldots, A_n\}, \text{Set of agents} \]
Agent \( A_i \) enters the market with house \( h_i \)
\( \succ_i \) Strict preference of agent \( A_i \) on set of house
\( \succ_{-i} \) Preference profile of agents except \( A_i \)
\( \succ (\succ_1, \succ_2, \ldots, \succ_n) = (\succ_i, \succ_{-i}) \), Preference profile
\( U \) Set of all possible strict preference profiles, \( \succ \in U \)
\( x \) \( x : N \to H \), A house allocation
\( x(i) \) House allocated to \( A_i \)
\( X \) Set of all possible house allocations in which
each agent receives exactly one house and no house is allocated
to two distinct agents
\( f \) \( f : U \to X \), House allocation mechanism

<table>
<thead>
<tr>
<th>Table 2.6: Notation: house allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>H</strong></td>
</tr>
<tr>
<td><strong>N</strong></td>
</tr>
<tr>
<td><strong>( \succ_i )</strong></td>
</tr>
<tr>
<td><strong>( \succ_{-i} )</strong></td>
</tr>
<tr>
<td><strong>( \succ (\succ_1, \succ_2, \ldots, \succ_n) = (\succ_i, \succ_{-i}) )</strong></td>
</tr>
<tr>
<td><strong>U</strong></td>
</tr>
<tr>
<td><strong>x</strong></td>
</tr>
<tr>
<td><strong>x(i)</strong></td>
</tr>
</tbody>
</table>
| **X** | Set of all possible house allocations in which
each agent receives exactly one house and no house is allocated
to two distinct agents |
| **f** | \( f : U \to X \), House allocation mechanism |

**Definition 2.35** (Strategyproof (SP)). Let \( x = f(\succ_i, \succ_{-i}) \) and \( x' = f(\succ'_i, \succ_{-i}) \). Mechanism \( f \) is strategyproof if \( x(i) \succ_i x'(i) \), for all \( A_i \), and all \( \succ_{-i} \in U_{-i} \).

**Definition 2.36** (Individually Rational (IR)). A mechanism \( f \) is individually rational if \( x_i \succ_i h_i \), where \( x = f(\succ) \), for all \( \succ \in U \).

An allocation \( x \) is blocked by a coalition of agents \( S \subseteq N \), if there is a feasible allocation of the houses initially owned by agents in \( S \) amongst themselves that Pareto dominates, for agents in \( S \), the allocation \( x \).

**Definition 2.37** (Core). A mechanism \( f \) is core-selecting if allocation \( x = f(\succ) \) is not blocked by any coalition of agents, for any preference profile \( \succ \).

The core implies Pareto efficiency (by considering coalitions of size \( n \)) and IR (by considering coalitions of size one).

Membership to the core is a desirable property. However natural questions to ask are, does there always exists a reallocation of houses which is in the core? If yes, does there exist a core-selecting mechanism? The answers to these questions are in the affirmative.

Shapley-Scarf [4] proposed an algorithm for house allocation, the top trading cycle algorithm, and showed that it always selects a reallocation from core. In 1977, Roth and Posslethwaite [28] showed that the core of house allocation problem is a singleton if preferences are strict.

**Definition 2.38** (Top Trading Cycle Algorithm). Construct a directed graph as follows.

- One node for each agent.
• Each agent points to owner of its most preferred house among the available houses. That is, there is directed edge from node \( i \) to node \( j \) if \( h_j \) is most preferred house for \( A_i \) among the available houses.

• In this digraph, there will be at least one cycle (including self-loops). All the agents on cycle trade their houses. That is, each agent on a cycle receives the house of the agent it is pointing to.

• These agents are removed from the system.

• Recurse the procedure till no agent is left.

We explain this algorithm with the following example.

**Example 2.12.** Consider a housing market with 5 agents and their preferences as shown in Figure 2.11. So there are 5 nodes, \( A_1, A_2, A_3, A_4, \) and \( A_5 \). In the first step, all the agents point to their agents owning their most preferred house. Thus the edges would be: \( A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_2, \ A_4 \rightarrow A_5 \rightarrow A_1 \). In this digraph \( A_2 \rightarrow A_3 \) forms a cycle and hence they trade their houses (swap their ownerships). At the beginning of step 2, \( A_2 \) and \( A_3 \) are removed from the system. Now, the digraph contains only three nodes, \( A_1, A_4, A_5 \) and edges: \( A_1 \rightarrow A_5 \rightarrow A_1 \) and \( A_4 \rightarrow A_5 \). \( A_1 \) gets \( A_5 \)'s house, and \( A_5 \) gets \( A_1 \)'s. In the last step of TTCA, there is only one node \( A_4 \) with self-loop. Thus the final allocation by TTCA is \( (A_1 - h_5, A_2 - h_3, A_3 - h_2, A_4 - h_4, A_5 - h_1) \).

Figures 2.12 and 2.13 show the execution of TTCA.

Roth [29] studied incentive issues in the house allocation problem and proved the following result.

**Theorem 2.13** (Roth [29]). TTCA is strategyproof or it is a dominant strategy for each agent to reveal its preference truthfully to TTCA.

**Proof:**

Suppose TTCA is not strategyproof. That is, there exists a type profile, say \( \succ \succ (\succ_i, \succ_{-i}) \) and an agent \( A_i \) who can manipulate TTCA at \( \succ \). \( A_i \) benefits by lying its preference as, say \( \succ'_i \) instead of \( \succ_i \), when other agents report \( \succ_{-i} \).

- Let \( x = TTCA(\succ_i, \succ_{-i}) \) and \( y = TTCA(\succ'_i, \succ_{-i}) \)

- \( A_i \) can manipulate TTCA means,

\[
y(i) \succ_i x(i)
\]
Figure 2.11: House allocation problem and TTCA

Figure 2.12: Execution of TTCA: step 2.

- Let $N_k$ be the set of agents who receive the house in the $k^{th}$ iteration of execution of TTCA at preference profile $\succ$ and $A_i \in N_j$.

- At preference report $(\succ'_i, \succ_{-i})$, all the agents in $N_k$, $k < j$ receive the same house as the
preference profile $\succ$ and hence they do not trade with $A_i$. When $A_i$ reports truthfully, TTCA assures $A_i$ the best house amongst the houses owned by agents in $N \cup \bigcup_{k=1}^{j-1} N_k$.

- When $A_i$ reports $\succ_i'$, it receives some house amongst the houses owned by agents in $N \setminus \bigcup_{k=1}^{j-1} N_k$.

- Thus, $x(i) \succ_i y(i)$.

- This contradicts Equation (2.14.1).

- Hence, TTCA is strategyproof.

Thus, TTCA as a mechanism is strategyproof and selects reallocation from the core. What are other good mechanisms? The following theorem due to Ma [30] states TTCA is the unique such mechanism.

**Theorem 2.14.** Any mechanism for house allocation that satisfies strategyproofness, individual rationality, and Pareto efficiency under strict preferences, has to be necessarily TTCA.

For a proof of the above theorem the readers are referred to Ma [30]. We present some of the extensions to house allocation in the next subsection.
2.14.2 Extensions to the House Allocation Problem

In the classic house allocation problem, the agents enter the market with their houses, that is, the type of objects are similar. It may be the case that agents enter market with two different kinds of objects, say like house and car. Konishi (2001) [31] considered a multi-object version of the house allocation problem. Each agent enters into the market with \( k \)-types of objects and they have preferences over the \( k \)-tuples of these objects. He showed that, when \( k > 1 \) and \( n \geq 4 \), the core may be empty as well as it may happen that the strict core is multi-valued.

Abdulkadiroglu and Sonmez [32] consider the situation of reassignment of the school dorms wherein some students are leaving and the same number of newly admitted students are entering into this market. The students who are leaving are releasing their property rights and the students who are entering do not have any property rights. Serial dictatorship faces the problem of individual rationality for existing students. The authors combine ideas from TTCA [4] and serial dictatorship [26], by proposing a mechanism called as AS-TTCA in which, each agent is assigned a priority to demand a house. If a house having no tenant is requested, it is allocated. If the house of an existing tenant is requested, he/she is put at the top of priority queue. If cycles are formed, the assignments are done as in TTCA. They showed that this is strategyproof, individually rational, and Pareto efficient.

Kurino’s (2009) [33] paper on house allocation using a dynamic mechanism considers a model in which there are houses to be assigned. Each agent lives for a fixed duration after which it leaves. So in each period, there is a set of agents leaving and a set of equal number of new entrants. In the model in this paper, the agents’ preferences can be temporal, that is, preferences over houses for an agent may change across periods. For example, in the first period an agent may prefer house \( h_1 \) over \( h_2 \) and in next period \( h_2 \) over \( h_1 \). They consider two types of mechanisms, called as spot mechanism and futures mechanism. In spot mechanisms, each agent is supposed to reveal only current period preferences. In futures mechanism, agents are supposed to report preferences in all periods. They have showed that there is no mechanism which is Pareto efficient and strongly individually rational. Strongly individually rational (SIR) means it is individually rational in each period. They also proposed various mechanism based on serial dictatorship and TTCA which satisfy some properties like strategyproofness, Pareto efficiency, or strongly individual rationality but not all. For example, serial dictatorship-spot mechanism is Pareto efficient but not SIR. TTCA-spot mechanism based on AS-TTCA is SIR but in general not Pareto efficient.

2.15 A Summary of Mechanisms without Money

We summarize the results seen in Sections 2.11-2.13 in Table 2.7.
<table>
<thead>
<tr>
<th>Result/Algorithm</th>
<th>One-Sided/Two-Sided</th>
<th>IC</th>
<th>IR</th>
<th>Initial Endowments</th>
<th>Dynamic</th>
<th>Efficiency</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gale-Shapley</td>
<td>Two</td>
<td>IC for one side of the market</td>
<td>√</td>
<td>No</td>
<td>-</td>
<td>Core</td>
<td>Stable and Optimal for the one side</td>
</tr>
<tr>
<td>Deferred Acceptance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shapley-Scarf</td>
<td>One</td>
<td>✓</td>
<td>✓</td>
<td>Yes</td>
<td>-</td>
<td>Core</td>
<td>Authors attribute algorithm to Gale-Shapley</td>
</tr>
<tr>
<td>TTCA [3]</td>
<td>One</td>
<td>✓</td>
<td>✓</td>
<td>Yes</td>
<td>-</td>
<td>Core</td>
<td>TTCA is the unique mechanism</td>
</tr>
<tr>
<td>Ma [30]</td>
<td>One</td>
<td>✓</td>
<td>✓</td>
<td>Yes</td>
<td>-</td>
<td>Core</td>
<td></td>
</tr>
<tr>
<td>Svenssons [26]</td>
<td>One</td>
<td>✓</td>
<td>✓</td>
<td>No</td>
<td>-</td>
<td>-</td>
<td>This paper also presents simpler proof of Ma’s [30] result</td>
</tr>
<tr>
<td>Serial Dictatorship</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Konishi [31]</td>
<td>One</td>
<td>✓</td>
<td>✓</td>
<td>Yes</td>
<td>-</td>
<td>Core may be empty or multi-valued</td>
<td>$k$ types of objects ($k &gt; 1, n \geq 4$)</td>
</tr>
<tr>
<td>Abdulkadiroglu and Sonmez [32]</td>
<td>One</td>
<td>✓</td>
<td>✓</td>
<td>Yes*</td>
<td>Single Step**</td>
<td>Pareto Efficient</td>
<td></td>
</tr>
<tr>
<td>Kurino [33]</td>
<td>One</td>
<td>✓</td>
<td>No</td>
<td>Yes*</td>
<td>Fixed Steps**</td>
<td></td>
<td>No mechanism is SIR and Pareto efficient though one of them is possible</td>
</tr>
</tbody>
</table>

Table 2.7: Summary: mechanisms without money

*Some agents with initial endowments and some without
**Dynamic nature is induced because some agents leave the system
Part 1

Mechanisms with Money and Static Agents

In Part 1 of the thesis, we address the allocation of heterogeneous objects among competing agents which are intelligent, rational, and willing to pay for receiving the preferable objects. Also, all the agents are present while decisions are being made. That is, we address mechanism design problems with static agents and wherein monetary transfers are possible. In particular, we address design of an optimal combinatorial auction (Chapter 3), design of truthful Multi-Armed-Bandit mechanisms (Chapter 4), and redistribution mechanisms for allocation of heterogeneous objects (Chapter 5).
Chapter 3

Optimal Multi-Unit Combinatorial Auctions

The current art in optimal combinatorial auctions is limited to handling the case of a single unit each of multiple items, with each agent bidding on exactly one bundle (single minded bidders). This chapter extends the current art by proposing an optimal auction for buying multiple units of multiple items when the bidders are single minded. We develop a procurement auction that minimizes the cost of procurement while satisfying Bayesian incentive compatibility and interim individual rationality. Under appropriate regularity conditions, this optimal auction also satisfies dominant strategy incentive compatibility. The results presented here hold true for equivalent forward auction settings as well. For the two item, multi-unit procurement auction, we design an optimal auction in the presence of the single minded bidders offering volume discounts. When bidders submit XOR bids on two disjoint bundles, we show how the current state of the art can be used for developing optimal auctions. The work reported in this chapter is available in the form of papers \cite{34,35}.

3.1 Introduction

3.1.1 Motivation and Background

Auction based mechanisms are extremely relevant in modern day electronic procurement systems \cite{36,37}, since they enable a promising way of automating negotiations with suppliers and achieving the ideal goals of procurement efficiency and cost minimization. To procure a bundle of items, different approaches \cite{38} as shown below are possible: sequential auctions, parallel auctions, and combinatorial auctions. In many cases, it may be beneficial to allow the suppliers to bid on combinations of items rather than on a single item. Such auctions are called combinatorial auctions. A combinatorial auction is a mechanism where bidders can submit bids on combinations of items. Combinatorial auctions have emerged in recent times as an important mechanism, extremely useful in numerous e-business applications such as e-selling,
e-procurement, e-logistics, supply chain formation, and B2B exchanges.

Our work is motivated by practical situations occurring in the real world. For example, a company may wish to purchase different items, say $A$, $B$ and $C$ in large volumes. There are various vendors interested in supplying some of these items. Now, the company can set up an auction as shown in Figure 3.1. It will be required to solve the winner determination problem, that is, to select a winning set of bids such that each item to be bought is included in at least one of the selected bids, and the total cost of procurement is minimized. This type of auction is referred to as a reverse auction. In a reverse auction, a buyer sets up an auction and multiple sellers/suppliers participate.

If a seller sets up an auction and multiple buyers participate in the auction, then we have a forward auction. For example, consider a seller owning four machines which can perform various jobs. The jobs that could be assigned to different machines may be different.

The seller wishes to earn revenue by offering the services of these machines for multiple units of time slots. There are buyers who wish to avail the services of various machines for multiple units of time slots. The goal of the seller would be to maximize revenue by setting up an auction for the services offered by the machines.

In this chapter, our interest is in multi-unit combinatorial procurement auctions, where a buyer is interested in procuring multiple units of multiple items. The results presented here are applicable to analogous forward auction settings.

The costs of the items or bundles that the agents are going to supply is a private information with the agents. Being rational, they may misreport it for making more profit. Thus, mechanism
design theory plays an important role in the design of such auctions. In mechanism design literature, an optimal auction refers to an auction which optimizes a performance metric (for example maximize revenue to a seller or minimize cost to a buyer) subject to two essential properties:

1. Incentive compatibility
2. Individual rationality.

Incentive compatibility comes in two forms: dominant strategy incentive compatibility (DSIC) and Bayesian incentive compatibility (BIC). DSIC property guarantees that reporting true valuations (or costs as the case may be) is a best response for each bidder, irrespective of the valuations (or costs) reported by the other bidders. BIC is a much weaker property which ensures that truth revelation is a best response for each bidder whenever the other bidders are also truthful. Individual rationality (IR) is a property which guarantees non-negative utility to each participant in the mechanism thus ensuring their voluntary participation. The IR property may be (1) ex-ante IR (if the bidders decide on participation even before knowing their exact types (valuations or costs) ) or (2) interim IR (if the bidders decide on participation just after observing their types), or (3) ex-post IR (if the bidders can withdraw even after having participated). For more details on these concepts from mechanism design, the reader is referred to Chapter 2 of this thesis.

3.1.2 Contributions and Outline

In his seminal work, Myerson characterized an optimal auction for selling a single unit of a single item. Extending his work has been attempted by several researchers and there have been some generalizations of his work for multi-unit single item auctions. However, designing an optimal combinatorial auction in general settings is still elusive.

Armstrong characterized an optimal auction for two objects where type sets are binary. Malakhov and Vohra studied an optimal auction for a single item multi-unit procurement auctions using a network interpretation. An implicit assumption in the above papers is that the sellers have limited capacity for the item. They also assume that the valuation sets are discrete. Iyengar and Kumar and Gautam et al. have proposed an optimal auction for multi-unit, single item procurement.

Ledyard has looked at single unit combinatorial auctions in the presence of single minded bidders. A single minded bidder is one who only bids on a particular subset of the items. In this paper, there is a seller who wishes to sell single units of different items. The interested bidders bid on particular subsets of these items. The auction rules are such that it maximizes the revenue to the seller and the auction is BIC and individually rational. Ledyard’s auction, however, does not take into account multiple units of multiple items and this motivates our
current work which extends Ledyard’s auction to the case of procuring multiple units of multiple items. The following are our specific contributions.

1. We characterize Bayesian incentive compatibility and interim individual rationality for procuring multiple units of multiple items when the bidders are single minded, by deriving a necessary and sufficient condition.

2. We design an optimal auction that minimizes the cost of procurement while satisfying Bayesian incentive compatibility and interim individual rationality. We refer to this auction as OCAS (Optimal Combinatorial Auction with Single minded bidders).

3. We show, under appropriate regularity conditions, that the proposed optimal auction also satisfies dominant strategy incentive compatibility.

4. When the bidders are willing to provide volume discounts, we propose an auction, namely VD-OCAS and conjecture that it provides an optimal mechanism for the two item multi-unit combinatorial auction in the presence of single minded, capacitated bidders.

5. We introduce the notion of an XOR minded bidder (a bidder who submits XOR bid). We provide some insights on how an optimal auction can be designed when agents are either single minded or XOR minded.

The rest of the chapter is organized as follows. First, we will explain our model in Section 3.2 and describe the notation that we use. We also outline certain essential technical details of optimal auctions from the literature. In Section 3.3, we develop the OCAS auction. We describe in Section 3.4, how OCAS can be used for constructing an optimal auction when the bidders are willing to offer volume discounts. In Section 3.5, we describe how we can generalize the approach when bidders are XOR minded. Section 3.6 concludes the chapter.

### 3.2 The Model

We consider a scenario in which there is a buyer and multiple sellers. The buyer is interested in procuring a set of distinct objects, $I$. She is interested in procuring multiple units of each object. She specifies her demand for each object. The sellers are single minded.

**Definition 3.1 (Single Minded Bidder).** In a combinatorial auction, we say an agent is single minded if he is interested in selling one specific bundle of items.

Though in combinatorial auctions, the sellers can bid on any of non-empty subsets of items, that is, they can submit $2^{|I|} - 1$ bids, when the sellers are single minded, they submit only
one bid for supplying a particular bundle of the objects. We illustrate this through an example below.

**Example 3.1.** Consider a buyer interested in buying 100 units of $A$, 150 units of $B$, and 200 units of $C$. Assume that there are three sellers. Seller 1 might be interested in providing 70 units of bundle $\{A, B\}$, that is, 70 units of $A$ and 70 units of $B$ as a bundle. Because he is single minded, he does not bid for any other bundles. We also assume that he would supply equal numbers of $A$ and $B$. Similarly, seller 2 may provide a bid for 100 units of the bundle $\{B, C\}$. The bid from seller 3 may be 125 units of the bundle $\{A, C\}$.

We state below important assumptions in the model.

- The sellers are single minded.
- The bundle for which each seller is going to bid for, is common knowledge.
- The sellers can collectively fulfill the demands specified by the buyer.
- The sellers are capacitated i.e. there is a maximum quantity of the bundle of interest they can supply. The bid therefore specifies a unit cost of the bundle and the maximum quantity that can be supplied. After receiving these bids, the buyer will determine the allocation and payment as per auction rules.
- The seller will never inflate his capacity, as it can be detected. If he fails to supply the quantity exceeding his capacity, he incurs a penalty which is a deterrent on inflating his capacity.
- Whenever a buyer buys anything from a seller, she will procure the same number of units of each of the items from the seller’s bundle of interest.
- If the buyer procures more than required quantity of an item, he can freely dispose it off. (Free disposal property).

It should be noted that the capacity of the sellers is private information to the sellers. So private information with the sellers is multi-dimensional. Table 3.1 shows the notation that will be used in the rest of the chapter. When the arguments for the functions, $U_i$, or $u_i$ are clear from the context, we use just $U_i(\cdot)$, $u_i(\cdot)$ instead of $U_i(b_i, \theta_i)$, $u_i(b, \theta)$.

We design an optimal reverse auction with the above assumptions. It is to be noted that our results hold true for equivalent forward auction settings as well. Before we delve into the
| $I$ | Set of items the buyer is interested in buying, \{1, 2, \ldots, m\} |
| $D_j$ | Demand for item $j$, $j = 1, 2, \ldots, m$ |
| $N$ | Set of sellers, \{1, 2, \ldots, n\} |
| $c_i$ | True cost of production of one unit of bundle of interest to \(i\), $c_i \in [\bar{c}_i, \hat{c}_i]$ |
| $q_i$ | True capacity for bundle which seller \(i\) can supply, $q_i \in [\bar{q}_i, \hat{q}_i]$ |
| $\hat{c}_i$ | Reported cost by the seller \(i\) |
| $\hat{q}_i$ | Reported capacity by the seller \(i\) |
| $\theta_i$ | True type i.e. cost and capacity of the seller \(i\), $\theta_i = (c_i, q_i)$ |
| $b_i$ | Bid of the seller \(i\). $b_i = (\hat{c}_i, \hat{q}_i)$ |
| $b$ | Bid vector, $(b_1, b_2, \ldots, b_n)$ |
| $b_{-i}$ | Bid vector without the seller \(i\), i.e. $(b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$ |
| $t_i(b)$ | Payment to the seller \(i\) when submitted bid vector is \(b\) |
| $T_i(b_i)$ | Expected payment to the seller \(i\) when he submits bid \(b_i\). Expectation is taken over all possible values of $b_{-i}$ |
| $x_i = x_i(b)$ | Quantity of the bundle to be procured from the seller \(i\) when the bid vector is \(b\) |
| $X_i(b_i)$ | Expected quantity of the bundle to be procured from the seller \(i\) when he submits bid \(b_i\). Expectation is taken over all possible values of $b_{-i}$ |
| $f_i(c_i, q_i)$ | Joint probability density function of $(c_i, q_i)$ |
| $F_i(c_i, q_i)$ | Cumulative distribution function of $f_i(c_i, q_i)$ |
| $f_i(c_i|q_i)$ | Conditional probability density function of production cost when it is given that the capacity of the seller \(i\) is \(q_i\) |
| $F_i(c_i|q_i)$ | Cumulative distribution function of $f_i(c_i|q_i)$ |
| $H_i(c_i, q_i)$ | Virtual cost function for seller \(i\), $H_i(c_i, q_i) = c_i + \frac{F_i(c_i|q_i)}{f_i(c_i|q_i)}$ |
| $\rho_i(b_i)$ | Expected offered surplus to seller \(i\), when his bid is \(b_i\) |
| $u_i(b, \theta_i)$ | Utility to seller \(i\), when bid vector is \(b\) and his type is $\theta_i$ |
| $U_i(b_i, \theta_i)$ | Expected utility to the seller \(i\), when he submits bid \(b_i\) and his type is $\theta_i$. Expectation is taken over all possible values of $b_{-i}$ |

Table 3.1: Notation: optimal combinatorial auctions in the presence of single minded bidders

technical details, we will explain an equivalent forward auction setting. In the reverse auction settings, the buyer is interested in procuring at least a specified quantity of each of the objects.
In the forward auction setting, the seller is interested in selling at most some specified quantity of each of the objects in the auction. The buyers are interested in a specific bundle of items, that is buyers are single minded. Also, the buyers need at least some fixed units of a specific bundle of items and this quantity is private information to the buyers. If they do not receive the desired units, the total utility to them is zero. The buyers do not deflate the need. We will explain this with an example (Example 3.2).

Example 3.2. Consider a seller having four robots, namely A, B, C and D. He can offer service of A for 150 time slots, service of B for 100 time slots, C for 100 time slots and D for 120. There are four buyers for these services. Buyer 1 is interested in utilizing 100 time slots of robots A and B. Buyer 2 needs 75 time slots of robots B and C; buyer 3 needs 60 time slots of A, B and C; buyer 4 is interested in utilizing all four robots for 90 time slots. In this setting, the buyers are single minded meaning they need to utilize a specific subset of robots for the specified time slots. Buyer 1 would not be interested in 90 time slots of A and 110 time slots of B. So, if she has to avail the services, she should receive at least 100 time slots of A and B.

3.2.1 Some Preliminaries

The problem of designing an optimal mechanism was first studied by Myerson [2] and Riley and Samuelson [45]. Myerson’s work is more general and considers the setting of a seller trying to sell a single unit of a single object to one of several possible buyers. In the rest of the chapter, the auctioneer will be a buyer and her objective will be to minimize the cost of procurement. In this chapter, we refer to an optimal auction as follows.

Definition 3.2 (An Optimal Auction). An auction is called as an optimal auction for a buyer if it minimizes the expected cost of procurement and satisfies Bayesian incentive compatibility, interim individual rationality, and for each item, at least the desired number of units are procured.

Note that, in this particular subsection, while explaining Myerson’s auction, unlike the rest of chapter, the auctioneer is the seller and his objective is to maximize the revenue.

Myerson’s Optimal Auction

First we will see Myerson’s optimal auction. In this particular setting, as per notation defined in Table 3.1, \( m = 1, D_1 = 1 \). (So, \( q_i \) will be 1 for all the agents and no longer a private information). \( F_i, H_i \), defined in Table 3.1, will be functions of single variable. The buyer’s private information will be the maximum cost she is willing to pay, which we will denote as \( \theta_i \). \( \theta_i \in \Theta_i = [\theta_i, \overline{\theta}_i] \).
Myerson \cite{2} characterizes all auction mechanisms that are Bayesian incentive compatible and interim individually rational in this setting. From this, he derives the allocation rule and the payment function for the optimal auction mechanism, using an interesting notion called the virtual cost function, defined as follows:

\[
H_i(\theta_i) = \theta_i - 1 - \frac{1 - F_i(\theta_i)}{f_i(\theta_i)}
\]

He has shown that an optimal auction is one with allocation rule \(x_i(.)\) satisfying:

\[
\text{maximize } \int \cdots \int \left[ \sum_{i=1}^{n} x_i(\theta_i) H_i(\theta_i) \right] \left[ \prod_{i=1}^{n} f_i(\theta_i) \right] d\theta_1 d\theta_2 \cdots d\theta_n \tag{3.1}
\]

subject to
1. \(X_i(\theta_i)\) is non-decreasing in \(\theta_i\) \(\forall \ i \in N\).
2. \(\sum_{i=1}^{n} x_i(.) \leq 1\).
3. And the payment rule for each \(i \in N\) is given by,

\[
T_i(\theta_i) = E_{b_{-i}}(u_i(\theta) - \theta_i(x_i(\theta)))
= U_i(\theta_i) - \theta_i X_i(\theta_i)
= \int_{\theta_i}^{\theta_{i}} X_i(s) ds - \theta_i X_i(\theta_i) \tag{3.2}
\]

Any auction for a single unit of a single item which satisfies Equation (3.1) and Equation (3.2) is optimal i.e. maximizes seller’s revenue and is BIC and IIR.

**Regularity Assumption**: If \(H_i(\theta_i)\) is increasing with respect to \(\theta_i\), then we say, the virtual cost function is regular or regularity condition holds true. Under this assumption one such optimal auction is, for each \(i \in N\),

\[
x_i(\theta) = 1 \text{ if } H_i(\theta_i) > \max \left\{ 0, \max_{j \neq i} H_j(\theta_j) \right\}
= 0 \text{ otherwise} \tag{3.3}
\]

The payment rule is given by,

\[
t_i(\theta_i, \theta_{-i}) = \left( \int_{\theta_i}^{\theta_{i}} x_i(s, \theta_{-i}) ds \right) - \left( \theta_i x_i(\theta) \right) \forall \theta \in \Theta
\]

The above auction can be described algorithmically as follows:

1. Collect bids from the buyers
2. Sort them according to their virtual costs

3. If the highest virtual cost is positive, allocate the object to the corresponding bidder

4. The winner, say $i$, will pay $t_i(\theta_{-i})$
   
   $$= \inf \{ \theta_i | H_i(\theta_i) > 0 \text{ and } H_i(\theta_i) > H_j(\theta_j) \forall j \neq i \}$$

From the payment rule, it is a dominant strategy for each bidder to bid truthfully under the regularity assumption. When bidders are symmetric, i.e. $F_i$ is the same for all bidders $i \in N$, then the above optimal auction is precisely Vickrey’s second price auction.

Myerson’s work can be easily extended to the case of multi-unit auctions with unit demand. But problems arise when the unit-demand assumption is relaxed. We move into a setting of multi-dimensional type information which makes truth elicitation non-trivial. Several attempts have addressed this problem, albeit under some restrictive assumptions. It is assumed, for example, that even though the seller is selling multiple units (or even objects), the type information of the entities is still one dimensional.

Researchers have also worked on extending Myerson’s work for an optimal auction for multiple objects. The private information in this setting may not be single dimensional. Armstrong has solved this problem for the two objects case, when type sets are binary, by enumerating all incentive compatibility conditions. Manelli and Vincent take a different approach. They define a feasible set of utility functions and show that the optimal mechanism needs to be necessarily an extreme point of this set and every extreme point is an optimal auction under certain distributions over types of the agents.

Ledyard has characterized an optimal multi-object single unit auction, when bidders are single minded. He shows that an optimal auction is similar to Myerson’s optimal auction. However, while solving the optimization problem, the seller has to ensure the feasibility constraints arising due to single units of multiple items and the bidders being single minded.

We now develop an optimal auction for multi-unit multi-item procurement in the presence of single minded and capacitated bidders.

### 3.3 Optimal Multi-Unit Combinatorial Procurement Auction

We will start this section with an example to illustrate that in a multi-unit, multi-item procurement auction, the suppliers may have an incentive to misreport their costs or capacities.

**Example 3.3.** Suppose the buyer has a requirement for 1000 units of an item. Also, suppose that there are four suppliers with $(c_i, q_i)$ values of $S_1 : (10, 500)$, $S_2 : (8, 500)$, $S_3 : (12, 800)$ and $S_4 : (6, 500)$. Suppose the buyer conducts the classic $k^{th}$ price auction, where the payment
to a supplier is equal to the cost of the first losing supplier. In this case, the sellers will be able
to do better by misreporting types. To see this, consider that all suppliers truthfully bid both
the cost and the quantity bids. The allocation then would be $S_1: 0, S_2: 500, S_3: 0, S_4: 500$
and this minimizes the total payment. Under this allocation the payment to $S_4$ would be
$10 \times 500 = 5000$ currency units. However, if he bids his quantity to be 490, then the allocation
changes to $S_1: 10, S_2: 500, S_3: 0, S_4: 490$ giving him a payment of $12 \times 490 = 5880$ currency
units and thus incentive compatibility does not hold. Thus it is evident that such uniform price
mechanisms are not applicable to the case where both unit cost and maximum quantity are
private information. The intuitive explanation for this could be that by under reporting their
capacity values, the suppliers create an artificial scarcity of the resources in the system. Such
fictitious shortages force the buyer to pay overboard for use of limited resources.

We also make another observation here. Suppose, $S_4$ bids $(6, 600)$. Then the buyer will
order from him 600 units at the cost of 10 per unit. Since his capacity is 500, he would not be
able to supply the remaining 100 units. If he bids $(6, 1000)$, then he will be paid only 8 per unit
and the buyer will be ordering from him 1000 units. This clearly vindicates our assumption that
a seller will not inflate his capacity.

We are interested in designing an optimal mechanism, for a buyer, that satisfies Bayesian
incentive compatibility (BIC) and individual rationality (IR). BIC means that the best response
of each seller is to bid truthfully if all the other sellers are bidding truthfully. IR implies the
players have non-negative payoff by participating in the mechanism. More formally, these can
be stated as (see Table 3.1 for notation),
\[
\forall i \in N \text{ and } \forall \theta_i \in [c_i, \bar{c}_i] \times [\bar{q}_i, \bar{q}_i]
\]
\[
U_i(\theta_i, \theta_i) \geq U_i(b_i, \theta_i) \forall b_i, \quad \text{(BIC)}
\]
\[
U_i(\theta_i, \theta_i) \geq 0 \quad \text{(IR)}
\]

The IR condition above corresponds to interim individual rationality (IIR).

3.3.1 Necessary and Sufficient Conditions for BIC and IR

To make the sellers report their types truthfully, the buyer has to offer them incentives. We
propose the following incentive, motivated by paying a seller higher than what he claims to be
the total cost of the production for the ordered quantity. $\forall i \in N,$
\[
\rho_i(b_i) = T_i(b_i) - \hat{c}_i X_i(b_i), \text{ where } b_i = (\hat{c}_i, \hat{q}_i)
\]
This implies,

\[ U_i(b_i, \theta_i) = T_i(b_i) - c_i X_i(b_i) \]
\[ = \rho_i(b_i) - (c_i - \hat{c}_i) X_i(b_i) \]  \( (3.6) \)

With the above offered incentive, we now state and prove the following theorem.

**Theorem 3.1.** Any mechanism in the presence of single minded, capacitated sellers is BIC and IR iff \( \forall i \in N \),

1. \( \rho_i(b_i) = \rho_i(\hat{c}_i, \hat{q}_i) + \int_{\hat{c}_i}^{\bar{c}_i} X_i(t, \hat{q}_i) dt \)
2. \( \rho_i(b_i) \) non-negative, and non-decreasing in \( \hat{q}_i \) \( \forall \hat{c}_i \in [c_i, \bar{c}_i] \)
3. The quantity which seller \( i \) is asked to supply, \( X_i(c_i, q_i) \) is non-increasing in \( c_i \) \( \forall q_i \in [\bar{q}_i, \hat{q}_i] \).

**Proof:**
A similar theorem is presented by Iyengar and Kumar [41] for the case of multi-unit single item procurement auctions. Our theorem is for the multi-item setting with single minded bidders. To prove the necessity part of the theorem, we first observe that,

\[ U_i(b_i, \theta_i) = U_i(\hat{c}_i, \hat{q}_i, c_i, q_i) \]
\[ = T_i(b_i) - c_i X_i(b_i) \]  \( (3.7) \)

From Equation \( (3.7) \) with appropriate rearrangement of terms,

\[ U_i(\hat{c}_i, q_i, c_i, q_i) = U_i(\hat{c}_i, \hat{q}_i, \hat{c}_i, q_i) + (\hat{c}_i - c_i) X_i(\hat{c}_i, q_i) \]

BIC implies

\[ U_i(\hat{c}_i, \hat{q}_i, c_i, q_i) \leq U_i(c_i, q_i, c_i, q_i), \quad \forall (\hat{c}_i, \hat{q}_i) \text{ and } (c_i, q_i) \in \Phi_i \]

In particular,

\[ U_i(\hat{c}_i, q_i, c_i, q_i) \leq U_i(c_i, q_i, c_i, q_i) \]

Without loss of generality, we assume \( \hat{c}_i > c_i \). Rearrangement of these terms yields

\[ \frac{U_i(\hat{c}_i, q_i, \hat{c}_i, q_i) - U_i(c_i, q_i, c_i, q_i)}{\hat{c}_i - c_i} \leq -X_i(\hat{c}_i, q_i) \]

Similarly, using \( U_i(c_i, q_i, \hat{c}_i, q_i) \leq U_i(\hat{c}_i, q_i, \hat{c}_i, q_i) \), we obtain,

\[ -X_i(c, q) \leq \frac{U_i(\hat{c}_i, q_i, \hat{c}_i, q_i) - U_i(c_i, q_i, c_i, q_i)}{\hat{c}_i - c_i} \leq -X_i(\hat{c}_i, q_i). \]  \( (3.8) \)
Taking limit $\hat{c}_i \to c_i$, we get,
\[
\frac{\partial U_i(c_i, q_i, c_i, q_i)}{\partial c_i} = -X_i(c_i, q_i).
\] (3.9)

Equation (3.8) implies $X_i(c_i, q_i)$ is non-increasing in $c_i$. This proves necessity of condition 3 of the theorem. When the seller bids truthfully, from Equation (3.6),
\[
\rho_i(c_i, q_i) = U_i(c_i, q_i, c_i, q_i).
\] (3.10)

For BIC, Equation (3.9) should be true. So,
\[
\rho_i(c_i, q_i) = \rho_i(c_i, q_i) + \int_{c_i}^{\hat{c}_i} X_i(t, q_i)dt
\] (3.11)

This proves necessity of condition 1 of the theorem. BIC also requires
\[
q_i \in \arg \max_{\hat{q}_i \in [q_i, \bar{q}_i]} U_i(c_i, \hat{q}_i, c_i, q_i) \forall c_i \in [c_i, \bar{c}_i]
\]
(Note that $\hat{q}_i \in [q_i, q_i]$ and not $\in [q_i, \bar{q}_i]$ as it is assumed that a bidder will not over-report his capacity). This implies, $\forall c_i$, $\rho_i(c_i, q_i)$ should be non-decreasing in $q_i$'s. The IR conditions (Equations (3.5) and (3.10)) imply
\[
\rho_i(c_i, q_i) \geq 0.
\]

This proves necessity of condition 2 of the theorem. Thus, these three conditions are necessary for BIC and IR properties. We now prove that these are sufficient conditions for BIC and IR.
Assume that all three conditions are true. Then
\[
U_i(\theta_i, \theta_i) = \rho_i(c_i, q_i) \geq 0.
\]

So the IR property is satisfied.
\[
U_i(b_i, \theta_i) = \rho_i(\hat{c}_i, \hat{q}_i) + (\hat{c}_i - c_i)X_i(\hat{c}_i, \hat{q}_i)
\]
\[
= \rho_i(\hat{c}_i, \hat{q}_i) + \int_{c_i}^{\hat{c}_i} X_i(t, \hat{q}_i)dt + (\hat{c}_i - c_i)X_i(\hat{c}_i, \hat{q}_i)
\]
\[
= \rho_i(\hat{c}_i, \hat{q}_i) + \int_{c_i}^{\hat{c}_i} X_i(t, \hat{q}_i)dt - \int_{c_i}^{\hat{c}_i} X_i(t, \hat{q}_i)dt + (\hat{c}_i - c_i)X_i(\hat{c}_i, \hat{q}_i)
\]
\[
\leq \rho_i(c_i, \hat{q}_i) \quad \text{as } X_i \text{ is non-increasing in } c_i
\]
\[
\leq \rho_i(c_i, q_i)
\]
\[
= U_i(\theta_i, \theta_i) \quad \text{as } \rho_i \text{ is non-decreasing in } q_i
\]

Thus BIC property is also satisfied. This proves the sufficiency part of the theorem. 

\[\blacksquare\]
3.3.2 Allocation and Payment Rules

The buyer’s problem is to solve

\[ \min \ E_b \sum_{i=1}^{n} t_i(b) \]

subject to,

1. \[ t_i(b) = \rho_i(b) + \tilde{c}_i x_i(b) \]
2. All three conditions in Theorem 3.1 hold true.
3. She procures at least \( D_j \) units of each item \( j \).

