Stochastic Mechanisms for Truthfulness and Budget Balance in Computational Social Choice

by

Lachlan Thomas Dufton

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis, we examine stochastic techniques for overcoming game theoretic and computational issues in the collective decision making process of self-interested individuals. In particular, we examine truthful, stochastic mechanisms, for settings with a strong budget balance constraint (i.e. there is no net flow of money into or away from the agents). Building on past results in AI and computational social choice, we characterise affine-maximising social choice functions that are implementable in truthful mechanisms for the setting of heterogeneous item allocation with unit demand agents. We further provide a characterisation of affine maximisers with the strong budget balance constraint. These mechanisms reveal impossibility results and poor worst-case performance that motivates us to examine stochastic solutions.

To adequately compare stochastic mechanisms, we introduce and discuss measures that capture the behaviour of stochastic mechanisms, based on techniques used in stochastic algorithm design. When applied to deterministic mechanisms, these measures correspond directly to existing deterministic measures. While these approaches have more general applicability, in this work we assess mechanisms based on overall agent utility (efficiency and social surplus ratio) as well as fairness (envy and envy-freeness).

We observe that mechanisms can (and typically must) achieve truthfulness and strong budget balance using one of two techniques: labelling a subset of agents as "auctioneers" who cannot affect the outcome, but collect any surplus; and partitioning agents into disjoint groups, such that each partition solves a subproblem of the overall decision making process. Worst-case analysis of random-auctioneer and random-partition stochastic mechanisms show large improvements over deterministic mechanisms for heterogeneous item allocation. In addition to this allocation problem, we apply our techniques to envy-freeness in the

room assignment-rent division problem, for which no truthful deterministic mechanism is possible. We show how stochastic mechanisms give an improved probability of envy-freeness and low expected level of envy for a truthful mechanism. The random-auctioneer technique also improves the worst-case performance of the public good (or public project) problem.

Communication and computational complexity are two other important concerns of computational social choice. Both the random-auctioneer and random-partition approaches offer a flexible trade-off between low complexity of the mechanism, and high overall outcome quality measured, for example, by total agent utility. They enable truthful and feasible solutions to be incrementally improved on as the mechanism receives more information and is allowed more processing time.

The majority of our results are based on optimising worst-case performance, since this provides guarantees on how a mechanism will perform, regardless of the agents that use it. To complement these results, we perform empirical, average-case analyses on our mechanisms. Finally, while strong budget balance is a fixed constraint in our particular social choice problems, we show empirically that this can improve the overall utility of agents compared to a utility-maximising assignment that requires a budget imbalanced mechanism.

Acknowledgements

I would like to thank my supervisor, Kate Larson, for her support and guidance throughout my PhD studies. Kate always gave me great advice and suggestions when I needed it, for my research and other aspects of my graduate studies. I am grateful to my departmental committee members, Robin Cohen and Peter van Beek, who gave valuable feedback on the direction of my research topic from the early stages of this work. I'd like to thank my internal-external committee member, Chaitanya Swamy, for useful technical feedback on some results in this work. I am also thankful to my external committee member, Ulle Endriss, not only for feedback on this work, but whose previous work on computational social choice got me interested in further studying the area.

I'd also like to thank Zinovi Rabinovich, Victor Naroditskiy, Maria Polukarov and Mingyu Guo, for very interesting research collaborations during my graduate studies.

I must also acknowledge my officemates, Ahmed Ataullah, John Champaign, John Doucette and Simina Brânzei, for great company in the office, and on countless tea and coffee breaks. The many truly awesome friendships I formed at the University of Waterloo made the last few years an amazing period in my life, that I will fondly look back on. A very special thanks to Sarah Lim, for her love, support and patience, even when my days and nights were full of thesis editing and defence preparation.

Finally, a huge thank you to all my family, especially my parents, Leonie and Michael, and brother, Cameron. Their constant love and support have allowed me to pursue so many exciting opportunities in my life.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}$	Set of all real numbers
$\mathbf{N}$	Set of agents
n	Number of agents $n = \|\mathbf{N}\|$
$\mathbf{N}_X$	Set of ignored agents
$\mathbf{M}$	Set of items / resources
m	Number of items / resources $m = \|\mathbf{M}\|$
$\mathbb{X}$	Set of outcomes
$a \in \mathbb{A}$	Allocation (and set of allocations)
$v \in V$	Value profile of all agents
$v_i \in V_i$	Value profile of agent $i \in \mathbf{N}$
$v_{-i} \in V_{-i}$	Value profile of all agents except $i \in \mathbf{N}$
	$v = (v_1, \ldots, v_n) = (v_i, v_{-i})$
$\hat{v}, \hat{v}_i, \hat{v}_{-i}$	Reported values
$\theta \in \Theta$	Type profile
$\theta_i \in \Theta_i$	Type (and type space) of agent i
$\theta_{-i} \in \Theta_{-i}$	Type of all agents except i
$\hat{\theta}, \hat{\theta}_i, \hat{\theta}_{-i}$	Reported types

u, u_i	Utility function (all agents, single agent i)
f	Social choice function
t, t_i	Payment function (all agents, single agent i)
T^{VCG}	Total VCG payment/surplus
$h_i(v_{-i})$	Additional payment/rebate for agent i
T	Total payment/rent
$\mathcal{M}$	Deterministic mechanism
$\hat{\mathcal{M}}$	Set of deterministic mechanisms
$\Delta\hat{\mathcal{M}}$	Stochastic/randomised mechanism
p_k	Probability of selecting mechanism/outcome k
r	Social surplus ratio
r_{WCE}	Worst-case expected social surplus ratio
p_{EF}	Probability of efficiency
p_{NF}	Probability of envy-freeness
E_{NF}	Expected envy-freeness

Chapter 1

Introduction

In this work, we examine stochastic techniques for overcoming game theoretic and computational issues in the collective decision making process of self-interested individuals. We assess the efficacy of these techniques, with a focus on strongly budget balanced, truthful mechanisms.

Problems where multiple individuals with their own preferences must come to some collective decision are ubiquitous. A website displaying advertisements must choose from different advertisers, each with different values for the location and pages on which those ads appear. A government licensing bands of the electromagnetic spectrum seeks to provide licences to bands and locations to companies that value them the most. In political elections, each citizen has some ordering over each candidate and the electoral system aggregates these preferences to determine who is elected. Each of these problems requires the elicitation of privately known preferences by some decision making process where individuals may misreport their preferences if this can improve their outcome. A bidder in an auction, for example, may be able to pay less by reporting a lower value for the resource being auctioned. Such problems are examined in social choice theory and mechanism design.

Computational social choice is a field of study combining artificial intelligence and computer science with social choice theory [Chevaleyre *et al.*, 2007; Brandt *et al.*, 2012]. Social choice theory aggregates a set of individual preferences or utilities in order to make some sort of collective decision. The goal of the decision making process is, typically, to maximise some collective utility of the group of individuals. For example, a utilitarian decision maker would measure and maximise collective utility as the sum of individual utility, while an egalitarian decision maker would attempt to maximise the utility of the worst-off agent in the group. This decision making process may also have other requirements such as fairness, that may ensure agents have roughly equal utility, or individual rationality, which makes sure no agent has a lower utility after participating in a decision making process. Since each individual has private preferences, a common requirement is truthfulness, or incentive compatibility, whereby an agent is guaranteed to maximise its own utility when reporting its private preferences truthfully. Social choice also draws from other fields such as game theory, decision theory and welfare economics.

While social choice theory ignores computational issues, computational social choice looks at issues such as tractability and complexity to perform the decision making process. Such computational costs may be used to provide additional guarantees on a decision process, which may be theoretically manipulable, but only at some intractably high computational cost. Computational social choice also applies algorithm design and verification techniques from computer science to the development of social choice systems. Computational social choice also covers the application of social choice techniques to the interactions and negotiations between automated agents in artificial intelligence, and in particular multiagent systems.

Randomisation and approximation are two tools frequently used in algorithm design. A randomised (or probabilistic) algorithm makes use of random numbers or bits during computation. For a single input, the performance and/or output of a random algorithm can be modelled by a random variable dependent on these random bits, rather than a single, deterministic value. Randomisation can be used to obtain a higher *expected* performance compared to deterministic algorithms, or to provide an ideal solution with some probability, where a deterministic algorithm is unable to do so for all inputs.

One example of using randomisation and approximation in an algorithm to overcome impossibility results is in the consensus problem [Pease *et al.*, 1980]. In this problem, a group of isolated processors each have some private information, but an unknown subset are faulty. The non-faulty processors must come to some group agreement in the presence of these faulty processors. While this problem is impossible to solve for several processor models, randomisation has been used to circumvent these impossibility results [Aspnes, 2003]. In other problems, a correct result may be unachievable deterministically, for all inputs, given computational or other restrictions. However, randomised algorithms exist that return a correct answer with some probability, p, or are known to produce an answer within some approximation factor, α, of the ideal solution. A classic example of this is the Miller-Rabin primality test [Miller, 1976; Rabin, 1980], to determine whether a given input is prime or composite. The algorithm's runtime increases linearly with the choice of parameter k, while the probability of returning an incorrect input is bounded to be at most 4^{-k}. These tools are also used to overcome computational complexity issues when a correct answer is always returned. For example, consider the convex hull problem, which, given a set of points P, finds the smallest convex set containing P [Chan, 1996]. Based on an adversarial argument, a deterministic algorithm finding the convex hull of a set of n points, processed serially, requires $\Omega(n^2)$ steps in the worst case. Randomisation is able to overcome this adversary argument and allows an expected running time of $O(n \log n)$ for this problem [Motwani and Raghavan, 1996].

Randomisation has been used in voting theory to overcome impossibility results regarding manipulation by voters. The Gibbard-Satterthwaite theorem shows that if a voting method is free from tactical or strategic voting then it must be either dictatorial, have some candidate that can never win, or be non-deterministic [Gibbard, 1977]. If a voting method allows tactical voting, then a voter may achieve a more desirable outcome by stating preferences other than his or her true preferences. A dictatorial voting system contains a single voter (the dictator) who can choose the winner. A simple example of a voting method that uses randomisation to be simultaneously free from strategic voting and non-dictatorial while not preventing any candidate from winning, is the random ballot. In a random ballot election, one ballot is randomly selected and the candidate ranked first on that ballot is the winner. While this voting system becomes dictatorial after the random choice is made, it does not need to define a set dictator in advance. It also means that the probability of a candidate winning is directly proportional to the number of voters who most support that candidate, which is untrue of dictatorial systems. There has been some recent work on randomisation in computational social choice [Alon *et al.*, 2011; Procaccia, 2010], and it is an area with many interesting open problems.

In this work we examine different techniques using randomisation and approximation in computational social choice problems. There are several classes of problems in computational social choice, and the primary focus of this study is on problems from multiagent resource allocation and fair division. These problems deal with the assignment of finite, indivisible goods to a group of agents, where each agent has his or her own preferences over bundles of goods. Solutions are evaluated on notions of collective utility, efficiency (Pareto optimality), and fairness [Chevaleyre *et al.*, 2006]. We focus on mechanisms that are truthful, so agents have no incentive to misreport their true preferences. The problems we consider have a strong budget balance constraint, where any payments made by agents, due to rules of the mechanism, must sum to zero. This means that we allow payments between agents, but there is no external party (residual claimant) to fund or collect surplus payments from the group of agents participating in the social choice problem. If there is no party to collect any surplus revenue, this surplus must somehow be discarded. While a payment surplus has been dealt with in previous work by "money burning" [Hartline and Roughgarden, 2008] for example, this may not always be possible. Laws or rules on the unit of currency may prohibit its destruction, or if the units are resources such as gold or time, the destruction may be physically impossible. Also, for currencies with small money supply, such as shares in an asset, money burning can affect a currency's value and have a non-negligible effect on agents' utilities, harming truthfulness. In settings where a mechanism is performed iteratively on a set of agents, even a small budget imbalance could result in completely draining the agents of their money supply, given enough iterations. The combination of truthfulness and strong budget balance can be quite restricting, however, as shown by the Green-Laffont impossibility theorem [Green and Laffont, 1979] (see Subsection 3.2.2).

The first problem we consider is the allocation of heterogeneous items to agents with unit demand (see Section 2.1). In particular, performing such allocations under a strong budget balance constraint. Consider the following two scenarios. First, a group of friends has won tickets to various events at an international sporting event. While each person would like to see any one event, they each have different preferences over the events and the seat locations. The group decides that the tickets should be given so that, if possible, each person gets to attend their most preferred event and that no one should get more than one ticket. The group decides to run an auction for the tickets amongst themselves, but since there is no outside auctioneer, all payments must be made between themselves. In the second scenario, a department has recently been allocated new office space and the employees have to decide amongst themselves how the offices will be allocated. They decide to run an auction, with each employee "paying" by offering to personally deliver a cup of gourmet coffee for a certain number of mornings. Those stuck with their lower preferences for offices will be compensated by having hot beverages hand delivered by those in the more desirable offices.

In both of these scenarios, we have a problem of allocating a set of indivisible goods, resources or tasks to a group of agents, such that each individual receives at most one item. Every agent may not get an item (e.g. there are not enough tickets) or there may be goods left unassigned (e.g. there are extra offices). This unit demand setting makes sense, for example, when individual agents have a need for only one item, such as tickets to an event or time slots in a schedule. It also serves as a simplification when agents have near-zero marginal utility for a bundle of items over the highest-valued item in that bundle. For example, an advertiser may have a higher utility for an advertisement placed on the top two positions of a website compared to just the top position, but the difference in utility is likely to be a small fraction of the agent's overall utility. In this way, an agent's preferences can be expressed simply as its value for each individual item, rather than values for all combinations of items, which greatly simplifies the amount of information needed to capture preferences. The two scenarios above require strong budget balance. Since there is no auctioneer for the tickets, any payments made must be distributed back to the participants. In the office setting, for every cup of coffee that one person must deliver, there must be someone set to receive a coffee (the receiver delivers a negative number of cups).

Related to heterogeneous item allocation is the room assignment–rent division problem

[Su, 1999] (see Section 2.2), which is a classic problem in multiagent resource allocation and fair division. Consider a group of friends who will rent a house together. They must decide both who gets which room, and what share of the rent each person will pay. Each friend will want to be allocated just one room and there should be no surplus or deficit when meeting the total rent. Each individual has his or her own preferences on which room is best, such as preferring the largest room, or the room with the best view. The goal is to assign individuals to rooms and divide the rent in such a way that no one prefers another room. More generally, this is a problem of allocating a set of indivisible, heterogeneous items (i.e. the rooms) along with a share of a divisible resource (i.e. the rent), such that all items are allocated and each agent gets exactly one indivisible item. The resources can provide positive or negative utility to the agents. In these settings, we are interested in more than just the total utility of agents, but also some notion of fairness. In this work we focus on *envy* and *envy-freeness* as measures of fairness. An agent is envious if it views another agent's bundle item and payment as strictly better than its own and an envy-free mechanism provides an allocation where no agent is envious. As well as fairness, envy-freeness gives a form of stability to an allocation. Under an envy-free allocation, no agent will want to force another agent to swap bundles.

Finally, we examine the public good (or public project) problem (see Section 2.3). A common example for this problem is a community deciding on whether or not to collectively purchase a resource that can be used by all, such as a new bridge. Based on how much they anticipate using this bridge, each community member has some private value for how much it is worth. The bridge should only be built if the total benefit it provides outweighs the total cost, which the community must pay. Furthermore, if the bridge is built, we should not expect any individuals to have to pay more than their value for the bridge. Agents in this setting are single-parameter, in that their preferences are captured by a single value:

the value for the public good.

In all these problems, the agents' preferences are private information, which the decision maker cannot assume will be reported honestly. Since agents can misreport their true preferences, or may have limited information about other agents' preferences, we focus our attention on truthful mechanisms, where an agent maximises its utility by reporting truthfully, regardless of the behaviour of other agents. This prevents individual agents from misreporting their preferences in order to increase their utility, at the expense of other agents. It also means agents also do not need to invest effort in determining an optimal strategy or predicting other agents' types, since the optimal strategy is always truthful reporting.

We examine existing solutions to these three problems, using deterministic and stochastic mechanisms. Due to impossibility results, these mechanisms must sacrifice some desirable property such as truthfulness or budget balance. Our work continues on from previous work (e.g. Roberts [1979] and Bikhchandani *et al.* [2006]) by further examining the space of mechanisms for these problems.

The focus of this work is on truthful mechanisms that have no surplus or deficit in total payments. We restrict our attention to affine-maximising mechanisms, but one of our contributions is a characterisation that shows truthful mechanisms for the heterogeneous item allocation problem, with an independence constraint must be affine-maximising. This is initially just for deterministic mechanisms, but we extend this to cover stochastic mechanisms. These new results are an important contribution to the multiagent systems literature, since they assist future work on similar problems to determine the space of possible mechanisms, and determine optimality within this space. We use these characterisations as part of a foundation for our investigation of truthful mechanisms for the room assignment–rent division problem.

As a result of these characterisations, we observe that truthfulness and strong budget balance for affine-maximising mechanisms can (and typically must) be achieved using techniques of either ignoring some agents, or partitioning agents into groups that each solve a sub-problem of the overall decision making process. We then apply these techniques to develop mechanisms, deterministic and randomised, for our three social choice problems.

For problems where randomisation has been seldom or never used, a comparison based on existing performance measures and desiderata is not always suitable, and this work demonstrates the limitations of these measures. A contribution of this work is the definition and discussion of appropriate comparison measures that appropriately capture the behaviour of randomised mechanisms. Using our characterisations, we provide upper bounds on what is achievable for truthful mechanisms given our constraints. We also present mechanisms that have worst-case bounds matching these upper bounds and are thus optimal according to these measures. The improved performance we demonstrate in randomised mechanisms encourages the use of these techniques in future work in multiagent systems research.

Communication and computational complexity are two other important concerns of computational social choice. While the techniques of ignoring agents and partitioning were motivated by the goal of improving social welfare measures, we examine the efficacy of these techniques on improving complexity issues. In particular, we show that both techniques offer a flexible trade-off between low complexity of the mechanism, and high overall utility of the agents.

The majority of our results are based on optimising worst-case performance, since this provides guarantees on how a mechanism will perform, regardless of the agents that use it. To complement these results, we perform empirical, average-case analyses on our mechanisms. These empirical results provide insight into the rarity of value profiles that lead to worst-case results from a mechanism. Worst-case bounds on their own make no distinction between a mechanism where all value profiles lead to the same level of performance, and a mechanism with only a single poor-performing value profile.

Finally, while strong budget balance is a constraint in our social choice problems, we show that this can improve the overall utility of agents. We empirically compare mechanisms that enforce a utility-maximising assignment with those that enforce strong budget balance. These results show that a budget imbalance can lower the utility of the group of agents by more than the loss of utility due to non-utility-maximising assignment.

This thesis is structured as follows. In Chapter 2 we introduce general notation and concepts required for our investigation of computational social choice problems. We then provide a description of our three social choice settings in Sections 2.1 to 2.3.

Next, in Chapter 3 we cover both existing and new deterministic solutions to our social choice problems, as well as related problems. In Section 3.1 we cover measures used to assess the quality of deterministic solutions followed by relevant characterisations in Section 3.2. This includes our novel characterisation of truthful and strongly budget balanced heterogeneous item allocation mechanisms, under an independence constraint, in Subsections 3.2.4 and 3.2.5. We then present existing and new deterministic mechanisms in Sections 3.3 to 3.5. Along with mechanisms, we provide upper bounds on the performance of deterministic mechanisms that motivate the examination of stochastic mechanisms.

In Chapter 4 we examine stochastic mechanisms for our three social choice settings. We introduce and discuss our measures for examining such mechanisms in Section 4.1, and Section 4.2 covers related work on the use of randomisation in other social choice problems. Sections 4.3 to 4.5 examine our randomisation techniques applied to our different settings, examining the worst-case performance of these mechanisms.

We look at average case results and complexity of the mechanisms in Chapter 5. In Section 5.1 we show the results of empirical tests on the average case performance of our stochastic mechanisms. Section 5.2 shows the trade off between enforcing efficient outcomes and enforcing strong budget balance. Next, in Sections 5.3 and 5.4, we show the flexible trade off between high solution quality and low communication or computational complexity of our stochastic mechanisms. Finally, in Chapter 6 we present our conclusions and directions for future work.