Expectation being a linear operator, the buyer’s problem is to minimize \( \sum_{i=1}^{n} E_b T_i(\tilde{c}_i, \tilde{q}_i) \). Condition 1 of the theorem has to hold true, which will imply the \( i^{th} \) term in the summation is given by

\[
\int_{\tilde{c}_i}^{\tilde{c}_i} \left( c_i X_i(c_i, q_i) + \rho_i(\tilde{c}_i, q_i) + \int_{\tilde{c}_i}^{\tilde{c}_i} X_i(t, q_i) dt \right) f_i(c_i, q_i) dc_i dq_i
\]

However,

\[
\int_{\tilde{c}_i}^{\tilde{c}_i} \left( \int_{\tilde{c}_i}^{\tilde{c}_i} X_i(t, q_i) dt \right) f_i(c_i, q_i) dc_i = \int_{\tilde{c}_i}^{\tilde{c}_i} X_i(c_i, q_i) F_i(c_i|q_i) f_i(q_i) dc_i
\]

Condition 2 of Theorem 3.1 requires \( \rho_i(\tilde{c}_i, q_i) \geq 0 \) and the buyer wants to minimize the total payment to be made. So, she has to assign \( \rho_i(\tilde{c}_i, q_i) = 0, \forall q_i, \forall i \). So her problem is to solve

\[
\min \sum_{i=1}^{n} \int_{\tilde{c}_i}^{\tilde{c}_i} \left( c_i + \frac{F_i(c_i|q_i)}{f_i(c_i|q_i)} \right) X_i(c_i, q_i) f_i(c_i, q_i) dc_i dq_i
\]

That is,

\[
\min \sum_{i=1}^{n} \int_{\tilde{c}_i}^{\tilde{c}_i} H_i(c_i, q_i) X_i(c_i, q_i) f_i(c_i, q_i) dc_i dq_i
\]

where, \( H_i(c_i, q_i) \) is the virtual cost function, defined in Table 3.1.

Define

\[
\tilde{c} = (\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n) \\
c = (c_1, c_2, \ldots, c_n) \\
\mathcal{C} = (c_1, c_2, \ldots, c_n).
\]

Similarly, define \( \tilde{q} \), \( q \) and \( \mathcal{Q} \). Let

\[
dc = dc_1 dc_2 \ldots dc_n \\
dq = dq_1 dq_2 \ldots dq_n \\
f(c, q) = \prod_{i=1}^{n} f_i(c_i, q_i)
\]
Her problem now reduces to

\[
\min \int_{q}^{c} \left( \sum_{i=1}^{n} H_i(c_i, q_i) x_i(c_i, q_i) \right) f(c, q) dc dq \\
\text{s.t.}
\]

1. \( \forall i, X_i(c_i, q_i) \) is non-increasing in \( c_i, \forall q_i \).
2. The Buyer’s minimum requirement of each item is satisfied.

The payments are calculated as

\[
T_i(c_i, q_i) = c_i X_i(c_i, q_i) + \int_{c_i}^{q_i} X_i(s, q_i) ds
\]

This is an optimal auction for the buyer in the presence of the single minded sellers. We refer to this auction as OCAS (Optimal Combinatorial Auction with Single minded bidders).

The above discussion can be summarized as the following theorem.

**Theorem 3.2.** In the presence of single minded, capacitated bidders (sellers), the OCAS is an optimal multi-unit, multi-item procurement auction. That is, an optimal auction for the buyer buying multiple units of multiple items is obtained by solving the optimization problem (3.12).

In the next subsection, we will see an optimal auction under regularity conditions.

### 3.3.3 Optimal Auction under Regularity Assumption

First, we make the assumption that,

\[
H_i(c_i, q_i) = c_i + \frac{F_i(c_i|q_i)}{f_i(c_i|q_i)}
\]

is non-increasing in \( q_i \) and non-decreasing in \( c_i \). This regularity assumption is the same as the one made by Iyengar and Kumar [41]. With this assumption, we now state the following theorem.

**Theorem 3.3.** Under regularity assumption, the buyer’s optimal auction when bidder \( i \) submits bid \( (c_i, q_i) \) is,

\[
\min \sum_{i=1}^{n} x_i H_i(c_i, q_i)
\]

subject to

1. \( 0 \leq x_i \leq q_i \), where \( x_i \) denotes the quantity that seller \( i \) has to supply of bundle \( \bar{x}_i \).
2. Buyer’s demands are satisfied.
Proof:

**Step 1:** The first condition in (3.12), that is, \( X_i(c_i, q_i) \) is non increasing in \( c_i \), \( \forall q_i \) and \( \forall i \) is satisfied when the regularity assumption holds.

**Step 2:** Thus any feasible solution to (3.14) is also a feasible solution to (3.12).

**Step 3:** The objective function in (3.14) is minimized for every \( c_i, q_i \) and \( \forall i \). Thus it minimizes the objective function in (3.12).

After this problem has been solved, the buyer pays each seller \( i \) the amount

\[
t_i = c_i x_i^* + \int_{c_i}^{c_i} x_i(t, q_i) dt
\]

where \( x_i^* \) is the quantity that agent \( i \) has to supply after solving the above problem.

We exemplify the above optimal mechanism under regularity assumption with an example. Before looking into the example, we would like to add a comment. The regularity assumption holds true means, for each of the agents virtual cost is increasing with true cost and non-decreasing with the capacity. Generally, with most of the standard probability density functions, this assumption holds true. With this optimization problem (3.12) becomes much easier to solve, in particular, solution to optimization problem (3.14) is a solution to optimization problem (3.12) too.

**Example 3.4.** Suppose the buyer is interested in buying 100 units of \( \{A, C, D\} \) and 250 units of \( \{B\} \). There are four sellers. Now, Seller 1 (S1) is interested in providing \( q_1 = 100 \) units of bundle \( \{A, B\} \); Seller 2 (S2): \( q_2 = 100 \) units of \( \{B\} \); Seller 3 (S3): \( q_3 = 150 \) units of \( \{B, C, D\} \); and Seller 4 (S4) is interested in up to \( q_1 = 120 \) units of \( \{A, B, C, D\} \). Assume for each agent that cost of supplying each unit of his bundle is independent of quantity he can supply, and is uniformly distributed over \([0,200]\). Then, \( H_i(c_i, q_i) = c_i + \frac{F(c_i|q_i)}{f(c_i|q_i)} = 2c_i \forall i \). Thus, the regularity assumption holds true.

Let the unit costs of the respective bundles be \( c_1 = 100, c_2 = 50, c_3 = 70 \) and \( c_4 = 110 \). Each seller will submit his bid \( (c_i, q_i) \). After receiving the bids, buyer will solve,

\[
\text{minimize } x_1 H_1(100, 100) + x_2 H_2(50, 100) + x_3 H_3(70, 150) + x_4 H_4(110, 120)
\]
subject to,

\begin{align*}
  x_1 & \leq 100 \\
  x_2 & \leq 100 \\
  x_3 & \leq 150 \\
  x_4 & \leq 120 \\
  x_1 + x_4 & \geq 100 \quad \text{(3.16)} \\
  x_1 + x_2 + x_3 + x_4 & \geq 250 \quad \text{(3.17)} \\
  x_3 + x_4 & \geq 100 \quad \text{(3.18)} \\
  x_i & \geq 0 \quad i = 1, 2, 3, 4.
\end{align*}

Equation (3.16) is required to be satisfied as at least 100 units of \( A \) have to be procured. Equation (3.17) is for procuring at least 250 units of \( B \), and Equation (3.18) is for procuring at least 100 units of \( C \) and \( D \). After solving this optimization problem, the buyer will determine the payment according to Equation (3.15).

In the above optimal auction, the buyer will procure 100 units of \( \{A, B\} \) from S1, 50 units of \( \{B\} \) from S2 and 100 units of \( \{B, C, D\} \) from S3. The buyer will pay 11000 to S1, 3500 to S2 and 11000 to S3.

We now state an important observation about optimal auction under regularity condition.

**Lemma 3.1.** The optimal multi-unit combinatorial auction (3.14) in the presence of single minded and capacitated bidders, under regularity condition is DSIC.

**Proof:**

It can be seen that for each seller \( i \), the best response is to bid truthfully irrespective of whatever the others are bidding. Thus, this mechanism enjoys the stronger property, namely dominant strategy incentive compatibility (DSIC), though we started with designing a Bayesian incentive compatible (BIC) mechanisms.

The above property is a direct consequence of the result proved by Mookherjee and Stefan [50] who have provided sufficient conditions for DSIC implementation of a BIC mechanism. These conditions state that, if BIC allocation rules satisfy a certain monotonicity, we can as well implement it in DSIC with appropriate money transfers without affecting expected utilities of the participating agents. Under these regularity assumptions, \( x_i \)'s (allocation rules) satisfy these conditions. So we have a DSIC mechanism.
3.4 Volume Discounts

We have seen optimal multi unit combinatorial procurement auction in the presence of single minded bidders. In real life, the suppliers may offer volume discounts. That is, if the buyer is asking for more quantity, the suppliers would be able to exploit economies of scale and hence provide volumes at lower per unit cost. We assume that the bids are in the form of a cost-demand curve, that is, marginal cost per unit at a particular demand. We assume that these curves are decreasing. We will explain this with one example.

Example 3.5. Assume that a supplier can supply a particular subset of items as a bundle. Let the cost-demand curve for a bidder be as specified in the Figure 3.2. The bidder can supply the first 100 units of the bundle at $50 per unit; next 50 units of the bundle at $40 per unit, etc. For a demand of 150-200 units, the per unit cost is $25, and $15 for number of units between 200-300. He can supply a maximum 300 units. Suppose, the buyer asks him to supply 170 units of his bundle, she has to pay him at least $50 \times 100 + $40 \times 50 + $25 \times 20 = $7500.

Volume discount bids introduce an additional level of complexity in the buyer’s supplier selection or bid selection problem and the payments. In this setting, ensuring incentive compatibility becomes a much harder problem. Gautam et al. have considered the single item multi-unit auction with volume discounts. In general, generalizing the technique developed in...
for volume discounts may not work for multi-item auctions. In this section, we show that their technique can be generalized for two item multi-unit auction with volume discounts in the presence of single minded bidders.

Say the two items are $A$ and $B$. As the bidders are single minded, the bidders will be of three types. Either (i) bidding for supplying item $A$, or (ii) supplying item $B$, or (iii) supplying the bundle \{A, B\}. Assume that the bidders are bidding truthfully. Then we argue that there is a cost minimizing procurement in which all the bidders who are supplying items, will supply to their full capacity except possibly one from each type of the bidder among (type I, II, and III). Before that, we will define a few terms.

**Definition 3.3** (Optimal Allocation). We call an assignment of the items and the number of units supplied by the agents as an optimal allocation if it minimizes the cost of procurement.

**Definition 3.4** (Active Agent). We say an agent is active agent if he is supplying strictly positive units of the bundle of his own interest, that is the bundle for which he bids for. Otherwise, if he is not supplying any item, we say the agent is inactive.

**Definition 3.5** (Saturated Agent). We say an active agent $i$ is a saturated agent if he is supplying the bundle of items which he bids for, to his full capacity. That is, the agent $i$ supplying $\hat{q}_i$ units of his bundle. Otherwise we say the agent is unsaturated.

**Proposition 3.1.** There exists an optimal allocation such that, all the active bidders are saturated except possibly one in each of the types (that is I, II and III).

**Proof:**
Suppose in an optimal allocation for type I bidders, there are two or more, say $k$, active agents who are not saturated. For the other types of agents, the same arguments hold true.

**Claim 3.1.** In an optimal allocation, the cost per unit for the last unit supplied by all the active and unsaturated type I bidders should be the same.

Suppose not. Then, we can ask the active and unsaturated type I bidder with the lowest marginal cost to supply one more unit and one less from any one of the other active type I agent having higher cost per unit. As the cost-demand curve is non-increasing, this would lead to decrease in the cost, contradicting optimality of the allocation. Hence the claim.

Now, we can un-assign some units from $k$ and ask agent 1 to supply more units. This would not increase the cost as cost curves are non-increasing. This procedure can continue until either agent 1 is saturated or agent $k$ becomes inactive. If agent 1 becomes saturated, we can continue
with agent 2 and if agent \( k \) is inactive, we can similarly reassign the allocation for the remaining agents until all the agents are saturated, with the possibility that at most one is left unsaturated.

We see an optimal auction for the above setting in the next subsection.

### 3.4.1 Optimal Combinatorial Auction with Volume Discounts

First we make the observation that for saturated agents, it does not matter what the cost-demand curve is, but what matters is only the average cost per unit and the capacity. Based on this observation, we propose the following auction which we conjecture to be an optimal auction for multi item multi unit combinatorial auction with volume discounts in the presence of single minded bidders.

| VD-OCAS |

| Step 1: For each bidder \( i \) with capacity \( q_i \), calculate his total cost of supplying \( Tc_i \), when he is saturated. Let \( \tilde{c}_i = \frac{Tc_i}{q_i} \). Let \( \tilde{F}(\tilde{c}_i|q_i) \) be distribution for \( \tilde{c}_i \) given that the agent \( i \) has capacity \( q_i \) and \( \tilde{f}(c_i|q_i) \) be the corresponding density function. |

| Step 2: Calculate |

\[
\tilde{H}_i(\tilde{c}_i, q_i) = \tilde{c}_i + \frac{\tilde{F}(c_i|q_i)}{\tilde{f}(c_i|q_i)}
\]  

(3.19)

| Step 3: Now solve the OCAS with \( \tilde{c}_i \) and \( \tilde{H}_i \). That is, solve the optimization problem (3.12). |

**Conjecture 3.1.** *VD-OCAS is an optimal mechanism.*

*Outline of a Possible Proof:* The VD-OCAS auction reduces the auctioneers problem to OCAS with appropriately modified virtual cost functions. Using arguments similar to Gautam et al [40] paper, our mechanism is an optimal auction for two-item, multi unit combinatorial procurement auction in the presence of capacitated, single minded bidders with volume discounts. We only conjecture that the VD-OCAS is an optimal auction. The reason is, we are at this point not clear if the arguments in the paper [40] can be extended to the situation here, though we strongly believe extension.
We also provide an intuitive argument why the VD-OCAS may be an optimal auction. That is, we need to show that, VD-OCAS satisfies BIC and IIR properties as well as minimizes the expected cost of procurement for the buyer. Recall, the Myerson auction, the Ledyard auction or the OCAS. In all optimal auctions in the respective papers, the allocation rule is such that the buyer minimizes the virtual cost of procurement and pays the agents by solving an appropriate integral. Thus, the virtual costs incentivize the agents optimally. That is, it is a best response for each bidder to submit his true cost and capacity, and minimizes the cost of procurement for the buyer. Based on this, we may expect the optimal auction for the settings in this section, the buyer has to minimize appropriate virtual cost of procurement and calculate payments in similar fashion as in other optimal auctions. By Proposition 3.1 and the above observation, only the average cost per unit for all the bidders, possibly except one in each of the type i, ii and iii; matters in the auction. So, we believe that the virtual cost function defined in Equation (3.19) serves the purpose of optimal auction.

3.4.2 VD-OCAS Under Regularity Assumption

We assume, for each agent $i$, the virtual cost function in Equation (3.19) is non-increasing in $q_i$ and non-decreasing in $\tilde{c}_i$. This is called as regularity assumption similar to Section 3.3.3. If regularity assumption holds true, we have an algorithmic description for an optimal mechanism.

The algorithm partitions the agents in three groups for type i, ii, and iii, say, $P_1$, $P_2$, and $P_3$. It sorts the agents according to increasing $\tilde{H}$ in each group and then greedily procures the items from the agents which we describe in the following algorithm.

VD-OCAS with regularity assumption (Description)

Find out whether it is better to get next the unit of items from type i and type ii agents or get the bundle from type iii agent. Continue till $D_A$ and $D_B$ units are procured. It may happen that, for item A or B, more items are procured because of type iii agents. After allocating greedily, if one of the items is procured more than needed, say item A, then deallocate the items from type i agents in reverse order of assignments such that only $D_A$ units are procured. The payments are calculated as in optimal auction (3.15).

The complexity of the above algorithm is $O(n \log n)$. In general we cannot solve the OCAS in polynomial time as the optimization problem (3.12) is at least as hard as the set cover problem which is NP-Hard. We have a polynomial time algorithm because we have assumed only the two items case.
To handle volume discounts, the technique used here for the two item case need not extend when there are more than two items. We are working towards designing optimal auctions with volume discounts with three or more items. In the next section we consider XOR bidding with unit demand case.

3.5 An Optimal Auction when Bidders are XOR-Minded

Consider the situation where a supplier can produce some of the items required by the buyer, say $A, B, C, D$. However, with the infrastructure he has, at a time either he can produce $A, D$ or $B, C$ but not any other combinations simultaneously. Thus he can either supply $A, D$ as a bundle or $B, C$ as a bundle but not both. That is, he is interested in XOR bidding.

**Definition 3.6 (XOR Minded Bidder).** We say a bidder is an XOR minded if he is interested in supplying either of two disjoint subsets of items but not both.

To simplify the analysis, in this section, we restrict ourselves to the unit demand case. That is the buyer is interested in buying a single unit of each of the items from $I$. Hence there are no capacity constraints. We formally state the assumptions.

- We assume that the bidders are XOR minded.
- For each bidder, his costs of the two bundles of his interest are independent.
- The two bundles for which each seller submits an XOR bid are known.
- The sellers can collectively supply the items required by the buyer.
- The buyer and the sellers are strategic.
- Free disposal. That is, if the buyer procures more than one unit of an item, he can freely dispose it off.

With the above assumptions, we now discuss an extension of the current art of designing optimal auctions for combinatorial auctions in the presence of XOR minded bidders. Though we assume the bidders are XOR minded, the BIC characterization and the auction presented here work even if each bidder is either single minded or XOR minded.

3.5.1 Notation

As, $q_i = 1$ for each bidder, we drop capacity from the types and bids for all the agents. Each agent will be reporting the costs for each bundle of his interest, he will be bidding two non-negative real numbers. And we need to calculate virtual costs on both the bundles. Thus, we
need appropriate modifications in some of the notation used in the chapter. We summarize the new notation for this section in Table 3.2. Each agent is submitting two different bids on two different bundles. We will use \( j \) to refer to the bundle.

| \( j \) | \( j = 1 \) or \( 2 \). Bundle index. |
| \( B_{ij} \) | The \( j^{th} \) bundle of items for which the agent \( i \) is bidding. \( j = 1, 2 \) |
| \( c_{ij} \) | True cost of production of \( B_{ij} \) to the seller \( i \). \( c_{ij} \in [c_i, \bar{c}_i] \) |
| \( c_i \) | \( = (c_{i1}, c_{i2}) \) |
| \( \theta_i \) | True type i.e. costs for \( i \). \( \theta_i = (c_{i1}, c_{i2}) \) |
| \( b_i \) | Bid of the seller \( i \). \( b_i = (\hat{c}_{i1}, \hat{c}_{i2}) \) |
| \( x_{ij} = x_{ij}(b) \) | Indicator variable to indicate whether \( B_{ij} \) is to be procured from the seller \( i \) when the bid vector is \( b \) |
| \( X_{ij}(b_i) \) | Probability that \( B_{ij} \) is procured from the seller \( i \) when he submits bid \( b_i \). Expectation is taken over all possible values of \( b_{-i} \) |
| \( f_{ij}(c_{ij}) \) | Probability density function of \( (c_{ij}) \) |
| \( F_{ij}(c_{ij}) \) | Cumulative distribution function of \( c_{ij} \) |
| \( H_{ij}(c_{ij}) \) | Virtual cost function for seller \( i \), for bundle \( B_{ij} \). \( H_{ij}(c_{ij}) = c_{ij} + \frac{F_{ij}(c_{ij})}{f_{ij}(c_{ij})} \) |

Table 3.2: Notation: XOR minded bidders

### 3.5.2 Optimal Auctions when Bidders are XOR Minded

First we characterize the BIC and IIR mechanisms for the settings under consideration in the next subsection. We design an optimal auction in Subsection 3.5.2.

**BIC and IIR: Necessary and Sufficient Conditions**

The utility for agent \( i \) is

\[
U_i(b_i, \theta_i) = -c_{i1}X_{i1} - c_{i2}X_{i2} + T_i(b_i, \theta_i)
\]

Using arguments similar to those in the proof of Theorem 3.1, for any mechanism in the presence of XOR minded bidders, the necessary condition for BIC is,

\[
\begin{align*}
\frac{\partial U_i(\cdot)}{\partial c_{i1}} &= -X_{i1}(c_{i1}, c_{i2}) \\
\frac{\partial U_i(\cdot)}{\partial c_{i2}} &= -X_{i2}(c_{i1}, c_{i2})
\end{align*}
\]

(3.20)
and \(X_{ij}(c_{i1}, c_{i2})\) should be non-increasing in \(c_{ij}, j = 1, 2\). We make an assumption that,

\[
\frac{\partial X_{i1}(\cdot)}{\partial c_{i2}} = \frac{\partial X_{i2}(\cdot)}{\partial c_{i1}}
\]

(3.21)

In general, the above assumption is not necessary for the mechanism to be truthful. However, if we assume that Equation (3.21) is true, we can solve PDE (3.20) analytically. Now we can state the following theorem,

**Theorem 3.4.** *With assumption (3.21), a necessary and sufficient condition for a mechanism to be BIC and IIR in the presence of XOR minded bidders is,

1. \(T_i(\cdot) = c_{i1}X_{i1}(\cdot) + c_{i2}X_{i2}(\cdot) + \int_{(c_{i1}, c_{i2})} U_i(\cdot) d\theta_i\)
2. \(X_{ij}(c_{i1}, c_{i2})\) should be non-increasing in \(c_{ij}, j = 1, 2\).
3. \(U_i(\overline{c}_{i1}, \overline{c}_{i2}) \geq 0\).*

### Optimal Auction with Regularity Assumption

Suppose we assume that \(H_{ij}\) is non-decreasing in \(c_{ij}\) for each \(i, j\). This is the same regularity assumption as in Myerson [2]. Now, following a similar treatment as for the buyers problem in Section 3.3.3, the buyer’s problem reduces to:

\[
\min \sum_{i=1}^{n} \sum_{j=1}^{2} x_{ij} H_{ij}(c_{ij})
\]

subject to

1. \(x_{ij} \in 0, 1\), where \(x_{ij}\) indicates whether supplier \(i\) is supplying his \(j^{th}\) or no.
2. \(x_{i1} + x_{i2} \leq 1\). (XOR minded bidder).
3. All the items are procured.

(3.22)

Now, we show that at optimal allocation in Problem (3.22), the assumption (3.21) holds true. For an agent \(i\), fix \(\theta_{-i}\) and consider the square of his types \([c_{i1}, \overline{c}_{i1}] \times [c_{i2}, \overline{c}_{i2}]\). When he bids \(b_i = (\overline{c}_{i1}, \overline{c}_{i2})\), he does not win any item. However, if he decreases his bid on \(B_{ij}\), it may be possible that, at bid \(c_{ij} < \overline{c}_{i}\), he wins the bundle \(B_{ij}\). Now, at any lower bid than \(c_{ij}\) for bundle \(B_{ij}\), he continues to win \(B_{ij}\). Also, he being XOR minded, he cannot win both the bundles. Thus, the type set’s square can be partitioned into three regions, \(R_1, R_2,\) and \(R_3\) as shown in Figure 4.3. When his type is in region \(R_j, j = 1, 2\), he is asked to supply \(B_{ij}\) and when it is in \(R_3\) he is not in the list of winning agents. Now, except on the boundary between \(R_1\) and \(R_2\), the assumption (3.21) holds true. Hence, though we are not using (3.21) as a necessary condition, it is getting satisfied in optimization problem (3.22). Thus OCAX is an optimal combinatorial auction for the buyer in the presence of XOR minded bidders.
The Case when Regularity Assumption is not Satisfied

Though we do not solve the buyer’s problem of optimal mechanism design without the regularity assumption, we highlight some thoughts on this. If we can assume (3.21), then we can design an optimal auction very similar to OCAS, in the presence of XOR minded bidders. The challenge is, we cannot use (3.21) as a necessary condition nor can we assume it. However, it may happen that in an optimal auction, the condition (3.21) will hold.

3.6 Conclusion

In this chapter,

- we have stated and proved the necessary and sufficient condition for incentive compatible and individually rational multi-unit multi-item auctions in the presence of single minded, capacitated buyers.

- We have provided a blueprint of an optimal mechanism, OCAS, for a buyer seeking to procure multiple units of multiple items in the presence of single minded and capacitated sellers. We also have shown that the mechanism minimizes the cost subject to DSIC and IIR if the virtual cost functions satisfy the regularity assumption.

- We have designed an optimal auction VD-OCAS for two item, multi-unit combinatorial auction when the single minded bidders give volume discounts.
• When bidders are XOR minded, under certain regularity conditions, we have designed an optimal OCAX.
Chapter 4

Truthful Multi-Armed Bandit Mechanisms for Multi-Slot Sponsored Search Auctions

In pay-per click sponsored search auctions which are currently extensively used by search engines, the auction for a keyword involves a certain number of advertisers (say \( k \)) competing for available slots (say \( m \)) to display their ads. This auction is typically conducted for a number of rounds (say \( T \)). There are click probabilities \( \mu_{ij} \) associated with each agent-slot pairs. The goal of the search engine is to maximize social welfare of the advertisers, that is, the sum of values of the advertisers. The search engine does not know the true values advertisers have for a click to their respective ads and also does not know the click probabilities \( \mu_{ij} \)s. A key problem for the search engine therefore is to learn these click probabilities during the \( T \) rounds of the auction and also to ensure that the auction mechanism is truthful. Mechanisms for addressing such learning and incentives issues have recently been introduced and are aptly referred to as multi-armed-bandit (MAB) mechanisms. When \( m = 1 \), characterizations for truthful MAB mechanisms are available in the literature and it has been shown that the regret for such mechanisms will be \( \Theta(T^{2/3}) \). In this chapter, we seek to derive a characterization in the realistic but non-trivial general case when \( m > 1 \) and obtain several interesting results. Our contributions include: (1) When \( \mu_{ij} \)s are unconstrained, we prove that any truthful mechanism must satisfy strong pointwise monotonicity and show that the regret will be \( \Theta(T) \) for such mechanisms. (2) When the clicks on the ads follow a certain click precedence property, we show that weak pointwise monotonicity is necessary for MAB mechanisms to be truthful. (3) If the search engine has a certain coarse pre-estimate of \( \mu_{ij} \) values and wishes to update them during the course of the \( T \) rounds, we show that weak pointwise monotonicity and Type-I separatedness are necessary and weak pointwise monotonicity and Type-II separatedness are sufficient conditions for the MAB mechanisms to be truthful. (4) If the click probabilities are
separable into agent specific and slot specific terms, we provide a characterization of MAB mechanisms that are truthful in expectation. The work reported in this thesis is available as [51].

4.1 Introduction

In the last two decades, the use of Internet has increased exponentially. No wonder Internet advertising has become the most effective way of advertisements due to its outreach to billions of people. These web sites display advertisements of various business organizations at various places on their web sites. These ads may be in the banner ads format. In this format, an ad is a long thin strip of information which may be either static or may include a hyper-link to the advertiser’s web page. The advertiser pays an online company in return for space to display the banner ad on one or more of the on-line company’s web pages. Advertisers may pay email service providers for displaying their ads in e-mail newsletters, e-mail marketing campaigns, and other commercial e-mail communication. The importance of such ads can be noted by the fact that Google makes revenue more than any newspaper agency or any TV channel through Internet advertisements. For more about Internet advertising; see [52, 53, 54, 55].

4.1.1 Sponsored Search Auctions

The most popular web sites are search engines. The Internet users need a search engine to locate the desired web sites. The most popular search engines, viz Google, Yahoo!, Bing, and many more offer the search facility for free. Whenever a user searches any set of keywords on a search engine, along with the search results, called as organic results, the search engine displays advertisements related to those keywords on the right side of the organic results or at the top of the organic results (see Fig 4.1). Such ads carry a hyper-link to the advertiser’s web site, called sponsored link. The search engine charges the advertiser for displaying her ad. One model of payment is the pay per impression model in which the search engine charges the advertiser each time it displays her ad. The more prevalent model is pay-per-click. In the pay-per-click model, the search engine charges an advertiser for displaying her ad only if a user clicks on her ad.

Typically, there will be more advertisers than slots available for displaying ads. Considering the competition for the sponsored links, the search engine goes for an auction mechanism. The decision regarding which ads are to be displayed and their respective order is based on the bids submitted by the advertisers indicating the maximum amount they are willing to pay per click. So for each keyword, there is an auction that takes place in the background. These auctions are called sponsored search auctions or pay-per click auctions. A variety of auction mechanisms exist for sponsored search auctions. Each auction mechanism has an allocation rule and a
The allocation rule determines which advertisements will be chosen to appear in the available slots and the actual allocation of slots to advertisers. The payment rule determines the amount that an allocated advertiser will pay each time the corresponding slot is clicked by a user.

The mechanism that was first used is called the generalized first price (GFP) mechanism. The advertising slots are allocated to the advertisers in descending order of their bids. If two advertisers place the same bid, then the tie is broken by an appropriate rule. Every time a user clicks on a sponsored link, an advertiser’s account is automatically billed the amount of the advertiser’s bid.

Most search engines currently use a mechanism that is based on the generalized second price (GSP) mechanism. In the GSP auction mechanism, every time a user clicks on a sponsored link, an advertiser’s account is automatically billed the amount of the advertiser’s bid who is just below him in the ranking of the displayed advertisements plus a minimum increment (typically $0.01). Many other mechanisms have also been discussed in the literature. For more details on the sponsored search auctions, the reader is referred to [13].

To perform any optimizations, such as maximizing social welfare or maximizing revenue to the search engine, the true valuations of the advertisers are needed. Being rational, the advertisers may actually manipulate their bids and therefore a primary goal of the search engine...
is to design an auction for which it is in the best interest of each advertiser to bid truthfully irrespective of the bids of the other advertisers. Such an auction is said to be dominant strategy incentive compatible (DSIC), or truthful.

4.1.2 Multi-Armed Bandit Mechanisms

The click on a displayed ad by the end user is a random event. The probability of the ad getting clicked depends upon the advertiser as well as where the ad is displayed. These click probabilities or clickthrough rates (CTRs), play a crucial role in these auctions. Given an agent $i$ and a slot $j$, the click probability $\mu_{ij}$ is the probability with which the ad of an agent $i$ will be clicked if the ad appears in the slot $j$. If the search engine knows the CTRs, then its problem is only to design a truthful auction. However, the search engine may not know the CTRs beforehand. Thus the problem of the search engine is twofold: (1) learn the CTR values (2) design a truthful auction.

Typically, the same set of agents compete for the given set of keywords. The search engine can exploit this fact to learn the CTRs by initially displaying ads by various advertisers. Also it is reasonable to assume that the advertisers may not revise their bids frequently. If the advertisers were bidding true values, the search engine’s problem would have been the same as that of a multi-armed bandit (MAB) problem [3] for learning the CTRs. Since the agents may not report their true values, the problem of the search engine can be described as one of designing an incentive compatible MAB mechanism. In the initial rounds, the search engine displays advertisements from all the agents to learn the CTRs. This phase is referred to as exploration phase. Then it uses the information gained in these rounds to maximize the social welfare. The latter phase is referred to as exploitation. The search engine will invariably lose a part of social welfare for the exploration phase. The difference between the social welfare the search engine would have achieved with the knowledge of CTRs and the actual social welfare achieved by a MAB mechanism is referred to as regret. Thus, regret analysis is also important while designing an MAB mechanism.

4.1.3 Related Work

The problem where the decision maker has to optimize his total reward based on gained information as well as gain knowledge about the available rewards is referred to as Multi-Armed Bandit (MAB) problem. The MAB problem was first studied by Robbins [3] in 1952. After his seminal work, MAB problems have been extensively studied for regret analysis and convergence rates. Readers are referred to [56] for regret analysis in finite time MAB problems. However, when a mechanism designer has to consider strategic behavior of the agents, these bounds on regret would not work. Recently, Babaioff, Sharma, and Slivkins [5] have derived a characterization for
truthful MAB mechanisms in the context of pay-per-click sponsored search auctions if there is only a single slot for each keyword. They have shown that any truthful MAB mechanism must have at least $\Omega(T^{2/3})$ worst case regret and also proposed a mechanism that achieves this regret. Here $T$ indicates the number of rounds for which the auction is conducted for a given keyword, with the same set of agents involved.

Devanur and Kakade \[6\] have also addressed the problem of designing truthful MAB mechanisms for pay-per-click auctions with a single sponsored slot. Though they have not explicitly attempted a characterization of truthful MAB mechanisms, they have derived similar results on payments as in \[5\]. They have also obtained a bound on regret of a MAB mechanism to be $\Theta(T^{2/3})$. Note that the regret in \[6\] is regret in the revenue to the search engine, as against regret analysis in \[5\] which is for social welfare of the advertisers. In this chapter, unless explicitly stated, when we refer to regret, we mean loss in the social welfare as compared to the social welfare that could have been obtained with the known CTRs.

In both of the above papers, only a single slot for advertisements is considered. Though the results in both papers are very important, in the real world sponsored search auctions typically there are multiple slots available for displaying the ads. Generalization of their work to the more realistic case of multiple sponsored slots is non-trivial and our chapter seeks to fill this research gap.

Prior to the above two papers, Gonen and Pavlov \[57\] had addressed the issue of unknown CTRs in multiple slot sponsored search auctions and proposed a specific mechanism. Their claim that their mechanism is truthful in expectation has been contested by \[5, 6\]. Also Gonen and Pavlov do not provide any characterization for truthful multi-slot MAB mechanisms.

The notion of truthfulness used in the papers \[5, 6\] is a dominant strategy incentive compatibility. Zoeter \[58\] considers the learning of CTRs when agents can reincarnate themselves and considers the weaker notion of incentive compatibility, namely Bayesian incentive compatibility, to optimally learn the CTRs and avoid reincarnation by the advertisers with a new identity. However, they also consider a single slot case and do not perform any regret analysis.

### 4.1.4 Our Contributions

In this chapter, we extend the results of Babaioff, Sharma, and Slivkins \[2\] and Devanur and Kakade \[6\] to the general case of two or more sponsored slots. The precise question we address is: which MAB mechanisms for multi-slot pay-per-click sponsored search auctions are dominant strategy incentive compatible? We describe our specific contributions below.

In the first and most general setting (Section 4.3.1), we assume no knowledge of click through rate ($\mu_{ij}$) values or any relationships among $\mu_{ij}$ values. We refer to this setting as the “unknown and unconstrained CTR” setting. Here we show that any truthful mechanism must satisfy a
highly restrictive property which we refer to as *strong pointwise monotonicity* property. We show that all mechanisms satisfying this property will however exhibit a high regret, which is $\Theta(T)$. This immediately motivates our remaining Sections 4.3.2, 4.3.3, and 4.3.4, where we explore the following variants of the general setting to obtain more specific characterizations.

First, in Section 4.3.2, we consider a setting where the realization is restricted according to a property that we call the *Higher Slot Click Precedence* property (a click in a lower slot will automatically imply that a click is received if the same ad is shown in any higher slot). For this setting, we provide a weaker necessary condition than strong pointwise monotonicity. Finding a necessary and sufficient condition however remains open.