Chapter 2

Problems in Computational Social Choice

The field of computational social choice covers a wide range of problems from social choice theory [Chevaleyre *et al.*, 2007; Brandt *et al.*, 2012]. Social choice problems involve multiple agents in a collective decision making process, the outcome of which affects all agents. In this chapter, we formally define each of the problems that we examine in this study, along with required definitions. We focus on a selection of problems from multiagent resource allocation [Chevaleyre *et al.*, 2006]. These problems deal with multiple autonomous entities called **agents**. These agents can represent, for example, people, autonomous software programs, computer systems, companies, robots, network links or a mix of these. In each of the problems we examine, we have a set of n agents, $\mathbf{N}$, where each agent $i \in \mathbf{N}$ has a **type** $\theta_i \in \Theta_i$ that captures all the relevant information about the agent. The **type space** of an agent, Θ_i, determines all the possible preferences that agent can hold, while $\Theta = \Theta_1 \times \cdots \times \Theta_n$ denotes all possible sets of agents. We assume that an agent's type is private information, known only to that agent. However, as we explain in the rest of this section, we can design mechanisms where a rational, self-interested agent will truthfully reveal its private information.

Each agent $i \in \mathbf{N}$ has a **utility function** $u_i : \Theta_i \times \mathbb{X} \to \mathbb{R}$ that captures an agent's preferences over the set of all possible **outcomes** $\mathbb{X}$, given its type. For any two outcomes $X_1, X_2 \in \mathbb{X}$, if $u_i(\theta_i, X_1) > u_i(\theta_i X_2)$ then agent i of type θ_i prefers outcome X_1 to X_2 (denoted $X_1 \succ_{\theta_i} X_2$), while if $u_i(\theta_i, X_1) = u_i(\theta_i, X_2)$ then the agent is indifferent between the two (denoted $X_1 \sim_{\theta_i} X_2$). Note that while agent' types are private, the utility functions are known for all agents. Since we consider settings where agents may make or receive payments, we model agents with **quasi-linear utilities**. That is, an agent's utility is linear in the payment it receives. For simplicity, the unit of the payment commodity forms the numeraire for all utility functions.

We assume agents act **rationally**, in the way described by Von Neumann and Morgenstern [1953]. Thus, where an agent i's actions can influence the outcome, it is assumed to act such as to maximise the (expected) value of its utility u_i. In stochastic settings, where agents use an *expected utility*, we assume **risk neutrality**. Suppose an agent is given the choice between a guaranteed payment of \$10, or a lottery giving \$20 with a 50% chance and \$0 otherwise. Both give an expected payment of \$10, so a risk-neutral agent would be indifferent between the two choices. Alternatively, a risk-averse agent would prefer the guaranteed \$10 while a risk-seeking agent would prefer the lottery.

Several of these agents as we described are making their collective decision according to some previously agreed upon rules. We capture this decision making process as a **mechanism**, which determines an outcome $X \in \mathbb{X}$, from some space of possible outcomes, $\mathbb{X}$, given a set of agents Θ. In this work, we focus our attention on *direct-revelation mechanisms* [Myerson, 1981]. Unless otherwise specified, we use the term *mechanism* to refer to a (deterministic) direct-revelation mechanism.

Definition 2.1 (Direct-revelation mechanism)**.** *A **direct-revelation mechanism** $\mathcal{M} = (f, t)$, is a decision making process that restricts the actions of the agents to each make a single (confidential) report of their private type. The **social choice function** (SCF)*[1] *$f : \Theta \to \mathbb{X}$ then determines an outcome $X \in \mathbb{X}$, based on the (reported) types of agents, while the **payment function** $t : \Theta \to \mathbb{R}^n$ determines the payments* received[2] by *each agent.*

The payment function can be equivalently expressed as n payment functions, denoted $t = (t_1, \ldots, t_n)$, where $t_i : \Theta \to \mathbb{R}$ is the payment made to agent i.

In these mechanisms, agents report their private types θ to the mechanism. While mechanisms may request alternative information from the agents, the **Revelation Principle** [Gibbard, 1973; Green and Laffont, 1977; Myerson, 1979; Myerson, 1981] states that if these mechanisms implement a SCF in a (Bayesian) Nash equilibrium, there is a direct-revelation mechanism that implements the same SCF in a (Bayesian) Nash equilibrium. Thus, it is sufficient to assume the mechanism requests an agent reveal its type rather than some other information. A **Nash equilibrium** is a solution concept that proposes an equilibrium set of strategies for a set of rational agents. In our settings, an agent's strategy is simply the type it chooses to report. If all agents are acting according to a Nash equilibrium, then no single agent can increase its utility by changing its strategy (i.e. misreporting). A mechanism implementing an SCF in a Nash equilibrium means that

[1] Also called a **decision function**

[2] In this work, all payment functions define the amount the mechanism pays to each agent, while in related work, it is also common for the payment function to represent the amount paid by each agent to the mechanism.

all agents reporting truthfully is a Nash equilibrium, where all agents know each other's reported types.

In this work, we examine both deterministic and stochastic mechanisms.

Definition 2.2 (Stochastic mechanism)**.** *We represent a* ***stochastic mechanism*** *(or* ***randomised mechanism****)* $\Delta\hat{\mathcal{M}}$ *as a random distribution over a set of deterministic mechanisms (or sets of social choice functions and payment functions)* $\hat{\mathcal{M}}$*. Let* p^k *denote the probability of selecting deterministic mechanism* $\mathcal{M}^k = (f^k, t^k) \in \hat{\mathcal{M}}$*. We also use* $k \in \hat{\mathcal{M}}$ *to denote a set of indices of these deterministic mechanisms. A valid mechanism must have* $\sum_{k\in\hat{\mathcal{M}}} p_k = 1$*, and* $p_k \in [0, 1]$, $\forall k \in \hat{\mathcal{M}}$*.*

A random process first selects the mechanism k *to use, then the deterministic social choice and payment functions,* (f^k, t^k) *are applied to the agents' reported types. Since the random process is performed independently of agent types, each of the deterministic mechanisms that the process can select from (i.e. the* ***support*** *of the stochastic mechanism) must have the same type space* Θ *and outcome space* $\mathbb{X}$*.*

The agents' types (which capture their values for the outcomes) are private information, which the mechanism cannot assume will be reported honestly. In fact, we assume an agent *will* misreport if it is in its own best interest. Since agents can misreport their true preferences, or may have limited information about other agents' preferences, we focus our attention on mechanisms that are dominant strategy incentive compatible. A mechanism that is **dominant strategy incentive compatible** (DSIC)[3] guarantees that any agent will maximise its own utility when reporting its type truthfully, regardless of the behaviour of all other agents. That is, for all agents $i \in N$, and for all other agents'

[3] Also called **strategy-proof**

strategies $\theta_{-i} \in \Theta_{-i}$, the agent's true type θ_i satisfies:

$$\theta_i \in \{\hat{\theta}_i \in \Theta_i : u_i(\theta_i, f(\hat{\theta}_i, \theta_{-i})) + t(\hat{\theta}_i, \theta_{-i}) \geq u_i(\theta_i, f(\hat{\theta}_i{}', \theta_{-i})) + t(\hat{\theta}_i{}', \theta_{-i})), \forall \hat{\theta}_i{}' \in \Theta_i, \} \tag{2.1}$$

Under a DSIC mechanism, all agents acting truthfully is a Nash equilibrium, but is a stronger solution concept since it does not require any agent to know the other agents' actions.

A stochastic mechanism is DSIC (in the universal sense) if and only if it is a randomisation over deterministic DSIC mechanisms. Thus, even if an agent is aware of the random choice, truthful reporting remains as its dominant strategy. This is in contrast to the weaker requirement of **truthfulness in expectation** (TIE), where an agent's *expected* utility is maximised when reporting truthfully. All DSIC mechanisms are TIE, but a TIE mechanism may not be DSIC. If the random choice is known in advance to one or more of the agents, then a TIE mechanism will not necessarily have truth-telling as its dominant strategy. If, for some reason, an auction had to be re-run (e.g. an agent noticed a mistake in its submitted bid), a DSIC mechanism remains truthful if the same random choice is kept, but a TIE mechanism must make a new random choice to remain truthful. Since a DSIC mechanism is truthful in dominant strategies, information acquired before the auction is re-run will not cause individual agents to misreport (unless there is collusion). Firstly, truthfulness is desirable since this prevents individual agents from misreporting their preferences in order to increase their utility, at the expense of other agents. Truthfulness also means that agents do not need to invest effort in determining an optimal strategy or predicting other agents' types, since the optimal strategy is always truthful reporting.

Individual rationality (also known as **voluntary participation**) is a common constraint in auctions, resource allocation and other mechanism design problems. It ensures that a rational agent would always choose to participate in the mechanism, regardless of other agents participating.

Definition 2.3 (Individual Rationality (IR))**.** *A mechanism is* ***individually rational*** *if no agent is worse off after participating in the mechanism. That is, all agents are guaranteed to have non-negative utility. So, for any* $\theta \in \Theta$*:*

$$u_i(\theta_i, f(\theta)) + t_i(\theta) \geq 0 \ , \ \forall i \in \mathbf{N} \tag{2.2}$$

A weaker notion is IR in expectation, where an agent's expected utility is guaranteed to be non-negative.

In the rest of this chapter each of the following sections describes one of our social choice settings in detail.

2.1 Unit Demand Heterogeneous Item Allocation

Heterogeneous item allocation is a well studied problem in multiagent resource allocation [Ausubel, 2006; Bansal and Garg, 2000; Bhattacharya *et al.*, 2010; Cavallo, 2006; Garg and Mishra, 2002; Gershkov and Moldovanu, 2009; Gujar and Yadati, 2011; Guo, 2012; Jain and Varaiya, 2005; Kojima, 2009]. Heterogeneous item allocation deals with a group of individuals deciding who should get which items, from a set of different, indivisible items.

In this work, we focus on agents who have **unit-demand**, where each agent desires at most one item at a time, or is unable to receive multiple items at once[4]. If agents are graduate students being assigned desks in offices, for example, each student only needs one

[4]This unit-demand form of item allocation is also known as the **assignment problem**.

desk and department regulations may only permit them to have at most one. Svensson [1983] argues that an agent requiring exactly one item is natural in settings where the indivisible items come at some large cost relative to the agent's total resources, such as a full-time job [Leonard, 1983] or house. In these cases, the agent's utility for two items would be significantly lower than the sum of utilities for the items individually. The following example illustrates this problem.

Example 2.1. *Four colleagues, Alex, Lou, Sam and Vic, are deciding who should borrow which company car over the upcoming long weekend. Available to them are a luxury sedan, a small hatchback, two vans and an SUV. They all have different requirements and preferences for the vehicles. Alex, for example will be buying large items for a renovation, so needs the space of the SUV or van, while Lou is going on a family trip, so would like the space of the SUV or comfort of the sedan. They each work out their preferences, as summarised in Table 2.1, and then must decide who gets which vehicle.*

	Sedan	Hatch	Van 1	Van 2	SUV
Alex	\$30	\$10	\$60	\$50	\$130
Lou	\$90	\$20	\$40	\$35	\$100
Sam	\$90	\$50	\$10	\$10	\$40
Vic	\$85	\$40	\$15	\$10	\$70

Table 2.1: *Item allocation example agents. Values agents receive for hiring each of the five vehicles.*

Formally, we have a set of n agents, $\mathbf{N}$, and a set of m heterogeneous, indivisible items, $\mathbf{M}$. Each agent i has a value for each item j, denoted $v_i(j)$. The values for all agents is captured in a **value profile** $v \in V$. The outcomes that the SCF $f : V \to \mathbb{A}$ can choose from is the space of all possible allocations of items to agents, $\mathbb{A}$. Since agents have unit demand, and can only receive at most one item, each **allocation** $a \in \mathbb{A}$ is a function $a : \mathbf{N} \to \mathbf{M} \cup \{0\}$. Thus, $a(i)$ is the item agent i receives, or 0 if the agent receives no item (if $n > m$). Since two different agents cannot receive the same item, we require $a(i) \neq a(j)$ if $i \neq j$, unless $a(i) = a(j) = 0$.

Each agent $i \in \mathbf{N}$ has a value associated with being assigned item $j \in \mathbf{M}$. This value is independent of other items and agents, and is private information to the agent. An agent's type $\theta_i \in \Theta_i$ captures its value for each of the m items. Additionally, these values are non-negative since we assume **free disposal**, where agents can freely discard items that would otherwise provide negative utility. Since agents do not care about externalities, which is a reasonable assumption whenever agents are primarily focussed on their own resources, these item values are the only information needed to describe an agent's type, thus $\Theta_i = \mathbb{R}^m_{\geq 0}$. To simplify notation, we use $v_i(j)$ to denote the value agent i receives on being assigned item j, and denote $v_i(a) = v_i(a(i))$. The values for all agents excluding i is denoted v_{-i}, and so $v = (v_i, v_{-i})$.

The set of all possible value profiles is $V = \mathbb{R}^{n \times m}_{\geq 0}$, where $v = (v_1, \ldots, v_n)$, for $v \in V$, $v_i \in V_i$. Agents report their item values $\hat{v}_i$ (which are not necessarily truthful) to the allocation mechanism, which determines the item each agent receives, along with agents' payments. Since agents are risk-neutral with **quasi-linear utilities**, the utility of agent $i \in \mathbf{N}$, given mechanism $\mathcal{M} = (f, t)$ when agents report values $v \in V$ is:

$$u_i = v_i(a(i)) + t_i(v) \tag{2.3}$$

where $a = f(v)$. When we talk about the value of an allocation a, this is the total of all agents' values for that allocation $\sum_{i \in \mathbf{N}} v_i(a)$.

We study DSIC mechanisms while imposing a strong budget balance (SBB) constraint on the payment function.

Definition 2.4 (Strong Budget Balance (SBB)). *A mechanism $\mathcal{M} = (f, t)$ is* ***strongly budget balanced*** *if all payments always sum to zero. So, for all $v \in V$, $\sum_{i \in \mathbf{N}} t_i(v) = 0$.*

A mechanism designer may have SBB as a desirable property since it prevents one potential loss of utility for the group of agents. Alternatively, it may be an explicit constraint, since budget imbalance requires either an external funding source, or some way to dispose of a surplus in a neutral way, which may not be possible. Donating a surplus to charity, for example, alters agents' utility functions, and may give some agents incentive to misreport in order to increase the donation. Destroying the surplus through money burning, if it is even possible, also alters agents' utility functions as it changes the value of remaining money. A virtual currency for a specific situation, for example, can have very small money supply, amplifying the effect of money burning. Finally, if a mechanism allocates resources in an iterated scenario, a budget imbalance could soon drain all money from the group of agents.

While an ideal property in some settings, achieving SBB adds constraints to valid payment functions that, when combined with requirements for truthfulness, can preclude properties such as efficiency [Green and Laffont, 1979] (see Subsection 3.2.2). Since truthfulness and efficiency are often hard constraints of the mechanism designer, the SBB constraint is typically relaxed to weak budget balance.

Definition 2.5 (Weak Budget Balance[5] (WBB)). *A mechanism $\mathcal{M} = (f, t)$ is* ***weakly budget balanced*** *if the sum of all payments received by agents is always non-positive. So, for all $v \in V$, $\sum_{i \in \mathbf{N}} t_i(v) \leq 0$.*

[5] Also known as **no subsidy** since the mechanism will not need to subsidise the agents in order to carry out the resource assignment.

While strong (or weak) budget balance and DSIC are constraints for our problem, when we compare mechanisms, we evaluate by efficiency and social surplus ratio. We cover these measures in detail in Section 3.1 before we introduce deterministic solutions to this problem.

2.1.1 Task Imposition

A congruent problem to the allocation of heterogeneous items to unit demand agents is that of **task imposition** [Leonard, 1983; Demange *et al.*, 1986] (or **job assignment**). Instead of $\mathbf{M}$ containing a set of desirable resources, it contains items for which the agents receive negative utility and have no free disposal. This may be the assignment of computational tasks to a set of processors, chores to housemates, or waste-disposal sites to communities. Our model remains the same, except agents' values, $V = \mathbb{R}_{\leq 0}^{n \times m}$, are non-positive rather than non-negative. The free disposal assumption does not apply in this setting; otherwise, the problem would be trivial. Unless some (possibly fixed) external subsidy is allowed, mechanisms for this problem cannot be individually rational. All agents receive negative utility for the tasks they are assigned, so to ensure each agent has positive utility overall, all agents must receive some positive payment.

2.2 Room Assignment–Rent Division

The **Room Assignment–Rent Division** (RARD) problem has a setting that is closely related to heterogeneous item allocation, with much previous work in the literature [Svensson, 1983; Maskin, 1986; Alkan *et al.*, 1991; Tadenuma and Thomson, 1991; Tadenuma and Thomson, 1995; Su, 1999; Meertens *et al.*, 2002; Willson, 2003; Abdulkadiroğlu *et al.*, 2004;

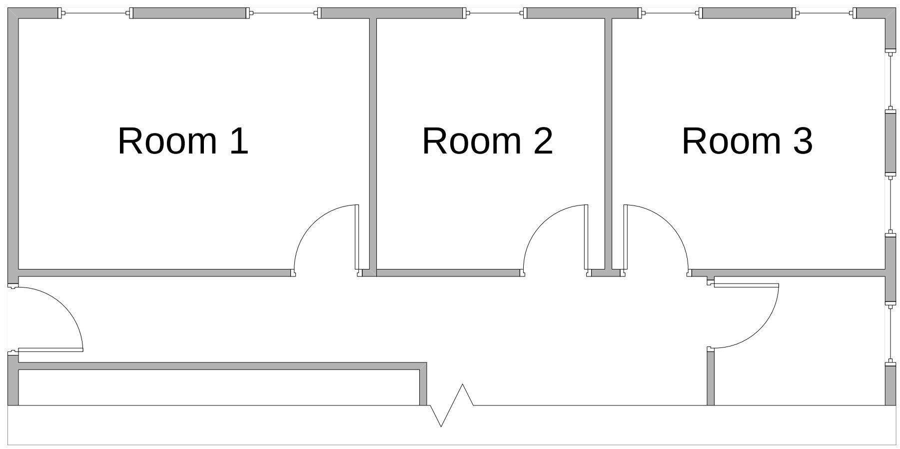


Figure 2.1: *Partial floor plan showing the three bedrooms for the room assignment–rent division example.*

Sakai, 2007; Kojima, 2009; Yenmez, 2012]. We illustrate this problem with the following example.

Example 2.2. *Consider a group of students, Morgan, Sasha, and Taylor, who are renting a house together (illustrated in Figure 2.1). They must decide among themselves who should get which room, given that they each have independent preferences over the rooms. Morgan likes rooms with plenty of space, such as room 1, while Taylor likes windows with a good view, like room 3. The renters must also decide how to divide the $1000 monthly rent. To avoid tension amongst the roommates, rather than only optimising total utility, we try to avoid cases were one renter is envious of another renter's room and rent. Consider the agents' with values for each room as summarised in Table 2.2 (e.g. Morgan would be prepared to pay $500 per month for room 1). If we set the rent for the three rooms as $400, $200 and $400, with Morgan in room 1, Sasha in room 2, and Taylor in Room 3, the three renters have utilities for each room as summarised in Table 2.3. Since everyone's utility* *for their own assigned rooms (in bold) is the maximum of the three rooms, they all believe they are in the best room for the current prices. This room assignment and rent division is said to be* envy-free *(see Section 3.1).*

	Room 1	Room 2	Room 3
Morgan	$500	$200	$400
Sasha	$400	$350	$400
Taylor	$350	$300	$500

Table 2.2: *RARD example agents. Amounts that agents would be willing to pay as monthly rent for rooms in the example house.*

	Room 1 ($400)	Room 2 ($400)	Room 3 ($200)
Morgan	**$100**	$0	$0
Sasha	$0	**$150**	$0
Taylor	-$50	$100	**$100**

Table 2.3: *RARD example agents. Agents' utilities for each of the three rooms, with the specified rent on each room. With agents in their assigned rooms (in bold) the assignment and division is envy-free.*

While room assignment is the classic and easy to explain example for this model, it has much more general applicability and is a well studied problem. In a task imposition problem (see Subsection 2.1.1), this model is appropriate for a fair division of both labour and compensation. Instead of a rent to be paid, there is compensation for the workers on task completion. The role of tasks and workers can be reversed, for example, with agents each owning tasks to be assigned to a heterogeneous set of equipment. Other scenarios from the literature include land distribution, inheritance division, corporate partnership dissolution and the assignment of parking spaces to employees.

As with the heterogeneous item allocation problem, in RARD we assign a set of indivisible, heterogeneous items (e.g. rooms in a shared house), $\mathbf{M}$, to a set of agents, $\mathbf{N}$ such that all agents receive exactly one item. However, there are no surplus or deficit items, so $|\mathbf{N}| = |\mathbf{M}|$. Additionally, there is also some total payment T of a divisible resource (e.g. rent) to be completely divided among the agents. This allocation and division is performed simultaneously. Each agent $i \in \mathbf{N}$ has a value for each item $j \in \mathbf{M}$, denoted as $v_i(j)$ with the unit of the divisible resource as the numeraire. Note that T may be negative, as in the case where it represents rent, or it may be positive, in which case there is some total payment that the agents have received as a group that must be divided.