In Section 4.3.3, we provide a complete characterization of MAB mechanisms which are *truthful in expectation* under a stochastic setting where a coarse estimate of $\mu_{ij}$ is known to the auctioneer and to the agent $i$, perhaps from some database of past auctions. Under this setting, the auctioneer updates his database of $\mu_{ij}$ values based on the observed clicks, thereby improving his estimate and maximizing revenue.

Finally, in Section 4.3.4, we derive a complete characterization of truthful multi-slot MAB mechanisms for a stochastic setting where we assume that the $\mu_{ij}$s are separable into agent-dependent and slot-dependent parts. Here, unlike the previous setting, we do not assume existence of any information on agent-dependent click probabilities.

For all the above multi-slot sponsored search auction settings, we show that the slot allocation in truthful mechanisms must satisfy some notion of monotonicity with respect to the agents’ bids and a certain weak separation between exploration and exploitation. Our results are summarized in Table 4.1.

Our approach and line of attack in this chapter follow that of [5] where the authors use the notions of *pointwise monotonicity*, *weakly separatedness*, and *exploration separatedness* quite critically in characterizing truthfulness. Since our chapter deals with the general problem of which theirs is a special case, these notions continue to play an important role in our chapter. However, there are some notable differences as explained below. We generalize their notion of *pointwise monotonicity* in two ways. The first notion we refer to as strong pointwise monotonicity and the second one as weak pointwise monotonicity. In addition to this, we introduce the key notions of *influential set*, *i-influentiality* and *strongly influential*. We use these new notions to define a non-trivial generalization of their notion of *weakly separatedness*, into two types of separatedness, Type-I and Type-II separated. The characterization of truthful mechanisms for a single parameter was provided by [59, 2]. For deriving payments to be assigned to the agents for truthful implementation, we use the approach in [59, 2].

In Section 4.4, we provide some simple experimental results on regret analysis. We conclude the chapter in Section 4.5.
<table>
<thead>
<tr>
<th>Number of Slots ((m))</th>
<th>Nature of the learning parameter (\text{CTR})</th>
<th>Solution Concept</th>
<th>Allocation rule</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m = 1) ([ \geq 1])</td>
<td>Unrestricted</td>
<td>DSIC</td>
<td>Pointwise monotone and exploration separated</td>
<td>(\Theta(T^{2/3}))</td>
</tr>
<tr>
<td>(m &gt; 1)</td>
<td>Unrestricted</td>
<td>DSIC</td>
<td>Strongly pointwise monotone and Type-I separated</td>
<td>(\Theta(T))</td>
</tr>
<tr>
<td>Higher Slot Click Precedence</td>
<td>DSIC</td>
<td>Weakly pointwise monotone and Type-I separated (necessary condition)</td>
<td>regret analysis not carried out</td>
<td></td>
</tr>
<tr>
<td>CTR Pre-estimates available</td>
<td>Truthful in expectation</td>
<td>Weakly pointwise monotone and Type-I separated (necessary) and Type-II separated (sufficient)</td>
<td>regret analysis not carried out</td>
<td></td>
</tr>
<tr>
<td>Separable CTR</td>
<td>Truthful in expectation</td>
<td>Weakly pointwise monotone and Type-I separated (necessary) Type-II Separated (sufficient)</td>
<td>(\Omega(T^{2/3})) (experimental evidence)</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: MAB Results

4.2 System Setup and Notation

In the auction considered, there are \(k\) agents and \(m\) ad slots \((k \geq m)\). Each agent has a single advertisement that she wants to display and a private value \(v_i\) which is her value per click on the ad. The auctioneer, that is the search engine, wishes to distribute the ads among these slots. This scenario is captured in Figure 4.2. The advertisements have certain click probabilities which depend upon the agent as well as the slot with which the agent is associated. Let \(\mu_{ij}\) be the probability of an ad of an agent \(i\) receiving click in slot \(j\). Now, the goal of the search engine...
Figure 4.2: A sponsored search auction scenario

is to assign these agents to the slots in a way that the social welfare, which is the total value received by the bidders, is maximized. However, there are two problems, (i) the search engine does not know $v_i$, the valuations of the agents and (ii) the search engine may not know the click probabilities $\mu_{ij}$.

So, the goal of the search engine is: (i) to design a DSIC auction in which it is in the agents’ interest to bid their true values, $v_i$’s (ii) to estimate $\mu_{ij}$. We consider multi-round auctions, where the search engine displays the various advertisements repeatedly over a large number of rounds. The mechanism uses the initial rounds in an explorative fashion to learn $\mu_{ij}$ and then uses the other rounds exploitatively to gain value.

The system works as follows. At the start of the auction, each agent submits a sealed bid $b_i$. Based on this bid and the click information from previous rounds, the mechanism decides to allocate each ad slot to a particular agent and then displays the $m$ chosen ads. The users can now click on any number of these ads and this information gets registered by the mechanism for future rounds. At the end of $T$ rounds, depending on the bids submitted by the agents and the number of clicks received by each agent, the agents have to make a certain payment $P_i$ to the mechanism.

Note: $P_i$ and $C_i$ are functions of $b$ and $\rho$. Whenever the arguments are clear from the context,
we just refer to them as $P_i$ and $C_i$.

A mechanism can be formally defined as the tuple $(A, P)$ where $A$ is the allocation rule specifying the slot allocation and $P$ is the payment rule.

The important notation used in the chapter is summarized in Table 4.2. We now define the terms used in this chapter.

We assume $T > 1$ to avoid corner cases in the proofs.

### 4.2.1 Important Notions and Definitions

**Definition 4.1 (Realization $\rho$).** We define a realization $\rho$ as a vector $(\rho(1), \rho(2), \ldots, \rho(T))$ where $\rho(t) = [\rho_{ij}(t)]_{K \times M}$ is click information in round $t$. $\rho_{ij}(t) = 1$, if an agent $i$’s ad receives a click in slot $j$ in round $t$, else $0$.

It is to be noted that the mechanism observes only those $\rho_{ij}(t)$ where $A_{ij}(b, \rho, t) = 1$. Thus, $\rho$ is never completely known to the auctioneer even after execution of all $T$ rounds. Also, though we use $A_{ij}(b, \rho, t)$ to indicate whether agent $i$ is allocated in slot $j$, in round $t$, it cannot depend upon future realizations. To be more precise, it should be interpreted as $A_{ij}(b, \rho(1), \rho(2), \ldots, \rho(t-1), t)$.

**Definition 4.2 (Clickwise Monotonicity).** We call an allocation rule $A$ clickwise monotone if for each agent $i$, for a fixed $(b_{-i}, \rho)$, the number of clicks, $C_i(b_i, b_{-i}, \rho)$ is a non-decreasing function of $b_i$. That is, $\frac{dC_i(\cdot)}{db_i} \geq 0 \forall (b_{-i}, \rho)$.

**Definition 4.3 (Weak Pointwise Monotonicity).** We call an allocation rule weak pointwise monotone if, for each agent $i$, for any given $(b_{-i}, \rho)$, and bid $b_i^+ > b_i$, $A_{ij}((b_i, b_{-i}), \rho, t) = 1 \Rightarrow A_{ij'}((b_i^+, b_{-i}), \rho, t) = 1$ for some slot $j' \leq j$, $\forall t$.

---

**Example 4.1.** Consider a setting with 4 agents and 2 slots for advertisements for a particular keyword and $T = 1,000$. The click probabilities, CTRs, for each agent be decreasing. That is, for each agent, the lower slot has lower CTR. An allocation $A$ be as follows. For first 100 rounds the advertisements of the four agents are displayed in each slot in a round robin fashion. That is, in the first round, the ads of the agent 1 and 2 are shown in the slots 1 and 2 respectively followed by in the slots 2 and 1 in next round. Then the agents 3 and 4 for next 2 rounds and again 1 and 2 till 100 rounds. For the remaining 900 rounds, the advertisements are displayed that maximize the expected sum of valuations of the clicks, where expectation is taken over the estimated click probabilities. If any of the bidders increases her bid, she will be displayed in higher slot. Thus, the allocation rule $A$ described here is weakly pointwise monotone.
\( K = \{1, 2, \ldots, k\}, \text{Set of agents} \)
\( M = \{1, 2, \ldots, m\} \text{ Set of slots} \)
\( i \) Index of an agent, \( i = 1, 2, \ldots, k \)
\( j \) Index of a slot, \( j = 1, 2, \ldots, m \)
\( T \) Total number of rounds
\( t \) A particular round \( t \in \{1, 2, \ldots, T\} \)
\( \rho_{ij}(t) \) = 1 if agent \( i \) gets a click in slot \( j \) in round \( t \)
\( \rho(t) \) \( (\rho_{ij}(t))_{i \in K, j \in M} \)
\( \rho \) \( (\rho(1), \rho(2), \ldots, \rho(T)) \)
\( v_i \) Agent \( i \)'s valuation of a click to her ad
\( b_i \) Bid by agent \( i \)
\( b_{-i} \) Bid vector of bids of all the agents except \( i \)
\( b \) Bid vector, indicating bids of all the agents
\(=(b_i, b_{-i})=(b_1, b_2, \ldots, b_k)\)
\( A_{ij}(b, \rho, t) \) = 1 If an agent \( i \) is allocated slot \( j \) in round \( t \)
\( = 0 \) otherwise
\( A(b, \rho, t) \) \( (A_{ij}(b, \rho, t))_{i \in K, j \in M} \)
\( A(b, \rho) \) \( (A(b, \rho, 1), A(b, \rho, 2), \ldots, A(b, \rho, T)) \), Allocation rule
\( C_i(b, \rho) \) Total number of clicks obtained by an agent \( i \) in \( T \) rounds
\( P_i(b, \rho) \) Payment made by agent \( i \)
\( P(b, \rho) \) \( (P_1(\cdot), P_2(\cdot), \ldots, P_k(\cdot)) \), Payment rule
\( U_i(v_i, b, \rho) \) Utility of an agent \( i \) in \( T \) rounds
\( = v_i C_i(b, \rho) - P_i(b, \rho) \)
\( b_i^+ \) A real number \( > b_i \)
\( \alpha_i \) Click probability associated with agent \( i \)
\( \beta_j \) Click probability associated with slot \( j \)
\( \mu_{ij} \) The probability that an ad of an agent \( i \) receives click when the agent is allotted slot \( j \).
\( N(b, \rho, i, t) \) Set of slot agent pairs in round \( t \) that influence agent \( i \) in some future rounds
\( \text{CTR} \) Click Through Rate (Click Probability)
\( \text{DSIC} \) Dominant Strategy Incentive Compatible

| Table 4.2: Notation: Multi-Armed Bandit mechanisms |
Definition 4.4 (Influential Set). Given a bid vector, $b$, a realization $\rho$ and round $t$, an influential set $I(b, \rho, t)$ is the set of all agent-slot allocation pairs $(i, j)$, such that (i) $A_{ij}(b, \rho, t) = 1$ and (ii) a change in $\rho_{ij}(t)$ will result in a change in the allocation in some future round. $t$ is referred to as an influential round. Agent $i$ is referred to as an influential agent and $j$ as influential slot w.r.t round $t$.

As the mechanism learns CTRs while executing repeated auctions, it updates CTRs based on the observed clicks. As the mechanism updates CTRs, some of the allocations may change. The observed clicks which change some future allocation, are said to influence some future allocation. Agent-slot pair $(i, j)$ is influential in round $t$, if $A_{ij}(b, \rho, t) = 1$ and in some future round $t'$, the allocation is changed based on $\rho_{ij}(t) = 0$ or 1 if other observed clicks are same till $t'$.

Definition 4.5 ($i$-Influential Set). We define the $i$-influential set $N(b, \rho, i, t) \subseteq I(b, \rho, t)$ as the set of all influential agent-slot pairs $(i', j')$ such that change in $\rho_{i'j'}(t)$ will change the allocation of agent $i$ in some future round.

Definition 4.6 (Strongly Influential). We call a slot-agent pair $(i^*, j^*)$ strongly influential in round $t$ w.r.t. the realization $\rho(t)$, if changing the realization (toggling) in the bit $\rho_{i^*j^*}(t)$ changes the allocation in a future round. We call such a set $(i^*, j^*, t)$ strongly $i$-influential if one of its influenced agents is $i$.

Definition 4.7 (Type-I Separated). We call an allocation rule Type-I separated if for a given $(b_{-i}, \rho)$, if $N((b_i, b_{-i}), \rho, i, t)$ is $i$-influential set, then $\forall (i', j') \in N((b_{i}, b_{-i}), \rho, i, t)$, $A_{i'j'} = 1$ when the agent $i$ increases her bid to $b_i^+$. This means that when an agent $i$ increases her bid, while the other parameters are kept fixed, the allocation in the originally influential slots does not change.

Definition 4.8 (Type-II Separated). We call an allocation rule Type-II separated if for a given $(b_{-i}, \rho)$ and two bids of agent $i$, $b_i$ and $b_i^+ > b_i$, $N((b_i, b_{-i}), \rho, i, t) \subseteq N((b_i^+, b_{-i}), \rho, i, t)$. This means that when an agent $i$ increases her bid, while the other parameters are kept fixed, the allocation in the originally influential slots does not change and they remain influential. We will explain the above definitions with an example.

Example 4.2 (Weakly Separated Allocation Rule). Suppose there are four agents competing for displaying their ads in any of the two slots available for two rounds. An allocation rule $A$ is defined as, in the first round, the ad of the agent $i$ is displayed in the slot $i$, $i = 1, 2$. If any
of the ads receives a click, that ad is retained in round the 2. If the ad in slot \(i\) is not clicked and if \(b_i < b_{i+2}\), then in round 2, the ad of the agent \(i + 2\) is displayed in slot \(i\), else the original ad is retained. Now, assume, \(b_1 < b_3\) and \(b_2 < b_4\). Thus, if the agent 1 receives the click in the round 1, then she retains the slot in second round else she loses the slot. Thus, she is influencing herself in round 2. Similarly, agent 2 is influential for herself. Thus, when \(b_1 < b_3\) and \(b_2 < b_4\), \(I(b, 1) = \{(1, 1), (2, 2)\} \) and \(N(b, 1, 1) = \{(1, 1)\}\). If the agent 1 or 2 increases her bid, still in round 1, she retains her slot. So, \(A\) is Type-I separated. However, if \(b_1 > b_3\), then she is not influential for herself. Thus, \(A\) is not Type-II separated.

Note, Type-II separated rule is also a Type-I separated rule. We show that any truthful allocation rule must be Type-I separated. However, for sufficiency, we use Type-II separatedness property of an allocation rule.

We continue to use definitions of Normalized Mechanism and Non-degeneracy from [5].

**Definition 4.9** (Non-degeneracy [5]). An allocation rule is said to be non-degenerate if for any given realization \(\rho\) and bid profile \((b_i, b_{-i})\) there exists a finite interval \(X\) around \(b_i\) such that the allocation in all rounds is the same for any bid profile \((x, b_{-i})\) where \(x \in X\).

**Definition 4.10** (Normalized Mechanism [5]). A mechanism is said to be normalized if the payment rule is defined such that each agent \(i\) pays at most \(b_i\) for each click that she gets.

Note, the definitions 4.9 - 4.9 are the properties of an allocation rule while definition 4.10 is a property of a mechanism.

With these preliminaries, we are now ready to characterize truthful MAB mechanisms for various settings in the next section.

### 4.3 Characterization of Truthful MAB Mechanisms

Before stating our results, we prove a minor claim that we will use to develop our characterizations. We will use this claim implicitly in our proofs.

**Claim 4.1.** Given \((b, (\rho(1), \rho(2), \ldots, \rho(t - 1)))\), if \((i^*, j^*)\) is \(i\)-influential in round \(t\), then \(\exists \rho^*(t)\) such that \((i^*, j^*)\) is also strongly \(i\)-influential w.r.t. \(\rho^*(t)\) in round \(t\).

**Proof:**
Suppose the claim is false. Let the \(i\)-influential set of slots in round \(t\) be \(N(b, \rho, i, t) = \{(i^1, j^1), (i^2, j^2), \ldots, (i^l, j^l), (i^*, j^*)\}\). \(N(b, \rho, i, t) \neq \emptyset\) since it has at least one element \((i^*, j^*)\).
Since we have assumed our claim to be false, \((i^*, j^*)\) is not strongly \(i\)-influential for any realization \((\rho_{i^*j^*}(t), \rho_{j^*j^*}(t), \ldots, \rho_{j^*j^*}(t))\) or the allocation of agent \(i\) in future rounds is the same whether \(\rho_{i^*j^*}\) is 0 or 1 for every given \((\rho_{i^*j^*}(t), \rho_{j^*j^*}(t), \ldots, \rho_{j^*j^*}(t))\). This means that the allocation of agent \(i\) is the same in future rounds for all realizations \((\rho_{i^*j^*}(t), \rho_{j^*j^*}(t), \ldots, \rho_{j^*j^*}(t))\). But this contradicts the fact that \(\{(i^1, j^1), (i^2, j^2), \ldots, (i^l, j^l), (i^*, j^*)\}\) is the set of \(i\)-influential slot-agent pairs in round \(t\). This proves our claim.

In our characterization of truthfulness under various settings, we show that a truthful allocation rule \(A\) must be Type-I separated. Though the proofs look similar, there are subtle differences in each of the following subsections. In our proofs, we start with the assumption that a truthful allocation rule \(A\) is not Type-I separated. That is,

\[
\exists b_i < b_i^+, b_{-i}, \rho, t, t', (i^*, j^*) \in N((b_i, b_{-i}), \rho, t, i) \quad \text{with influenced round } t' \text{ and } A_{i^*j^*}((b_i^+, b_{-i}), \rho, t) = 0
\]

(4.1)

Subsequently, we show that this leads to a contradiction in each of the subsections, implying the necessity of Type-I separatedness.

### 4.3.1 Unknown and Unconstrained CTRs

In this setting, we do not assume any previous knowledge of the CTRs although we do assume that such CTRs exist. Here, we show that any mechanism that is truthful under such a setting must follow some very rigid restrictions on its allocation rule.

**Definition 4.11** (Strong Pointwise Monotonicity). An allocation rule is said to be strongly pointwise monotone if it satisfies: For any fixed \((b_{-i}, \rho)\), if an agent \(i\) with bid \(b_i\) is allocated a slot \(j\) in round \(t\), then \(\forall b_i^+ > b_i\), she is allocated the same slot \(j\) in round \(t\). That is if the agent \(i\) receives a slot in round \(t\), then she receives the same slot for any higher bid. For any lower bid, either she may receive the same slot or may loose the impression.

A strongly pointwise monotone allocation rule is also a weakly pointwise monotone. However, strongly pointwise monotone is much stronger notion.

**Theorem 4.1.** Let \((A, P)\) be a deterministic, non-degenerate mechanism for the MAB, multi-slot sponsored search auction, with unconstrained and unknown \(\mu_{ij}\). Then, mechanism \((A, P)\) is DSIC iff \(A\) is strongly pointwise monotone and Type-I separated. Further, the payment scheme is given by,

\[
P_i((b_i, b_{-i}), \rho) = b_iC_i((b_i, b_{-i}), \rho) - \int_0^{b_i} C_i((x, b_{-i}), \rho)dx.
\]
Proof:

The proof is organized as follows. In step 1, we show the necessity of the payment structure. In step 2, we show the necessity of strong pointwise monotonicity. As there are similarities in the proof of necessity of Type-I separatedness in this theorem as well as in Proposition 4.1, we defer the proof to Appendix. Finally in step 3, we prove that the above payment scheme in conjunction with strong pointwise monotonicity and Type-I separatedness imply that the mechanism is DSIC.

Step 1: The utility structure for each agent $i \in N$ is

$$U_i(v_i, (b_i, b_{-i}), \rho) = v_i C_i((b_i, b_{-i}), \rho) - P_i((b_i, b_{-i}), \rho)$$

The mechanism is DSIC iff it is the best response for each agent to bid truthfully. That is, by bidding truthfully, each agent's utility is maximized. Under non-degenerate allocation rule assumption, by Myerson' theorem for single parameter agents, we need $\frac{dC_i}{db_i} \geq 0$, which is the clickwise monotonicity condition.

And for $(A, P)$ to be DSIC, we need

$$P_i((b_i, b_{-i}), \rho) = b_i C_i((b_i, b_{-i}), \rho) - \int_0^{b_i} C_i((x, b_{-i}), \rho)dx + P_i((0, b_{-i}), \rho) \quad \text{and} \quad \frac{dC_i}{db_i} \geq 0 \quad \forall (b_i, b_{-i}, \rho) \quad (4.2)$$

For a mechanism to be normalized we need $P_i((0, b_{-i}), \rho) = 0$. And hence the necessity of the payment structure specified in the theorem.

Step 2: We first prove the necessity of strong pointwise monotonicity by contradiction. We have seen from (1) that $\frac{dC_i}{db_i} \geq 0 \quad \forall (b_i, b_{-i}, \rho)$ is necessary for DSIC of $A$. We show that if $A$ is not strongly pointwise monotone, then there exists some allocation and realization $\rho$ for which $\frac{dC_i}{db_i} < 0$. If $A$ is not strongly pointwise monotone, there exists $(b_i, b_i^+, b_{-i}, \rho, t) \ni A_{ij_1}((b_i, b_{-i}), \rho, t) = 1$ and $A_{ij_2}((b_i^+, b_{-i}), \rho, t) = 1$

where $j_1 \neq j_2 \quad (4.3)$

Over all such counter-examples, choose the one with the minimum $t$. By this choice, we ensure that in this example $\forall t' < t$, we have $A_{ij}(b_i, t') = A_{ij}(b_i^+, t')$. The only difference occurs in round $t$. Now, consider the game instance where $\rho_{ij_1}(t) = 1$, $\rho_{ij_2}(t) = 0$, $\rho_{ij}(\tau) = 0 \forall \tau > t$. The occurrence of such $\rho$ has non-zero probability. Now, under $(b_{-i}, \rho)$, agent $i$ has the same allocation and the same number of clicks until round $(t - 1)$ independent of whether she bids $b_i$ or $b_i^+$. However, in round $t$ with bid $b_i$, she receives a click and with bid $b_i^+$ she does not,
implying for this case that $\frac{dC_i}{db_i} < 0$. This violates the click monotonicity requirement. So, strong pointwise monotonicity is indeed a necessary condition for truthful implementation of MAB mechanisms under this setting.

Step 3: Finally, we show that strong pointwise monotonicity and Type-I separatedness are sufficient conditions for clickwise monotonicity and computability of the payments and hence for truthfulness. Suppose $A$ is a strongly pointwise monotone and Type-I separated allocation rule. So, it clearly satisfies the clickwise monotonicity. Now, by the Type-I separatedness condition, we already have all the information required to calculate the allocation of agent $i$ in every round for every bid $x < b_i$. This is because for bids $x < b_i$, by the strong pointwise monotonicity condition, in each round $t$, either agent $i$ keeps the same slot she had in the observed game instance $((b_i, b_{-i}), \rho)$ or loses the impression altogether, that is, does not get a click. Hence, we have all the information required to compute $P_i((b_i, b_{-i}), \rho)$ as per Equation 4.2. This completes the sufficiency part of the theorem.

**Implications of Strong Pointwise Monotonicity**

For a given round $t$, if an agent $i$ is allocated a slot $j$, then by the definition of strong pointwise monotonicity she receives the same slot for any higher bid that she places. If she lowers her bid, she may either retain the slot $j$, or lose the impression entirely. This leads to the strong restriction that an agent’s bid can only decide whether or not she obtains an impression, and not which slot she actually gets. As we shall show below, this restriction has serious implications on the regret incurred by any truthful mechanism.

**Regret Estimate**

In the single slot case it is a known result that the worst case regret is $\Theta(T^{2/3})$ [5]. So, for the multi-slot case, the regret is $\Omega(T^{2/3})$. We show here that the worst case regret generated in the multi-slot general setting by a truthful mechanism is in fact $\Theta(T)$. We show this for the 2 slot, 3 agent case with an intuitive argument, which can be generalized.

Consider a setting with two slots and three competing agents, that is $m = 2$, $k = 3$. Let the agents be $A_1$, $A_2$ and $A_3$. By Theorem 4.1, any truthful mechanism has to be strongly pointwise monotone. That is, in any round, the bids of the agents only determine which agents will be displayed and not the slots they obtain.

Suppose, $A_3$’s bid $b_3 < \min(b_1, b_2)$ in addition to having low CTRs. In this case, any mechanism that grants $A_3$ an impression $\Theta(T)$ times, will have regret $\Theta(T)$.

So, we can assume that $A_3$’s ad gets an impression for a very small number of times when compared with $T$. Thus, ads by $A_1$ and $A_2$ will appear $\Theta(T)$ times. In each round, $A_1$ will get either slot 1 or slot 2 independent of her bid, while the other slot is assigned to $A_2$. 115
In any strongly pointwise monotone mechanism, either $A_1$ is assigned a slot 1 $\Theta(T)$ times or slot 2.

Without loss of generality, we assume that $A_1$ is assigned slot 1 $\Theta(T)$ times. So, the allocation (slot 1, slot 2)↔($A_1$,$A_2$) is made $\Theta(T)$ times. Consider a game instance where this is not the welfare maximizing assignment, that is, the relation $(\mu_{11}b_1 + \mu_{22}b_2) < (\mu_{12}b_1 + \mu_{21}b_2)$ holds true. Since the slot allocation does not depend on the individual bids, such an instance can occur. In such a setting ($A_2$, $A_1$) would have been optimal assignment. As a result, each round having the allocation ($A_1$, $A_2$) incurs constant non-zero regret. Since such an allocation occurs $\Theta(T)$ times, the mechanism has a worst case regret of $\Omega(T)$. However, there are $T$ rounds, the regret is $O(T)$. Hence any truthful mechanisms under the unrestricted CTR setting exhibit a high $\Theta(T)$ regret.

Since the strong monotonicity condition places such a severe restriction on $A$ and also leads to a very high regret, in the following sections we explore some relaxations on the assumption that $\mu_{ij}$’s are unrelated. With such settings which are in fact practically quite meaningful, we are able to prove more encouraging results.

4.3.2 Higher Slot Click Precedence

This setting is similar to the general one discussed above in that we do not assume any knowledge about the CTRs. However, we impose a restriction on the realization $\rho$ that it follows higher slot click precedence defined below.

Definition 4.12. A realization $\rho$ is said to follow Higher Slot Click Precedence if $\forall i \in K, \forall t = 1, 2, \ldots, T$,

$$\rho_{ij_1}(t) = 1 \Rightarrow \rho_{ij_2}(t) = 1 \forall j_2 < j_1$$

Higher slot click precedence implies that if an agent $i$ obtains a click in slot $j_1$ in round $t$, then in that round, she receives a click in any higher slot $j_2$. This assumption is in general valid in the real world since any given user (fixed by round $t$) who clicks on a particular ad when it is displayed in a lower slot would definitely click on the same ad if it was shown in a higher slot.

We show, under this setting, that weak pointwise monotonicity and Type-I separatedness are necessary conditions for truthfulness. They are, however, not sufficient conditions. Clearly, strong pointwise monotonicity and Type-I separatedness will still be sufficient conditions. A weaker sufficient condition for truthfulness under this setting is still elusive.

Implications of the Assumption

Observe that a slot-agent pair $(i, j)$ is influential in some round $t$ only if changing the realization in the entry $\rho_{i,j}(t)$ for some realization $\rho$ results in a change in allocation in some future round.
Crucial to the influentiality is the fact that $\rho_{ij}(t)$ can change.

Now, consider the following situation: it has been observed that in the game instance $((b_i, b_{-i}), \rho)$, we have $\rho_{ij}(t) = 0$ where agent $i$ obtains slot $j_1$ in round $t$. We are interested in the game instance $((x, b_{-i}), \rho)$ where agent $i$ gets slot $j_2 > j_1$ where $x < b_i$ and in knowing whether $(i, j_2)$ is an influential pair in round $t$ for some influenced agent. Now, since $\rho_{ij_1} = 0$ and $j_1 < j_2$, by our defining assumption, we conclude that $\rho_{ij_2}(t) = 0 \forall x < b_i$. Hence, our mechanism knows that in all the relevant cases, the realization in the given slot-agent pair never changes. Hence, $(i, j_2)$ cannot be an influential pair for any $j_2 > j_1$ in round $t$. We will use this observation in the proof of necessity characterization.

**Proposition 4.1.** Consider the setting in which realization $\rho$ follows Higher Slot Click Precedence. Let $(A, P)$ be a deterministic non-degenerate DSIC mechanism for this setting. Then the allocation rule $A$ must be weak pointwise monotone and Type-I separated. Further, the payment scheme is given by,

$$P_i(b_i, b_{-i}; \rho) = b_iC_i(b_i, b_{-i}; \rho) - \int_0^{b_i} C_i(x, b_{-i}; \rho)dx$$

**Proof:**

The proof for the payment scheme is identical to that in Theorem 4.1. Here, we prove the necessity of weak pointwise monotonicity and for Type-I separatedness, we defer it to Appendix.

We prove the necessity of weak pointwise monotonicity, in a very similar fashion to that of the necessity of strong pointwise monotonicity in Theorem 4.1. The crucial difference is, while constructing $\rho$, we have to ensure that it satisfies the higher order click precedence. Suppose $A$ is truthful but not weakly pointwise monotone, that is, $\exists (b_i, b_{i+}, b_{-i}, \rho, t)$ and $A_{ij_1}((b_i, b_{-i}), \rho, t) = 1$ and $A_{ij_2}((b_i^{+}, b_{-i}), \rho, t) = 1$ for some $j_1 < j_2$. Over all such examples, choose the one with the least $t$. By this choice, we ensure that in this example, $\forall t' < t$, we have $A_{ij}(b_i, t') = A_{ij}(b_i^{+}, t')$. The only difference occurs in round $t$. Now, consider the game instance where $\rho_{ij_1}(t) = 1$ and $\rho_{ij_2}(t) = 0$. Such realization has a non-zero probability of occurrence. Now, under $(b_{-i}, \rho)$, agent $i$ gets the same allocation and the same number of clicks until round $(t - 1)$ independent of whether she bids $b_i$ or $b_{i+}$. However, in round $t$ with bid $b_i$ she gets a click and with bid $b_i^{+}$ she does not, implying for this case that $\frac{dC_i}{db_i} < 0$. This leads to a contradiction. So, weak pointwise monotonicity is a necessary condition. If $A$ is not strongly pointwise monotone, does not violate clickwise monotonicity. That is, for truthful $A$, it may possible that, $A_{ij_1}((b_i, b_{-i}), \rho, t) = 1$ and $A_{ij_2}((b_i^{+}, b_{-i}), \rho, t) = 1$ where $j_2 < j_1$. Thus, for $A$ to be truthful, strong pointwise monotonicity may not be necessary.
4.3.3 When CTR Pre-estimates are Available

In this setting, we assume the existence of some previous database or pre-estimate of CTR values but no restriction on $\rho$. That is, $\mu_{ij} = \frac{X_{ij}}{Y_{ij}}$ where $X_{ij}$ is the number of clicks obtained by agent $i$ in slot $j$ out of the $Y_{ij}$ times she obtained the slot $j$ over all past auctions. Here, in general, $\mu_{i1} \geq \mu_{i2} \geq \ldots \geq \mu_{im}$. For our characterization, we assume that each $\mu_i = (\mu_{i1}, \mu_{i2}, \ldots, \mu_{im})$ is known to the agent $i$ and the auctioneer.

In this setting, the auctioneer uses explorative rounds to improve his estimate of the CTRs and updates the database. Then, he makes use of his new knowledge of the CTRs in the exploitative rounds. The payment scheme, however, only makes use of the old CTR matrix. Under this scheme, we derive the conditions required for a mechanism to be truthful in expectation over $\mu$, defined as follows.

**Definition 4.13 (Truthful in Expectation).** A mechanism is said to be truthful in expectation over $\mu$, the CTR pre-estimate, if each of the agents believes that the number of clicks she obtains is indeed $\sum_t \sum_j (\mu_{ij} A_{ij}(b, \rho, t))$, which is the number of clicks she will obtain if the CTR pre-estimate is perfectly accurate.

### Fairness

For this characterization, we need the notion of fair allocation rules, as defined below.

**Definition 4.14 (Fair Allocation).** Consider two game instances $((b_i, b_{-i}), \rho)$ and $((b_i', b_{-i}), \rho)$ having the same slot-agent-round triplets, $(i', j', t')$ as strongly $i$-influential. Let $(i^*, j^*, t)$ be such triplet with the smallest $t'$ in which $i$ is influenced. Consider the realization $\rho'$ differing from $\rho$ only in this influential element $\rho_{i^*,j^*}(t)$. Then, the allocation rule $A$ is said to be fair if for every such pair of games it happens that

$$\sum_j \mu_{ij} A_{ij}((b_i, b_{-i}), \rho, t') \geq \sum_j \mu_{ij} A_{ij}((b_i, b_{-i}), \rho', t') \iff \sum_j \mu_{ij} A_{ij}((b_i', b_{-i}), \rho, t') \geq \sum_j \mu_{ij} A_{ij}((b_i', b_{-i}), \rho', t')$$

The intuition behind fair allocations is that changing the realization only in a fixed strongly $i$-influential slot generally changes agent $i$’s allocation in a predictable fashion independent of her own bid, either improving her slot or worsening it in the earliest influenced round, irrespective of the allocation or realization in the rest of the game. For example, if agent $i$’s chief competitor agent, $i'$, is strongly $i$-influential, then $i'$ not getting a click in the influential round will generally mean that agent $i$ will go on to get a better slot than if agent $i'$ got a click, independent of $b_i$. 
Truthfulness Characterization

Here, the expected utility for the agent $i$, $U_i(v_i, b, \rho) = \left( \sum_{t=1}^{T} \sum_{j=1}^{m} \mu_{ij} A_{ij}(b, \rho, t)v_i \right) - P_i(b, \mu)$ (4.4)

Proposition 4.2. Let $(A, P)$ be a normalized mechanism under this setting. Then, the mechanism is truthful in expectation over $\mu$ iff $A$ is weakly pointwise monotone and the payment rule is given by

\[ P_i(b, \mu) = \sum_{t=1}^{T} \sum_{j=1}^{m} \mu_{ij} \{b_i A_{ij}(b, \mu, t) - \int_{0}^{b_i} A_{ij}(x, b_{-i}, \mu, t)dx\} \]

and payments are computable.

Proof:

In Step 1, we prove the necessity and sufficiency of the payment structure. For the mechanism to be implemented, we need to compute the payments of all the agents uniquely. That is, $P_i$s need to be computable for all agents $i$. In Step 2, we show weak pointwise monotonicity is equivalent to the second order condition which is clickwise monotonicity in the context of this chapter.

Step 1: The expected utility of an agent $i$ is given by (4.4).

Under non-degeneracy, by Myerson’s theorem we get, the $(A, P)$ to be truthful, the payment structure should be,

\[ P_i(b, \mu) = \sum_{t=1}^{T} \sum_{j=1}^{m} \mu_{ij} \{b_i A_{ij}(b, \mu, t) - \int_{0}^{b_i} A_{ij}(x, b_{-i}, \mu, t)dx\} \]

and

\[ \forall i, \sum_{t} \sum_{j} \mu_{ij} \frac{dA_{ij}}{db_i} \geq 0 \] (4.5)

Step 2: We show, (4.3) ⇔ weak pointwise monotonicity.

(i) It is obvious that weak pointwise monotonicity $\Rightarrow \sum_{t} \sum_{j} \mu_{ij} \frac{dA_{ij}}{db_i} \geq 0$. An increase in $b_i$ under a weakly pointwise monotone $A$ would result in a better slot allocation for agent $i$. This in turn, would result in an increase in $\sum_{j} \mu_{ij} A_{ij}$ in each round.

(ii) Now we prove the converse. Suppose $A$ is not weakly pointwise monotone. That is, $\exists i, b_i, b_i^+, b_{-i}, \rho, \mu, t \ni A_{ij}(b_i, b_{-i}, \rho, \mu, t) = 1$ and $A_{ij'}(b_i^+, b_{-i}, \rho, \mu, t) = 1$ where $j' > j$. Consider

\footnote{Note, the characterization in this section would hold even if $\mu_{ij}$ are arbitrary weights. However, while using arbitrary weights, mechanism may charge some agents more than their actual willingness to pay. Also regret in the revenue, that is loss in the revenue to the search engine will be trivially $\Theta(T)$.}
the smallest such \( t \). Allocation in this round does not depend upon the realization of this round or of future rounds. We consider the instance of the game where \( \rho_{ij}(t) = 1 \) and \( \rho_{ij'}(t) = 0 \) and \( t \) is the last round. Such an instance has a non-zero probability and for this instance, \( \sum_t \frac{d}{db_i} \sum_j \mu_{ij} A_{ij} < 0 \). This proves the equivalence claim.

Note, it is crucial that each \( \mu_{ij} \) is a previously known constant and cannot be defined as \( \mu_{ij} = X_{ij}/Y_{ij} \) based on the clicks in the current \( T \) rounds post facto. If we do so, \( X_{ij}/Y_{ij} \) can change with the allocation of agent \( i \) in a particular game and hence, \( \mu_{ij} \) would become a function of \( b_i \) and the mechanism would be no longer truthful.

For truthful implementation, the payments need to be computable and computing the payments may involve the unobserved part of \( \rho \). In the next theorem, we show that Type-I separatedness is necessary and Type-II separatedness is sufficient for computation of these payments. So, along with the computation of payments and the above proposition, we get,

**Theorem 4.2.** Let \((A, P)\) be a mechanism for this stochastic multi-round auction setting where \( A \) is a non-degenerate, deterministic and fair allocation rule. Then, \((A, P)\) is truthful in expectation over \( \mu \) if \( A \) is weakly pointwise monotone and Type-II separated and the payment scheme is given by,

\[
P_i(b, \rho) = \sum_{t=1}^{T} \sum_{j=1}^{m} \mu_{ij} \{b_i A_{ij}(b, \rho, t) - \int_{0}^{b_i} A_{ij}(x, b_{-i}, \rho, t) dx\}
\]

Also, if a mechanism \((A, P)\) is truthful, then it is weakly pointwise monotone, Type-I separated, the payment is given as above and is computable.

**Proof.**

This setting/characterization works best with old advertisers who have already taken part in a large number of auctions. As we already have proved Proposition 4.2, we just need to show that Type-I separatedness is in fact a necessary and Type-II separatedness is sufficient condition for the computability of payments, that is, for each agent \( i \), computability of \( \sum_{j=1}^{m} \mu_{ij} \int_{0}^{b_i} A_{ij}(x, b_{-i}, \rho, t) dx \).

Step 1: We first provide the proof for the sufficiency of Type-II separatedness. Suppose \( A \) is Type-II separated. The mechanism observes and knows all allocations and the observed realization for the game instance carried out with the original bid vector \((b_i, b_{-i})\). Specifically, it knows \( N((b_i, b_{-i}), i, \rho, t) \) for all rounds \( t \) and the respective realizations in these slots. Now, in the game instance \((x, b_{-i})\) where \( x \leq b_i \), by Type-II separatedness, we have \( N((x, b_{-i}), i, \rho, t) \subseteq \)
We prove the necessity of Type-I separatedness by contradiction. That is, we assume \( N((b_i, b_{-i}), i, \rho, t) \). This means that the allocation in \( i \)-influential slots for game instance \(((x, b_{-i})t)\) is a subset of that in observed game instance \(((b_i, b_{-i}), \rho)\). So, the mechanism already knows all the click information in the \( i \)-influential slots for the game instance \(((x, b_{-i}), \rho)\). Since the payment scheme is only interested in the allocation of agent \( i \), the realization in the unobserved slots is unimportant and can be assumed arbitrarily. Thus, the mechanism has complete information to compute \( P_i((b_i, b_{-i}), \rho, t) \).

**Step 2:** Next, we prove the necessity of Type-I separatedness by contradiction. That is, we assume (\( \mathcal{II} \)) is true. Consider a complete realization \( \rho(t) \) in round \( t \) for which \((i^*, j^*)\) is strongly \( i \)-influential (such a realization exists by our previous claim \( \mathcal{II} \)) and construct the two complete realizations \( \rho \) and \( \rho' \) from \((\rho(1), \rho(2), \ldots, \rho(t - 1), \rho(t))\) which only differ in \( \rho_{i^*j^*}(t)\). Over all choices of counter-examples \((b_i, t, \rho(t), i^*, j^*)\), we choose the one which has the smallest influenced round \( t' \). Now, we compare the payment that the mechanism has to make for this game instance at the end of \( t' \) rounds under the two different realizations \( \rho \) and \( \rho' \).

Let \( \varphi \in \{\rho, \rho'\} \). By the strong \( i \)-influence of \((i^*, j^*, t)\), the agent \( i \) gets different allocations in round \( t' \) under the different realizations \( \rho \) and \( \rho' \). This implies,

\[
\sum_j \mu_{ij} A_{ij}((b_i, b_{-i}), \rho, t') \neq \sum_j \mu_{ij} A_{ij}((b_i, b_{-i}), \rho', t').
\]

Without loss of generality,

\[
\sum_j \mu_{ij} A_{ij}((b_i, b_{-i}), \rho, t') > \sum_j \mu_{ij} A_{ij}((b_i, b_{-i}), \rho', t') \quad (4.6)
\]

(or agent \( i \) gets a higher slot under realization \( \rho \) than \( \rho' \)).

By the non-degeneracy of \( A \), there exists a finite interval of bids about \( b_i \) such that for every bid \( x \) in this interval,

\[
A_{ij}((x, b_{-i}), \varphi, t') = A_{ij}((b_i, b_{-i}), \varphi, t') \forall j \quad (4.7)
\]

Suppose \( x' \in (0, b_i^+) \) is another bid such that the same slot-agent-round set \((i^*, j^*, t)\) is strongly \( i \)-influential with the same influenced round \( t' \) for the game \(((x', b_{-i}), \rho)\). Then by the fairness of \( A \),

\[
\sum_j \mu_{ij} A_{ij}((x', b_{-i}), \rho, t') \geq \sum_j \mu_{ij} A_{ij}((x', b_{-i}), \rho', t') \quad (4.8)
\]

From (\( \mathcal{II} \)), (\( \mathcal{II} \)), and (\( \mathcal{II} \)) and using the fact that \( t' \) the smallest influenced round that is strongly \( i \)-influenced by the bit \( \rho_{i^*j^*}(t) \) which is the only differing bit between \( \rho \) and \( \rho' \), we can see that \( \forall x' \in (0, b_i^+) \)

\[
\sum_j \mu_{ij} A_{ij}((x', b_{-i}), \rho, t') \geq \sum_j \mu_{ij} A_{ij}((x', b_{-i}), \rho', t') \quad (4.9)
\]
and \( \exists \) a finite interval \( X \) around bid \( b_i \) such that \( \forall x \in X \), we have,

\[
\sum_j \mu_{ij} A_{ij}((x, b_{-i}), \rho, t') > \sum_j \mu_{ij} A_{ij}((x, b_{-i}), \rho', t')
\]  

From equations (4.9) and (4.10), and the fact that agent \( i \)'s allocation is the same under both realizations \( \rho \) and \( \rho' \) until round \( t' \) (from smallest influenced round choice), we conclude that,

\[
\sum_{t'=1}^{t'} \int_0^{b_i^+} \sum_j \mu_{ij} A_{ij}((x, b_{-i}), \rho, t) dx > \sum_{t'=1}^{t'} \int_0^{b_i^+} \sum_j \mu_{ij} A_{ij}((x, b_{-i}), \rho', t) dx
\]

Additionally, we can assume that there are no clicks after round \( t' \). As a result, we have \( P_i((b_i^+, b_{-i}), \rho) \neq P_i((b_i^+, b_{-i}), \rho') \). However, the mechanism cannot distinguish between the two realizations \( \rho \) and \( \rho' \) as the only differing bit \( r_{i,j*}(t) \) is unobserved. Hence, the mechanism fails to assign a unique payment to agent \( i \). This is a consequence of our initial assumption (4.1). Thus if \( A \) is not Type-I separated the payments are not computable. This completes the proof. \( \blacksquare \)

### 4.3.4 When CTR is Separable

In the previous setting we assumed that some pre-estimate on the CTR matrix \( [\mu_{ij}] \) existed. In real world applications, however, it is very often the case that the slot-dependent probabilities are known while the agent dependent probabilities are unknown. To leverage this fact, we make a widely accepted assumption: we assume that the click probability due to the slot is independent of the click probability due to the agent. That is, we assume that \( \mu_{ij} = \alpha_i \beta_j \), where \( \alpha_i \) is the click probability associated with agent \( i \) and \( \beta_j \) is the click probability associated with slot \( j \). We also assume that the vector \( \beta = (\beta_1, \beta_2, \ldots, \beta_m) \) is common knowledge. In general \( \beta_1 \geq \beta_2 \ldots \geq \beta_m \).

Here, any mechanism will use the explorative rounds to try to learn the values of \( \alpha_i \) as accurately as possible.

Let \( Y_{ij} \) denote the number of times that agent \( i \) obtains the impression for slot \( j \), and \( X_{ij} \) denote the corresponding number of times she obtains a click. Then, we define \( \alpha'_i = \text{avg}_j \{ \frac{1}{\beta_j} X_{ij} \} \) and \( \mu'_{ij} = \alpha'_i \beta_j \).

In this section, we assume \( \frac{d\alpha'_i}{db_i} = 0 \) or that \( \alpha'_i \) does not change with bid \( b_i \). We are justified in making this assumption since \( \alpha'_i \) is a good estimate of \( \alpha_i \) which is independent of which slot agent \( i \) obtains how many times. By changing her bid \( b_i \), agent \( i \) can only alter her allocations which should not predictably or significantly affect \( \alpha' \). It is trivial to see that \( \frac{d\alpha'_i}{db_i} = 0 \Rightarrow \frac{d\mu'_{ij}}{db_i} = \frac{b_j d\alpha'_i}{db_i} = 0 \).

We model truthfulness based on the utility gained by each agent in expectation over this \( \mu'_{ij} \). That is, utility to an agent \( i \) is given by equation (4.4), with \( \mu \) being replaced by \( \mu' \). With
the above setup, it can easily be seen that truthfulness mechanisms under this setting have the same characterization as the truthful mechanisms with a pre-estimate of CTR.

**Theorem 4.3.** Let \((A, P)\) be a mechanism for the stochastic multi-round auction setting where \(A\) is a non-degenerate, deterministic and fair allocation rule. Then, \((A, P)\) is truthful in expectation over \(\mu'\) if \(A\) is weakly pointwise monotone and Type-II separated and the payment scheme is given by,

\[
P_t(b, \rho) = \sum_{i=1}^{m} \sum_{j=1}^{m} \mu'_{ij} \{b_i, A_{ij}(b, \rho, t) - \int_0^{b_i} A_{ij}(x, b_{-i}, \rho, t) dx\}
\]

Also, if a mechanism \((A, P)\) is truthful, then it is weakly pointwise monotone, Type-I separated, the payment is given as above and is computable.

**Proof:**
This theorem can be proven using similar arguments as used in the proof of Theorem 4.2, with \(\mu\) being replaced by \(\mu'\).

**4.4 Experimental Analysis**

Since the single slot setting is a special case of the multi-slot setting, we obtain \(\Omega(T^{2/3})\) as a lower bound for the regret incurred by a truthful multi-slot sponsored search mechanism.

We have characterized truthful MAB mechanisms in various settings in the previous section. However, we have not studied MAB mechanisms in multi-slot auctions for regret estimation in such mechanisms (except the \(\Theta(T)\) worst case bound we showed for the unconstrained case in Section 4.3.1). In this section, we present a brief experimental study on the regret of an truthful MAB mechanism for multi-slot sponsored search auction under separable CTR case.

For our study, we have picked a simple mechanism belonging to the separable CTR case. In the simulation, we displayed the agents in the available slots in a round robin fashion for the first \(T^{2/3}\) rounds. Then, we used the observed information on the clicks to estimate the \(\mu_{ij}\) values. The payments were computed as per Theorem 4.3.

We performed simulations for various \(T\) values with \(k = 4\) and \(m = 2\). For a fixed \(T\), we generated \(100T\) different instances, and estimated the average case as well as worst case regrets. In each instance, we generate CTRs and bids randomly. Figure 4.3 depicts \(\ln(\text{worst case regret})\) and \(\ln(\text{average case regret})\). It is observed that \(\ln(\text{worst case regret})\) is closely approximated by \(\ln(T^{2/3})\) while \(\ln(\text{average case regret})\) is closely approximated by \(\ln(T^{2/3})\), clearly showing that the worst case regret is \(\Omega(T^{2/3})\) and the average case regret is upper bounded by \(O(T^{2/3})\).
4.5 Conclusion

In this chapter, we have provided characterizations for truthful multi-armed bandit mechanisms for various settings in the context of multi-slot pay-per-click auctions, thus generalizing the work of \cite{1, 2} in a non-trivial way. The first result we proved is a negative result which states that under the setting of unrestricted CTRs, any strategyproof allocation rule is necessarily strongly pointwise monotone. We also showed that every strategyproof mechanism in unrestricted CTR setting will have $\Theta(T)$ regret. By weakening the notion of unrestricted CTRs, we were able to derive a larger class of strategyproof allocation rules. Our results are summarized in Table 4.1.

In the auctions that we have considered, the auctioneer cannot vary the number of slots he wishes to display. One possible extension of this work could be in this direction, that is, the auctioneer can dynamically decide the number of slots for advertisements. We assume that the bidders bid their maximum willingness to pay at the start of the first round and they would not change their bids till $T$ rounds. Another possible extension would be to allow the agents to bid before every round. We could also be exploring the cases where the bidders have budget constraints.
Appendix: Proofs

We show the necessity of Type-I separatedness for Theorem 4.1 and Proposition 4.1. We prove the necessity of the Type-I separatedness condition by contradiction. That is we assume (4.1). Over all such possible counter-examples of \((b_i, b_i^+, b_{-i}, \rho, t, t')\), choose the one with the least \(t'\).

We can assume, \(\rho_{ij}(\tau) = 0 \forall \tau > t'\) as the clicks in future rounds do not affect decisions in the current round. Let \(\rho'\) be the realization that differs from \(\rho\) only in bit \(\rho_{i^*j^*}(t)\). We show that under these two realizations the payment charged to the agent \(i\) differ at bid \(b_i^+\). However, since \((i^*, j^*)\) is not part of the allocation in round \(t\) under the bid \(b_i^+\), the difference between \(\rho\) and \(\rho'\) is not observed by the mechanism.

Agent \(i\)'s allocation and click information differs only in round \(t'\) under the two different realizations \(\rho\) and \(\rho'\), by minimality of \(t'\). Now, for the proof of Theorem 4.1, by strong monotone property agent is either displayed in a particular slot, say \(j\) or not displayed at all. \((i^*, j^*)\) being \(i\)-influential, in round \(t'\) agent \(i\)'s allocation differs in \(\rho\) and \(\rho'\). Let,

\[
0 = A_{ij}((b_i, b_{-i}), \rho, t') < A_{ij}((b_i, b_{-i}), \rho', t') = 1
\]

Otherwise we can swap \(\rho\) and \(\rho'\). Now,

\[
A_{ij}((x, b_{-i}), \rho, t') \leq A_{ij}((x, b_{-i}), \rho', t') \forall x < b_i^+.
\]

(4.11)

If above claim is not true we can get violation of strong pointwise monotone necessity either at \(\rho\) or \(\rho'\). Because of non-degeneracy, there exists interval \(X\) such that

\[
A_{ij}((x, b_{-i}), \rho, t') < A_{ij}((x, b_{-i}), \rho', t') \forall x \in X.
\]

Hence \(P_i((b_i^+, b_{-i}), \rho) < P_i((b_i^+, b_{-i}), \rho)\) proving the claim.

Now, for the proof of Proposition 4.1, \((i^*, j^*)\) being \(i\)-influential, in round \(t'\), agent \(i\)'s allocation differs in \(\rho\) and \(\rho'\). Let

\[
A_{ij_1}((b_i, b_{-i}), \rho, t') = 1 \quad \text{and} \quad A_{ij_2}((b_i, b_{-i}), \rho', t') = 1
\]

Here since the two differ only in \(j_1 \neq j_2\), without loss of generality, let \(j_1 < j_2\) (or \(j_1\) be the better slot, since it is possible that one of the realizations may lead to no slot allocation).

Now, similar to observation in equation 4.11, we can argue that, for all \(x < b_i^+\), the agent \(i\) is allocated a better slot than \(j_2\) under realization \(\rho\). Due to higher slot click precedence and non-degeneracy of allocation rule, \(P_i((b_i^+, b_{-i}), \rho) < P_i((b_i^+, b_{-i}), \rho)\).

Thus we have seen that, if the allocation rule is not Type-I separated in the context of
either Theorem 4.1 or Proposition 4.1, the mechanism cannot uniquely determine the truthful payments for the agents. Hence, Type-I separatedness is necessary for truthful implementation of a clickwise monotone allocation rule.
Chapter 5

Redistribution Mechanisms for Assignment of Heterogeneous Objects

There are $p$ heterogeneous objects to be assigned to $n$ competing agents ($n > p$) each with unit demand. It is required to design a Groves mechanism for this assignment problem satisfying weak budget balance, individual rationality, and minimizing the budget imbalance. This calls for designing an appropriate rebate function. When the objects are identical, this problem has been solved by Moulin [8] and Guo and Conitzer [9]. We measure the performance of such mechanisms by redistribution index. We first show an impossibility theorem which rules out linear rebate functions with non-zero redistribution index in heterogeneous object assignment. Motivated by this theorem, we explore two approaches to get around this impossibility. In the first approach, we show that linear rebate functions with non-zero redistribution index are possible when the valuations for the objects have some relationship and design a mechanism with linear rebate function that is optimal on worst case analysis. In the second approach, we show that rebate functions with non-zero efficiency are possible if linearity is relaxed. We extend the rebate functions of [8] and [9] to heterogeneous objects assignment and conjecture them to be worst case optimal. The results reported in this chapter are available as [60, 61].

5.1 Introduction

Consider that $p$ resources are available and each of $n > p$ agents is interested in utilizing one of them. Naturally, we should assign these resources to those agents who value them the most. Since Vickery, Clarke and Groves mechanisms [19, 20, 21] have attractive game theoretic properties such as dominant strategy incentive compatibility (DSIC) and allocative efficiency (AE), Groves mechanisms are quite appealing to use in this context. However, in general, a
The Groves mechanism need not be budget balanced. That is, the total transfer of money in the system may not be zero. So the system will be left with a surplus or deficit. Using Clarke’s mechanism \cite{Clarke}, we can ensure under fairly weak conditions, that there is no deficit of money (that is the mechanism is weakly budget balanced). In such a case, the system or the auctioneer will be left with some money.

Often, the surplus money is not really needed in many social settings such as allocations by the Government among its departments, etc. Since strict budget balance cannot coexist with DSIC and AE (Green-Laffont theorem \cite{GreenLaffont}), we would like to redistribute the surplus to the participants as far as possible, preserving DSIC and AE. This idea was originally proposed by Laffont \cite{Laffont}. The total payment made by the mechanism as a redistribution will be referred to as the rebate to the agents.

In this chapter, we consider the following problem. There are $n$ agents and $p$ heterogeneous objects ($n \geq p > 1$). Each agent desires one object out of these $p$ objects. Each agent’s valuation for any of the objects is independent of his valuations for the other objects. Valuations of the different agents are also mutually independent. Our goal is to design a mechanism for assignment of the $p$ objects among the $n$ agents which is allocatively efficient, dominant strategy incentive compatible, and maximizes the rebate (which is equivalent to minimizing the budget imbalance). In addition, we would like the mechanism to satisfy feasibility and individual rationality. Thus, we seek to design a Groves mechanism for assigning $p$ heterogeneous objects among $n$ agents satisfying:

1. Feasibility (F) or weak budget balance. That is, the total payment to the agents should be less than or equal to the total received payment.

2. Individual Rationality (IR), which means that each agent’s utility by participating in the mechanism should be non-negative.


We call such a Groves mechanism that redistributes Clarke’s Payment as Groves redistribution mechanism or simply redistribution mechanism. Designing a redistribution mechanism involves the design of an appropriate rebate function. If in a redistribution mechanism, the rebate function for each agent is a linear function of the valuations of the remaining agents, we refer to such a mechanism as a linear redistribution mechanism (LRM). In many situations, design of an appropriate LRM reduces to a problem of solving a linear program.

Due to the Green-Laffont theorem \cite{GreenLaffont}, we cannot guarantee 100% redistribution at all type profiles. So a performance index for the redistribution mechanism would be the worst case redistribution. That is, the fraction of the surplus which is guaranteed to be redistributed...
irrespective of the bid profiles. This fraction will be referred to as redistribution index in the rest of the chapter. The advantage of worst case analysis is that, it does not require any distributional information on the type sets of the agents. It is desirable that the rebate function is deterministic and anonymous. A rebate function is said to be anonymous if two agents having the same bids get the same rebate. So, the aim is to design an anonymous, deterministic rebate function which maximizes the redistribution index and satisfies feasibility and individual rationality.

Our work in this chapter seeks to non-trivially extend the results of Moulin [8] and Guo and Conitzer [9] who have independently designed a Groves mechanism in order to redistribute the surplus when objects are identical (homogeneous objects case). Their mechanism is deterministic, anonymous, and has maximum redistribution index over all possible Groves redistribution mechanisms. We will refer to their mechanism as the worst case optimal (WCO) mechanism. The WCO Mechanism is a linear redistribution mechanisms. In this chapter, we concentrate on designing a linear redistribution mechanism for the heterogeneous objects case.

5.1.1 Relevant Work

As it is impossible to achieve allocative efficiency, DSIC, and strict budget balance simultaneously, we have to compromise on one of these properties. Faltings [63] and Guo and Conitzer [64] achieve budget balance by compromising on AE. If we are interested in preserving AE and DSIC, we have to settle for a non-zero surplus or a non-zero deficit of the money (budget imbalance) in the system. To reduce budget imbalance, various rebate functions have been designed by Bailey [65], Cavallo [66], Moulin [8], and Guo and Conitzer [9]. Moulin [8] and Guo and Conitzer [9] designed a Groves redistribution mechanism for assignment of $p$ homogeneous objects among $n > p$ agents with unit demand. Guo and Conitzer [9] generalize their work in [67] for multi-unit demand of identical items. In [67], Guo and Conitzer designed a redistribution mechanism which is optimal in the expected sense for the homogeneous objects setting. Thus, it will require some distributional information over the type sets of the agents. Clippel et al [68] use the idea of destroying some of the items to maximize the agents’ utilities. The preliminary version of the results presented here appeared in our papers [60] and [61].

5.1.2 Contributions and Outline

Our objective in this chapter is to design a Groves redistribution mechanism for assignment of heterogeneous objects with unit demand. To the best of our knowledge, this is the first attempt to design a redistribution mechanism for assignment of heterogeneous objects.

First, we investigate the question of existence of a linear rebate function for redistribution of surplus in assignment of heterogeneous objects. Our result shows that in general, when the
domain of valuations for each agent is $\mathbb{R}^p_+$, it is impossible to design a linear rebate function, with non-zero redistribution index, for the heterogeneous settings. However, we can relax the assumption of independence of valuations of different objects to get a linear rebate function with non-zero redistribution index. Another way to get around the impossibility theorem is to relax the linearity requirement of a rebate function. In particular, our contributions in this chapter can be summarized as follows.

- We first prove the impossibility of existence of a linear rebate function with non-zero redistribution index for the heterogeneous settings when the domain of valuations for each agent is $\mathbb{R}^p_+$ and the valuations for the objects are independent.

- When the objects are heterogeneous but the values for the objects of an agent can be derived from one single number, we design a Groves redistribution mechanism which is linear, anonymous, deterministic, feasible, individually rational, and efficient. In addition, the mechanism is worst case optimal with non-zero redistribution index.

- We show the existence of a non-linear rebate function that has non-zero redistribution index.

- We propose a mechanism, HETERO, which extends Moulin/WCO mechanism for heterogeneous settings. We conjecture HETERO to have non-zero redistribution index and to be worst case optimal.

The chapter is organized as follows. In Section 5.2, we introduce the notation followed in the chapter and describe some background work from the literature. We also explain WCO mechanism there. In Section 5.3, we state and prove the impossibility result. We derive an extension of the WCO mechanism for heterogeneous objects but with single dimensional private information in Section 5.4. The impossibility result does not rule out possibility of non-linear rebate functions with strictly positive redistribution index. We show this with a redistribution mechanism, BAILEY-CAVALLO, which is Bailey’s mechanism [65] applied to the settings under consideration in Section 5.5. We design another non-linear rebate function, namely HETERO. This rebate function matches with Moulin’s rebate function when the objects are identical. We describe the construction of HETERO in Section 5.5. We have performed simulation to provide empirical evidence for our conjecture regarding HETERO. The experimental setup and results are described in Section 5.6. We will conclude the chapter in Section 5.7 with the directions for future work. In out analysis, we need an ordering of the bids of the agents which we define in Appendix 1. The proofs of the lemmas in the chapter are presented in Appendix 2.
5.2 Preliminaries and Notation

5.2.1 The Model and Notation

The notation used is summarized in Table 5.1. Note that, where the context is clear, we will use $t, t_i, r_i, k,$ and $v_i$ to indicate $t(b), t_i(b), r_i(b), k(b),$ and $v_i(k(b))$ respectively. In this chapter, we assume that the payment made by agent $i$ is of the form $t_i(\cdot) - r_i(\cdot),$ where $t_i(\cdot)$ is agent $i$’s payment in the Clarke pivotal mechanism [20]. We refer to $\sum_i t_i,$ as the total Clarke payment or the surplus in the system.

In general, we assume there are $n$ agents and $p$ distinct objects. We also assume that, the allocation rule satisfies property, Allocative Efficiency (AE).

5.2.2 Important Definitions

Definition 5.1 (Redistribution Mechanism). We call a Groves mechanism as Groves redistribution mechanism or simply redistribution mechanism, if it allocates objects to the agents in an allocatively efficient way and redistributes the Clarke surplus in the system in the form of rebates to the agents such that the net payment made by each agent still follows Groves payment structure.

Definition 5.2 (Linear Rebate Function). We say a rebate to an agent is linear rebate function, if it is linear combination of bid vectors of all the remaining agents. Moreover, if a redistribution mechanism uses linear rebate functions for all the agents, we say the mechanism is linear redistribution mechanism.

Definition 5.3 (Redistribution Index). A redistribution index of a redistribution mechanism is defined to a worst case fraction of Clarke’s surplus that gets redistributed among the agents. That is,

$$e = \inf_{b: t(b) \neq 0} \frac{\sum r_i(b)}{t(b)}$$

5.2.3 Optimal Worst Case Redistribution when Objects are Identical

When the objects are identical, every agent $i$ has the same value for each object, call it $v_i$. Without loss of generality, we will assume, $v_1 \geq v_2 \geq \ldots \geq v_n$. In Clarke’s pivotal mechanism, the first $p$ agents will receive the objects and each of these $p$ agents will pay $v_{p+1}$. So, the surplus in the system is $pv_{p+1}$. For this situation, Moulin [8] and Guo and Conitzer [9] have independently designed a redistribution mechanism.

Guo and Conitzer [9] maximize the worst case fraction of the total surplus which gets redistributed. This mechanism is called the WCO mechanism. Moulin [8] minimizes the ratio of budget imbalance to the value of an optimal allocation, that is the value of an allocatively
Table 5.1: Notation: redistribution mechanisms

<table>
<thead>
<tr>
<th>n</th>
<th>Number of agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Set of the agents = {1, 2, \ldots, n}</td>
</tr>
<tr>
<td>p</td>
<td>Number of objects</td>
</tr>
<tr>
<td>i</td>
<td>Index for an agent, (i = 1, 2, \ldots, n)</td>
</tr>
<tr>
<td>j</td>
<td>Index for object, (j = 1, 2, \ldots, p)</td>
</tr>
<tr>
<td>(\mathbb{R}_+)</td>
<td>Set of positive real numbers</td>
</tr>
<tr>
<td>(\Theta_i)</td>
<td>The space of valuations of agent (i), (\Theta_i = \mathbb{R}_+^p)</td>
</tr>
<tr>
<td>(b_i)</td>
<td>Bid submitted by agent (i), (b_i = (b_{i1}, b_{i2}, \ldots, b_{ip}) \in \Theta_i)</td>
</tr>
<tr>
<td>(b)</td>
<td>((b_1, b_2, \ldots, b_n)), the bid vector</td>
</tr>
<tr>
<td>(K)</td>
<td>The set of all allocations of (p) objects to (n) agents, each getting at most one object</td>
</tr>
<tr>
<td>(k(b))</td>
<td>An allocation, (k(\cdot) \in K), corresponding to the bid profile (b)</td>
</tr>
<tr>
<td>(k^*(b))</td>
<td>An allocatively efficient allocation when the bid profile is (b)</td>
</tr>
<tr>
<td>(k^*_{-i}(b))</td>
<td>An allocatively efficient allocation when the bid profile is (b) and agent (i) is excluded from the system</td>
</tr>
<tr>
<td>(v_i(k(b)))</td>
<td>Valuation of the allocation (k) to the agent (i), when (b) is the bid profile</td>
</tr>
<tr>
<td>(v)</td>
<td>(v : K \rightarrow \mathbb{R}), the valuation function, (v(k(b)) = \sum_{i \in N} v_i(k(b)))</td>
</tr>
<tr>
<td>(t_i(b))</td>
<td>Payment made by agent (i) in the Clarke pivotal mechanism, when the bid profile is (b), (t_i(b) = v_i(k^<em>(b)) - (v(k^</em>(b)) - v(k^*_{-i}(b))))</td>
</tr>
<tr>
<td>(t(b))</td>
<td>The Clarke payment, that is, the total payment received from all the agents, (t(b) = \sum_{i \in N} t_i(b))</td>
</tr>
<tr>
<td>(t^{-1})</td>
<td>The Clarke payment received in the absence of the agent (i)</td>
</tr>
<tr>
<td>(r_i(b))</td>
<td>Rebate to agent (i) when bid profile is (b)</td>
</tr>
<tr>
<td>(e)</td>
<td>The redistribution index of the mechanism, (e = \inf_{b, t(b) \neq 0} \frac{\sum_{i \in N} r_i(b)}{t(b)})</td>
</tr>
</tbody>
</table>

efficient allocation. The WCO mechanism coincides with Moulin’s feasible and individually rational mechanism. Both the above mechanisms work as follows. After receiving bids from the agents, bids are sorted in decreasing order. The first \(p\) agents receive the objects. Each agent’s Clarke payment is calculated, say \(t_i\). Every agent \(i\) pays, \(p_i = t_i - r_i\), where, \(r_i\) is the rebate function for an agent \(i\).

\[
\begin{align*}
    r_i^{WCO} &= c_{p+1}v_{p+2} + c_{p+2}v_{p+3} + \ldots + c_{n-1}v_n & i = 1, \ldots, p + 1 \\
    r_i^{WCO} &= c_{p+1}v_{p+1} + \ldots + c_{i-1}v_{i-1} + c_iv_{i+1} + \ldots + c_{n-1}v_n & i = p + 2, \ldots, n
\end{align*}
\] (5.1)
where,

\[
    c_i = \frac{(-1)^{i+p-1}(n-p)}{i \binom{n-1}{i} \sum_{j=p}^{n-1} \binom{n-1}{j} \sum_{j=1}^{n-1} \binom{n-1}{j}} \sum_{j=p}^{n-1} \binom{n-1}{j} \left\{ \sum_{j=1}^{n-1} \binom{n-1}{j} \right\} ; \quad i = p+1, \ldots, n-1 \tag{5.2}
\]

Suppose \( y_1 \geq y_2 \geq \ldots \geq y_{n-1} \) are the bids of the \((n-1)\) agents excluding the agent \( i \), then equivalently the rebate to the agent \( i \) is given by,

\[
    r_i^{WCO} = \sum_{j=p+1}^{n-1} c_j y_j \tag{5.3}
\]

The redistribution index of this mechanism is \( e^* \), where \( e^* \) is given by,

\[
    e^* = 1 - \frac{\binom{n-1}{p}}{\sum_{j=p}^{n-1} \binom{n-1}{j}}
\]

This is an optimal mechanism, since there is no other mechanism which can guarantee more than \( e^* \) fraction redistribution in the worst case.

Before we proceed to present our impossibility theorem we state the following theorem by Guo and Conitzer [9] which will be used to design our mechanism.

**Theorem 5.1** ([9]). For any \( x_1 \geq x_2 \geq \ldots \geq x_n \geq 0 \),

\[
    a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \geq 0 \quad \text{iff} \quad \sum_{i=1}^{j} a_i \geq 0 \quad \forall \; j = 1, 2, \ldots, n
\]

### 5.3 Impossibility of Linear Rebate Function with Non-Zero Redistribution Index

We have just reviewed the design of a redistribution mechanism for homogeneous objects. We have seen that the WCO mechanism is a linear function of the types of agents. We now explore the general case. In the homogeneous case, the bids are real numbers which can be arranged in decreasing order. The Clarke surplus is a linear function of these ordered bids. For the heterogeneous scenario, this would not be the case. Each bid \( b_i \) belongs to \( \mathbb{R}_+^p \); hence, there is no unique way of defining an order among the bids. Moreover, the Clarke surplus is not a linear
function of the received bids in the heterogeneous case. So, we cannot expect any linear/affine rebate function of types to work well at all type profiles. We will prove this formally.

We first generalize a theorem due to Guo and Conitzer \[9\]. The context in which Guo and Conitzer \[9\] stated and proved the theorem is in the homogeneous setting. We show that this result holds true in the heterogeneous objects case also. The symbol \(\succ\) denotes the order over the bids of the agents, as defined in the Appendix.

Theorem 5.2. In Groves redistribution mechanism, any deterministic, anonymous rebate function \(f\) is DSIC iff,

\[
    r_i = f(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n) \forall i \in N \tag{5.4}
\]

where, \(v_1 \succ v_2 \succ \ldots \succ v_n\).

Proof:

- The “if” part: If \(r_i\) takes the form given by equation (5.4), then the rebate of agent \(i\) is independent of his valuation. The allocation rule satisfies allocative efficiency. So, the mechanism is still Groves and hence DSIC. The rebate function defined is deterministic. If two agents have the same bids, then, as per the ordering defined in Appendix, \(\succ\), they will have the same ranking. Suppose agents \(i\) and \(i+1\) have the same bids. Thus \(v_i \succ v_{i+1}\) and \(v_{i+1} \succ v_i\). So, \(r_i = f(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)\) and \(r_{i+1} = f(v_1, v_2, \ldots, v_i, v_{i+2}, \ldots, v_n)\). Since \(v_i = v_{i+1}\), \(r_i = r_{i+1}\). Thus the rebate function is anonymous.

- The “only if” part: For mechanism to be strategyproof, the rebate function for agent \(i\) should be independent of his bid. So, \(r_i\) should depend on only \(v_{-i}\). So, for deterministic rebate function, \(r_i = f_i(v_{-i})\). Now, we desire anonymous rebate function. That is, rebate should be independent of the identity of the agent. Thus, if \(v_i = v_j\), then \(r_i = r_j\). With out loss of generality, say \(v_i = v_{i+1}\), then \(v_{-i} = v_{-(i+1)}\). So, \(r_i = r_{i+1}\) implies, \(f_i = f_{i+1}\). Similarly \(f_{i+1} = f_{i+2}\) and so on. Thus, \(r_i = f(v_{-i}) \forall i \in N\).

We now state and prove the main result of this chapter.

Theorem 5.3. If a redistribution mechanism is feasible and individually rational, then there cannot exist a linear rebate function which satisfies all the following properties:

- DSIC
- deterministic
- anonymous
• non-zero redistribution index.

**Proof**: Assume that there exists a linear function, say $f$, which satisfies the above properties. Let $v_1 \succ v_2 \succ \ldots \succ v_n$. Then according to Theorem 5.2, for each agent $i$,

$$r_i = f(v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n) = (c_0, e_p) + (c_1, v_1) + \ldots + (c_{n-1}, v_n)$$

where, $c_i = (c_{i1}, c_{i2}, \ldots, c_{ip}) \in \mathbb{R}^p$, $e_p = (1, 1, \ldots, 1) \in \mathbb{R}^p$, and $(\cdot, \cdot)$ denotes the inner product of two vectors in $\mathbb{R}^p$. Now, we will show that the worst case performance of $f$ will be zero. To this end, we will study the structure of $f$, step by step.

**Observation 1**: Consider type profile $(v_1, v_2, \ldots, v_n)$ where $v_1 = v_2 = \ldots = v_n = (0, 0, \ldots, 0)$. For this type profile, the total Clarke surplus is zero and $r_i = (c_0, e_p) \forall i \in N$. Individual rationality implies,

$$(c_0, e_p) \geq 0 \quad (5.5)$$

Feasibility implies the total redistributed amount is less than the surplus, that is,

$$\sum_i r_i = n(c_0, e_p) \leq 0 \quad (5.6)$$

From, (5.5) and (5.6), it is easy to see that, $(c_0, e_p) = 0$.

**Observation 2**: Consider type profile $(v_1, v_2, \ldots, v_n)$ where $v_1 = (1, 0, 0, \ldots, 0)$ and $v_2 = \ldots, v_n = (0, 0, \ldots, 0)$. For this type profile, $r_1 = 0$ and if $i \neq 1$, $r_i = c_{11} \geq 0$ for individual rationality. For this type profile, it can be seen through straightforward calculations that the Clarke surplus is zero. Thus, for feasibility, $\sum_i r_i = (n - 1)c_{11} \leq t = 0$. This implies, $c_{11} = 0$.

In the above profile, by considering $v_1 = (0, 1, 0, \ldots, 0)$, we get $c_{12} = 0$. Similarly, one can show $c_{13} = c_{14} = \ldots = c_{1p} = 0$.

**Observation 3**: Continuing like above with, $v_1 = v_2 = \ldots = v_i = e_p$, and $v_{i+1} = (1, 0, \ldots, 0)$ or $(0, 1, 0, \ldots, 0)$, \ldots or $(0, 0, \ldots, 0, 1)$, we get, $c_{i+1} = (0, 0, \ldots, 0) \forall i \leq p - 1$. Thus,

$$r_i = \begin{cases} (c_{p+1}, v_{p+2}) + \ldots + (c_{n-1}, v_n) & : i \leq p + 1 \\ (c_{p+1}, v_{p+1}) + \ldots + (c_{i-1}, v_{i-1}) \\ +(c_i, v_{i+1}) + \ldots + (c_{n-1}, v_n) & : \text{otherwise} \end{cases} \quad (5.7)$$

Thus a rebate function in any linear redistribution mechanism has to be necessarily of the form in the Equation (5.7). We now claim that the redistribution index of such mechanism is
zero. For any individually rational redistribution mechanism, trivial lower bound on redistribution index is zero. We prove that in a linear redistribution mechanism, there exists a type profile, at which the fraction of the Clarke surplus that gets redistributed is zero. Consider the type profile:

\[ v_1 = (2p - 1, 2p - 2, \ldots, p + 1, p) \]
\[ v_2 = (2p - 2, 2p - 3, \ldots, p, p - 1) \]
\[ \vdots \]
\[ v_{p-1} = (p + 1, p, \ldots, 3, 2) \]
\[ v_p = (p, p - 1, \ldots, 2, 1) \]
\[ v_{p+1} = v_{p+2} = \ldots = v_n = (0, 0, \ldots, 0). \]

Now it can be seen through straightforward calculations of Clarke’s payment, with this type profile, agent 1 pays \((p - 1)\), agent 2 pays \((p - 2)\), …, agent \((p - 1)\) pays 1 and the remaining agents pay 0. Thus, the Clarke payment received is non-zero but it can be seen that \(r_i = 0\) for all the agents. Hence, redistribution index for any linear redistribution mechanism has to be zero.