A mechanism for this problem, $\mathcal{M} = (f, t)$, determines a room assignment through $f : V \to \mathbb{A}$, and a rent division through $t : V \to \mathbb{R}^n$. As before, we assume agents' types are private, so an RARD mechanism receives reported agent values $v \in V$ and produces an allocation, $a = f(v)$, where $a : \mathbf{N} \to \mathbf{M}$, and a rent division $t(v)$. A valid allocation, a, must be bijective so every agent receives one item, every item is assigned to one agent. To ensure agents don't misreport, we focus on DSIC mechanisms. Similar to SBB, we require $\sum_{i \in \mathbf{N}} t_i(v) = T$, $\forall v \in V$, since agents cannot be short on rent, and any surplus would have to be destroyed. Agents have quasi-linear utilities, so an agent's utility is calculated the same as in Equation (2.3), that is:

$$u_i = v_i(a) + t_i(v) \tag{2.4}$$

where $v_i(a) = v_i(a(i))$.

2.2.1 Multi-house Room Assignment–Rent Division

The RARD model involves the division of exactly 1 divisible resource, such that there is no surplus of deficit. The model can be extended to allow multiple divisible resources. A similar extension was done in fair division, where Cloutier *et al.* [2010] generalised the cake-cutting problem to one that fairly divides multiple cakes instead of just one.

The multi-house room assignment–rent division scenario is an extension to RARD, where each agent is allocated a room from one of two or more houses, as well as a share of the rent for the particular house they occupy. Only the agents in house A contribute to the rent of house A, while those in house B contribute to the rent of house B. Other applications include allocation of tasks to workers, where each task belongs to a department with its own funding; or the allocation of computational jobs to a set of heterogeneous devices where the movement of jobs between devices carries a high cost.

As before, we have agents $\mathbf{N}$ with preferences over rooms $\mathbf{M}$ and a mechanism $\mathcal{M} = (f, t)$ determines the room assignment and rent division. However, we now have additional constraints on t to ensure the total rent for each house is met. Given $k > 1$ houses, each house j has a total amount of divisible resource T_j. We also partition the rooms $\mathbf{M}$ into k disjoint, non-empty sets $\mathbf{M}_1 \ldots \mathbf{M}_k$, where each set represents a house. For each house j under allocation $a \in \mathbb{A}$, we require $\sum_{i \in \mathbf{N}_j(a)} t_i(v) = T_j$, where $\mathbf{N}_j(a) = \{i \in \mathbf{N} : a(i) \in \mathbf{M}_j\}$ (i.e. the set of agents assigned to house j under allocation a).

As with the single house setting, it is desirable to find an allocation rule such that no agent is envious of another agent's room and rent (see Section 3.1).

2.3 Public Good

The **public good** (or public project) problem [Foley, 1967; Chen, 2008; Apt *et al.*, 2008; Apt and Estévez-Fernández, 2009] is a setting where a group of agents decide whether or not to purchase some resource or undertake some project. If the project goes ahead, all agents benefit to varying degrees, but they must collectively fund it. A common example is a decision made by a community on whether or not to build a bridge. Some members of the community are more likely to use the bridge than others, so would be willing to pay more to fund the project. The community should only go ahead with construction of the bridge if the total reported benefit of the bridge exceeds the total cost.

Example 2.3. *Three graduate students, Simina, John and Lachlan, share an office together and see an ad for a \$50 refrigerator. While they agree it will make a good addition to the office, they must decide whether the price makes the purchase worthwhile. Simina typically brings lunch from home while John and Lachlan usually buy from the cafeteria, so their individual valuations are \$25, \$10 and \$5, respectively. In this case, it would not make sense to purchase, since the collective benefit is less than the total cost.*

If, instead, Lachlan started taking lunch every day, so saw a \$30 benefit to having the fridge, the total benefit, \$65, outweighs the cost, so the fridge should be purchased.

Again, we have a set of agents, $\mathbf{N}$, but the possible outcomes $\mathbb{X} = \{X_0, X_1\}$ are to either acquire the public good (X_1) or to forego it (X_0). The public good comes at a price of T, which is to be paid by all n agents. Note that in our model of the public good problem, agents receive $\frac{T}{n}$ when the good is not purchased. This implies that the funding for the good has already been procured, and is refunded to the agents should the mechanism decide not to purchase. An agent $i \in N$ of type θ_i has a private value for the good, denoted v_i. An agent's utility (excluding additional payments from the mechanism) for the two outcomes is

$$u_i(\theta_i, X_0) = \frac{T}{n} \tag{2.5}$$

$$u_i(\theta_i, X_1) = v_i \tag{2.6}$$

since the payment for the public good is distributed evenly over all agents. The total utility of all agents is T if the project is not completed, and if the project is completed the total utility is

$$\sum_{i \in \mathbf{N}} v_i \tag{2.7}$$

A mechanism for this problem, $\mathcal{M} = (f, t)$ takes the set of reported agent values, and $f : V \to \mathbb{X}$ determines whether or not the good is acquired or project is completed, while $t : V \to \mathbb{R}^n$ determines any payment received by the agent. Since the payment of $\frac{T}{n}$ is already incorporated into each agent's utility function, this is not included in the payment function. Agents may misreport their true value, so we again focus on DSIC mechanisms.

An alternative public good model defines utility functions as:

$$u_i(\theta_i, X_0) = 0 \tag{2.8}$$

$$u_i(\theta_i, X_1) = v_i - \frac{T}{n} \tag{2.9}$$

This model implies agents are charged equally $\frac{T}{n}$ only when the good is to be purchased. This simply shifts all agents' utilities or payment functions by a constant amount, so will not affect the space of mechanisms. In this work, however, we define utilities as described in Equations (2.5) and (2.6), since this ensures all agents have positive utilities, which simplifies equations in later chapters.

The version of the public good problem we examine is single-parameter, as there is only one good to choose from. A recently studied extension to this is the combinatorial public project problem [Papadimitriou *et al.*, 2008; Buchfuhrer *et al.*, 2010], where there are multiple public goods, and agents have preferences over subsets of these goods. A simple example of this is a community deciding whether to build (and pay for) either a swimming pool, a tennis court, both or neither.

Chapter 3

Deterministic Solutions

The focus of this work is on examining stochastic solutions to computational social choice problems, along with techniques appropriate for assessing these measures. In this chapter, we cover existing deterministic solutions and we also introduce some important new solutions and results. It is these deterministic solutions that that we build upon to get stochastic solutions in Chapter 4. Before presenting the deterministic mechanisms, we introduce the measures that are used to evaluate these solutions in Section 3.1. This chapter also includes relevant characterisations on the space of possible mechanisms and impossibility results in order to demonstrate worst-case optimal mechanisms. We present an important new characterisation of deterministic DSIC mechanisms for heterogeneous item allocation, with an independence constraint, in Subsection 3.2.4, and extend this to cover DSIC and SBB mechanisms in Subsection 3.2.5. From this, we focus on affine-maximising social choice functions.

Before introducing the stochastic mechanisms in Chapter 4 for the problems we described in Chapter 2, in this chapter we cover deterministic solutions for these problems in Subsections 3.3 to 3.5. In addition to providing and surveying mechanisms, we present worst-case bounds on the performance of deterministic mechanisms for these problems. It is the poor worst-case performance and impossibility results when using deterministic mechanisms that motivates us to examine stochastic solutions in this work.

3.1 Deterministic Measures

For multiagent resource allocation problems where the focus is on maximising the welfare of the agents receiving the items rather than the profit of the auctioneer, mechanism designers typically focus on two deterministic properties. These are achieving allocative efficiency and the worst-case level of social surplus ratio.

Definition 3.1 (Allocative Efficiency)**.** *An allocation $a \in \mathbb{A}$ is* ***allocatively efficient***[1] *(or simply* ***efficient****) for value profile $v \in V$ if*

$$\sum_{i \in \mathbf{N}} v_i(a) \geq \sum_{i \in \mathbf{N}} v_i(a'), \forall a' \in \mathbb{A} \tag{3.1}$$

That is, allocation a maximises the sum of all agents' utilities (excluding payments).

An **efficient mechanism** will always choose an efficient outcome, for any value profile. We denote the efficient allocation for a value profile v as $a^*(v)$, or simply a^* where the value profile is clear from context.

$$a^*(v) \in \arg\max_{a \in \mathbb{A}} \sum_{i \in \mathbf{N}} v_i(a) \tag{3.2}$$

When the focus is on the social choice function, efficiency is a natural requirement and corresponds to a utilitarian maximisation of the collective social welfare in the chosen

[1] Sometimes referred to as Pareto efficient.

outcome. This is also referred to as a **utilitarian social choice function**. If the outcome changes from an efficient to a non-efficient outcome, some agents will lose utility while other agents may gain utility. However, by definition, the total loss of utility outweighs the total gain.

For the second property, we examine the worst-case ratio of the sum of all agents' utilities (including payments) in the chosen outcome, to the value of the efficient allocation.

Definition 3.2 (Social Surplus Ratio [Guo and Conitzer, 2009; Moulin, 2009])**.** *The* ***social surplus ratio*** *is the ratio of the total utility achieved by the mechanism for a value profile* $v \in V$ *against the best possible total utility. This is calculated as:*

$$r(v) = \frac{\sum_{i \in \mathbf{N}} v_i(f(v)) + \sum_{i \in \mathbf{N}} t_i(v)}{\sum_{i \in \mathbf{N}} v_i(a^*(v))} \tag{3.3}$$

For the purposes of worst case ratio, we exclude the value profile $v = \mathbf{0}$, where no agent is interested in any item. Otherwise this would give an undefined ratio. It is possible, however, for agents to have infinitesimal values for each item, or some agents to have zero values. The ratio measure is independent of the scale of the units of the agents' utilities. It captures a measure of how close the mechanism comes to achieving the maximum possible total utility for a group of agents. A worst-case ratio of 1, for a WBB mechanism, represents a mechanism that is always efficient and strongly budget balanced. Any weakly budget balanced mechanism cannot have a ratio exceeding 1 for any value profile, since this would require a positive net payment. A loss of total utility, lowering the ratio below 1, either comes from an inefficient allocation, or from non-budget balanced payments that are received by some outside entity. A mechanism with a ratio of 0 will have some value profile where the total payment cancels out the value of the allocation. Finally, an individually rational mechanism cannot ever have a negative ratio, since a negative ratio implies the total value of the outcome is less than the total payments. A non-negative worst-case ratio does not necessarily imply IR, however.

When we only look at SBB mechanisms, we have $\sum_{i \in \mathbf{N}} t_i(v) = 0$, so the only loss of social surplus comes from inefficient allocation, $f(v) \neq a^*(v)$. Thus, for SBB mechanisms, the social surplus ratio is the ratio of the sum of agents' values for $f(v)$ compared to the sum of values in the efficient allocation, $a^*(v)$.