The above theorem provides disappointing news. It rules out the possibility of a linear redistribution mechanism for the heterogeneous settings which will have non-zero redistribution index. However, there are two ways to get around it.

1. The domain of types under which Theorem 5.3 holds is, \(\Theta_i = \mathbb{R}_+^p, \forall i \in N\). One idea is to restrict the domain of types. In Section 5.4, we design a worst case optimal linear redistribution mechanism when the valuations of agents for the heterogeneous objects have a certain type of relationship.

2. Explore the existence of a rebate function which is not a linear and yields a non-zero performance. We explore this in Section 5.5.

It should be noted that our impossibility result holds true when we are defining a linear rebate functions as in Definition 5.2. Our result may not hold for other types of linearity. For example, sort bid components of other \((n - 1)\) agents and define rebate function to be linear combination of these \((n - 1)p\) elements. At this point, we have not explored such linear rebate functions.

### 5.4 A Redistribution Mechanism for Heterogeneous Objects when Valuations have a Scaling Based Relationship

Consider a scenario where the objects are not identical but the valuations for the objects are related and can be derived by a single parameter. As a motivating example, consider the website
somefreeads.com and assume that there are $p$ slots available for advertisements and there are $n$ agents interested in displaying their ads. Naturally, every agent will have a higher preference for a higher slot. Define click through rate of a slot as the number of times the ad is clicked, when the ad is displayed in that slot, divided by the number of impressions. Let the click through rates for slots be $\alpha_1 \geq \alpha_2 \geq \alpha_3 \ldots \geq \alpha_p$. Assume that each agent has the same value for each click by the user, say $v_i$. So, the agent’s value for the $j^{th}$ slot will be $\alpha_j v_i$. Let us use the phrase *valuations with scaling based relationship* to describe such valuations. We define this more formally below.

**Definition 5.4.** We say the valuations of the agents have scaling based relationship if there exist positive real numbers $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_p > 0$ such that, for each agent $i \in N$, the valuation for object $j$, say $\theta_{ij}$, is of the form $\theta_{ij} = \alpha_j v_i$, where $v_i \in \mathbb{R}^+$ is a private signal observed by agent $i$.

Without loss of generality, we assume, $\alpha_1 \geq \alpha_2 \geq \alpha_3 \ldots \geq \alpha_p > 0$. (For simplifying equations, we will assume that there are $(n - p)$ virtual objects, with $\alpha_{p+1} = \alpha_{p+2} = \ldots = \alpha_n = 0$). We immediately note that the homogeneous setting is a special case that arises when $\alpha_1 = \alpha_2 = \alpha_3 = \ldots = \alpha_p > 0$.

For the above setting, we design a Groves mechanism which is almost budget balanced and optimal in the worst case. Our mechanism is similar to that of Guo and Conitzer [9] and our proof uses the same line of arguments.

### 5.4.1 The Proposed Mechanism

We will use a linear rebate function. We propose the following mechanism:

- The agents submit their bids.
- The bids are sorted in decreasing order.
- The highest bidder will be allotted object 1, the second highest bidder will be allotted object 2, and so on.
- Agent $i$ will pay $t_i - r_i$, where $t_i$ is the Clarke payment and $r_i$ is the rebate.

$$t_i = \sum_{j=i}^{p} (\alpha_j - \alpha_{j+1})v_{j+1}$$

- Let agent $i$’s rebate be,

$$r_i = c_0 + c_1 v_1 + \ldots + c_{i-1} v_{i-1} + c_i v_{i+1} + \ldots + c_{n-1} v_n$$
\( c_i \)'s are defined as follows.

The mechanism is required to be individually rational and feasible.

- The mechanism will be individually rational iff \( r_i \geq 0 \ \forall \ i \in N \). That is, \( \forall i \in N \)
  \[
  c_0 + c_1 v_1 + \ldots + c_{i-1} v_{i-1} + c_i v_{i+1} + \ldots + c_{n-1} v_n \geq 0.
  \]

- The mechanism will be feasible if the total redistributed payment is less than or equal to the surplus. That is, \( \sum_i r_i \leq t = \sum_i t_i \) or \( t - \sum_i r_i \geq 0 \), where,
  \[
  t = \sum_{j=1}^p j(\alpha_j - \alpha_{j+1}) v_{j+1}.
  \]

With the above setup, we now derive \( c_0, c_1, \ldots, c_{n-1} \) that will maximize the fraction of the surplus which is redistributed among the agents.

**Step 1:** First, we claim that, \( c_0 = c_1 = 0 \). This can be proved as follows. Consider the type profile, \( v_1 = v_2 = \ldots = v_n = 0 \). For this type profile, individual rationality implies \( r_i = c_0 \geq 0 \) and \( t = 0 \). So for feasibility, \( \sum_i r_i = nc_0 \leq t = 0 \). That is, \( c_0 \) should be zero. Similarly, by considering type profile \( v_1 = 1, v_2 = \ldots = v_n = 0 \), we get \( c_1 = 0 \).

**Step 2:** Using \( c_0 = c_1 = 0 \),

- The feasibility condition can be written as:
  \[
  \sum_{j=2}^{n-1} \left\{ (j-1)(\alpha_{j-1} - \alpha_j) - (j-1)c_{j-1} - (n-j)c_j \right\} v_j - (n-1)c_{n-1} v_n \geq 0 \quad (5.8)
  \]

- The individual rationality condition can be written as
  \[
  c_2 v_2 + \ldots + c_{i-1} v_{i-1} + c_i v_{i+1} + \ldots + c_{n-1} v_n \geq 0 \quad (5.9)
  \]

**Step 3:** When we say our mechanism’s redistribution index is \( e \), we mean, \( \sum_i r_i \geq et \), that is,

\[
\sum_{j=2}^{n-1} \left( -e(j-1)(\alpha_{j-1} - \alpha_j) + (j-1)c_{j-1} + (n-j)c_j \right) v_j + (n-1)c_{n-1} v_n \geq 0 \quad (5.10)
\]

**Step 4:** Define \( \beta_1 = \alpha_1 - \alpha_2 \), and for \( i = 2, \ldots, n - 1 \), let \( \beta_i = i(\alpha_i - \alpha_{i+1}) + \beta_{i-1} \). Now, inequalities (5.8), (5.9), and (5.10) have to be satisfied for all values of \( v_1 \geq v_2 \geq \ldots \geq v_n \geq 0 \).
By Theorem (5.1), we need to satisfy the following set of inequalities:

\[
\sum_{i=2}^{j} c_i \geq 0 \quad \forall j = 2, \ldots, n - 1 \\
e \beta_1 \leq (n - 2)c_2 \leq \beta_1 \\
e \beta_{i-1} \leq n \sum_{j=2}^{i-1} c_j + (n - i)c_i \leq \beta_{i-1} \quad i = 3, \ldots, p \\
e \beta_p \leq n \sum_{j=2}^{i-1} c_j + (n - i)c_i \leq \beta_p \quad i = p + 1, \ldots, n - 1 \\
e \beta_p \leq n \sum_{j=2}^{n-1} c_j \leq \beta_p
\]

Now, the mechanism designer wishes to design a mechanism that maximizes \( e \) subject to the above constraints.

Define \( x_j = \sum_{i=2}^{j} c_i \) for \( j = 2, \ldots, n - 1 \). This is equivalent to solving the following linear program.

\[
\begin{align*}
\text{maximize} & \quad e \\
\text{s.t.} & \quad e \beta_1 \leq (n - 2)x_2 \leq \beta_1 \\
& \quad e \beta_{i-1} \leq ix_{i-1} + (n - i)x_i \leq \beta_{i-1} \quad i = 3, \ldots, p \\
& \quad e \beta_p \leq ix_{i-1} + (n - i)x_i \leq \beta_p \quad i = p + 1, \ldots, n - 1 \\
& \quad e \beta_p \leq nx_{n-1} \leq \beta_p \\
& \quad x_i \geq 0 \quad \forall i = 2, \ldots, n - 1
\end{align*}
\] (5.11)

So, given \( n \) and \( p \), the social planner will have to solve the above optimization problem and determine the optimal values of \( e, c_2, c_3, \ldots, c_{n-1} \). It would be of interest to derive a closed form solution for the above problem.

The discussion above can be summarized as the following theorem.

**Theorem 5.4.** When the valuations of the agents have scaling based relationship, for any \( p \) and \( n > p + 1 \), the linear redistribution mechanism obtained by solving LP (5.11) is worst case optimal among all Groves redistribution mechanisms that are feasible, individually rational, deterministic, and anonymous. This mechanism is an example of mechanism having non-zero redistribution index.

**Proof:**

The worst case optimality of the mechanism can be proved following the line of arguments of Guo and Conitzer [9].

As per the impossibility Theorem 5.3, there is no linear redistribution mechanism for general heterogeneous setting having non-zero efficiency. However, when objects have scaling based relation, the linear redistribution mechanism, that is obtained by solving LP (5.11) has non-zero
efficiency at-least for some \((n, p)\) instances. This is obtained by actually solving the LP using matlab for various \(n\) and \(p\). This certainly proves that, at least for \(n = 10, 12, 14, p = 2, 3, 4, \ldots, 8\) and when valuations have scaled based correlation, the worst case optimal mechanism given by the LP (5.11) has non-zero redistribution index. Now we prove the upper bound on the redistribution index of a redistribution mechanism LP (5.11).

**Claim 5.1.** If \(e^*\) is the solution of the LP (5.11), then

\[ e^* \leq \min \left\{ \frac{A}{B}, \frac{B}{A} \right\} \]

where, \(A = \sum_{i=1,3,5,\ldots}^{n} \beta_{i-1} \left( \frac{n}{i} \right) \) and \(B = \sum_{i=2,4,6,\ldots}^{n-1} \beta_{i-1} \left( \frac{n}{i} \right) \).

The LP (5.11) can be written as

maximize \(e\)

s.t.

\(e\beta \leq Mx \leq \beta\)

\(x \geq 0\)

where \(x = (x_2, x_3, \ldots, x_{n-1}) \in \mathbb{R}_+^{n-2}\) and \(\beta = (\beta_1, \beta_2, \ldots, \beta_p, \beta_p, \ldots, \beta_p) \in \mathbb{R}_+^{n-1}\) and

\[
M = \begin{bmatrix}
  n - 2 & 0 & 0 & \cdots & 0 \\
  3 & n - 3 & 0 & \cdots & 0 \\
  0 & 4 & n - 4 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & n - 1 & 1 \\
  0 & 0 & n \\
\end{bmatrix}
\]

Now, \(y = (y_1, y_2, \ldots, y_{n-1}) \in \mathbb{R}^{n-1}\) is in the range of \(M\) iff

\[
\binom{n}{2} y_1 + \binom{n}{4} y_3 + \cdots = \binom{n}{3} y_2 + \binom{n}{5} y_4 + \cdots
\] (5.12)

Now, \(e^* \beta_i \left( \frac{n}{i+1} \right) \leq \left( \frac{n}{i+1} \right) (Mx)_i \leq \left( \frac{n}{i+1} \right) \beta_i \forall i \in \{1, 2, 3, \ldots, n - 1\}\). Now summing these inequalities for odd \(i\)s and using (5.12), we get \(e^*A \leq B\) and summing over even \(i\)s we get \(e^*B \leq A\). and hence the claim.

We verified using matlab for \(n = 10, 12, 14\) and \(p = 2, 3, \ldots, 8\), the redistribution index of the proposed mechanism is in fact, \(e^* = \min \left\{ \frac{A}{B}, \frac{B}{A} \right\}\)
5.5 Non-linear Redistribution Mechanisms for the Heterogeneous Setting

We should note that the homogeneous objects case is a special case of the heterogeneous objects case in which each bidder submits the same bid for all objects. Thus, we cannot expect any redistribution mechanism to perform better than the homogeneous objects case. For \( n \leq p + 1 \), the worst case redistribution is zero for the homogeneous case and so will be for the heterogeneous case \([9, 8]\). So, we assume \( n > p + 1 \). In this Section we propose two redistribution mechanisms with non-linear rebate functions. We construct a redistribution scheme by applying the mechanism proposed by Bailey \([65]\) to the heterogeneous settings. We refer to this proposed mechanism on heterogeneous objects as \textit{BAILEY-CAVALLO} redistribution mechanism. It is crucial to note that the non-zero redistribution index of the BAILEY-CAVALLO mechanism does not trivially follow from that of the mechanism in \([65]\). We rewrite WCO mechanism and extend the rebate functions to heterogeneous objects settings. We call this mechanism as \textit{HETERO}.

In each of the mechanisms, namely BAILEY-CAVALLO and HETERO, the objects are assigned to those agents who value them most. The Clarke’s payments are collected from the agents and the surplus is redistributed among the agents according to the rebate functions defined in the mechanism. Hence, both are Groves redistribution mechanisms and hence DSIC.

As stated above, for \( n \leq (p + 1) \), redistribution index for any redistribution mechanism has to be zero. For the case \( n > p + 1 \), the redistribution index for any linear redistribution mechanism has to be zero (Theorem 5.3). We prove that for \( n \geq (2p + 1) \), BAILEY-CAVALLO has non-zero redistribution index. We only conjecture that HETERO is worst case optimal, that is no mechanism can have better redistribution index than HETERO as well we conjecture that its redistribution index is the same as WCO, which is non-zero when \( n > (p + 1) \). Thus, for \( n \in \{p + 2, p + 3, \ldots, 2p\} \), there is still no redistribution mechanism for which non-zero redistribution index is proved.

5.5.1 BAILEY-CAVALLO Mechanism

First, consider the case when \( p = 1 \). Let the valuations of the agents for the object be, \( v_1 \geq v_2 \geq \ldots \geq v_n \). The agent with the highest valuation will receive the object and would pay the second highest bid. Cavallo \([66]\) proposed the rebate function as,

\[
\begin{align*}
r_1 &= r_2 = \frac{1}{n}v_3 \\
r_i &= \frac{1}{n}v_2 \quad i > 2
\end{align*}
\]
Similar mechanism was independently proposed by Porter et al [19]. Motivated by this scheme, we propose a scheme for the heterogeneous setting. Suppose agent \( i \) is excluded from the system. Then let \( t^{-i} \) be the Clarke surplus in the system (defined in Table 5.1). Define,

\[
 r^B_i = \frac{1}{n} t^{-i} \quad \forall i \in N
\]

(5.13)

- As the Clarke surplus is always positive, \( r^B_i \geq 0 \) for all \( i \). Thus, this scheme satisfies individual rationality.

- \( t^{-i} \leq t \quad \forall i \) (revenue monotonicity). So, \( \sum_i r^B_i = \sum_i \frac{1}{n} t^{-i} \leq n \frac{1}{n} t = t \). Thus, this scheme is feasible.

We now show that the BAILEY-CAVALLO scheme has non-zero redistribution index if \( n \geq 2p + 1 \). First we state two lemmas. The proof will be given in Appendix 5.7. These lemmas are useful in designing redistribution mechanisms for the heterogeneous settings as well as in analysis of the mechanisms. Lemma 5.2 is used to show non-zero redistribution index of the BAILEY-CAVALLO mechanism. Lemma 5.1 is used to find an allocatively efficient outcome for the settings under consideration. Also, this lemma 5.1 is useful in determining the Clarke payments.

**Lemma 5.1.** If we sort the bids of all the agents for each object, then

1. An optimal allocation, that is an allocatively efficient allocation, will consist of the agents having bids among the \( p \) highest bids for each object.

2. Consider an optimal allocation \( k^* \). If any of the \( p \) agents receiving objects in \( k^* \) is dropped, then there always exists an allocation \( k^*_{-i} \) that is an optimal allocation (on the remaining \( n-1 \) agents) which allocates objects to the remaining \( (p-1) \) agents. The objects that these \( p-1 \) agents receive in \( k^*_{-i} \), may not however be the same as the objects they are allocated in \( k^* \).

**Lemma 5.2.** There are at most \( 2p \) agents involved in deciding the Clarke payment.

*Note:* When the objects are identical, the bids of \( (p+1) \) agents are involved in determining the Clarke payment.

Now, we show non-zero redistribution index of the BAILEY-CAVALLO redistribution mechanism.

**Theorem 5.5.** When there are sufficient number of agents, (that is \( n > 2p \)), the BAILEY-CAVALLO redistribution mechanism has non-zero redistribution index.
Proof: In Lemma 5.2, we have shown that there will be at most 2\(p\) agents involved in determining the Clarke surplus. Thus, given a type profile, there will be \((n - 2p)\) agents, for whom, \(t^{-i} = t\) and this implies that at least \(\frac{n-2p}{n}\) will be redistributed. That is the redistribution index of the mechanism is at least \(\frac{n-2p}{n} > 0\).

Note that the above mechanism may not be worst case optimal. As when objects are identical, WCO mechanism performs better on worst case analysis than the above mechanism. So, we suspect that in heterogeneous settings as well the above mechanism would not be optimal on the worst case analysis. In next subsection, we explore another rebate function, namely HETERO.

5.5.2 A Redistribution Mechanism for the Heterogeneous Setting - HETERO

We propose a redistribution mechanism. We will be referring to it as HETERO.

When the objects are identical, the WCO mechanism is given by equation (5.3). We give a novel interpretation to it. Consider the scenario in which one agent is absent from the scene. Then the Clarke’s payment received is either \(pv_{p+1}\) or \(pv_{p+2}\) depending upon which agent is absent. If we remove two agents, the surplus is \(pv_{p+1}\) or \(pv_{p+2}\) or \(pv_{p+3}\), depending upon which two agents are removed. Till \((n-p-1)\) agents are removed, we get non-zero surplus. If we remove \((n-p)\) or more agents from the system, there is no need for any mechanism for assignment of the objects. So, we will consider the cases when we remove \(k\) agents, where, \(1 \leq k < n-p\).

Now let \(t^{-i,k}\) be the average payment received when agent \(i\) is removed along with \(k\) other agents that is, a total of \((k+1)\) agents are removed comprising of \(i\). The average is taken over all possible selections of \(k\) agents from the remaining \((n-1)\) agents. We can rewrite the WCO mechanism in terms of \(t^{-i}, t^{-i,k}\). Observe that, \(t^{-i}, t^{-i,k}\) can be defined in heterogeneous settings as well. We propose to use a rebate function defined as,

\[
r_i^H = \alpha_1 t^{-i} + \sum_{k=2}^{k=n-p-1} \alpha_k t^{-i,k-1}
\]

where \(\alpha_k\) are the suitable weights assigned to the surplus generated when a total of \(k\) agents are removed from the system. By using different \(\alpha_k\)s, we get different mechanisms. However, we prefer to choose \(\alpha_k\)s as the following.

The Equivalence of HETERO and WCO when Objects are Identical

It is desirable that HETERO should match with the WCO mechanism when the objects are homogeneous. So we choose \(\alpha\)s in Equation (5.14) in a way ensures that when the objects are identical, at all type profiles \(r_i^H\) in equation (5.14) is equal to \(r_i^{WCO}\) in equation (5.3). Since the rebate is a function of the remaining \((n-1)\) bids, we can write it as, \(r_i = f(x_1, x_2, \ldots, x_{n-1})\)
where $x_1, x_2, \ldots, x_{n-1}$ are bids without the agent $i$, in decreasing order. Note, in this case, that each bidder will be submitting a bid $b_i \in \mathbb{R}_+$. Now, we can write, $t^{-i,k}$, $r^H_i$, and $r_i$ in terms of $x_1, x_2, \ldots, x_{n-1}$, as,

$$t^{-i,k-1} = \sum_{l=0}^{k-1} \binom{p+l}{p} \binom{n-p-2-l}{k-1-l} x_{p+1+l}$$

$$r^H_i = \sum_{k=1}^{k=n-p-1} \alpha_k t^{-i,k-1}$$

$$r^WCO_i = \sum_{l=0}^{n-p-1} c_{p+1+l} x_{p+1+l}$$

(5.15)

(5.16)

where, $c_i$, $i = p+1, p+2, \ldots, n-1$ are given by equation (5.2).

Consider the type profile $(x_1 = 1, x_2 = 1, \ldots, x_{p+1} = 1, x_{p+2} = 0, \ldots, x_{n-1} = 0)$. For HETERO to agree with WCO, the coefficients of $x_{p+1}$ in equation (5.15) and equation (5.16) should be the same. Now consider the type profile $(x_1 = 1, x_2 = 1, \ldots, x_{p+2} = 1, x_{p+3} = 0, \ldots, x_{n-1} = 0)$. As the coefficients of $x_{p+1}$ in equation (5.15) and equation (5.16) are the same, the coefficients of $x_{p+2}$ should also be equal in equation (5.15) and equation (5.16).

Thus, the coefficients of $x_{p+1}, x_{p+2}, \ldots, x_{n-1}$ in equation (5.16) and equation (5.17) should agree.

Let $L = n - p - 1$. Thus, for $i = p+1, \ldots, n-1$,

$$c_i = \sum_{k=0}^{n-i-1} \alpha_{L-k} \binom{i-1}{p} \binom{n-i-1}{k} \binom{n-1}{p+1+k}$$

(5.17)

The above system of equations yields, for $i = 1, 2, \ldots, L$,

$$\alpha_i = \frac{(-1)^{(i+1)(L-i)}p!}{(n-i)!} \binom{L-i}{j} \sum_{j=0}^{L-i} \chi \sum_{l=p+i+j}^{n-1} \binom{n-1}{l}$$

(5.18)

where $\chi$ is given by,

$$\chi = \frac{n-p}{\sum_{j=p}^{n-1} \binom{n-1}{j}}$$

144
As the HETERO mechanism matches with the WCO when objects are identical, the HETERO mechanism satisfies individual rationality and feasibility in the homogeneous case. These two properties, however, remain to be shown in the heterogeneous case.

Properties of HETERO

**Conjecture 5.1.** The HETERO mechanism satisfies individual rationality, feasibility, is worst case optimal, and has redistribution index same as WCO.

Intuition Behind Individual Rationality of HETERO

We have to show that for each agent \( i \), \( r_i^H \geq 0 \) at all type profiles. For convenience, we will assume \( i \) implicitly. So, say, \( r_i^H = r \) and \( \Gamma_1 = t^{-i} \), \( \Gamma_j = t^{-i(j-1)} \), \( j = 2, \ldots, L \). Now, the rebate is given by the equation, \( r = \sum_j \alpha_j \Gamma_j \). We have to show that \( r \geq 0 \). Note that, \( \Gamma_1 \geq \Gamma_2 \geq \ldots \geq \Gamma_L \geq 0 \). So, if \( \sum_{j=1}^{L} \alpha_i \geq 0 \) \( \forall \) \( j = 1 \to L \), individual rationality would follow from Theorem 5.1. We observe that, in general, this is not true. The important observation is, though \( \Gamma_i \)'s are decreasing positive real numbers, they are related. For example, we can show that if \( \Gamma_1 > 0 \), then \( \Gamma_2 > 0 \). In our experiments, which we describe in the next section, we keep track of \( \frac{\Gamma_2}{\Gamma_1} \). We observed that this ratio is in \([0, 1]\). For Theorem 5.1 to be applicable, this ratio can be \([0, 1]\).

Thus, though \( \alpha \)'s are alternately positive and negative, the relation among \( \Gamma \)'s would not make \( r \) to go negative and it will be within limits in such a fashion that total rebate to the agents will be less than or equal to total Clarke’s payment. It remains to show individual rationality analytically in the general case. However, we are only able to show in the following cases.

1. Consider the case when \( p = 2 \). (i). If \( n = 4 \), \( \alpha_1 = \frac{1}{4} \). (ii). If \( n = 5 \), \( \alpha_1 = 0.27273 \), \( \alpha_2 = -0.18182 \). (iii). If \( n = 6 \), \( \alpha_1 = 0.29487, \alpha_2 = -0.25641, \alpha_3 = 0.12821 \).

2. Consider the case when \( p = 3 \). (i). If \( n = 5 \), \( \alpha_1 = \frac{1}{5} \). (ii). If \( n = 6 \), \( \alpha_1 = 0.21875 \), \( \alpha_2 = -0.15625 \). (iii). If \( n = 7 \), \( \alpha_1 = 0.23810, \alpha_2 = -0.21429, \alpha_3 = 0.11905 \).

By Theorem 5.1, it follows that for the above cases, the proposed mechanism satisfies the individual rationality.

Feasibility and Worst Case Optimality of HETERO

Similarly, we also believe that, \( \alpha \)'s adjust rebate functions optimally such that, HETERO remains feasible and is worst case optimal and has same redistribution index as WCO. Though, we do not have analytical proof, we provide some empirical evidence for the conjecture in Section 5.6.
5.6 Experimental Analysis

5.6.1 Empirical Evidence for Individual Rationality of HETERO

Solving equations $(5.18)$ is a challenging task. Though the new mechanism is the extension of the Moulin or the WCO mechanism, yet, we are not able to prove individual rationality and feasibility of HETERO analytically. We therefore seek empirical evidence.

Simulation 1

We consider various combinations of $n$ and $p$. For each agent, and for each object, the valuation is generated as a uniform random variable in $[0, 100]$. We run our simulation for the following combinations of $n$ and $p$.

For $p = 2$, $n = 5, 6, \ldots, 14$, for $p = 3$, $n = 7, 8, \ldots, 14$ and for $p = 4$, $n = 9, 10, \ldots, 14$. For each combination of $n$ and $p = 2$, we generated randomly 100,000 bid profiles and evaluated our mechanism. We also kept track of the worst case performance of our mechanism over these 100,000 bid profiles. Our mechanism was feasible and individually rational in these 100,000 bid profiles. The redistribution index of our mechanism is upper bounded by that of the WCO mechanism. We observed that the worst case performance over these 100,000 random bid profiles was the same as that of WCO. This is a strong indication that our mechanism will perform well in general.

Simulation 2: Bidders with Binary Valuation

Suppose each bidder has valuation for each object, either 0 or 1. Then there are $2^{np}$ possible bid profiles. We ran an experiment to evaluate our mechanism with all possible bid profiles of agents with binary valuations. We considered $p = 2$ and $n = 5, 6, \ldots, 12$. We found that the mechanism is feasible, individually rational, and the worst case performance is the same as that of the WCO mechanism. Note, as indicated earlier, no mechanism can perform better than the WCO mechanism in the worst case. And our mechanism performs as well as the WCO. Thus, though there is no analytical proof with us, for binary valuation settings, for $p = 2$ and $n = 5, 6, \ldots, 12$, our mechanism is worst case optimal.

5.6.2 BAILEY-CAVALLO vs HETERO

In this subsection, we compare, the worst case redistribution index of BAILEY-CAVALLO vs worst case redistribution index of HETERO for varying number of objects when there are 10 agents in the system. That is, we study worst case redistribution index for various $p$ when $n = 10$. The worst case is taken over randomly generated 50K bid profiles. The comparison is depicted
in the figure 5.1. The redistribution index of WCO is an upper bound on any Redistribution Mechanism for heterogeneous settings. However, the simulations not being exhaustive, the worst case performance of the mechanisms can be better than WCO. Exact worst case may be worse than WCO. However, in simulations, we never encountered a situation where HETERO is worse than WCO. We can see from the figure 5.1, BAILEY-CAVALLO mechanism’s worst case performance is better than HETERO, for $p = 3, 4, 5, 6, 7$. This worst case is worst over the randomly generated 50K bid profiles in our simulations.

The other observation we made in our simulations is that most of the time (70%), BAILEY-CAVALLO redistributes more VCG surplus than HETERO ever though worst case performance is poorer than HETERO.

These observations also lead to a question that Cavallo [41] raised in the context of dynamic redistribution mechanisms. Do we really need highly sophisticated mechanism, that is worst case optimal, when simple mechanism perform quite well in general.

![Figure 5.1: Redistibution index vs number of objects when number of agents = 10](image-url)
5.7 Conclusion

We addressed the problem of assigning $p$ heterogeneous objects among $n > p$ competing agents. When the valuations of the agents are independent of each other and their valuations for each object are independent of valuations on the other objects, we proved the impossibility of the existence of a linear redistribution mechanism with non-zero redistribution index (Theorem 5.3). Then we explored two approaches to get around this impossibility.

- In the first approach, we showed that linear rebate functions with non-zero redistribution index are possible when the valuations for the objects have scaling based relationship. For these settings, we proposed a strategyproof linear redistribution mechanism that is optimal on worst case analysis, individually rational and feasible (Theorem 5.4).

- In the second approach, we relaxed linearity requirement. We showed that non-linear rebate functions with non-zero redistribution index are possible by applying BAILEY-CAVALLO mechanism to the settings (Theorem 5.5).

- We proposed a mechanism, namely HETERO, for general settings when the objects are heterogeneous and private values of an agent for these objects are independent of each other. The mechanism is deterministic, anonymous, and DSIC. The HETERO mechanism extends the Moulin /WCO mechanism. Though we have not analytically proved feasibility and individual rationality, we have sufficient empirical evidence to conjecture that our mechanism is feasible and individually rational (Conjecture 5.1).

It would be interesting to see if we can characterize the situations under which linear redistribution mechanisms with non-zero redistribution indices are possible for heterogeneous settings.

An interesting research direction is to investigate the individual rationality and feasibility for the proposed HETERO mechanism. Also, we strongly believe that the new mechanism is a worst case optimal. An immediate future direction is to prove this fact or design a mechanism which is worst case optimal.

Another interesting problem to explore is to characterize all redistribution mechanisms that are worst case optimal for heterogeneous settings.

APPENDIX 1: Ordering of the Agents Based on Bid Profiles

We will define a ranking among the agents. This ranking is used in proving a Theorem on rebate function. This theorem is similar to Cavallo’s theorem on characterization of DSIC, deterministic, anonymous rebate functions for homogeneous objects. We would not be actually
computing the order among the bidders. We will use this order for proving impossibility of the linear rebate function with the desired properties.

5.7.1 Properties of the Ranking System

When we are defining ranking/ordering among the agents, we expect the following properties to hold true:

- Any permutation of the objects and the corresponding permutation on bid vector \((b_{i1}, b_{i2}, \ldots, b_{ip})\) for each agent \(i\), should not change the ranking. That is, the ranking should be independent of the order in which the agents are expected to bid for this objects.
- Two bidders with the same bid vectors should have the same rank.
- By increasing the bid on any of the objects, the rank of an agent should not decrease.

5.7.2 Ranking among the Agents

This is a very crucial step. First, find out all feasible allocations of the \(p\) objects among the \(n\) agents, each agent receiving at most one object. Sort these allocations, according to the valuation of an allocation. Call this list \(L\). To find the ranking between \(i\) and \(j\), we uses the following algorithm.

1. \(L_{ij} = L\)

2. Delete all the allocations from \(L_{ij}\) which contain both \(i\) and \(j\).

3. Find out the first allocation in \(L_{ij}\) which contains one of the agent \(i\) or \(j\). Say \(k'\).

   3.1. Suppose this allocation contains \(i\) and has value strictly greater than any of remaining allocations from \(L_{ij}\) containing \(j\), then we say, \(i \succ j\).
   3.2. Suppose this allocation contains \(j\) and has value strictly greater than any of remaining allocations from \(L_{ij}\) containing \(i\), then we say, \(j \succ i\).

4. If the above step is not able to decide the ordering between \(i\) and \(j\), let \(A = \{k \in K | v(k) = v(k')\}\). Update \(L_{ij} = L_{ij} \setminus A\) and recur to step (2) till EITHER
   - there is no allocation containing the agent \(i\) or \(j\) OR
   - the ordering between \(i\) and \(j\) is decided.

5. If the above steps do not give either of \(i \succ j\) or \(j \succ i\), we say, \(i \equiv j\) or \(i \succeq j\) as well as \(j \succeq i\).
Before we state some properties of this ranking system $\succ$, we will explain it with an example. Let there be two items A and B, and four bidders. That is, $p = 2, n = 4$ and let their bids be:

$b_1 = (4, 5), b_2 = (2, 1), b_3 = (1, 4), \text{ and } b_4 = (1, 0)$.

Now, allocation $(A = 1, B = 3)$ has the highest valuation among all the allocations. So,

agent 1 $\succ$ agent 2
agent 1 $\succ$ agent 4
agent 3 $\succ$ agent 2
agent 3 $\succ$ agent 4

Now, in $L_{13}$ defined in the procedure above, the allocation $(A = 2, B = 1)$ has strictly higher value than any other allocation in which the agent 3 is present. So,

agent 1 $\succ$ agent 3.

Thus,

agent 1 $\succ$ agent 3 $\succ$ agent 2 and
agent 1 $\succ$ agent 3 $\succ$ agent 4

In $L_{24}$, the allocation $(A = 2, B = 1)$ has strictly higher value than any other allocation in which the agent 4 is present. Thus, the ranking of the agents is,

agent 1 $\succ$ agent 3 $\succ$ agent 2 $\succ$ agent 4

It can be seen that the ranking defined above, satisfies the following properties.

1. $\succ$ defines a total order on the set of bids.
2. $\succ$ is independent of the order of the objects.
3. If two bids are the same, then they are equivalent in this order.
4. By increasing a bid, no agent will decrease his rank.

If agent $i \succ$ agent $j$, we will also say $v_i \succeq v_j$.

APPENDIX 2: Some Proofs

5.7.3 Proof of Lemma 1

• Suppose an optimal allocation contains an agent whose bid for his winning object, say $j$, is not in the top $p$ bids for the $j^{th}$ object. There are other $(p - 1)$ winners in an optimal
allocation. So, there exists at least one agent whose bid is in the top \( p \) bids for the \( j^{th} \) object and does not win any object. Thus, allocating him the \( j^{th} \) object, we have an allocation which has higher valuation than the declared optimal allocation.

- Suppose an agent \( i \) who receives an object in an optimal allocation is removed from the system. The agent will have at most one bid in the top \( p \) bids for each object. So, agents now having bids in the top \( p \) bids, will be at the \( p^{th} \) position. It can be seen that there will be at most one agent in an optimal allocation who is on the \( p^{th} \) position for the object he wins. If there is more than one agent in an optimal allocation on the \( p^{th} \) position for the object they win, then we can improve on this allocation. Hence, after removing \( i \), there will be at most one more agent who will be a part of a new optimal allocation.

5.7.4 Proof of Lemma 2

The argument is as follows.

1. Sort the bids of the agents for each object.

2. The optimal allocation consists of agents having bids in the \( p \) highest bids for each of the objects (Lemma 5.1).

3. For computing the Clarke payment of the agent \( i \), we remove the agent and determine an optimal allocation. And, using his bid, the valuation of optimal allocation with him and without him will determine his payment. This is done for each agent \( i \). As per Lemma 5.1, if any agent from an optimal allocation is removed from the system, there exists a new optimal allocation which consists of at least \( (p - 1) \) agents who received the objects in the original optimal allocation.

4. There will be \( p \) agents receiving the objects and determining their payments will involve removing one of them at a time, there will be at most \( p \) more agents who will influence the payment. Thus, there are at most \( 2p \) agents involved in determining the Clarke payment.
In Part 1 of the thesis, we have addressed the allocation of heterogeneous objects among competing agents which are intelligent, rational, and willing to pay for receiving the preferable objects. Also, all the agents are present in the system while the allocation takes place. In Part 2 of the thesis, we address mechanism design problems in the context of dynamic agents without monetary transfers. In particular, we address two-sided matchings (Chapter 6), house allocation (Chapter 7), and allocation of objects to dynamically arriving agents (Chapter 8).
Chapter 6

Dynamic Stable Matching in Two-Sided Markets

We study dynamic matching without money when one side of the market is dynamic with arrivals and departures and the other is static and agents have strict preferences over agents on the other side of the market. In enabling stability properties, so that no pair of agents can usefully deviate from the match, we consider the use of a fall-back option where the dynamic agents can be matched, if needed, with a limited number of agents from a separate “reserve” pool. We introduce a mechanism GSODAS (Generalized Stable Online Deferred Acceptance with Substitutes), which is truthful for agents on the static side of the market and stable. In simulations, we establish that GSODAS outperforms in terms of rank-efficiency, a pair of randomized mechanisms that operate without the use of a fall-back option. In addition, we demonstrate good rank-efficiency in comparison to a non-truthful mechanism that employs online stochastic optimization. The work reported here is available as [71].

6.1 Introduction

For motivation, we can consider the campus recruitment job market. Companies visit colleges in various time slots during the year, while students are seeking a position throughout the year. In our terms, this is a two-sided matching problem in which the companies are “dynamic” with arrival and departure times while the students are “static” and always present in the market, although perhaps already matched.

Further, let us suppose that students may seek to obtain a better match by strategic mis-reporting of their preferences over companies, while companies report true preference rankings on students. We can assume this because it is generally known what skill sets companies want (e.g., which grades, in which kinds of classes, etc.). Student preferences on companies may be
predetermined or determined dynamically as companies arrive, as long as preference orderings over earlier companies are not changed by subsequent arrivals. For companies, it is probably easiest to think that their preferences over students are determined upon arrival and company cannot re-enter the market once departed.