Efficiency may be ideal from a collective viewpoint, but may result in an individual agent seeing a large loss so that a large number of other agents can see a marginal gain. Thus, in some settings we are interested in not just an efficient allocation, but also some notion of fairness. In this work we focus on *envy* and *envy-freeness* as measures of fairness for allocation problems. An allocation assigns to each agent a bundle, which contains an item (or set of items) along with some price or payment. For a particular allocation of bundles to agents, an agent is *envious* if it views another agent's bundle as strictly better than its own. Envy-freeness was introduced by Foley [1967], and Brams and Taylor [1996] discuss envy-freeness and other measures of fairness in the setting of fair division.

Definition 3.3 (Envy)**.** *Given an outcome consisting of allocation $a \in \mathbb{A}$ and payments $t(v)$, agent $i \in \mathbf{N}$ is **envious** of agent $j \in \mathbf{N}$ if*

$$v_i(a(i)) + t_i(v) < v_i(a(j)) + t_j(v) \tag{3.4}$$

Definition 3.4 (Envy-free)**.** *An outcome consisting of allocation $a \in \mathbb{A}$ and payments $t(v)$ is **envy-free** if there is no agent than envies another agent. That is:*

$$v_i(a(i)) + t_i(v) \geq v_i(a(j)) + t_j(v) \ , \forall i, j \in \mathbf{N} \tag{3.5}$$

An **envy-free mechanism** will always produce an envy-free outcome, for all value profiles $v \in V$. Envy-free outcomes mean that every agent, according to its own utility function, finds its current assignment to be the best of all the agents' assignments. As well as being fair, an envy-free outcome can also be desirable due to the stability it provides. No agent, or group of agents, would attempt to deviate from the mechanism's prescribed bundles as all agents would be worse off. Consider, the envy-free allocation in Example 2.2 of Section 2.2. While Sasha and Taylor both prefer Room 3 to Room 2, at the current rent, Sasha would be worse off by forcing Taylor to swap, because, with payments included, Sasha receives higher overall utility in Room 2.

The results we present next in Section 3.2 focus on efficiency but are directly relevant to envy-free mechanisms. Envy-freeness is a stronger requirement than efficiency as every envy-free allocation is also an efficient allocation in our RARD settings [Svensson, 1983; Alkan *et al.*, 1991]. The requirement of envy-freeness adds constraints on the possible payments received under the efficient allocation.

3.2 Characterisations

To motivate the use of randomisation, in this section we present some existing and new characterisations on DSIC mechanisms satisfying efficiency or SBB. We describe Roberts' well known characterisation of truthful social choice functions where agents have a complete domain of preferences. We then provide a new characterisation of truthful mechanisms for heterogeneous item allocation under an additional constraint of *independence*, then refine this further to cover SBB mechanisms. These characterisations lead to poor worst-case performance or impossibility results for deterministic mechanisms, which motivate us to examine randomised approaches in Chapter 4.

3.2.1 Groves Mechanisms

An important and influential result in mechanism design was the definition of the Groves class of mechanisms [Vickrey, 1961; Clarke, 1971; Groves, 1973]. This class defines mechanisms which are efficient and DSIC for agents with quasi-linear utility.

Definition 3.5 (Groves mechanism)**.** *A mechanism $\mathcal{M} = (f, t)$ belongs to the **Groves class of mechanisms** if*

- *Social choice function f produces the efficient outcome (maximising the sum of all agents' utilities):*

$$f(v) \in \arg\max_{X \in \mathbb{X}} \sum_{i \in \mathbf{N}} v_i(X) \tag{3.6}$$

- *Given $X = f(v)$, agents receive payments according to payment function:*

$$t_i(v) = \sum_{j \neq i} v_j(X) + h_i(v_{-i}) \ , \forall i \in \mathbf{N} \tag{3.7}$$

  for some additional payment functions $\{h_i(v_{-i})\}_{i \in \mathbf{N}}$.

Since an agent receives a payment equal to the sum of all other agents' utilities for the selected outcome, its total utility including payment matches the sum maximised by the SCF, plus some additional payment $h_i(v_{-i})$. Thus, if an agent tries to misreport its own type to change the outcome selected by f, it will only lower its own utility. The rebate function $h_i(v_{-i})$ can be any function independent of agent i's reported type, thus making it a constant payment from agent i's perspective. Thus, the form of the payment functions ensures the mechanism is DSIC.

The Groves class of mechanisms are particularly important as they are the *only* mechanisms that are both efficient and DSIC for agents with quasi-linear utilities [Green and

Laffont, 1977]. So when examining DSIC and efficient mechanisms, the only degree of freedom is to change the additional payment functions. However, this choice can greatly affect the mechanism on measures such as the social surplus ratio. For example, simply setting $h_i(v_{-i}) = 0$ leads to a violation of WBB, since all agents are receiving positive payments from the mechanism.

3.2.1.1 The Vickrey–Clarke–Groves Mechanism

Of all the mechanisms in the Groves class, the **Vickrey-Clarke-Groves** (VCG) mechanism (also known as the **pivotal mechanism** or **Clarke mechanism**) is perhaps the most well known. Being from the Groves class, it is DSIC and the SCF chooses the efficient outcome. Agents make payments so as to maximise the total payments made to the mechanism, while still maintaining individual rationality [Clarke, 1971]. The additional payments received by the agents are calculated as

$$h_i(v_{-i}) = -\sum_{j \neq i} v_j(X_{-i}) \tag{3.8}$$

where

$$X_{-i} \in \arg\max_{X \in \mathbb{X}} \sum_{j \in \mathbf{N} \setminus \{i\}} v_j(X) \tag{3.9}$$

The outcome X_{-i} is the efficient outcome when excluding agent i. Combined with the rest of the Groves payment, an agent is paying the total loss in utility the other agents have experienced due to its participation (i.e. the agent pays its externality). Agents whose participation does not affect the outcome end up paying nothing. For the VCG mechanism to be individually rational in our settings, we require every agent $i \in \mathbf{N}$ to have $v_i(X_{-i}) \geq 0$. This mechanism is WBB if there is no agent whose removal can result in a net loss for the remaining agents. This condition is known as the no single-agent effect [Parkes, 2001].

Definition 3.6 (No single-agent effect)**.** *A mechanism $\mathcal{M} = (f, t)$ has the* ***no single-agent effect*** *if, for all $v \in V$*

$$\sum_{j \neq i} v_j(f(v)) \leq \sum_{j \neq i} v_j(X_{-i}) \qquad \forall i \in \mathbf{N} \tag{3.10}$$

The no single-agent effect holds in all the settings we examine where agents have non-negative utilities. When allocating items or rooms, for example, removing an agent will increase the set of available items for the remaining agents, so they are guaranteed to be at least as well off.

While VCG seems like an ideal mechanism as it is efficient, individually rational and weakly budget balanced, the potential budget imbalance leads to very poor worst-case total utility. In the public project setting, VCG gives a worst-case social surplus ratio of $\frac{1}{n}$ [Guo *et al.*, 2011] (see Section 3.5). Under an item allocation setting, the worst-case performance is even more dire, with a ratio of 0 (see Section 3.3).

3.2.2 Green-Laffont Impossibility Theorem

While the VCG mechanism is DSIC and always produces an efficient outcome, the loss of utility from budget imbalance can cancel out the gains from the efficient allocation. The ideal case is a Groves mechanism that achieves SBB; however, Green and Laffont [1979] show that this is impossible for quasi-linear agents with a complete domain of preferences (unrestricted value functions). This result is widely referred to as the Green-Laffont Impossibility Theorem.

Theorem 3.1. *[Green and Laffont, 1979] There exists no Groves mechanism (i.e. efficient and DSIC) such that* $\sum_{i \in \mathbf{N}} t_i(v) = 0$ *for all* $v \in V$*, where* $V = \mathbb{R}^{n \times |\mathbb{X}|}$.

This means that, generally, a social surplus ratio of one is not possible for DSIC, WBB mechanisms, since it cannot be simultaneously efficient and SBB.

3.2.3 Roberts' Characterisation

The space of Groves mechanisms are the DSIC mechanisms that implement an efficient social choice function. Such SCFs typically cannot be budget balanced however (due to Theorem 3.1), and so an inefficient SCF may provide a better overall social surplus ratio. Roberts [1979] examined the space of social choice functions that can be implemented in a DSIC mechanism given agents with quasilinear utility functions and a complete domain of preferences. With a complete domain of preferences, agents' types assign a separate value for each possible outcome. This means that Robert's characterisation does not apply in a unit demand item allocation setting, for example, since agents are necessarily indifferent between two outcomes if their own items do not change. Roberts' characterisation relies on the property of positive association of differences in the social choice function.

Definition 3.7 (Positive Association of Differences (PAD) [Roberts, 1979]). *We say that* f *satisfies* ***positive association of differences*** *if, for any two value profiles* $v, v' \in V$ *where* $X = f(v)$,

$$v'_i(X) - v_i(X) > v'_i(Y) - v_i(Y) \qquad \forall i \in \mathbf{N}, \forall Y \in \mathbb{X} \setminus \{X\} \tag{3.11}$$

implies $f(v') = X$.

PAD requires that a social choice function keep the same outcome if all agents increase their *relative* value for that outcome compared to all other outcomes. This is the primary property that determines whether an SCF can be implemented in a DSIC mechanism. In particular, these DSIC mechanisms require an affine maximising SCF.

Definition 3.8 (Affine Maximiser [Roberts, 1979]). *A social choice function f is an* ***affine maximiser*** *if, for any $v \in V$:*

$$f(v) \in \arg\max_{a \in \mathbb{A}'} \left(U_0(a) + \sum_{i \in \mathbf{N}} \left(\gamma_i v_i \left(a \right) \right) \right) \tag{3.12}$$

for some constants $(\mathbb{A}', U_0, \gamma)$, where $\mathbb{A}' \subseteq \mathbb{A}$ is the space of possible outcomes, $U_0 : \mathbb{A} \to \mathbb{R}$ is a set of weights for each outcome, and $\gamma_1, \ldots, \gamma_n \geq 0$ are weights for each agent, where $\exists i \in \mathbf{N}$ s.t. $\gamma_i > 0$.