Each company seeks to match with a single student, and a match (if any) must be assigned by the mechanism by the end of the company’s time slot. We assume, however, that each company has the opportunity to adopt a “fall-back” option, selected from its own reserve pool of students and providing (if necessary) a match for the company that is just as good as that from the primary matching market. This option can be exercised by the company if a matched student subsequently becomes unavailable because the mechanism later decommits and rematches the student to another company. But the reserve pool should be used in a limited way — we assume that it is more costly, and therefore less desirable for the company.

Within Artificial Intelligence (AI) community, this work is situated in the subfield of multi-agent resource allocation, and for example motivated by an interest in developing AI for crowdsourcing and task-sourcing markets [72, 73]. For the present model, we have workers on one side that seek a match with a new task (e.g., every week a match is formed for the subsequent week) and an uncertain and dynamic supply of tasks, each with preferences over workers and requiring a match to be assigned for the next week by its own deadline. Note here workers also have preferences over talks. For ex-positional purposes only, we refer to the static and strategic side of the market as Men and the dynamic and truthful side of the market as Women. A constraint imposed by the dynamics of the problem is that the match to a woman must be made before her departure (although with a chance to decommit from this and use a fall back option). Each man insists on receiving a match only by the final time period, beyond which no additional women will arrive.

6.1.1 Contributions

We introduce the GSODAS (Generalized Online Deferred Acceptance with Substitutes) mechanism, which makes use of the fall-back option, also referred to as substitute agents. GSODAS is dominant-strategy incentive compatible for static agents and also stable, such that no man-woman pair would prefer to deviate from their respective matches and re-match between themselves. Such a man-woman pair if it exists is called a blocking pair. The blocking pair includes a man-woman pair where the woman has matched with a fall-back option and that for any such pair, the woman prefers this man to the fall-back option (assumed equivalent in rank-preference to her original match) and the man prefers the woman than his match. The number of fall-back options required by GSODAS is worst-case optimal across online mechanisms that provide stability.
We compare the match quality from GSODAS with two randomized, truthful matching mechanisms that operate without using a fall-back option. For match quality, we consider both the stability (measuring the average number of men that are in at least one blocking pair) and the rank-efficiency of the mechanisms. The rank-efficiency measures the average preference rank-order achieved by agents in the match, with a rank-order of 1 for the most preferred and \( n \) for least-preferred, where there are \( n \) agents on each side of the market. For this, we interpret the rank-order for a woman matched with a substitute as equivalent to that for the man with which she was first matched, but ignore the substitute himself in determining rank-efficiency. In addition, the preference of any man that goes unmatched is accounted as a rank of \( n + 1 \).

For a rank-efficiency baseline, we also consider the performance of a non-truthful algorithm, namely, Consensus, that employs online stochastic optimization in determining dynamic matches. This provides a strong, baseline target for rank-efficiency. In simulation, we demonstrate that GSODAS has rank-efficiency better than the randomized mechanisms but dominated by Consensus, while the randomized mechanisms and also Consensus also suffer from poor stability and many blocking pairs. GSODAS requires on average around 20% of the women to be matched with fall-back options for two period problems, increasing to an average of 30% of the men for longer 12-period problems (in which the women are present in the market for around 3-4 periods). The most compelling direction for future work is to find an appropriate relaxation of stability for dynamic problems, and look to see whether this can provide a significant reduction in the use of substitutes.

### 6.1.2 Related Work

The classic matching algorithm is the deferred acceptance algorithm \( [10] \). This algorithm is strategyproof for one-side of the market and produces a stable match with respect to reported preferences. Moreover, there does not exist a stable matching mechanism that is strategyproof for all agents \( [23] \). We are only aware of one other paper on dynamic matching with incentive and stability considerations: Compte and Jehiel \( [34] \) consider a different dynamic matching problem to the one studied here, with a static population but agents that experience a preference shock, and impose an individual rationality constraint across periods so that no agent becomes worse off as the match changes in response to a shock. The authors demonstrate how to modify the deferred acceptance algorithm to their problem. For more background on the matching literature, readers are referred to a survey by Sönmez and Ünver \( [22] \). Karp et al. \( [75] \) consider the algorithmic problem of online matching, but without strategic considerations. Awasthi and Sandholm \( [10] \) consider a dynamic kidney exchange problem, but for a satisfying (rather than strict preference) model and without consideration of incentive or stability constraints. Parkes \( [77] \) provides a survey of dynamic auction mechanisms with money.
6.2 Preliminaries

Consider a market with $n$ men (set $M$) and $n$ women (set $W$). The men are static and the women are dynamic, with woman $i \in W$ having arrival $a_i$ and departure $d_i$, with $a_i, d_i \in \{1, \ldots, T\}$ where $T$ is the number of time periods. Each agent has a strict preference profile $\succ_i$ on agents on the other side of the market, and prefers to be matched than unmatched. We write $w_1 \succ_m w_2$ to indicate a strict preference by man $m$ for woman $w_1$ over woman $w_2$. A match to a man can be made in any of the $T$ periods, and preferences may be determined dynamically as women arrive as long as the preference rank on earlier arrivals is unchanged. Similarly, we write $m_1 \succ_w m_2$ to denote a preference by woman $w$ for man $m_1$ over $m_2$. For a woman, a match (if any) must be made between $a_i$ and $d_i$ and preferences must be well defined upon arrival.

Let $M(t)$ and $W(t)$ respectively denote the set of men and women available for matching in period $t$. Let $AW(t)$ denote the set of women to arrive in $t$, $DW(t)$ the set of women to depart in $t$, and $W'(t)$ the set of women yet to arrive. Let $\mu$ denote a match, with $\mu(m) \in W \cup \{\phi\}$ the match to man $m$ and $\mu(w) \in M \cup \{\phi\}$ the match to woman $w$, with $\mu(i) = \phi$ to indicate that agent $i$ is unmatched. A woman is available for matching while present, and a match $\mu(w) \neq \phi$ to a woman must be finalized by period $d_i$. Upon the departure of woman $w$ with $\mu(w) \neq \phi$, then the matched man $\mu(w) \in M$ ordinarily becomes unavailable for matching and $M(t)$ is updated. On the other hand, when we allow for a fall-back option the mechanism may decommit from the match $\mu(w)$ and allow a man to re-match.

For static settings, Gale-Shapley’s deferred-acceptance (DA) algorithm yields a stable matching. In this chapter we adopt the man-proposing DA algorithm as a building block:

**Definition 6.1.** Man-proposing DA. Each man proposes to his most preferred woman. Each woman keeps the best match and rejects other men. All rejected men then propose to their next preferred woman. The procedure continues until there are no more rejections.

We denote $DA(M, W)$ as male proposing DA with set of men $M$ and set of women $W$. The DA algorithm terminates in a finite number of steps because every man proposes to a finite number of women.

Let $\succ = (\succ_i)_{i \in M \cup W}$. We also write $\succ_i = (\succ_i, \succ_{-i})$, where $\succ_{-i}$ denotes the preferences of all the agents except $i$. Let $\rho = \{(a_i, d_i) : i \in W\}$ denote the arrival and departure periods of the women. An online matching mechanism $f$ selects a matching $\mu = f(\succ, \rho)$. To be feasible, we require that $f(\succ, \rho)$ is invariant to information about later arrivals, so that $\mu(w)$ is invariant to preferences of men about women $w'$ to arrive after $w$ departs or to the preferences, arrival or departure times of later arrivals $w'$. In particular, $\mu(w)$ must be determined by period $d_i$ at which a woman departs. Note, in our settings, the static side, $M$, is always available for matching. So, it depends upon the rules of the mechanism whether or not a man be allowed for
recomputing the match.

**Definition 6.2.** An online mechanism \( f \) is truthful (or strategyproof) for men if for each man \( m \), for all arrival-departure schedules \( \rho \), and for all preferences \( \succ_{\neg m} \) except \( m \),

\[
\mu'(m) \neq \mu(m),
\]

where \( \mu' = f(\succ'_m, \succ_{\neg m}, \rho) \).

In evaluating the performance of a mechanism, we follow Budish and Cantillon [78] and assume risk neutral agents with a constant difference in utility across the matches that are adjacent in their preference list. The rank of an agent \( i \) for a matching \( \mu \), written \( \text{rank}_i(\mu) \), is the rank order of the agent with whom he or she is matched. A match by \( i \) with the most-preferred agent in \( \succ_i \) receives rank order 1 and with the least-preferred receives rank order \( n \). If \( \mu(i) = \phi \) then the rank-order is \( n + 1 \). Based on this, the rank of a matching \( \mu \) is

\[
\text{rank}(\mu) = \frac{1}{2n} \sum_{i \in M \cup W} \text{rank}_i(\mu).
\]

To define the rank-efficiency of a mechanism we assume that \( (\succ, \rho) \) is distributed with a distribution function \( \Phi \) and compute the expected rank over the induced distribution on matches:

**Definition 6.3.** The rank-efficiency of an online mechanism \( f \), given distribution function \( \Phi \), is

\[
\text{rank}^f = \mathbb{E}_{(\succ, \rho) \sim \Phi}[\text{rank}(f(\succ, \rho))].
\]

The rank-efficiency is the expected rank of a matching produced by \( f \), where the expectation is with respect to preference profiles \( (\succ) \) and arrival/departure schedules \( (\rho) \).

To gain some intuition for the dynamic matching problem, we can consider simply running a man-proposing DA on unmatched men and women in the system at the departure of one or more women. As well as fixing the match for any such departing woman, it also sets the match for any man matched to a departing woman. The set of men still available for matching in the future is updated.

**Example 6.1.** Consider \( M = \{m_1, m_2, m_3\} \) and \( W = \{w_1, w_2, w_3\} \). Suppose the preferences
and arrival/departure periods are as follows:

\[ m_1 : w_3 \succ_m w_1 \succ_m w_2 \]
\[ m_2 : w_2 \succ_m w_1 \succ_m w_3 \]
\[ m_3 : w_1 \succ_m w_2 \succ_m w_3 \]
\[ w_1 : m_1 \succ w_1 m_2 \succ w_1 m_3, a_{w_1} = 1, d_{w_1} = 1 \]
\[ w_2 : m_1 \succ w_2 m_2 \succ w_2 m_3, a_{w_2} = 1, d_{w_2} = 2 \]
\[ w_3 : m_1 \succ w_3 m_2 \succ w_3 m_3, a_{w_3} = 2, d_{w_3} = 2 \]

If the agents are truthful, the mechanism will match \( m_1 \) with \( w_1 \), \( m_2 \) with \( w_2 \) and \( m_3 \) with \( w_3 \). While reporting truthfully, \( m_1 \) has to leave the system with \( w_1 \). However, \( m_1 \) can report his preference as \( w_2 \succ_{m_1} w_1 \). With this manipulation, he will get matched with \( w_2 \) in period 1, and remain available to match in period 2 with \( w_3 \), his most preferred woman. Thus, this greedy DA mechanism is manipulable.

This illustrates the basic challenge to be addressed: how to achieve strategyproofness in an online matching mechanism when departures on one side of the market require decisions to be made without knowledge of possible future arrivals. We will also be interested in achieving stability and high rank-efficiency.

### 6.3 Introducing a Fall-Back Option

A fall-back option allows a mechanism to decommit from a match made in an earlier period to a departed woman because the woman is assumed to have access to a fall-back option or substitute. Such a substitute is assumed to be at least as preferred as the match provided by the mechanism. On the other hand, substitutes are assumed to be costly to use and thus a woman would prefer to receive her match from the matching market.

Let \( R \) denote the set of substitutes. We now allow for a matching \( \mu \) to allocate \( \mu(m) \in W \cup \{\phi\} \) and \( \mu(w) \in M \cup R \cup \{\phi\} \). For each substitute \( r \in R \), we say that \( r \) is equivalent to man \( m \in M \) for woman \( w \in W \), if \( m' \succ_w m \iff m' \succ r \) for all \( m' \in M \setminus \{m\} \); i.e., as long as \( r \) is equivalent in terms of preference rank to \( m \) for woman \( w \). In extending the notion of rank-efficiency, the rank order to a woman for a substitute is that of the man \( m \) replaced by the substitute while the rank of the substitute himself is not included in \( \text{rank}(\mu) \).

**Definition 6.4.** Matching \( \mu \) is stable if there does not exist a blocking pair \((m, w)\), where \((m, w)\) is a blocking pair for \( \mu \) if either:

\[ 
\]
(1) \( w \succ_m \mu(m) \) and \( m \succ_w \mu(w) \), or 
(2) if \( w \) receives a substitute \( r \) that is equivalent to man \( m' \), then, \( w \succ_m \mu(m) \) and \( m \succ_w m' \).

When \( m \) is part of any blocking pair, we say \( m \) is unstable. Else we say \( m \) is stable.

Here is a simple idea for how one might initially think about using substitutes, with the goal of achieving truthfulness and stability:

- Maintain a match \( \mu \) in each period between men and women that have arrived so far, including matches to substitutes, if any. One or more men may have a match with a woman who has already departed.

- Upon the arrival of one or more women in period \( t \), run the man-proposing DA(\( M, W(t) \)) for men \( M \) and women \( W(t) \) present, and modified so that each man in \( M \) only proposes to women that are more preferred than his current match (if any). The match for women \( W(t) \) (and corresponding men) is adopted into match \( \mu \). If this decommits a man \( m \) from a match with a previously departed woman then this woman uses a substitute.

Example 6.2. Consider men \( M = \{m_1, m_2, m_3\} \) and \( W = \{w_1, w_2, w_3\} \). Suppose the preferences and arrival/departure periods are as follows:

\[
\begin{align*}
m_1 &: w_1 \succ_{m_1} w_2 \succ_{m_1} w_3 \\
m_2 &: w_2 \succ_{m_2} w_1 \succ_{m_2} w_3 \\
m_3 &: w_1 \succ_{m_3} w_2 \succ_{m_3} w_3 \\
w_1 &: m_2 \succ_{w_1} m_1 \succ_{w_1} m_3, a_{w_1} = d_{w_1} = 2 \\
w_2 &: m_1 \succ_{w_2} m_2 \succ_{w_2} m_3, a_{w_2} = d_{w_2} = 1 \\
w_3 &: m_1 \succ_{w_3} m_2 \succ_{w_3} m_3, a_{w_3} = d_{w_3} = 1
\end{align*}
\]

At \( t=1 \), \( w_2 \) and \( w_3 \) are present. So, \( m_1 \) is matched with \( w_2 \) and \( m_2 \) is matched with \( w_3 \). At \( t=2 \), \( w_1 \) arrives. So \( m_2 \) gets matched with \( w_1 \) as he prefers \( w_1 \) over \( w_3 \) and \( w_3 \) receives a substitute instead of \( m_2 \). The final match is \( m_1 - w_2, m_2 - w_1, m_3 - \text{none} \). Suppose instead that \( m_1 \) reports his preference as \( m_1 : w_1 \succ_{m_1} w_3 \succ_{m_1} w_2 \). At \( t=1 \), he gets matched with \( w_3 \) and \( m_2 \) gets matched with \( w_2 \). At \( t=2 \), only \( w_1 \) is present and \( m_1 \) gets matched with \( w_1 \). The final match is \( m_1 - w_1, m_2 - w_2, m_3 - \text{none} \). Thus \( m_1 \) can get a better match by misreporting his preferences.

This manipulation is possible because a man can get matched with a less preferred woman in the current period (in the above example, \( m_1 \) gets matched with \( w_3 \) instead of \( w_2 \) in period 1), which allows another man to be matched to the match that would otherwise be assigned (\( m_2 \).
to \( w_2 \) in this case), and so that this man no longer competes for a new arrival in the next period (\( m_2 \) does not propose to \( w_1 \) in this example).

\[
\begin{array}{c|c|c}
  m_1 & (T, \ldots, 2, 1, w) & w_1 & (1, 2, \ldots, T, m), a = d = 1 \\
  m_2 & (T, \ldots, 2, 1, w) & w_2 & (\mu(w_1), 1, 2, \ldots, T, m), a = d = 2 \\
  \vdots & \vdots & \vdots & \vdots \\
  m_T & (T, \ldots, 2, 1, w) & w_T & (\mu(w_{T-1}), 1, 2, \ldots, T, m), a = d = T \\
  m_{T+1} & (2T, \ldots, T+2, T+1, w) & w_{T+1} & (T+1, T+2, \ldots, 2T, m), a = d = 1 \\
  m_{T+2} & (2T, \ldots, T+2, T+1, w) & w_{T+2} & (\mu(w_{T+1}), T+1, \ldots, 2T, m), a = d = 2 \\
  \vdots & \vdots & \vdots & \vdots \\
  m_{2T} & (2T, \ldots, T+2, T+1, w) & w_{2T} & (\mu(w_{2T-1}), T+1, \ldots, 2T, m), a = d = T \\
  \vdots & \vdots & \vdots & \vdots \\
  m_{(\alpha-1)T+1} & (\alpha T, \ldots, (\alpha-1)T+2, (\alpha-1)T+1, w) & w_{(\alpha-1)T+1} & (\mu(w_{(\alpha-1)T+1}), \ldots, \alpha T, m), a = d = 1 \\
  \vdots & \vdots & \vdots & \vdots \\
  m_{\alpha T} & (\alpha T, \ldots, \alpha T, w) & w_{\alpha T} & (\mu(w_{\alpha T}), \ldots, \alpha T, m), a = d = 1 \\
\end{array}
\]

Table 6.1: Construction of agent preferences used for worst-case substitutes requirement in online matching mechanisms

### 6.3.1 GSODAS

Recall that \( W(t) \) is the set of women present in period \( t \). The GSODAS algorithm works as follows, where \( \max_m(w_1, w_2) \) denotes the woman of \( \{w_1, w_2\} \) most preferred by man \( m \):

- For periods \( t \in \{1, \ldots, T\} \), maintain **provisional match** \( \mu^t(m) \in W \cup \{\phi\} \), for every \( m \in M \). Initialize \( \mu^0(m) = \phi \).

- Maintain a **committed match** \( \mu^*(m) \) for every \( m \in M \), initialized to \( \mu^*(m) = \phi \) for all \( m \).

- In every period \( t \) in which at least one woman departs,
  
  (i) run \( DA(M, W(t)) \), and let \( \mu^t \) denote this match
  
  (ii) update \( \mu^t(m) := \max_m(\mu^{t-1}(m), \mu^t(m)) \) for every \( m \)
  
  (iii) if the assignment changes in \( \mu^t(m) \) from \( \mu^{t-1}(m) \) for man \( m \), where \( \mu^*(m) \neq \phi \), then woman \( \mu^*(m) \) is matched with a substitute for \( m \) and \( \mu^*(m) \leftarrow \phi \) with \( m \) no longer committed.

  (iv) \( \mu^*(m) := \mu^t(m) \) if woman \( \mu^t(m) \) departs in the current period.
• The final match $\mu_G$ has men matched as in $\mu^*(m)$ (along with corresponding $\mu(w)$ for matched women $w$), and with any other woman who received a substitute in step (iii) matched to this substitute, or otherwise unmatched.

GSODAS maintains a sequence of provisional matches $\mu^t$ in each period $t$, but the matches are only committed (and may even be subsequently decommitted) as women depart. A match is valid when no man is matched to multiple women and no woman is matched to multiple men.

Claim 6.1. The GSODAS algorithm is strategyproof for men and generates a valid match.

Proof:
Fix man $m$. Strategyproofness follows immediately from the strategyproofness of man-proposing DA when one notices that the preferences reported by other agents in DA($M, W(t)$) in period $t$ are independent of the report of man $m \in M$. Moreover, man $m$ receives the woman that is most preferred across all runs of the man-proposing DA, across all periods.

To establish that the final match is valid, suppose for contradiction that there is some woman $w = \mu_G(m_1) = \mu_G(m_2)$ for $m_1 \neq m_2$. Suppose that $w$ is matched with $m_1$ at $t_1$ and $m_2$ at $t_2$. Assume first that $m_1 >_w m_2$. At $t_2$, $w$ is matched with $m_2$, which implies that $m_1$ did not propose to her and received a better match at $t_2$ than $w$. But then we would not have $w = \mu_G(m_1)$ because this is the best match across all periods for $m_1$. Similarly, if $m_2 >_w m_1$ then at $t_1$, when $w$ is matched with $m_1$, $m_2$ must have received a better match than $w$ and we again have a contradiction.

The manipulation in Example 6.2 is not possible here. Suppose $m_1$ tries to manipulate; now, in period 2 DA, $\mu(m_2) = w_1$ and $w_2$ being preferred, $\mu^*(m_2) = w_2$ and $w_1$ will be provided with a substitute. And $\mu^*(m_1) = w_3$. Instead when he reports truthfully, he will be matched with $w_2$.

Note: GSODAS matches every woman, either with a man $m \in M$ or with a substitute. Because $|M| = |W|$ and some women receive a substitute, the number of unmatched men equals the number of substitutes adopted in the mechanism.

6.3.2 Stability

Stability requires that there is no blocking pair, i.e., no man-woman pair that would both prefer to match with each other than their match from the mechanism.

Claim 6.2. The GSODAS algorithm is stable.

Proof:
We prove the claim by a contradiction. Suppose a pair, $(m, w)$ blocks the final match $\mu_G$ yielded by GSODAS. For each man, the final match is a woman most preferred among all his provisional
matches (perhaps $\phi$). Because $w \succ_m \mu_G(m)$, then $m$ was never matched with $w$ in a provisional match. Let $a$ be the arrival time of $w$ and $d$ the departure time. Let $M(w) = \{m_a, m_{a+1}, \ldots, m_d\}$ denote the set of men with whom $w$ is matched (if any) in the provisional match in each period $t \in \{a, \ldots, d\}$. Because the match generated in each period is stable, then $m' \succ_w m$ for all $m' \in M(w)$. In particular, we have $m_d \succ_w m$ and $\mu_G(w) \succ_w m$ (including the case where $w$ later receives a substitute), and $(m, w)$ is not a blocking pair.

**Claim 6.3.** The worst case substitute requirement in GSODAS for a $T$ period problem, with $n = \alpha T$ men and women, for $\alpha \in \{1, 2, \ldots\}$, is $\alpha(T - 1)$.

**Proof:**

Let $k$ denote the number of matches between men and women, so that $n - k$ is the number of matches between women and substitutes. For $k$ matches with (non-substitutes) men, we can have at most $(T - 1)k$ substitutes, occurring when a better match is found for each of the $k$ matched men in each round. We require $k$ plus the total number of substitutes to be at least $n$, since all women will always receive some match. Therefore $k + (T - 1)k \geq n$, and $k \geq n/T$. From this, the maximum number of substitutes, $n - k \leq n - n/T = (T-1)n = \alpha(T - 1)$.

To see that this bound is tight, consider the following example. Consider an instance in which in every period, exactly $\alpha$ women arrive and the $j^{th}$ woman in that period indicates $m_j$ to be her best match. Each woman departs immediately. That is, in period 1, women $w_1, w_2, \ldots, w_\alpha$ arrive and depart. In period $i$, $w_{(i-1)\alpha+1}, \ldots, w_{i\alpha}$ arrive and depart. $w_1, w_{\alpha+1}, w_{2\alpha+1}, \ldots, w_{(T-1)\alpha+1}$ indicate $m_1$ as the most preferred match. $w_2, w_{\alpha+2}, w_{2\alpha+2}, \ldots, w_{(T-1)\alpha+2}$ indicate $m_2$ as the most preferred match, and so forth. Each $m_j$ has preference as $(w_{(T-1)\alpha+j}, \ldots, w_{\alpha+j}, w, w)$, where $w$ is a placeholder for all other women (in arbitrary sequence). Each $m_j, j = 1, 2, \ldots, \alpha$ invokes the need for a substitute at every time $t = 2, 3, \ldots, T$, and therefore the total number of substitutes is $\alpha(T - 1)$.

Thus, GSODAS has a large worst-case cost in terms of the number of substitutes required. We will evaluate an average-case cost in simulation.

Comparing GSODAS with other algorithms, we establish a worst-case trade-off between the number of substitutes and the number of men that can be part of a blocking pair. For this, define for matching $\mu$ the quantity,

$$S(\mu) = |\text{unstable men in } \mu| + |\text{substitutes used}|,$$

where an unstable man is part of at least one blocking pair.

**Proposition 6.1.** For any online matching algorithm, for every problem with $T$ periods, $n = \alpha T$ men and women, and $\alpha \in \{1, 2, \ldots\}$, there exists an instance in which $S(\mu) \geq \alpha(T - 1)$. For GSODAS, we have $S(\mu) \leq \alpha(T - 1)$, with $S(\mu) = \alpha(T - 1)$ in the worst case.
Proof:
Consider agent preferences in Table 6.1. The preference profile of a man \( m \) is denoted by the indices of women in decreasing order of preference; e.g., preference profile \( w_2 \succ_m w_4 \succ_m w_1 \succ w \) will be denoted as \((2, 4, 1, w)\). The \( w \) at the end of the list indicates all other women in some arbitrary order. A similar convention is adopted for the preferences of women. The agents are grouped into \( \alpha \) blocks, each consisting of \( T \) men and women. In each period, one woman from each block arrives and departs immediately. The groups are defined so that the men in each group prefer the women in the same group more than any woman in any other group. The same is true for the women, except that for any woman, \( w_iT+j \) for \( i \in \{0, \ldots, \alpha - 1\} \) and \( j \in \{2, \ldots, T\} \), her most-preferred man is set to be the match \( \mu(w_iT+j-1) \) to the preceding woman in the block when this woman receives a match, and this match is not a substitute.

We argue that each of \( w_iT+j \) in groups \( i \in \{0, \ldots, \alpha - 1\} \) for \( j \in \{1, \ldots, T - 1\} \) contributes a count of 1 to \( S(\mu) \). If such a woman receives a substitute, then she contributes 1 to this sum. Similarly, for every woman unmatched, at least one additional man is unmatched and part of a blocking pair (e.g., with the unmatched woman). Now suppose that \( w_iT+j \) is matched with man \( m_{i'T+j'} \) where \( i' \neq i \). There must be some \( w_{i'T+k} \) for \( k \in \{1, \ldots, T\} \) not matched with a man in the \( i' \)th group. But then \((m_{i'T+j'}, w_{i'T+k})\) is a blocking pair because the man prefers any woman in \( i' \) to woman \( w_iT+j \) and woman \( w_{i'T+k} \) prefers a man in group \( i' \) over a match from any other group, noting that for \( k > 1 \) she cannot be matched to her most-preferred man for \( \mu(w_{i'T+k-1}) \neq \phi \) because this man is matched with the preceding woman in the group. In the other case, when \( i' = i \), then \((m_{i'T+j'}, w_{i'T+j+1})\) is a blocking pair. This is because every man in group \( i \) prefers a later woman in the group over an earlier woman, and woman \( w_{i'T+j+1} \) has \( m_{i'T+j'} \) as her most-preferred match. Noting that for each such woman, \( w_iT+j \), the blocking pair involves the man with whom she is matched, then we add 1 to \( S(\mu) \).

In GSODAS, the number of unstable men = 0. And by Claim 6.2, the number of substitutes \( \leq \alpha(T-1) \) and hence for GSODAS, \( S(\mu) \leq \alpha(T-1) \).

We see that there is a trade-off, in the worst-case, between the stability of an online algorithm and the number of substitutes. There exist instances where every substitute below \( \alpha(T - 1) \) leads to one additional man part of a blocking pair. For stability, then in the worst-case there is a need for at least as many substitutes as in GSODAS. An online algorithm that does not use substitutes will, in the worst-case, have a shrinking fraction \( \alpha/n = 1/T \) of men that are not part of blocking pairs as \( T \) increases.

### 6.3.3 Randomized Online Matchings

In this section, we introduce two additional mechanisms, that are truthful for men but without using the fall-back option. These are Random Online Matching Algorithms (ROMA). In the first
variation, ROMA1, every woman is matched with some man from the set $M$ while in the second variation, ROMA2, not all the women are matched. The algorithms make different trade-offs between stability and rank-efficiency.

For ROMA1, in every period $t$, if there are departing women then select $|DW(t)|$ men at random and run man-proposing DA using these men and $DW(t)$. Commit to this match. In periods without departing women, then with probability $p > 0$, run man-proposing DA with $W(t)$ women and $|W(t)|$ men selected at random. Commit to this match. Any match is final and these men and women are not considered for matching in future periods. For ROMA2, we define a threshold $\tau \geq 1$, and whenever the number of women present is $|W(t)| \geq \tau$, then select $|W(t)|$ men at random and run man-proposing DA. Commit to this match.

**Claim 6.4.** ROMA1 and ROMA2 are strategyproof for men.

**Proof:**

Men are randomly matched into a single instance of the man-proposing DA algorithm and cannot affect which instance they match to through misreports of preferences, and because the man-proposing DA is strategyproof for men.

ROMA1 and ROMA2 have an advantage over GSODAS in that they do not require the use of substitutes. On the other hand, they may well lead to a large number of blocking pairs and worse rank-efficiency because each man only participates in a single instance of DA.

### 6.3.4 Comparison with a Stochastic Optimization based Approach

To obtain a baseline performance for rank-efficiency, we adopt an online sample-based stochastic optimization algorithm, based on the Consensus approach of Van Hentenryck and Bent [79].

The algorithm is not truthful, but provides good rank-efficiency.

The Consensus approach adopts a generative model of the future to sample random future arrivals of agents on the dynamic side of the market, and uses these samples to guide match decisions for agents in the market. That is, it is assumed that the underlying statistical model of the arrival and departure of the agents and their preferences is known. Whenever, there is call to sample a future, the generative model generates an instance of arrival-departure schedule along with preferences for the remanining women. Generally, in the Consensus technique, large number of future samples are generated. Best possible decision pertaining to current period is taken for each sample generated. Then based on the frequencies of various current possible decision, the final current period decision is made. In regards to our matching problem, in every period in which at least one woman departs, Consensus samples multiple possible future arrivals and matches each departing woman with the man with which she is most frequently matched when running a man-proposing DA on each sample:
For any period $t$ in which at least one woman departs,

(i) generate $K$ samples of the preferences for $n - \ell$ women, where $\ell$ women have already arrived,

(ii) for each sample $W_k$, for $k \in K$, run man-proposing $\text{DA}(M(t), W(t) \cup W_k)$

(iii) for each woman $w \in W(t)$, let $L(w)$ denote the man most frequently matched with her in the result of running DA on each of the $K$ samples, breaking ties at random,

(iv) run man-proposing DA on the set of women, $W(t)$, and men in the set $\{L(w) | w \in W(t)\}$. Commit the matches in this DA that involve departing women, updating $M(t)$ accordingly.

Note that it is possible that $L(w_1) = L(w_2)$ for some $w_1 \neq w_2$, so that there are less men than women in step (iv) and some women may depart without a match.

### 6.4 Experimental Results

![Graph showing the number of substitutes required for men in GSODAS as $n$ increases, fixing $T = 2$.](image)

Figure 6.1: The number of substitutes required for men in GSODAS as $n$ increases, fixing $T = 2$.

We compare the rank-efficiency and stability of GSODAS, ROMA1, ROMA2, and Consensus (which is not truthful). In all simulations, we generate preference profiles uniformly at random for all men and women. In ROMA1, the value of parameter $p$ is set to be 0.3, which was found experimentally to provide good rank-efficiency for $T = 2$ and $T = 4$ for varying $n$. 

165
The threshold parameter $\tau$ in ROMA2 is similarly tuned to achieve the best performance for rank-efficiency, and we adopt $\tau = \max\{0.375n/T, 1\}$.

We first investigate the number of substitutes required in GSODAS. For this we consider a problem with two time periods, increasing the number of agents on each side of the market from $n = 2$ to 24. For each woman $i$, $a_i$ is either 1 or 2, both with equal probability and $d_i \in \{1, 2\}$ uniformly at random if $a_i = 1$, else $d_i = 2$. We also increase the number of periods $T$ from 2 to 12, holding $n = 20$, and generating the arrival time, $a_i$, for a woman uniformly between $[1, T]$, with departure time $d_i$ uniformly between $[a_i, a_i + T/3]$, with $d_i$ also capped at a maximum value of $T$. In both experiments, we determine worst-case and average case performance over 20,000 random instances.

The results are illustrated in Figures 6.1 and 6.2. For a problem with two time periods, we find that an average of $\approx 20\%$ of the number of men are required as substitutes, increasing to around $30\%$ for $T = 12$. For two period problems, in the worst case we need a substitute for as many as 1 in every 2 men in the market when $n \leq 10$; this fraction drops to $37\%$ for $n = 24$. For $n = 20, T = 12$, then as many as $55\%$ of the number of men are required as substitutes in the worst case.

We turn now to comparing rank-efficiency and stability in each of the mechanisms. For this,
Figure 6.3: The rank-efficiency (x-axis) vs. the number of unstable men (y-axis) for $n = 10$ and $T = 2$.

Figure 6.4: The rank-efficiency (x-axis) vs. the number of unstable men (y-axis) for $n = 20$ and $T = 4$. 
we determine the average rank-efficiency and average number of unstable men (i.e., number of men \( m \) for whom there exists a woman \( w \) such that \( (m, w) \) is a blocking pair). The results are again averaged over 20,000 instances. Figures 6.3 and 6.4 plot the average rank-efficiency (x-axis) against the average number of unstable men (y-axis) for \( n = 10, T = 2 \) and \( n = 20, T = 4 \). Recall that Consensus is not strategyproof, and that rank-efficiency assigns a rank of \((n + 1)\) to unmatched agents and ignores the rank preference of substitute agents. The results are encouraging for the GSODAS mechanism. We see that it dominates ROMA1 and ROMA2 in rank-efficiency while achieving perfect stability. This is even though we count \( n + 1 \) rank for the unmatched men in GSODAS, the number of which can be quite large due to the use of substitutes. Comparing with Consensus, we see that GSODAS has worse rank-efficiency, achieving a rank-efficiency that is situated between that of Consensus and the ROMA mechanisms.

6.5 Conclusions

In this chapter, we have initiated a study into dynamic matching problems in two-sided markets without money. One side of the market is static while the other side is dynamic, and we require truthfulness on the static side of the market. We achieve stability, and truthfulness on the static side, by allowing for the possibility of a fall-back option, so that the mechanism can decommit from some matches made to already departed agents, at which point a substitute is adopted. The GSODAS mechanism has better rank-efficiency than simpler methods that do not use substitutes, although with less rank-efficiency achieved by a non-truthful stochastic optimization approach.

Still, the use of substitutes in GSODAS is quite high, with 30% on average as the number of agents. As number of time periods increases (for uniform preferences), as many as 55% required in the worst-case experimental instances. This is likely unacceptable in many practical domains, yet we prove that better worst-case properties are unavailable if full stability is required. The most interesting future direction, then, will look to relax the requirement of offline stability. This precludes blocking pairs, irrespective of the timing of the agents that comprise a blocking pair in system and the information available at the time of a match. Perhaps by relaxing this requirement, then mechanisms with good rank-efficiency, acceptable stability, but less need for exercising the fall-back option can be developed.
Chapter 7

Dynamic House Allocation

We study a dynamic variant on the house allocation problem. Each agent owns a distinct object (a house) and is able to trade its house while present in the market. Agents have strict preferences over houses, and the market operates without payments. The goal is to enable an efficient reallocation of objects, along with strategyproofness and while satisfying participation constraints. We first establish conditions under which an online mechanism that allows an agent to influence the period in which it trades can be manipulated. This motivates partition mechanisms in which agents are divided online into disjoint feasible trading groups, with the top trading cycle algorithm (TTCA) run separately for each group. In particular, we demonstrate good rank-efficiency for a mechanism that adopts stochastic-optimization in determining how to partition agents.

7.1 Introduction

In the house allocation problem, each of a set of self-interested agents owns a distinct object (a house) and has strict preferences on houses[3]. The problem is to find a reallocation of objects amongst agents that is robust against misreports of preferences by agents while identifying beneficial trades and without using money. The top-trading cycle algorithm (TTCA) [29, 28] is strategyproof and finds an allocation in the core. An allocation in the core is stable, in the sense that no coalition of agents can block the outcome by reallocating their initial objects amongst themselves in a way that is weakly better for every agent and strictly better for at least one agent in the coalition. The TTCA is known to be essentially unique amongst mechanisms for the static house allocation problem with useful economic properties [29, 30].

In the dynamic model of the house allocation problem discussed here, each agent has an arrival period and a departure period and is only able to trade with other agents present simultaneously in the market at the same time. For a motivating example, consider college housing,
with students on different leases and willing to trade during the month before their lease expires. Another example is provided by the problem of reallocating offices in the workplace, or finding mutually-beneficial trades (e.g., swaps, three-cycles, etc) of vacation property such as time-shares when inventory is in the market at different times.

7.1.1 Contributions

We establish general conditions under which no mechanism in which an agent can influence the set of agents with which it participates in a TTCA (e.g., the period in which it trades and thus the other agents that it trades with) can be strategyproof. Given this, we study partition mechanisms, in which each agent is assigned online to a group of agents with which it will engage in a single TTCA.

We consider three variations of partition mechanisms:

(i) trade on departure (DO-TTCA)
(ii) trade on departure or when the population size is above some threshold (T-TTCA)
(iii) a stochastic optimization approach to determine a good partitioning of agents (SO-TTCA).

We adopt rank-efficiency as a measure of performance. Rank-efficiency is the preference rank of agents for allocated objects, averaged across all agents and across instances sampled from a distribution. This is a meaningful measure of performance for risk neutral agents, each of which has an equal difference in utility for successive houses in its preference list.

The experimental results show that SO-TTCA and T-TTCA outperform DO-TTCA by 15-20% in terms of rank-efficiency when there are large number of agents with uniform arrival time and the agents are patient. Furthermore, when the agents arrive according to a Poisson process, then SO-TTCA outperforms DO-TTCA by 4-10% in terms of rank-efficiency. For environments in which agents tend to be impatient, T-TTCA has performance competitive with that of SO-TTCA. But T-TTCA is less successful than SO-TTCA in identifying partitions in which each agent tends to have a large number of trading partners, which is useful when agents are patient.

7.1.2 Related Work

For an overview of mechanisms on matching and house allocation, the interested reader is referred to Sönmez and Ünver [22]. In previous models of multi-period house allocation problems, each agent expresses a preference on a sequence of allocations [32, 33]. Our model is different because an agent’s preferences are expressed only over its eventual allocation upon departure, that is, the final house it receives at the time of departure and not on the sequence of allocations made while present in the market. Another difference is that our agents bring a house to the market upon arrival and depart with a house, while previous models maintain the same pool
of objects throughout and agents arrive and leave empty-handed. Stochastic models of kidney exchanges, in which trades are identified amongst donor-patient pairs, are explored but without incentive considerations [76, 80]. For other dynamic models, there is existing work on matching (with strict preferences expressed in a bipartite graph) [71], assignment (which is different from house allocation because agents do not own an object upon arrival) [81], as well as a considerable literature on dynamic mechanisms with money (see Parkes [77] for a survey). For static problems, there is also computational work on the reallocation of objects through different types of local trades (e.g., swaps, 3-cycles, etc.), with a view to finding desirable outcomes such as efficient or fair allocations [52, 53].

7.2 The Model

Let $N = \{1, 2, \ldots, n\}$ denote the set of agents and $H = \{h_1, h_2, \ldots, h_n\}$ denote the set of distinct houses. Agent $i$ enters the market with house $h_i$ in period $a_i \in T$, where $T = \{1, 2, \ldots\}$, and departs in $d_i \in T$. Let $Sched_N$ denote the set of all possible arrival and departure times of $N$ agents. Agent $i$ has strict preferences $\succ_i$ on the house allocated upon departure. Let $(h \succ_i h') \equiv (h' \not\succ_i h)$. The preference profile of all the agents is denoted by $\succ = (\succ_1, \succ_2, \ldots, \succ_n) \in U$, where $U$ is the set of all possible strict preference profiles. Let $\succ_{-i}$ denote the preference profile of agents except agent $i$. An agent knows its preferences $\succ_i$ upon arrival into the market. We allow for arbitrary misreports of preferences, but assume that arrival and departure times are truthfully reported.

Let $x : N \rightarrow H$ denote a house allocation, with agent $i$ allocated to house $x(i)$. Let $X(\rho)$ denote the set of feasible allocations given arrival-departure schedule $\rho \in Sched_N$. An allocation is feasible for a given schedule $\rho$ - if there is a sequence of trades, one in each period,
- where the trade in any given period is only between those agents present
- and is feasible given the allocation determined through trade in the previous period, - with each agent allocated to exactly one house in every period.

An online house allocation mechanism $f(\succ, \rho) \in X(\rho)$, generates a feasible allocation given a reported type profile and arrival-departure schedule. Let $x = f(\succ, \rho)$ and $x' = f(\succ', \rho')$. The problem is online, in that an agent’s type $(a_i, d_i, \succ_i)$ is not available until period $t = a_i$ and therefore we require, for all $j \in N$, that $x(j) = x'(j)$ when $\succ, \succ'$ and $\rho, \rho'$ differ only in periods after agent $j$’s departure. An assignment of houses, $x \in X$, Pareto dominates $y \in X$, if $x \succeq_i y$ for all $i$ and $x \succ_j y$ for some $j$. An allocation $y \in X$ is Pareto efficient if there is no allocation $x \in X$ that Pareto dominates $y$.

**Definition 7.1** (Pareto efficient). A mechanism $f$ is Pareto efficient if allocation $x = f(\succ, \rho)$
is Pareto efficient for all preference profiles \( \prec \) and all schedules \( \rho \).

**Definition 7.2** (Strategyproof (SP)). Let \( x = f(\succ_i, \succ_{-i}, \rho) \) and \( x' = f(\succ'_i, \succ_{-i}, \rho) \). A online mechanism \( f \) is strategyproof if \( x(i) \succ_i x'(i) \), for all \( i \), all \( \rho \in \text{Sched}_N \) and all \( \succ_{-i} \in U_{-i} \).

**Definition 7.3** (Individually Rational (IR)). An online mechanism \( f \) is individually rational if \( x_i \succ_i h_i \), where \( x = f(\succ, \rho) \), for all \( \succ \in U \), all \( \rho \in \text{Sched}_N \).

An allocation \( x \) is blocked by a coalition of agents \( S \subseteq N \), if there is a feasible allocation of the houses initially owned by agents in \( S \) amongst themselves that Pareto dominates, for agents in \( S \), the allocation \( x \).

**Definition 7.4** (Core). An online mechanism \( f \) is core-selecting if allocation \( x = f(\succ, \rho) \) is not blocked by any coalition of agents, for any preference profile \( \succ \), and any schedule \( \rho \).

An allocation is in the core implies it is Pareto efficient (by considering coalitions of size \( n \)) and IR (by considering coalitions of size one).

To obtain a quantitative measure of efficiency, we assume in our experimental analysis that agents are risk neutral, and with a uniform difference in utility between successive houses in their preference orders. Based on this, we can compare the expected utility of two mechanisms in terms of rank-efficiency. Let \( \text{rank}_{f,i}(\succ, \rho) \) denote the rank that agent \( i \) assigns to the house allocated by mechanism \( f \) given \( \succ \) and \( \rho \), and define

\[
\text{RANK}_f(\succ, \rho) = \sum_i \text{rank}_{f,i}(\succ, \rho)
\]

**Definition 7.5.** The rank-efficiency of mechanism \( f \) is

\[
\text{rank}^f = \mathbb{E}_{\succ, \rho}[\text{RANK}_f(\succ, \rho)],
\]

where the expectation is taken with respect to a distribution on agent preferences and arrival-departure schedules.

**Lemma 7.1.** No online house allocation mechanism with three or more agents can be Pareto efficient as well as individually rational (IR).

**Proof:**
Suppose there exists an online mechanism \( f \), which is IR as well as Pareto efficient. Consider the following two period dynamic house allocation with 3 agents. (Note that, agent \( i \) owns house
In this instance, 1 - $h_3$, 2 - $h_1$, 3 - $h_2$ is the only feasible, Pareto efficient, and IR allocation. As agent 2 departs in period 1, $f$ should assign $h_1$ in period 1 only. If it happens that agent 3 : $h_3 \succ_3 h_1 \succ_3 h_2$, then the only allocation that is feasible, Pareto efficient, and IR is, 1 - $h_1$, 2 - $h_2$, 3 - $h_3$. For this allocation, $f$ should retain $h_2$ with agent 2 at $t = 1$. But no online mechanism can correctly decide at $t = 1$ whether to assign $h_1$ or $h_2$ to agent 2 because the preferences of agent 3 are unknown.

In contrast, a serial-dictatorship mechanism, in which agents release ownership of their house upon arrival and receive their most preferred house of those available upon departure (with ties broken at random), is Pareto efficient but not IR. To see that this is Pareto efficient, note that the second agent to be allocated cannot receive a better house without the first agent receiving a worse house. This argument continues inductively. Failure of IR is easy to understand.

Because core is a stronger property than Pareto efficiency and IR, we also know from Lemma 7.1 that no online mechanism can be core-selecting.

### 7.3 **Dynamic Top Trading Cycle Mechanisms**

In this section, we define the static Top Trading Cycle Algorithm (TTCA) and introduce dynamic generalizations, leading to a result that constrains the use of TTCA for dynamic house allocation problems.

#### 7.3.1 **The Static TTCA**

The Top Trading Cycle Algorithm (TTCA) [1] is strategyproof and selects a core allocation for the static house allocation problem.

**Definition 7.6** (Top Trading Cycle Algorithm). Every agent points to its most preferred house. There will be at least one cycle, and the agents on any such cycle (including self-loops) receive the house to which they point. These agents are removed from the system. Now each remaining
agent points to its most preferred remaining house. The procedure continues till there are no houses left to allocate.

---

**Example 7.1 (TTCA).** Consider a problem with 5 agents, with agent \( i \) owning house \( h_i \). Let the preferences of these agents over houses be:

1. \( h_2 \succ h_4 \succ h_3 \succ h_1 \succ h_5 \)
2. \( h_3 \succ h_4 \succ h_5 \succ h_1 \succ h_2 \)
3. \( h_2 \succ h_3 \succ h_1 \succ h_4 \succ h_5 \)
4. \( h_5 \succ h_2 \succ h_3 \succ h_4 \succ h_1 \)
5. \( h_1 \succ h_4 \succ h_2 \succ h_5 \succ h_3 \succ h_5 \)

The agents point to their most preferred house as: 1 → 2 → 3 → 2, 4 → 5 → 1. Now, agent 2 and agent 3 form a cycle, trade, and are removed. In the next round, the agents point to the houses as: 1 → 4 → 5 → 1. This being a cycle, agent 1 gets agent 4’s house, agent 4 gets agent 5’s and agent 5 gets agent 1’s. Thus the final allocation by TTCA is (1 − \( h_4 \), 2 − \( h_3 \), 3 − \( h_2 \), 4 − \( h_5 \), 5 − \( h_1 \)).

---

We are interested in exploring whether or not we can use TTCA as a building block for a family of strategyproof online mechanisms.

### 7.3.2 Online TTCA

We first consider the simplest possible idea, which is to run TTCA in every period in which at least one agent departs and commit to the allocation determined for departing agents. Call this mechanism O-TTCA.

**Lemma 7.2.** O-TTCA is not strategyproof when there are three or more agents.

**Proof:**
Consider an example with 3 agents $N = \{1, 2, 3\}$, with agent $i$ owning house $h_i$, and preferences:

1: $h_3 \succ_1 h_2 \succ_1 h_1$ ($a_1 = 1, d_1 = 3$)
2: $h_1 \succ_2 h_2 \succ_2 h_3$ ($a_2 = 1, d_2 = 1$)
3: $h_1 \succ_3 h_3 \succ_3 h_2$ ($a_3 = 2, d_3 = 3$)

If agent 1 reports truthfully, at $t = 1$, the $1 \leftrightarrow 2$ trade will occur and there will be no trade in periods $t \in \{2, 3\}$. But agent 1 can report $h_3 \succ'_1 h_1 \succ'_1 h_2$. Now no trade occurs in $t = 1$, and at $t = 3$, agents $1 \leftrightarrow 3$ trade and agent 1 will receive $h_3$ which is preferred to $h_2$.

### 7.3.3 Precluding Multiple Trades

A *sample path* $\omega = (\succ, \rho)$ is an instance of the dynamic house allocation problem. At each time $t$, let $\omega(t)$ denote the restriction of $\succ$ and $\rho$ to only those agents with $a_i < t$ and $\omega(t_1, t_2)$ denote the restriction to agents with $a_i \in \{t_1, \ldots, t_2\}$. Sample path $\omega(t, t')$ is a *valid continuation* of $\omega(t)$ if $\omega(t, t')$ is an instance of agents and reported preferences arriving in $[t, t']$. We denote $\omega(t, \infty)$ by $\omega(t_+)$. To avoid corner cases, in this section we consider a generalization of the model where there are $N$ classes of equivalent houses; houses in each class are identical. Agents have strict preference over classes of houses, and multiple agents may own houses in the same class. Agents owning the same house are said to be *similar*, though they might have different preference reports.

For TTCA to remain strategyproof, we need an arbitrary way to break ties among identical houses when looking at cycles. A natural way would be to break ties with the arrival order of agents, with house belonging to later arriving agents having higher priority (and otherwise at random). For example, $A$ owns $h_1$ and arrives before $B$, who owns $h_2$; $h_1$ and $h_2$ are in the same class. If $C$ most prefers that class, in TTCA he would point to $h_2$ first. If that is not available to him, then $C$ would point to $h_1$ and so on.

Given a generalized TTCA with such a tie breaking scheme, we may simulate it with a classical TTCA, cTTCA, where there are only distinct houses. Consider agent $A$ participating in TTCA with preference, $\succ_A = c_1 \succ_A c_2 \ldots$, where $c_i$ is a class of houses. We can construct an agent in cTTCA with preference among houses present in the system: $\succ'_A = c_{i1} \succ'_A c_{i2} \ldots \succ'_A c_{i21} \succ'_A c_{i22} \ldots$, where $c_{i1}$ is the house in class $i$ with the highest tie-break priority, i.e. belonging to the latest arriving agent. Running classical cTTCA on $\{\succ'\}$ is identical to running TTCA on $\{\succ\}$ with the tie breaking scheme and remains strategyproof.
We consider online mechanisms where agents trade through participation in TTCA cycles. In order to be feasible, if a TTCA cycle occurs at time \( t \), then all participating agents must be present at \( t \). An agent may participate in multiple TTCA cycles. However the example below shows how the ability to participate in multiple TTCA cycles can easily create incentives for agents to misreport their preferences. This is familiar from Kurino [33].

**Example 7.2.** Consider a scenario with three agents \((A, h_A), (B, h_B), \) and \((C, h_C)\). \( A \) arrives and departs in period 1 and reports preference \( h_C \succ_A h_A \succ_B h_B \). \( B \) arrives and departs in period 2 and reports preference \( h_A \succ_B h_B \succ_B h_C \). \( C \) arrives in period 1 and departs in period 2 and the mechanism allows \( C \) to participate in TTCA at both times. Suppose \( C \) has true preference \( h_B \succ_C h_C \succ_C h_A \). If \( C \) reports his true preference, then there would be no trade for any agent. But if \( C \) misreports \( h_B \succ_C h_A \succ_C h_C \), then he can obtain \( h_A \) in the first round, which he can use to obtain \( h_B \) in the second round.

In the rest of this section, we restrict our consideration to mechanisms that allow each agent to participate in at most one TTCA. Given any sample path \( \omega \), we may unambiguously state the participation time of an agent \( A \), denoted \( t(\succ_A) \) for report \( \succ_A \) as the time (if any) when it participates in TTCA. An agent present at time \( t \) and who has not participated in TTCA yet is said to be available. This is still a rich domain of mechanisms; in which TTCA an agent participate in, can depend on his arrival/departure and preference report.

A mechanism is simple if given a scenario \( \omega(0, t) \) and agent \((A, h_0, \succ_A, a, d)\) present and available at \( t \), there exists a set of agents \( \Lambda = \{i, h_i, \succ_i, a_i > t, d_i\} \), called the perfect match set for \( A \) in scenario \( \omega(0, t) \), such that (a) if a continuation \( \omega(t+) \) contains \( \Lambda \), then \( A \) receives his most preferred house according to \( \succ_A \) under the scenario \( \omega(0, t) + \omega(t+) \), and (b) if \( B \) is present in \( \omega(0, t) \) and not similar to \( A \) then \( B \) does not trade with any agent in \( \Lambda \).

We make two immediate observations about simple mechanisms:

- \([O1]\) Given a scenario \( \omega(0, t) \) and any agent \( A \) present and available in period \( t \) and with \( d_A > t \), there exists a continuation \( \omega_1 \) where the perfect match set for \( A \) arrives and \( A \) receives \( h_1 \), his most preferred house.

- \([O2]\) Given a scenario \( \omega(0, t) \), and an agent \( A \) present and available in \( t \) with \( d_A > t \), there exists a continuation where for each of the other present agents, \( B \), not similar to \( A \), a perfect match set for \( B \) arrives, and these are the only arriving agents. In this scenario, \( A \) would have no candidate for trade since all other originally present agents not similar to it would be matched up with perfect match sets and \( A \) cannot trade with another agent’s
perfect match sets by definition. A cannot trade with a similar agent since they have identical houses.

Later we will demonstrate that our mechanisms are simple. Now we state the requirements for simple mechanisms to be strategyproof.

**Lemma 7.3.** If an online house allocation mechanism is strategyproof and simple and agent A participates in TTCA in period \( t(\succ_A) \) for some report \( \succ_A \), then fixing scenario \( \omega(0,t) \) in regard to all agents except A, agent A continues to participate in period \( t(\succ_A) \) for all reports \( \succ'_A \).

**Proof:**
Consider any \( \omega \) (fixing the instance for all agents except A). Let \( t_1 \) be the earliest of \( \{t(\succ'_A)\} \) for all reports \( \succ'_A \) for agent A, and \( \succ_A \) be the corresponding preference report. Let \( \omega(0,t_1) \) be the restriction of \( \omega \) up to \( t_1 \). We want to prove that if there exists \( \succ_A \) with participation time \( > t_1 \), then we can construct a scenario where A has incentive to misreport. We consider various classes of \( \succ_A \):

1. If \( \succ_A \) has \( h_1 \succ_A h_0 \), then A reporting \( \succ_A \) must participate in TTCA at \( t_1 \). Suppose \( \succ_A \) does not participate at \( t_1 \). By [O2], there exists a continuation \( \omega_1(t_1+) \) from \( t_1 \) onward where A ends up keeping his original house \( h_0 \). In this scenario, A would have benefited from misreporting \( \succ_1 \neq \succ_A \) in order to obtain \( h_1 \).

2. Now consider if A has the true preference \( \succ_A = h_2 \succ_A \ldots \succ_A h_1 \succ_A \ldots h_0 \succ_A \ldots \), for any feasible \( h_2 \notin \{h_1, h_0\} \). By above, \( \succ_A \) participates at \( t_1 \). There are two possibilities.

   a. \((A, \succ_A)\) is allocated \( h_2 \) at \( t_1 \). By (1) this implies in particular that all \( \succ'_A \) that rank \( h_2 \) highest must trade at \( t_1 \), and not just preferences of type \( \succ_A \).

   b. \((A, \succ_A)\) is not allocated \( h_2 \) at \( t_1 \). Suppose there exists \( \succ'_A = h_2 \succ'_A \ldots \) that does not trade at \( t_1 \). Then there is a continuation \( \omega_2(t_1+) \) from \( t_1 \) onward where A reporting \( \succ'_A \) receives \( h_2 \) by [O1]. Then A would have benefited from reporting \( \succ'_A \) instead of \( \succ_A \) under this scenario. Since \( h_2 \) is arbitrary, this shows that all preference reports must participate at \( t_1 \).

This shows that a simple strategyproof mechanism cannot use agent’s reported preference to decide at which time to let it participate in TTCA.

It is possible for a simple strategyproof mechanism to use reported preferences to decide in which partition of TTCA to let an agent participate in, so long as the partitions are concurrent and always occur in the same round. Consider a family of such mechanisms as follows: at time \( t \), take all the present agents and arbitrarily divide into two groups P1 and P2 (can be more than two). For a new agent A with reported preference \( \succ \), simulate the TTCA of \( \succ \) in P1 and P2 and see under which partition, A would have received better allocation under \( \succ \) and then let A join that partition. Such mechanisms are strategyproof. Through simulation, we observe empirically that the performance of the mechanism improves when agents are allowed to participate in...
TTCA with a large number of other agents. So in practice, it is more effective to combine the partitions \( P_1 \) and \( P_2 \) into one large partition \( P \). We therefore examine in detail mechanisms where the TTCA an agent participates in does not depend on its reported preference.

### 7.4 Partition Mechanisms

Given the analysis in the previous section, we consider now a special class of mechanisms that ensure that each agent only participates in a single TTCA, and moreover determines the group of agents with which an agent can trade without considering the reported type profile of an agent.

**Definition 7.7** (Partition mechanism). *Partition the agents into groups, \( \{(P_1, t_1), (P_2, t_2), \ldots, (P_k, t_k)\} \), such that all the agents in \( P_j \subseteq N \) are present at \( t = t_j \) and each agent is in exactly one group. The partition is constructed in a way that is independent of the agents’ reported preferences. Run TTCA on trading set \( P_j \) in period \( t_j, j = 1, 2, \ldots, k \).*

**Lemma 7.4.** A partition mechanism is strategyproof.

This follows immediately from the strategyproofness of TTCA given that each agent is placed in a single TTCA event and this placement is independent of its report (and recall that arrival and departure times are not manipulable in our model).

#### 7.4.1 Simple Partition Mechanisms

One simple example of a partition mechanism is DO-TTCA, which runs TTCA only amongst the agents that depart in each period \( t \). This continues until all agents have arrived. At this period, TTCA is run with the remaining agents. One obvious flaw with DO-TTCA is that it forfeits the chance to execute trades amongst a large number of agents that are present but may depart at distinct times.

The *Threshold TTCA* (T-TTCA) is designed to address this problem by allowing large group of agents to trade if they are present simultaneously though depart in different periods. Let \( D(t) = \{i \in N : d_i = t\} \). An active agent in period \( t \) is an agent that is present in the market and has not yet participated in a TTCA. Let \( A(t) \) denote the active agents in period \( t \). The T-TTCA is described in Algorithm 1.

The \( \text{THRSHD} \) parameter of T-TTCA can be selected for a particular probabilistic model of agent preferences and arrival-departures to maximize system performance.
Algorithm 1 T-TTCA

INPUT: \((N, \succ, \rho)\)

OUTPUT: House allocation

1: while some agents are yet to arrive do
2:   At each time slot \(t\)
3:   if \(D(t) \neq \emptyset\) then
4:     if \(|A(t)| > \text{THRSHD}\) then
5:       Execute TTCA with the agents \(A(t)\)
6:       Mark all the agents in \(A(t)\) as inactive.
7:     else
8:       Execute TTCA with the agents \(D(t)\)
9:     end if
10:   end if
11: if all the agents have arrived then
12:   run TTCA with all the present agents
13: end if
14: end while

7.4.2 Stochastic Optimization

Mechanism SO-TTCA adopts a sample-based stochastic optimization method for partitioning the agents. Every agent in \(A(t) \cap D(t)\), is included in the trading group in period \(t\). For any other agent \(i \in A(t) \setminus D(t)\), the decision about whether to include the agent is made based on the solution to \(K\) (sampled) offline scheduling problems. Call this offline routine as \(\text{schedule}\).

The offline problem determines a partition of agents given knowledge of the arrival-departure schedule but without considering agent preferences. A reasonable heuristic first identifies the period in which there are maximum number of agents present, say \(t_1\). Consider these agents as one group in the partition, say \(P_1\), and then recurse on the remaining agents. The intuition is that it tends to be beneficial to move an agent from a smaller trading group into a larger trading group because this leads to more trading options for each participant. An empirical analysis of the rank-efficiency for different methods to construct partitions supports this intuition.

Let \(N_i\) denote the number of times agent \(i\) is scheduled into period \(t\) over these \(K\) offline problems, where for each of these problems, a possible arrival-departure schedule for the \(n_t\) agents still to arrive is sampled from a underlying probabilistic model of the domain. Agent \(i\) is allocated to the trading group in the current period if this count \(N_i\) is greater than \(K(1 - \frac{n_t}{N})\), where \(n_t\) is the number of agents that have arrived up to and including period \(t\). The effect of comparing to \(K(1 - \frac{n_t}{N})\) is that the mechanism is more likely to place an agent into the
trading group as the number of agents still to arrive decreases (and thus the opportunity to trade decreases).

**Algorithm 2 SO-TTCA**

**INPUT:** \((N, \succ, \rho)\)

**OUTPUT:** House allocation

1. while some agents are yet to arrive do
2. At each time slot \(t\)
3. if \(D(t) \neq \emptyset\) then
4. \(S = D(t) \cap A(t)\)
5. Generate \(K\) samples of arrival-departure for the agents still to arrive.
6. For each of these \(K\) scenarios, call schedule routine.
7. For each agent \(i \in A(t) \setminus D(t)\)
8. if \(|\{k \in K \text{ s.t. scheduled period of } i = t\}| > (1 - \frac{n_t}{n}) \cdot K\) then
9. \(S \leftarrow S \cup \{i\}\)
10. end if
11. Execute TTCA with the agents from \(S\)
12. end if
13. if all the agents have arrived then
14. run TTCA with all the present agents
15. end if
16. end while

### 7.4.3 Partition mechanisms are simple

**Lemma 7.5.** The partition mechanisms that we have seen are simple mechanisms.

**Proof:**
We prove the lemma by explicit construction of perfect sets.

- **DO-TTCA:** Given agent \((A, h_0, \succ_A, a, d)\), the perfect match set is a set of \(n'\) identical agents \((A', h', \succ_{A'}, d, d)\) such that \(h'\) is the most preferred house of \(A'\) according to \(\succ_{A'}\), and \(\succ_{A'}\) ranks \(h_0\) the highest and \(h'\) second. Here \(n'\) is the number of agents similar to \(A\) present in the system. In any continuation that contains such \(A'\)'s, \(A\) and \(A'\)'s will be in the same TTCA at time \(d\) and one \(A'\) will trade with \(A\). Only agents similar to \(A\) may trade with an \(A'\).

- **T-TTCA:** Given agent \((A, h_0, \succ_A, a, d)\) and any scenario \(\omega(0, t)\) such that \(A\) is available at \(t\), with \(t < d\). Fix some threshold, THRSHD. A perfect match set for \(A\) is a set of \(m\)
agents \(\{i, h_i, \succ_i, t+1, t+1\}\) who arrive and depart in time \(t+1\), where \(m > \text{THRSHD}\). The first \(n'\) of these agents are identical to \(A'\) constructed for DO-TTCA. All other agents rank their own house \(h_i\) first. They are dummy agents who do not trade and only serve to trigger the threshold. In any continuation that contains \(\{i\}\), at time \(t+1\), there are more available agents than THRSHD, and hence all present agents participate in TTCA. Agents \(A\) and one \(A'\) would trade and \(A\) gets his top house \(h'\).

- **SO-TTCA**: Given agent \((A, h_0, \succ_A, a, d)\) and any scenario \(\omega(0, t)\) such that \(A\) is available at \(t\), with \(t < d\). Suppose there are \(n_t\) agents still yet to arrive. A perfect match for \(A\) is the same set of agents \(\{i\}\) as constructed in the case of T-TTCA, with \(m = n_t\). In any continuation containing \(\{i\}\), there are no additional future arrivals. Hence in the offline scheduling problem, all agents present at \(t + 1\) would be put into the same partition. In this TTCA cycle, \(A\) would trade with \(A'\) and obtain \(h'\).

### 7.5 Simulation Results

We perform experiments to evaluate the rank-efficiency of O-TTCA, DO-TTCA, T-TTCA, and SO-TTCA under various simulated environments. The preferences over houses are sampled uniformly at random for all the agents except in environment [E4]. For each environment and each mechanism, all results are averaged over 2000 random problem instances. By “waiting time” (or patience) we mean the number of periods an agent is present in the market.

- **[E1]** A Poisson process with arrival rate \(\lambda\) is run until \(n\) agents arrive. Each agent’s waiting time is exponentially distributed with parameter \(\mu\). We adopt \(\lambda = \frac{n}{8}\), and \(\mu = 0.01\lambda\).

- **[E2]** Same as [E1] except \(\mu = 0.1\lambda\)

- **[E3]** Arrival time for every agent is uniformly distributed on \(\{1, \ldots, T\}\). The departure time for agent \(i\) is uniformly distributed on \(\{a_i, a_i + \frac{T}{8}\}\). We adopt \(T = 30\) and if it happens that, \(d_i > T\), we put \(d_i = T\).

- **[E4]** Non-uniform preferences: Some houses are in more demand than the other. We first associate a popularity index with each house, with the popularity index for house \(h_j\) defined as the probability density \(\phi(x)\) of a Normal distribution with mean 1, standard deviation 0.3, and evaluated at \(x = \frac{2j}{n}\). Given a popularity index assigned to each house, we generate a preference profile for agent \(i\) by sampling houses according to popularity, with houses sampled without replacement with probability proportional to the popularity of a house. The sample order defines an agent’s preferences, with the first house the most-preferred,
Figure 7.1: Rank-efficiency against the number of agents in environment [E1] for patient agents with $\lambda = n/8$ and $\mu = 0.01\lambda$.

the second house sampled the second most-preferred, and so on [SI]. The arrival-departure are set as in [E1].

In T-TTCA, we varied the THRSHD parameter around the expected number of the agents arriving in each period, and experimentally observed that the rank-efficiency is optimized with a threshold that is set to the expected number of agents arriving in each period. For [E3], this is $THRSHD = \frac{n}{T}$ and for [E1], [E2] and [E4] this is $THRSHD = \lambda$.

We plot the rank-efficiency, for environments [E1] to [E4] in Figures respectively. In each figure, we also include the rank-efficiency for the O-TTCA mechanism as a reference. Recall that the O-TTCA mechanism is not strategyproof, and significantly less constrained in that it allows for trading in every period. We observe that when agents arrive into the system by a Poisson process, SO-TTCA improves rank-efficiency over DO-TTCA by 4-10% depending upon the waiting times of the agents. Specifically, when $\mu = 0.1\lambda$, SO-TTCA improves by 4% and for $\mu = 0.01\lambda$ it improves over DO-TTCA by 10%.

T-TTCA performs well only when the waiting times are lower (i.e., for higher $\mu$, [E2]). When the waiting times are high (in experiment [E1]), T-TTCA fails to capture the future periods in which there may a be large number of agents accumulated. Even if there are many active agents

182
Figure 7.2: Rank-efficiency against the number of agents in environment [E2] for less patient agents with $\lambda = n/8$ and $\mu = 0.1\lambda$.

Figure 7.3: Rank-efficiency against the number of agents in environment [E3] with a uniform arrival-departure model.
Figure 7.4: Rank-efficiency against the number of agents in environment [E4] with preferences that are correlated across agents and depend on the popularity index of a house.

in the current period, they may be patient and willing to wait for some future period.

In [E3], the SO-TTCA and T-TTCA mechanisms have much better performance than DO-TTCA, improving rank-efficiency by 15-20% for large numbers of agents. The performance of SO-TTCA and T-TTCA are almost identical under this model. With the waiting time being relatively small as compared to experiments in [E1] and [E2], T-TTCA is able to capture the periods in which more agents are present simultaneously. In general, SO-TTCA is more robust to different arrival-departure models than T-TTCA. In addition, SO-TTCA requires no tuning, whereas T-TTCA requires that the threshold is appropriately set.

7.6 Conclusions

In this chapter we have considered a dynamic version of the house allocation problem. We identify a trade-off between efficiency and individual rationality in the online version of the house allocation problem, and identify a requirement that agents cannot trade with different subsets of agents by changing their type report. We consider a family of partition mechanisms in which an agent’s trading group is determined without regard to its type. A clear benefit is established for T-TTCA and SO-TTCA over a method that performs trading upon the departure of one or more agents, and the stochastic optimization approach (SO-TTCA) outperforming the
threshold approach (T-TTCA) for Poisson arrival processes.

An interesting aspect of the use of stochastic optimization is that it determines which subset of agents should trade but not what trade should occur, and without consideration of specific agent types. This preserves strategyproofness.

The most interesting immediate next steps would be to consider a generalization in which agents can misreport arrival and departure, while continuing to explore the role for stochastic optimization in identifying useful trading groups.
Chapter 8

Dynamic Allocation Mechanisms for Assignment of Heterogeneous Objects

Agents have arrivals and departures and strict preferences over items. Strategyproofness requires the use of an arrival-priority serial-dictatorship (APSD) mechanism, which is ex post Pareto efficient but has poor ex ante efficiency as measured through average rank-efficiency. We introduce the scoring-rule (SR) mechanism, which biases in favor of allocating items that an agent values above the population consensus. The SR mechanism is not strategyproof but has tolerable manipulability in the sense that: (i) if every agent optimally manipulates, it reduces to APSD, and (ii) it significantly outperforms APSD for rank-efficiency when only a fraction of agents are strategic. The performance of SR is also robust to mistakes by agents that manipulate on the basis of inaccurate information about the popularity of items.

8.1 Introduction

In this chapter, we are interested in assignment problems to a dynamic agent population. For example, consider the allocation of tasks to agents that arrive and depart and have preferences on tasks and a time window within which they can be allocated a task. The problem can also be one of resource allocation, where agents arrive and demand access to a resource before departure. Motivating domains include those of car pooling in which the agents are commuters and the resources are seats in shared cars, or science collaborators in which the agents are people looking to perform useful work for the community. We consider problems in which it is undesirable to use money; e.g., because of community norms, legal constraints, or inconvenience. Our contributions are as follows.
8.1.1 Contributions

For static assignment problems without money, the class of strategyproof mechanisms is more or less restricted to serial dictatorships in which each agent in turn selects its most preferred item of the unclaimed items. For the dynamic problem considered here, the additional requirement of strategyproofness with respect to arrival (i.e., preventing manipulation by reporting later arrival) necessitates the use of an arrival-priority serial dictatorship (APSD). Although APSD is ex post Pareto efficient, it is unsatisfactory in another, more refined sense. If we consider ex ante efficiency, that is average utility received by agents under the mechanism, then its performance is quite poor. For risk-neutral agents for which the difference in utility for any pair of consecutive items in a preference ordering is constant, then the expected average rank (or rank-efficiency) measures the ex ante efficiency. APSD has poor rank-efficiency because an early arrival may pick its most preferred item over its second most preferred item even if this leaves a later arrival with its least preferred item rather than its most preferred item.

This motivates our study of tolerable manipulability, an agenda for computational mechanism design first suggested by Feigenbaum and Shenker [84]. We introduce the scoring-rule (SR) mechanism [81], which biases in favor of allocating items that an agent values atypically highly. The SR mechanism is not strategyproof but has tolerable manipulability in the following sense: (i) if every agent optimally manipulates SR then it reduces to APSD, and thus the performance of the only strategyproof mechanism; and (ii) the SR mechanism significantly outperforms APSD for rank-efficiency when only a fraction of agents are strategic. The performance of SR is also robust in the following sense: the optimal manipulation is a dominant strategy, and thus invariant to strategies of other agents, and SR continues to outperform APSD even when agents have inaccurate information about the distribution on preferences in the population and thus the rules of the SR mechanism.

Our simulation results on the SR mechanism demonstrate that for 10 agents, SR has 10% greater rank-efficiency than APSD when all agents are truthful and non-strategic. When 5 of the 10 agents are strategic, SR still maintains 5% greater rank-efficiency than APSD. Furthermore, the advantage of SR over APSD increases as the number of agents increase. With 25 agents, SR has 19% greater rank-efficiency over APSD. To further benchmark the performance of the SR mechanism we also compare against the rank-efficiency of a sample-based stochastic optimization algorithm [79], namely, Consensus. When all the agents are truthful, SR outperforms Consensus by 4% when there are 10 agents. Even with up to four out of the ten agents acting strategically SR outperforms Consensus.
8.1.2 Related Work

For the house allocation problem, which is a totally static assignment problem in which agents and items are both fixed and each agent has strict preferences over the assignment of one item (or house), Svensson [26] establishes that the only non-bossy, neutral and strategyproof mechanisms (defined in the next section) are serial dictatorships. Papai [27] relaxes the requirement of neutrality and achieves a richer characterization.

Abdulkadiroglu and Loertscher [85] study a dynamic house allocation problem that is quite different from our problem. It is a two period problem in which the agents and items are fixed and dynamics occur because agent preferences in period two are unknown in period one. Kurino [33] considers a dynamic house allocation problem with fixed items and a dynamic (overlapping generations) agent population. His problem is again different from ours because each agent demands an item in every period. Another difference is that items are not consumed but rather returned to the market when an agent departs.

We are not aware of any prior work on tolerable manipulability in the context of dynamic mechanism design. Othman and Sandholm [86] define the notion of manipulation-optimal mechanisms, in which: (i) the mechanism is undominated by any strategyproof mechanism when all agents are rational; and (ii) the performance is better than any strategyproof mechanism if any agent fails to be rational in any way. Their results are essentially negative, in that a manipulation-optimal mechanism is impossible whenever there is an agent and a joint type profile of other agents, such that there are two types of the agent for which its best-response is to be non-truthful and moreover its optimal misreport depends on its true type. This is a very weak condition and holds for our problem. Othman and Sandholm [86] demonstrate positive results only in the case of an agent with two possible types. See also Conitzer and Sandholm [87] who first introduced an example to demonstrate the existence of a manipulation-optimal mechanism. Our definition of tolerable manipulability is weaker than that of manipulation-optimal mechanisms, in that we retain (i) but replace (ii) with (ii') the performance is better than any strategyproof mechanism if a sufficient fraction of agents fail to be rational by being truthful. With this approach we are able to achieve positive results for an interesting problem domain. In a dynamic auction setting with money, Lavi and Nisan [88], in studying the performance of mechanisms for a class of (set-Nash) equilibria, consider another form of tolerable manipulability in the sense that no strategyproof online algorithm has good properties but they demonstrate an online algorithm with good properties as long as agents play strategies from a set that is closed under rational behavior.
8.2 The Model

There are \( A = \{1, \ldots, n\} \) agents, \( I = \{I_1, \ldots, I_m\} \) items, and each agent \( k \in A \) has an arrival \( \alpha_k \in T \), departure \( \beta_k \in T \), demands one item and has preferences \( \phi_k \in \Phi \) on items, where \( T = \{1,2,\ldots\} \) is the set of discrete time periods and \( \Phi \) the set of preferences. Altogether this defines an agent’s type \( \theta_k = (\alpha_k, \beta_k, \phi_k) \in \Theta \) where \( \Theta \) is the set of types. Preferences \( \phi_k \) are strict and define a rank \( r(k,j) \in \{1, \ldots, m\} \), where \( r(k,1) \) is the index of the most preferred item and so on. We write \( \phi_k : I_{r(k,1)} \succ_k I_{r(k,2)} \succ_k \ldots \succ_k I_{r(k,m)} \) to denote an agent’s preferences. Each agent \( k \) only cares about its allocated item in interval \( \{\alpha_k, \ldots, \beta_k\} \), and is indifferent to the allocation to other agents. We consider a fixed set of items, all available from period one.