Since Equation (3.12) may not specify a unique outcome, we can model this SCF more precisely as a multi-round affine maximiser. A **multi-round affine maximiser** can be considered as a series of affine maximisations, $f^{(1)}, \ldots, f^{(j)}$ that are performed until there is a unique outcome selected. Each $f^{(k)}$ is an affine maximiser of the form of Equation (3.12), defined by $(\mathbb{A}^{(k)}, U_0^{(k)}, \gamma^{(k)})$. An agent will only have a non-zero weight $\gamma_i^{(k)}$ in at most one round. If there are multiple outcomes that satisfy $f^{(k)}$, then affine maximisation $f^{(k+1)}$ will perform an additional affine maximisation, where $\mathbb{A}^{(k+1)}$ is the set of allocations that maximised $f^{(k)}(v)$. These "tie-breaking" affine maximisations can even be constant orderings independent of all agents' types (all agents have zero-weight), or dictatorial (only a single agent with non-zero weight). For a DSIC mechanism implementing a multi-round affine-maximising SCF, each round of affine maximisation has payments of the form [Roberts, 1979] $\forall i \in \mathbf{N}$, $v \in V$:

$$t_i^{(k)}(v) = \begin{cases} \dfrac{1}{\gamma_i^{(k)}} \left(U_0^{(k)} \left(f^{(k)}(v) \right) + \displaystyle\sum_{j \neq i} \gamma_j v_j^{(k)} \left(f^{(k)}(v) \right) \right) + h_i^{(k)}(v_{-i}) & \gamma_i^{(k)} > 0 \\ h_i^{(k)}(v_{-i}) & \gamma_i^{(k)} = 0 \end{cases} \tag{3.13}$$

where $\{h_i^{(k)}(v_{-i})\}_{i\in\mathbf{N}}$ are additional payment functions that are independent of an agent's own declared value and thus do not affect the mechanism's truthfulness. Recall that these are payments *received* by the agents. The full payment function of the mechanism is thus $t_i(v) = \sum_k t_i^{(k)}(v)$. In this work, we typically focus on a single round of affine maximisation at a time, so for simplicity of notation, we omit the superscript denoting the round when the round number is unambiguous.

A Groves mechanism is an affine maximiser with the initial maximisation performed with $\mathbb{A}' = \mathbb{A}$, $U_0 = 0$ and $\gamma_i = 1$, $\forall i \in \mathbf{N}$. An extreme example of a multi-round affine maximiser is the serial dictatorship, where agents one-by-one choose their most preferred item from the remaining items, according to some fixed agent ordering. This is a series of affine maximisers on a shrinking set of possible outcomes, where each maximisation has a single non-zero-weighted agent, until only a single outcome remains.

For agents with $\gamma_i > 0$ in an affine maximising DSIC mechanism, Equations (3.12) and (3.13) combine to give the agent a utility of

$$u_i(v) = \frac{1}{\gamma_i} \max_{a\in\mathbb{A}'} \left(U_0(a) + \sum_{i\in\mathbf{N}} (\gamma_i v_i(a)) \right) + h_i(v_{-i}) \tag{3.14}$$

Theorem 3.2. *[Roberts, 1979] In an environment with quasilinear agents, a complete domain of preferences, and at least 3 outcomes:*

1. *If social choice function f is implementable in a deterministic DSIC mechanism then f satisfies PAD*
2. *f satisfies PAD if and only if f is an affine maximiser*

Roberts originally asserted that an affine-maximising SCF f also implied that f was implementable in a DSIC mechanism, but Carbajal *et al.* [2013] have since shown this to be false, unless f is from the more-restricted class of lexicographic (i.e. multi-round) affine maximisers.

While Theorem 3.2 is a strong result, it doesn't directly fit in any of the settings we examine. Under the item allocation and RARD settings, agents' preferences are not over the entire space of outcomes, since an agent is necessarily indifferent between two outcomes where it receives the same item. Under these settings, PAD is not a meaningful measure, since the inequality in Equation (3.11) can never hold. In a public good setting there are only two outcomes, so Theorem 3.2 does not apply.

If we allow agents to have full preferences over all allocations, such as in RARD if roommates could have preferences over their neighbours as well as their own room, then Theorem 3.2 would apply to that setting. In Subsection 3.2.4, we provide a new characterisation that does apply in a unit demand allocation setting, with similar requirements on the SCF to Theorem 3.2.

3.2.4 DSIC Unit Demand Allocation Mechanisms

In this work, when looking at DSIC mechanisms, we restrict our attention to affine maximisers with tie-breaking (multi-round affine maximisers). As Subsection 3.2.3 shows, this is an interesting and well-studied class of mechanisms, and includes all mechanisms in the Groves class. We adapt Roberts' characterisation [1979] of DSIC mechanisms to show sufficient conditions under which a DSIC item allocation mechanism must be an affine maximiser. We introduce some key definitions and previous results before providing a characterisation of all DSIC and SBB mechanisms. In particular, we will show that DSIC necessitates weak monotonicity in SCFs, and when there are more than two possible outcomes, if we additionally require independence, then the SCF must be an *affine maximiser*.

Definition 3.9 (Weak Monotonicity (W-MON) [Lavi *et al.*, 2003]). *Social choice function f satisfies **W-MON** if, for any $v \in V$, $i \in \mathbf{N}$, $v'_i \in V_i$: if $f(v) = a$ and $f(v'_i, v_{-i}) = b$, then:*

$$v'_i(b) - v_i(b) \geq v'_i(a) - v_i(a) \tag{3.15}$$

With a weakly monotonic SCF, for agent i to change the allocation from a to b by changing its reported value, its value for a cannot increase more than its increase for b.

Definition 3.10 (Independence [Lavi *et al.*, 2003]). *Social choice function f satisfies **independence**[2] if, for any $v, v' \in V$, if $f(v) = a$ and $f(v') = b \neq a$, then $\exists i \in \mathbf{N}$ s.t.*

$$v'_i(a) - v'_i(b) \neq v_i(a) - v_i(b) \tag{3.16}$$

If an allocation rule satisfies independence, it cannot change the allocation from a to b if no single agent changes its difference in value (relative preference) between allocation a and b.

Definition 3.11 (Strong Monotonicity (S-MON)). *Social choice function f satisfies **S-MON** if and only if f satisfies both W-MON and independence. Equivalently, Equation (3.15) must hold strictly if $a \neq b$.*

These properties are used in characterisations of which social choice functions can be implemented in DSIC mechanisms.

Definition 3.12 (Rich Domain [Bikhchandani *et al.*, 2006]). *A **rich domain** for a set of possible outcomes $\mathbb{A}$ has the set of agents' types as all $v_i \in \mathbb{R}_+^{|\mathbb{A}|}$ that are consistent with some partial order over $\mathbb{A}$.*

[2] Lavi *et al.* [2003] refer to this as **independence of irrelevant alternatives**.

Theorem 3.3. *[Bikhchandani* et al.*, 2006] If a mechanism $\mathcal{M} = (f, t)$ is DSIC, then the social choice function f is weakly monotone.*

Proof. Consider any $v \in V$, $i \in \mathbf{N}$, $v'_i \in V_i$ such that $f(v) = a$ and $f(v'_i, v_{-i}) = b$. Since the mechanism is DSIC, utility is maximised with truthful reporting. This means

$$v_i(a) - t(v) \geq v_i(b) - t(v'_i, v_{-i}) \tag{3.17}$$

$$v'_i(a) - t(v) \leq v'_i(b) - t(v'_i, v_{-i}) \tag{3.18}$$

$$\Rightarrow v_i(a) - v_i(b) \geq t(v) - t(v'_i, v_{-i}) \geq v'_i(a) - v'_i(b) \tag{3.19}$$

$$\Rightarrow v'_i(b) - v_i(b) \geq v'_i(a) - v_i(a) \tag{3.20}$$

Valid payments only exist if Equation (3.20) always holds, and if this always holds, then f is W-MON by definition. □

Theorem 3.4. *[Bikhchandani* et al.*, 2006] A mechanism implementing a social choice function f on a rich domain is DSIC if and only if f is weakly monotone.*

In our unit demand allocation setting we have a rich domain, as the only restriction on agent types is to force equality on outcomes where the agent receives the same item.

We show which social choice functions must be affine maximisers, then show that, with the additional requirement of independence, this includes all SCFs that can be implemented in a DSIC mechanism. Roberts' [1979] result (see Subsection 3.2.3) provided a proof for an unrestricted domain, showing under which conditions an SCF must be an affine maximiser and relying on *positive association of differences* (PAD). It also requires value profiles with strict inequalities between two outcomes for all agents, which does not make sense in the unit demand allocation setting, as agents must *necessarily* have the same value for allocations where they do not change items. We can formulate a slightly different theorem that requires agents have the same value for outcomes where they receive the same item and uses a property similar to PAD, which we present in Lemma 3.1.

For Lemma 3.1 to fit closely with Roberts' proof, in this subsection we alter our notation slightly, and define $V = \mathbb{R}^{|\mathbf{N}|\times|\mathbb{A}'|}$ as a set of agents' values over all possible *allocations* $\mathbb{A}' \subseteq \mathbb{A}$ (instead of values over all items). Thus, for any $x, y \in \mathbb{A}$, we need to explicitly enforce $v_i(x) = v_i(y)$ whenever $x_i = y_i$. The vector $v(x)$ holds the values of each agent for their allocated item in $x \in \mathbb{A}$. We define the relationship $v(x) \ll^{xy} v(y)$ (and similarly $\gg^{xy}$) between two vectors (where $v \in V$) to denote $v_i(x) < v_i(y)$ when $x_i \neq y_i$ (i.e., agent i changes item), while the two values are necessarily equal when $x_i = y_i$. This holds for all $i \in \mathbf{N}$. Note that $v(x) \ll^{xy} v(y) \equiv v(x) \ll^{yx} v(y)$, since x and y define agents whose values must be equal in this comparison.

Lemma 3.1. *If $f(v) = y$ and $v'(y) - v'(x) \gg^{xy} v(y) - v(x)$, $\forall x \in \mathbb{A}'$, $x \neq y$, for two distinct value profiles $v, v' \in V$, $v \neq v'$, and f satisfies S-MON, then $f(v') = y$.*

That is, if agents are allocated according to y for value profile v, and no agent's relative preference for an alternative item increases, then the allocation should stay as y.

Proof. Consider changing the values of agents, one by one, from v_i to v'_i. We start with $f(v) = y$, and at every step S-MON forces us to keep allocation y. Create a series of value profiles $v^j = (v_1, v_2, \ldots, v_j, v'_{j+1}, \ldots, v'_n)$ for $j \in [0, n]$, so $v = v^n$, $v' = v^0$. First, we have $f(v^n) = y$, then change agent n's value so we have v^{n-1}. Let $f(v^{n-1}) = z$, so by W-MON we have:

$$v'_n(z) - v'_n(y) \geq v_n(z) - v_n(y) \tag{3.21}$$

But given our conditions for this lemma, this can only be satisfied at equality, where agent

n keeps the same item, i.e., $y_n = z_n$. However, as all other agents keep the same value profile, no other agent can change items without violating independence, so $y = z$.