In a direct-revelation mechanism, the message space allows an agent \( k \) to report \( \theta'_k = (\alpha'_k, \beta'_k, \phi'_k) \neq \theta_k \) in some period \( t = \alpha'_k \). We assume \( \alpha'_k \geq \alpha_k \) and \( \beta'_k \leq \beta_k \), which together with \( \beta'_k \geq \alpha_k \) implies that \( t \in \{\alpha_k, \ldots, \beta_k\} \). The arrival assumption \( (\alpha'_k \geq \alpha_k) \) is standard in problems of dynamic mechanism design [77], and can be motivated easily if the arrival is the period in which an agent realizes its demand or discovers the mechanism. The departure assumption \( (\beta'_k \leq \beta_k) \) is made for convenience; all our mechanisms allocate upon reported departure in any case and so an agent never has a useful strategy that involves reporting \( \beta'_k > \beta_k \).

A mechanism is defined by a function \( f : \Theta^n \rightarrow X \), where \( x = (x_1, \ldots, x_n) \in X \) denotes an allocation of items \( x_k \in I \) to each agent \( k \) and \( X \) is the set of feasible allocations, such that \( x_k \neq x_\ell \) if \( k \neq \ell \). We only consider deterministic mechanisms. Let \( f_k(\theta) \in I \) denote the allocation to agent \( k \) in period \( t \in \{\alpha_k, \ldots, \beta_k\} \), where \( \theta = (\theta_1, \ldots, \theta_n) \) is the joint type profile. Let \( \theta_{-k} = (\theta_1, \ldots, \theta_{k-1}, \theta_{k+1}, \ldots, \theta_n) \) and \( f(\theta'_k, \theta_{-k}) \) is shorthand for \( f(\theta_1, \ldots, \theta_{k-1}, \theta'_k, \theta_{k+1}, \ldots, \theta_n) \).

To meet the requirements of an online mechanism, which cannot know future type reports, we require \( f_k(\theta_k, \theta_{-k}) = f_k(\theta'_k, \theta_{-k}) \) whenever \( \theta_{\leq k} = \theta'_{\leq k} \), for all \( k \in \{1, \ldots, n\} \), where \( \theta_{\leq t} \) is the restriction of type profile \( \theta \) to include only those agents that arrive no later than period \( t \).

Some desiderata of online assignment mechanisms:

- **Strategyproof**: Mechanism \( f \) is strategyproof if \( f_k(\theta_k, \theta_{-k}) \succ_k f_k(\theta'_k, \theta_{-k}) \) for all \( \theta'_k = (\alpha'_k, \beta'_k, \phi'_k) \) where \( \alpha'_k \geq \alpha_k \) and \( \beta'_k \leq \beta_k \), all \( k \), and all \( \theta \). Truthful reporting is a dominant-strategy equilibrium.

- **Non-Bossy**: Assignment mechanism \( f \) is non-bossy if \( (f_k(\theta_k, \theta_{-k}) = f_k(\theta'_k, \theta_{-k})) \Rightarrow (f(\theta_k, \theta_{-k}) = f(\theta'_k, \theta_{-k})) \), for all \( \theta'_k = (\alpha'_k, \beta'_k, \phi'_k) \) where \( \alpha'_k \geq \alpha_k \) and \( \beta'_k \leq \beta_k \), all \( k \), and all \( \theta \).

- **Neutrality**: Assignment mechanism \( f \) is neutral if \( f_k(\theta) = \pi_k(\pi_k(\theta)) \) for all agents \( k \), type profiles \( \theta \), and item permutations \( \pi \), where \( \pi : \{1, \ldots, m\} \rightarrow \{1, \ldots, m\} \) is bijective, \( \pi^{-1} \) is the inverse, and \( \pi_k(\theta) = \theta' \) is a type profile induced in the straightforward way by
the permutation on items, so that \( r'(k, j) = r(k, \pi(j)) \) for all agent \( k \) and all items \( I_j \), with arrival and departure times unchanged.

- **Pareto Efficient.** Assignment mechanism \( f \) is *ex post* Pareto efficient if for all \( \theta \in \Theta^n \), then there is no feasible allocation \( x' \) that is weakly preferred by all agents to \( x = f(\theta) \) and strictly preferred by at least one agent.

### 8.3 Strategyproof Mechanisms

A serial dictatorship has a priority ranking \( h : \{1, \ldots, n\} \to \{1, 2, \ldots, n\} \), such that agents are ordered \( h(1), h(2), \ldots \), and assigned the most preferred item still available given the allocation to higher priority agents.

**Definition 8.1.** An online serial dictatorship is a serial dictatorship in which \((h(i) < h(j)) \Rightarrow \alpha_i \leq \beta_j\).

This follows from the online setting and what we term *schedulability*, i.e., we can only guarantee agent \( i \) a higher priority in the sense of serial dictatorship than agent \( j \) if it arrives before agent \( j \) departs.

**Lemma 8.1.** Let \( f \) be a deterministic, strategyproof, neutral, non-bossy, online assignment mechanism. Then \( f \) must be an online serial dictatorship.

**Proof:**

Let \( f \) be any strategyproof, neutral, non-bossy online mechanism and given any reported schedule of agent arrival and departures, \( \sigma = \{\alpha'_1, \beta'_1, \ldots, \alpha'_n, \beta'_n\} \). Now \( f|\sigma(\phi_1, \ldots, \phi_n) = \hat{f}(\theta_1, \ldots, \theta_n) \) defines an off-line mechanism \( \hat{f} \), mapping a preference profile to an allocation of items. The restricted mechanism \( f|\sigma \) is non-bossy since \( f_k|\sigma(\phi_k, \phi_{-k}) = f_k|\sigma(\phi'_k, \phi_{-k}) = f_k(\theta_k, \theta_{-k}) = f_k(\theta'_k, \theta_{-k}) \). It is also neutral since any items permutation of \( f|\sigma \) is also an items permutation of \( f \). Lastly, \( f|\sigma \) is strategyproof since any profitable misreport of preferences would carry over into a manipulation of \( f \). Therefore by Svensson [26], \( f|\sigma \) must be an offline serial dictatorship. For, \( f \) to be online feasible, any agent \( k \) should receive an object before departure. So, all the agents having higher priority than agent \( k \) must be allocated before agent \( k \)'s departure. and the priority structure in \( f \) must be such that, if \((h(i) < h(j))\), then \( \alpha_i \leq \beta_j \). That is, \( f \) is online serial dictatorship.

**Definition 8.2.** The Arrival-Priority Serial Dictatorship (APSD) or Greedy mechanism assigns priority by arrival, with an earlier arrival having a higher priority and with ties broken arbitrarily. An item is assigned upon an agent’s arrival, and released (and thus allocated) upon departure.
Theorem 8.1. A deterministic online mechanism is strategyproof, neutral and non-bossy if and only if it is the APSD mechanism.

Proof:

($\Leftarrow$) It is immediate to check that APSD is neutral and non-bossy. To establish that it is strategyproof, note that since it is a serial dictatorship an agent $k$ should report its true preference $\phi_k$ whatever its $(\alpha'_k, \beta'_k)$ report. Moreover, for any reported arrival $\alpha'_k$, the outcome is invariant to its reported departure $\beta'_k \geq \alpha'_k$ (still with $\beta'_k \leq \beta_k$). Reporting a later arrival $\alpha'_k > \alpha_k$ only reduces its priority rank and reduces its choice set of items.

($\Rightarrow$) Consider a strategyproof mechanism that is neutral and non-bossy, and thus an online serial dictatorship by Lemma 8.1, but is not an arrival-priority serial dictatorship. In particular, there is some preference profile $\phi \in \Phi^n$ and some agent arrival/departure schedule such that $\alpha_i < \alpha_j$ but $h(i) > h(j)$ for some pair of agents $i, j$. To be schedulable, we must also have $\alpha_j \leq \beta_i$. But now, agent $i$ can report $\beta'_i = \alpha_i < \alpha_j$, and force $h(i) < h(j)$, again by schedulability. Now suppose in particular that $\phi_i = \phi_j$, so that the item agent $j$ receives when agent $i$ is truthful is strictly preferred by agent $i$ to its own allocation. Agent $i$ will receive an item that is at least as good as that allocated to agent $j$ when agent $i$ is truthful, and thus this is a useful deviation.

An online serial dictatorship is ex post Pareto efficient because agent preferences are strict. For agents except the top priority agent, a change in allocation to a more preferred item would require that an agent with higher priority be allocated a less preferred item than its current allocation.

Ex ante efficiency. A stronger performance requirement is that of ex ante efficiency. Informally, a mechanism is ex ante Pareto efficient if there is no function from types to allocations that induces higher expected utility than the distribution induced by a mechanism $f$ and the type distribution. In general, to make any claims about ex ante efficiency in a setting with ordinal preferences, we need to introduce some additional assumptions about agent utility functions. Following Budish and Cantillon [78], we assume risk neutral agents with a constant difference in utility across items that are adjacent in their preference list. Let $R(x) = \frac{1}{n} \sum_k r(k, x_k)$ define the average-rank score for allocation $x \in X$, where $r(k, x_k)$ is the agent $k$’s rank for his allocated item $x_k$. Under this assumption, then the expected average-rank score measures ex ante efficiency.

8.4 Heuristic Allocation Methods

In this section, we compare the performance of APSD with two other algorithms, the scoring-rule (SR) and Consensus algorithms, considering only truthful inputs and without concern to
manipulations. By assuming that ties are broken arbitrarily if multiple agents arrive together, we can keep the presentation simple and consider the case of only one agent arriving in each period.

**The Scoring-rule Algorithm.** The general idea of the scoring-rule (SR) algorithm is to bias in favor of allocating an item to an agent that he values above the population consensus. For this, we define the score of item $I_j$, given that there are $m$ items, as $S(I_j) = \sum_{k=1}^{m} Pr(j,k)k$, where $Pr(j,k)$ is the probability that item $I_j$ is ranked in $k$th position by a random agent. For a uniform preference model, $Pr(j,k) = 1/m$ for all items $I_j$, but this will typically be skewed in favor of some items over. The SR algorithm works as follows:

1. Suppose that agent $k$ arrives, with $\phi_k : I_{r(k,1)} \succ_k I_{r(k,2)} \succ_k \ldots \succ_k I_{r(k,m)}$.

2. Assign item $I_j$ to agent $k$, where $j \in \arg\min_{\ell \in \text{avail}} [r^{-1}(k,\ell) - S(I_\ell)]$, where avail is the set of available items given prior allocations and $r^{-1}(k,\ell)$ is the inverse rank function, giving agent $k$’s rank for item $I_\ell$. Release item $I_j$ to agent $k$ upon departure.

If agent $k$ has a less popular $I_\ell$ item as one of his top choices (low $r^{-1}(k,\ell) - S(I_\ell)$), then SR will tend to allocate this item and save popular items for other agents; this is how the algorithm is designed to improve \textit{ex ante} efficiency.

**The Consensus algorithm.** The Consensus algorithm is a sampled-based method of stochastic optimization [79].

1. When agent $k$ arrives, generate $L$ samples of the types of possible future agent arrivals; i.e., $L$ samples of the types of $n-k$ agents where agent $k$ is the $k$th agent to arrive. Let $A_\ell$ denote the set of agents in the $\ell$th such sample.

2. For each sample $\ell \in \{1, \ldots, L\}$, compute an allocation $x^*_\ell$ of the available items to agents $\{k\} \cup A_\ell$ to minimize the average rank of allocated items.

3. Determine the item $I_j$ that is allocated to agent $k$ most often in the $L$ solutions $\{x^*_1, \ldots, x^*_L\}$, breaking ties at random, and assign this item to the agent. Release the item to the agent upon its departure.

**8.4.1 Rank Efficiency Analysis**

To evaluate the rank-efficiency of APSD, SR, and the Consensus algorithm we adopt a simple model of the distribution on agent preferences that is parameterized by $(p_1, \ldots, p_m)$, with $p_j > 0$ to denote the popularity of item $I_j$. Given this, we have the following generative model for \textit{weighted-popularity} preferences: For each agent $k$: 

193
• Initialize \( R_1 = \{1, 2, ..., m\} \). In the first round, an item is selected at random from \( R_1 \) with probability \( \frac{p_j}{\sum_{i \in R_1} p_i} \). Let \( I_{k_1} \) denote the item selected in this round.

• Let \( R_2 = R_1 \setminus \{k_1\} \) describe the remaining items. In this round Item \( I_j \) is selected from \( R_2 \) with probability \( \frac{p_j}{\sum_{i \in R_2} p_i} \).

• Continue to construct a preference rank \( \phi_k : I_{k_1} \succ_k I_{k_2} \succ_k ... \succ_k I_{k_m} \) for the agent.

In our experiments, we adjust the popularity profile on items by adjusting a similarity parameter, \( z \). For each item \( I_j \), we set \( p_j = \Psi_z(\frac{2j}{m}) \), where there are \( m \) items and \( \Psi_z \) is the density function for Normal distribution \( N(1, z) \). High similarity corresponds to an environment in which all items are of similar popularity. Low similarity corresponds to an environment in which a few items are significantly more popular than other items. Rank-efficiency is evaluated over 1000 independent simulations, with the average rank for a run normalized to the rank of the optimal off-line allocation that minimizes the average rank based on true preferences. Smaller rank-efficiency is better, and 1 is optimal.

Figure 8.1 provides representative results for 10 agents and 10 items.\(^1\) SR outperforms APSD for all similarities, with improvement of at least 10\% for similarity \( z \leq 0.3 \). In absolute terms, an improvement of 10\% in (normalized) rank efficiency for 10 agents and 10 items corresponds to an average absolute rank improvement of 0.4; so, roughly equivalent to an agent expecting to receive an average improvement of rank position of one every other time. For high similarity the performance of all algorithms becomes quite similar. Figure 8.2 considers increasing the number of agents, holding the number of items equal to the number of agents, and for similarity 0.3. The SR algorithm again has the best average rank-efficiency for all numbers of agents. For \( n = 15 \) and \( n = 30 \), the performance of APSD (or Greedy) is 40\% and 60\% worse, respectively, than the off-line solution, while SR is 20\% and 35\% worse.

8.4.2 What is The Scoring Rule Doing Right?

When all items have equal popularity, the SR algorithm agrees exactly with APSD. In this case, the score is equal for every item and \( \text{arg min}_{\ell \in \text{avail}} [r^{-1}(k, \ell) - S(I_\ell)] = \text{arg min}_{\ell \in \text{avail}} r^{-1}(k, \ell) \), and selects the item for agent \( k \) with the smallest rank. Now consider a simple scenario where there are only two agents \( \{1, 2\} \) and two items \( \{I_1, I_2\} \). Agent 1 arrives first and has preference ranks \( \phi_1 : I_1 \succ_1 I_2 \). Suppose that SR allocates \( 1 \rightarrow I_1 \), so that \( r(1, 1) - S(I_2) < r(1, 2) - S(I_1) \). Because the score of an item is precisely the expected rank for allocation that item, \( S(I_1) = \mathbb{E}_{\phi_2} r(2, 1) \) and \( S(I_2) = \mathbb{E}_{\phi_2} r(2, 2) \). Now, we see that SR allocates \( I_1 \) to agent 1 exactly when

\(^1\)In all experiments, agents arrive in sequence, and this sequencing is sufficient to simulate; note that the performance of all algorithms are invariant to departure.
Figure 8.1: Rank-efficiency under truthful agents as the similarity in item popularity is adjusted, for 10 items and 10 agents.

\[ r(1,1) + S(I_1) = r(1,1) + E_{\phi_2}r(2,1) < r(1,2) + E_{\phi_2}r(2,2) = r(1,2) + S(I_2), \]

and makes the allocation decision to minimize total expected rank. To test this intuition, we track the first occasion when SR and APSD make different allocations. Let SR allocate \( I_1 \) while APSD allocates \( I_2 \). In being greedy, APSD achieves an average rank of 1.4 for \( I_2 \), but SR still achieves 1.62. On the other hand, while SR achieves an average rank of 3.54 on \( I_1 \), APSD struggles and achieves only 4.92 on this item. Looking at the score, we see that \( S(I_1) = 6.43 > S(I_2) = 3.09 \). SR made the right decision in allocating the less popular \( I_1 \) when a good opportunity arises while holding on to the more popular \( I_2 \).

### 8.5 Tolerable Manipulability

We focus now on the SR algorithm, which is effective in meeting the \textit{ex ante} efficiency performance target but is manipulable by agents. The question that we ask in this section is whether the mechanism is tolerably manipulable?

**Example 8.1.** Consider a simple example with three items \( \{I_1, I_2, I_3\} \). Most agents agree that \( I_1 \) is the best, \( I_2 \) is the second best, and \( I_3 \) is the least desirable. Therefore, \( S(I_1) = 1 + \epsilon, \)
\( S(I_2) = 2 \) and \( S(I_3) = 3 - \epsilon, \) for \( \epsilon > 0 \) small. Suppose the first agent to arrive, agent 1, has
preferences $\phi_1 : I_1 \succ_I I_3 \succ_I I_2$. The SR mechanism computes $\{1 - S(I_1), 2 - S(I_3), 3 - S(I, 2)\} = \{-\epsilon, -1 + \epsilon, 1\}$ and allocates $I_3$ to agent 1. This is the right decision. With high probability, the agents 2 and 3 will have the common preference $\phi_2(= \phi_3) : I_1 \succ I_2 \succ I_3$. Then the SR mechanism would have allocation $1 \rightarrow I_3, 2 \rightarrow I_1, 3 \rightarrow I_2$ for an average rank of 1.67. The APSD mechanism would have allocated $1 \rightarrow I_1, 2 \rightarrow I_2$ and $3 \rightarrow I_3$ for an average rank of 2. However, agent 1 could misreport his preferences to be $\phi'_1 : I_1 \succ'_1 I_2 \succ'_1 I_3 \ldots$. This is the right decision. With high probability, the agents 2 and 3 will have the common preference $\phi_2(= \phi_3) : I_1 \succ I_2 \succ I_3$. Then the SR mechanism would compute $\{1 - S(I_1), 2 - S(I_2), 3 - S(I, 3)\} = \{-\epsilon, 0, \epsilon\}$ and allocate $1 \rightarrow I_1$, which is his top choice. Thus, the SR mechanism is not strategyproof.

Let us now analyze the optimal manipulation for an agent $k$ in the SR mechanism. Suppose $\phi'_k = I_1 \succ'_k I_2 \succ'_k \ldots \succ'_k I_m$ yields the allocation of $I_j$, and that this is the best obtainable item for agent $k$ under SR. Clearly, $\phi''_k : I_j \succ''_k I_1 \succ''_k I_2 \ldots \succ''_k I_{j-1} \succ''_k I_{j+1} \succ''_k \ldots I_m$ also leads to the same allocation. In fact, we show that if the agent can win item $I_j$ with some misreport, then it can always win the item by placing it first, followed by the claimed items, followed by the other items in order of ascending score. Let $avail$ denote the set of available items, and $claimed$ the rest of the items. We propose the following manipulation algorithm for agent $k$:

1. Select the most preferred item $I_1 \in avail$. 

---

Figure 8.2: Rank-efficiency under truthful agents as the population increases, for 10 items, 10 agents and similarity of 0.3.
2. Consider a preference profile $\phi'_k$, with items ordered, from most preferred to least preferred, as

$$[I_1, \text{claimed, sorted}(\text{avail} \setminus \{I_1\})] \quad (8.1)$$

This reports $I_1$ as the most preferred item, followed by the claimed items in any order, followed by the rest of the items sorted in ascending order of score.

3. Apply the SR calculation to $\phi'_k$. If SR allocates $I_1$, then report $\phi'_k$. Else, repeat steps 1-2 for the second preferred item, $I_2$, third preferred item and so on until an item is obtained, and report the corresponding $\phi'_k$.

**Lemma 8.2.** The preference report generated is the agent’s best response to the current state of the SR mechanism and given a particular set of item scores.

**Proof:**

It is sufficient to show that for any $I_j$, the ordering of the remaining items $I_l$ will be optimal in the sense of minimizing their adjusted score $r^{-1}(k, l) - S(I_l)$. To see this, consider two adjacent items $\{I_1, I_2\}$ (not equal to $I_j$) in a reported preference order, for which $S(I_2) < S(I_1)$. Let $u$ denote the reported rank of the item in the first position. We claim that $\min(u - S(I_2), u + 1 - S(I_1)) > \min(u - S(I_1), u + 1 - S(I_2))$, and therefore it is best to report $I_2$ before $I_1$. This is by case analysis. Case (i): $S(I_1) - S(I_2) \leq 1$. Now we have $u - S(I_2) \leq u + 1 - S(I_1)$, and also $u - S(I_1) \leq u + 1 - S(I_2)$. Therefore, $\min(u - S(I_2), u + 1 - S(I_1)) = u - S(I_2) > u - S(I_1) = \min(u - S(I_1), u + 1 - S(I_2))$. Case (ii): $S(I_1) - S(I_2) > 1$. Now we have $u + 1 - S(I_1) \leq u - S(I_2)$ and $u - S(I_1) \leq u + 1 - S(I_2)$. Then, $\min(u - S(I_2), u + 1 - S(I_1)) = u + 1 - S(I_1) > u - S(I_1) = \min(u - S(I_1), u + 1 - S(I_2))$. Finally, it is easy to see that it is always just as good to list the unavailable items immediately after item $I_j$.

**Theorem 8.2.** If the difference in scores between successive items in SR is less than 1, and all agents are strategic, then the allocation under SR is identical to that under the APSD mechanism.

**Proof:**

We show that agent $k$ that follows its best response will receive its most preferred available item. The hardest case is when the agent’s top ranked item is also the minimum score item. Label this item $I_1$. The optimal reported preference order is to sort the items in order of ascending score, e.g. $\phi'_k : I_1 >'_k I_2 >'_k \ldots >'_k I_m$. By assumption about consecutive scores, we have $1 - S(I_1) < 2 - S(I_2) < 3 - S(I_3) < \ldots < m - S(I_m)$ and SR would allocate the agent $I_1$. If the agent’s top-ranked item is not the minimum score item, then the inequalities would still follow.
Figure 8.3: Rank-efficiency of SR as the fraction of strategic agents varies, compared to SR with entirely truthful agents and to Greedy. 10 agents, 10 items, similarity=0.3.

The sufficient condition on adjacent scores can be interpreted in the context of the weighted-popularity model. Sort the items, so that \( I_1 \) is the highest popularity, \( I_2 \) the second highest and so on. The condition on the gap between scores induces a simple requirement on popularity \( \{p_1, p_2, \ldots, p_m\} \):

**Proposition 8.1.** The scoring-gap condition is satisfied, and SR reduces to APSD under strategic behavior, in the weighted-popularity preference model if, for \( j = 1, 2, \ldots, m-1 \), we have

\[
\frac{p_j}{\sum_{i=j}^m p_i} < \frac{p_{j+1}}{\sum_{i=j+1}^m p_i}.
\]

The proof follows from series expansions.

### 8.5.1 Experimental Results

Having established the first criterion that we introduce for tolerable manipulability (that the mechanism reduces to APSD when every agent is strategic) we now establish the second criterion: that the *ex ante* efficiency is better under SR than under ASDP when a sufficient fraction of agents fail to be rational and are instead truthful. In fact, the results show that this holds for any fraction of strategic agents. We vary the fraction \( q \in [0, 1] \) of agents that are strategic.

Figure 8.3 gives results for a simulation of 10 agents with 10 items and similarity of 0.3. (averaged over 5000 independent runs). The average rank-efficiency of SR decreases as the
fraction of strategic agents increases, but is always better than APSD (i.e., Greedy) for any fraction $q < 1$ of strategic agents and equal to APSD when the fraction is one. We break down the performance of SR in terms of the average rank to strategic agents and to truthful agents. Strategic agents perform better than truthful agents and the gap is significant. However, when the fraction of strategic agents is less than a threshold (in this example 0.3), the truthful agents still do better than they would under APSD. Strategic agents receive better objects than they would receive in offline optimal.

Though the simulation results are for 10 objects, for intermediate values of $m$, SR continues to outperform. (e.g., by 8.3% rank-efficiency for 5 items and 10 agents, on agents receiving an item.) Also, SR outperforms APSD at least by 5% for similarity between $[0.1, 0.5]$.

We also consider the sensitivity of SR to strategic agents with imperfect information $\{\hat{p}_j\}$ about the relative popularity, and thus score, of items. Given a true profile of item popularities $\{p_1, \ldots, p_m\}$ we consider two types of perturbations: First experiment. For each manipulating agent, for each item $I_j$, perturb the popularity so that $\hat{p}_j/p_j = \Delta_j \sim \mathcal{N}(1, z')$, where the variance, $z' \in \{0.1, 0.3\}$. This models “local” errors, in which the estimate for the popularity of each item is off by some random, relative amount. Second experiment. The true popularity profile, $\{p_j\}$, is constructed by evaluating the Normal pdf $\Psi_z(\frac{2j}{m})$ for each of $j \in \{1, \ldots, m\}$, where $\Psi_z$ is the density function for Normal distribution $\mathcal{N}(1, z)$. Here, we construct $\hat{p}_j = \Psi'_z(\frac{2j}{m})$, where $\Psi'_z$ is the density function for Normal $\mathcal{N}(1 + \Delta, z)$ where $\Delta \sim U[-z''/2, z''/2]$. We consider $z'' \in \{0.1, 0.3\}$. This models “large-scale” errors, in which an agent has a systematic bias in popularity across all items.

Figure 8.4 presents the average rank-efficiency in the two experiments, for 10 agents, 10 items and similarity 0.3. We focus on $z' = 0.3$ and $z'' = 0.3$. The results are very similar for $z'$ and $z''$ set to 0.1. The local perturbation turns out to not affect the performance much since $\{\hat{p}_i\}$ introduces a multiplicative error with zero expected bias. On the other hand, the large-scale perturbation can significantly reduce the rank-efficiency of SR under manipulative agents, introducing an additive error with bias; even here, SR still outperforms Greedy with less than 40% strategic agents.

8.6 Conclusions

We have considered a dynamic mechanism design problem without money from the perspective of tolerable manipulability. This is an interesting domain because the unique strategyproof mechanism is easy to identify but has poor ex ante efficiency. We can also consider a domain with dynamic items and dynamic agents, where even APSD is no longer strategyproof and there can in fact be no reasonable, strategyproof mechanism. Tolerable manipulability is an interesting
Figure 8.4: Rank-efficiency of SR as the fraction of strategic agents varies and with errors in agent beliefs. 10 agents, 10 items, similarity =0.3.

direction for practical, computational mechanism design, relaxing from the worst-case requirements of strategyproofness and providing better performance when only a fraction of agents are strategic and the rest are truthful. An appealing direction is to achieve stronger guarantees, finding a compromise between the framework we adopt and Othman and Sandholm’s [86], which seems too exacting to achieve.
Chapter 9

Summary and Future Work

We have seen that there is a plethora of problems dealing with allocation of heterogeneous objects in the presence of strategic agents. The participating agents have preferences over these allocations. In the heterogeneous settings, the preference structure of the agents is multi-dimensional. It is a challenging task to choose a system wide good allocation, taking into account the preferences of the agents. The agents are strategic and may lie about their preferences. This makes the problems formidable. We addressed the allocation of heterogeneous objects in a mechanism design framework. In the real world, the agents may be dynamic. The situations may allow monetary transfers among the agents so as to achieve desirable game theoretic properties or it may be illegal/infeasible to use monetary transfers. Thus we can attribute two dimensions to these problems:

1. Possibility of monetary transfers among agents
2. Agents being static or dynamic.

![Mechanism design problem space classification](image_url)

Figure 9.1: Mechanism design problem space classification
As shown in Figure 9.1, according to these two attributes, the space of mechanism design problems is classified into four categories, namely, A, B, C, and D. Based on this classification, the thesis is divided into two parts. In Part-1 of the thesis, we addressed the problems of type A, that is with static agents and wherein monetary transfers are allowed. In Part-2, we studied the problems of type D, that is with dynamic agents and wherein monetary transfers among the agents are not possible or illegal. The current state of the art does not handle these problems. We proposed novel mechanisms for allocation of heterogeneous objects under various settings. We addressed the Problems 1–6 in Chapter 3–8 respectively. We concentrated on incentive compatible mechanisms. In most of the cases, we focused on dominant strategy incentive compatible mechanisms. Though in the optimal combinatorial auctions problem we started with Bayesian incentive compatible mechanisms, we showed that our mechanism is dominant strategy incentive compatible under fairly reasonable assumption, namely regularity assumption. In dynamic allocation mechanism, we proposed a mechanism that is not dominant strategy incentive compatible, but performs well.

There are many natural extensions to this thesis. The results presented in the thesis indicate that in many situations, the mechanism design problems need weaker solution concepts than dominant strategy incentive compatibility. In dynamic settings, it becomes difficult to achieve some of the game theoretic properties, like Pareto efficiency and stability which can be achieved in static settings. Thus there is a need to weaken these concepts when it comes to designing mechanisms in dynamic settings.

Below, we summarize our results chapter-wise and state immediate possible directions for future work for each problem.

9.1 PART 1: MECHANISMS WITH MONEY AND STATIC AGENTS

Chapter 3: Optimal Multi-Unit Combinatorial Auctions

In this chapter, we addressed the challenges involved in designing an optimal combinatorial procurement auction. We restricted our attention to the case of single minded capacitated bidders. We have characterized all incentive compatible and individually rational multi-unit multi-item auctions in the presence of single minded, capacitated buyers. With this characterization, we proposed an optimal auction, OCAS, for a buyer seeking to procure multiple units of multiple items in the presence of single minded and capacitated sellers. If virtual cost functions satisfy regularity conditions, we have designed an optimal auction that satisfies dominant strategy incentive compatibility and IIR. We developed an auction, VD-OCAS, for two item, multi-unit combinatorial auction when single minded bidders offer volume discounts. We conjecture it to
be an optimal auction. Under certain regularity, we have given a blueprint of an optimal auction when bidders are XOR minded.

The possible directions to extend our work is to address settings in which the sellers are willing to provide volume discounts in more than two items auctions. The other direction for future work is to design an optimal auction when bidders are willing to submit bids on multiple bundles.

Chapter 4: Truthful Multi-Armed Bandit Mechanisms for Multi-Slot Sponsored Search Auctions

In this chapter, we addressed the allocation of sponsored search slots to the advertisers who pay for each click received through such ads. The click probabilities play an important role in designing auctions for such allocation. Typically, these probabilities had to be learn over the repeated auctions. Thus there is need to combine techniques from mechanism design theory and machine learning theory. We have provided characterizations for truthful (dominant strategy incentive compatible) multi-armed bandit mechanisms for various settings in the context of multi-slot, pay-per-click auctions.

- The first result we proved is a negative result which states that under the setting of unrestricted CTRs, any strategyproof allocation rule is necessarily strongly pointwise monotone.

- We also showed that every strategyproof mechanism in unrestricted CTR setting exhibit a very high regret ($\Theta(T)$).

- By weakening the notion of unrestricted CTRs, we were able to derive a larger class of strategyproof allocation rules. Our results are summarized in Table 4.1.

In the auctions that we have considered, the auctioneer cannot vary the number of slots he wishes to display. One possible extension of this work could be that the auctioneer can dynamically decide the number of slots for advertisements. We assume that the bidders bid their maximum willingness to pay at the start of the first round and they would not change their bids till $T$ rounds. Another possible extension would be to allow the agents to bid before every round. One can also explore the cases where the bidders have budget constraints.

Typically in learning settings, the agents and the mechanism designer would have some prior beliefs regarding the CTRs as well as valuations of all the agents. Thus, rather than looking into dominant strategy incentive compatible mechanisms, one can look for Bayesian Incentive compatible mechanisms that learn CTRs optimally. Another direction is to look into randomized mechanisms.
Chapter 5: Redistribution Mechanisms for Assignment of Heterogeneous Objects

We addressed the problem of designing redistribution mechanism for the allocation of $p$ heterogeneous objects among $n$ competing agents. When the valuations of the agents are independent of each other and their valuations for each object are independent of valuations on the other objects, we proved the impossibility of the existence of a linear redistribution mechanism with non-zero redistribution index (Theorem 5.3). Then we explored two approaches to get around this impossibility.

- In the first approach, we showed that linear rebate functions with non-zero redistribution index are possible when the valuations for the objects have scaling based relationship ($n > (p + 1)$). For these settings, we proposed a linear redistribution mechanism that is optimal on worst case analysis, individually rational and feasible (Theorem 5.4).

- In the second approach, we relaxed the linearity requirement. We showed that non-linear rebate functions with non-zero redistribution index are possible by applying BAILEY-CAVALLO mechanism to the settings (Theorem 5.5).

- We proposed a mechanism, namely HETERO, for general settings when the objects are heterogeneous and private values of an agent for these objects are independent of each other. The mechanism is deterministic, anonymous, and dominant strategy incentive compatible. The HETERO mechanism extends the Moulin /WCO mechanism. Though we have not analytically proved feasibility and individual rationality, we have sufficient empirical evidence to conjecture that our mechanism is feasible and individually rational (Conjecture 5.1). We also conjecture HETERO to be worst case optimal having same redistribution index as the homogeneous settings.

It would be interesting to see if we can characterize the situations under which linear redistribution mechanisms with non-zero redistribution indices are possible for heterogeneous settings.

An interesting research direction is to investigate the individual rationality and feasibility for the proposed HETERO mechanism. Also, we strongly believe that the new mechanism is worst case optimal. An immediate future direction is to prove this fact or design a mechanism which is worst case optimal. Another interesting problem to explore is to characterize all redistribution mechanisms that are worst case optimal for heterogeneous settings.

We restricted our attention to deterministic mechanisms. It would be interesting to look for randomized mechanisms. Also, one can explore dominant strategy incentive compatible and budget balanced mechanisms by compromising on allocative efficiency.
Chapter 6: Dynamic Stable Matching in Two-Sided Markets

In this chapter, we initiated a study into dynamic matching problems in two-sided markets without money. One side of the market is static while the other side is dynamic, and we require truthfulness on the static side of the market. In general it is not possible to achieve stability in the dynamic settings. We proposed to allow agents to use option fall-back, that is some agents can use substitutes instead of their matches. Based on this substitute option, we proposed a dominant strategy incentive compatible, dynamic, and stable matching algorithm, namely, GSODAS. We showed that it achieves stability optimally. That is, on worst case analysis, no online matching algorithm can achieve stability with lesser substitutes than the GSODAS. It also performs better on rank-efficiency than simpler methods that do not use substitutes, although with less rank-efficiency achieved by a non-truthful stochastic optimization approach.

Though GSODAS is worst case optimal for number of substitutes required, the use of substitutes in GSODAS is quite high, on average 30% of the number of agents. As the number of time periods increases (for uniform preferences), as many as 55% of the number agents are required in the worst-case experimental instances. This is likely unacceptable in many practical domains, yet we prove that better worst-case properties are unavailable if full stability is required. The most interesting future direction, then, will be to relax the requirement of offline stability. This precludes blocking pairs, irrespective of the timing of the agents that comprise a blocking pair in system and the information available at the time of a match. Perhaps by relaxing this requirement, mechanisms with good rank-efficiency, acceptable stability, but less need for exercising the fall-back option can be developed.

Chapter 7: Dynamic House Allocation

In this chapter we considered a dynamic version of the house allocation problem. We showed that no individually rational dynamic house allocation mechanism can always achieve efficiency which is achieved by TTCA in static settings. We proved that, in strategyproof simple mechanisms the agents cannot trade with different subsets of agents by changing their type report. This motivated us to propose a family of strategyproof mechanisms called as partition mechanisms, in which the agents are partitioned into disjoint trading groups and this partitioning is determined without regard to their types. We proposed three partition mechanism, DO-TTCA, T-TTCA, and SO-TTCA. A clear benefit is established for T-TTCA and SO-TTCA over a method that performs trading upon the departure of one or more agents (DO-TTCA). In the experiments we performed, the stochastic optimization approach (SO-TTCA) outperformed the threshold approach (T-TTCA) for Poisson arrival processes.
An interesting aspect of the use of stochastic optimization is that it determines which subset of agents should trade but not what trade should occur, and without consideration of specific agent types. This preserves strategyproofness. The most interesting immediate next steps would be to consider a generalization in which agents can misreport arrival and departure, while continuing to explore the role for stochastic optimization in identifying useful trading groups.

Chapter 8: Dynamic Allocation Mechanisms for Assignment of Heterogeneous Objects

In this chapter we addressed the allocation of heterogeneous objects to dynamically arriving agents. We identified arrival priority serial dictatorship as the unique dominant strategy incentive compatible mechanism for the the problem. It has poor poor $ex$ $ante$ efficiency. So we considered a dynamic mechanism design problem without money from the perspective of tolerable manipulability. We proposed a heuristic based mechanism which we call as SR (Scoring Rule) which has better rank-efficiency than APSD if agents are truthful. It reduces to APSD when all the agents manipulate it optimally.

We can also consider a domain with dynamic items and dynamic agents, where even APSD is no longer strategyproof and there can in fact be no reasonable, strategyproof mechanism. Tolerable manipulability is an interesting direction for practical, computational mechanism design, relaxing from the worst-case requirements of strategyproofness and providing better performance when only a fraction of agents are strategic and the rest are truthful. An appealing direction is to achieve stronger guarantees, finding a compromise between the framework we adopt and Othman and Sandholm’s [86], which seems too exacting to achieve.
Bibliography