The procedure is continued, changing the value profile for each agent one by one, using $v^{n-1}, \ldots v^0$. In each case, the allocation cannot change without violating W-MON or independence. After each agent has changed its value, the value profile is v' and the allocation remains at y, and so $f(v') = f(v) = y$. □

Theorem 3.5. *If allocation function f for allocating heterogeneous items to unit demand agents satisfies S-MON, and there are at least 3 possible allocations ($|\mathbb{A}'| \geq 3$) then f is an affine maximiser.*

The proof of this theorem, which is provided in Appendix A, closely follows the proof of Theorem A in [Roberts, 1979], but with many adjustments due to our restrictions on agent types.

Theorem 3.6. *If a heterogeneous item allocation mechanism for unit demand agents is DSIC, and SCF f, with $|\mathbb{A}'| \geq 3$ satisfies independence, then f is an affine maximiser.*

Proof. By Theorem 3.4, f satisfies W-MON and since we require independence, by definition it also satisfies S-MON. Since f satisfies S-MON, by Theorem 3.5, f is an affine maximiser. □

Since Theorem 3.6 only requires DSIC mechanisms implement affine maximisation if there are at least 3 outcomes (as was also the case with Roberts' theorem [1979]), there will be other DSIC mechanisms that restrict the outcome space to two possibilities. Even with only two outcomes, a DSIC mechanism's SCF must still satisfy weak monotonicity, due to Theorem 3.4 and we discuss these SCFs later in Subsection 3.2.5.4. There also exist truthful mechanisms that violate independence, and thus are not required to be affine maximisers. However, for this work we focus on affine-maximising SCFs, and conjecture that, for our goals of efficiency and envy-freeness, these mechanisms are optimal among all truthful mechanisms.

3.2.5 Strongly Budget Balanced Mechanisms

We will now extend the results of Theorem 3.6 to show that when the SBB constraint is added to DSIC and independence, it places restrictions on the set of possible allocations, $\mathbb{A}' \subseteq \mathbb{A}$, and the set of agents $\mathbf{N}' = \{i \in \mathbf{N} : \gamma_i > 0\}$, whose reported values affect the affine maximisation. We focus on affine-maximising SCFs, and we now show that whether an affine maximising SCF of the form in Equation (3.12) can be implemented in a DSIC and SBB mechanism depends only on $\mathbb{A}'$ and $\mathbf{N}'$.

We let $a^* = f(v)$ be the "optimal" allocation[3] according to the maximisation in Equation (3.12). Because payments take the form of Equation (3.13), by strong budget balance, we require $\sum_{i\in\mathbf{N}} t_i(v) = 0$ so:

$$\sum_{i\in\mathbf{N}} h_i(v_{-i}) = -\left(\sum_{i\in\mathbf{N}'} \frac{1}{\gamma_i}\right)\left(U_0\left(a^*\right) + \sum_{i\in\mathbf{N}'} \gamma_i v_i\left(a^*\right)\right) + \sum_{i\in\mathbf{N}'} v_i\left(a^*\right) \tag{3.22}$$

When we have multiple rounds of affine maximisations, due to tie-breaking (i.e. lexicographic affine maximisers), it will be sufficient to consider a SBB mechanism as a series of SBB rounds. Every round involves additional payments $h_i(v_{-i})$ for all agents $i \in \mathbf{N}$ that "cancel out" payments required for DSIC. If we conceptualise the overall mechanism as a series of these rounds, we can move these additional payments between rounds, such

[3]Where there is no unique maximum, there is some (affine maximising) tie-breaking procedure (SCF is a multi-round affine-maximiser).

that they only cancel out payments for that round. By definition, an agent will only have a non-zero weight in a single round, so its reported value will only be used in the DSIC payments (see Equation (3.13)) in a single round.

For the purposes of SBB mechanisms, we can consider a smaller class of mechanisms. The mechanisms we look at for this problem are affine maximisers with payments in the form of Equation (3.13). However, when searching for possible affine-maximising SBB mechanisms, it is sufficient to consider those with $U_0 = 0$. Then, DSIC and SBB mechanisms with non-zero U_0 can be produced from these mechanisms by adding a constant to agents' reported v.

Lemma 3.2. *If we have a DSIC and SBB mechanism with SCF f for some $\mathbb{A}', U_0, \gamma$, then its payment functions, t, can be used to give SBB payments, t', for a DSIC mechanism with SCF f' defined by $\mathbb{A}', U_0', \gamma$.*

Proof. Equation (A.106) in the proof of Theorem 3.5 sets the structure of U_0 in S-MON affine maximisers, and is repeated here as Equation (3.23).

$$f(v) = x \Rightarrow \sum_i \gamma_i(v_i(x) - \delta_i(x)) \geq \sum_i \gamma_i(v_i(y) - \delta_i(y)) \ , \forall y \in \mathbb{A}' \tag{3.23}$$

This result shows that the constants U_0, U_0' can be defined relative to γ in the form

$$U_0(a) = -\sum_{i \in \mathbf{N}} \gamma_i \delta_i(a) \tag{3.24}$$

$$U_0'(a) = -\sum_{i \in \mathbf{N}} \gamma_i \delta_i'(a) \tag{3.25}$$

for some constants δ, δ'.

From this we can define SCF f' in terms of f:

$$f'(v) = \arg\max_{a \in \mathbb{A}'} \sum_{i \in \mathbf{N}} \gamma_i(v_i(a) - \delta'_i(a)) \tag{3.26}$$

$$= \arg\max_{a \in \mathbb{A}'} \sum_{i \in \mathbf{N}} \gamma_i(v_i(a) - \delta'_i(a) + \delta_i(a) - \delta_i(a)) \tag{3.27}$$

$$= f(v - \delta' + \delta) \tag{3.28}$$

Both mechanisms have payments of the form Equation (3.13), and the payments for (f, t) satisfy Equation (3.22), so for payment function t, we have additional payments $\{h_i(v_{-i})\}_{i \in \mathbf{N}}$ such that:

$$\sum_{i \in \mathbf{N}} h_i(v_{-i}) = -\left(\sum_{i \in \mathbf{N}'} \frac{1}{\gamma_i}\right)\left(\sum_{i \in \mathbf{N}'} \gamma_i \left(-\delta_i\left(f(v)\right) + v_i\left(f(v)\right)\right)\right) + \sum_{i \in \mathbf{N}'} v_i\left(f(v)\right) \tag{3.29}$$

And to get SBB payments for (f', t'), where $a' = f'(v) = f(v - \delta' + \delta)$, we require:

$$\sum_{i \in \mathbf{N}} h'_i(v_{-i}) = -\left(\sum_{i \in \mathbf{N}'} \frac{1}{\gamma_i}\right)\left(\sum_{i \in \mathbf{N}'} \gamma_i \left(-\delta'_i\left(a'\right) + v_i\left(a'\right)\right)\right) + \sum_{i \in \mathbf{N}'} v_i\left(a'\right) \tag{3.30}$$

$$= -\left(\sum_{i \in \mathbf{N}'} \frac{1}{\gamma_i}\right)\left(\sum_{i \in \mathbf{N}'} \gamma_i \left(-\delta_i(a') + (v_i(a') - \delta'_i(a') + \delta_i(a'))\right)\right) + \sum_{i \in \mathbf{N}'} v_i\left(a'\right) \tag{3.31}$$

$$= \sum_{i \in \mathbf{N}} h_i(v_{-i} - \delta'_{-i} + \delta_{-i}) + \sum_{i \in \mathbf{N}'} (\delta'_i(a') - \delta_i(a')) \tag{3.32}$$

So we get an SBB and DSIC mechanism for SCF f' using additional payments of $h_i(v_{-i} - \delta'_{-i} + \delta_{-i})$ for all agents $i \in \mathbf{N}$, with agents $i \in \mathbf{N}'$ also receiving the additional payment $(\delta'_i(a') - \delta_i(a'))$. □

Similarly we can remove another degree of freedom by normalising agent coefficients γ such that $\sum_{i \in \mathbf{N}} (\gamma_i)^{-1} = 1$.

Lemma 3.3. *A DSIC and SBB mechanism (f, t) with an affine maximising SCF defined by $\mathbb{A}', U_0 = 0, \gamma$ and additional payments $\{h_i(v_{-i})\}_{i\in\mathbf{N}}$ is equivalent to a mechanism (f', t') with the same $\mathbb{A}', U_0$, but with $\gamma' = \gamma(\sum_{i\in\mathbf{N}}(\gamma_i)^{-1})$.*

Proof. Since $U_0 = 0$, this positive constant scalar has no effect on the affine maximisation in Equation (3.12). Similarly, the scalar has no effect on any individual payment defined by Equation (3.13). Thus, both the allocation and payment functions are identical. □

If we have $U_0 = 0$ and $\sum_{i\in\mathbf{N}}(\gamma_i)^{-1} = 1$, then SBB payments must satisfy:

$$\sum_{i\in\mathbf{N}} h_i(v_{-i}) = -\sum_{i\in\mathbf{N}'} (\gamma_i - 1)\, v_i\, (a^*) \tag{3.33}$$

Finally, if we change γ in the SCF for a DSIC and SBB mechanism, we can construct SBB payments for this new SCF, provided the set of agents with non-zero γ_i, $\mathbf{N}'$, does not change.

Lemma 3.4. *If there exists a DSIC and SBB mechanism with affine-maximising SCF f for some $\mathbb{A}', U_0 = 0, \gamma$, where $\sum_{i\in\mathbf{N}}(\gamma_i)^{-1} = 1$. Then there exist SBB payments, t' for a DSIC mechanism with SCF f' defined by $\mathbb{A}', U_0, \gamma'$, where $\sum_{i\in\mathbf{N}}(\gamma_i')^{-1} = 1$, and $\gamma_i = 0$ if and only if $\gamma_i' = 0$.*

Proof. From mechanism (f, t), we have additional payments $\{h_i(v_{-i})\}_{i\in\mathbf{N}}$ that satisfy Equation (3.33). These additional payments calculate $f : \mathbf{N}' \to \mathbb{A}'$, which gives a^*, that is:

$$\sum_{i\in\mathbf{N}} h_i(v_{-i}) = -\sum_{i\in\mathbf{N}'} (\gamma_i - 1)\, v_i\, (f(v)) \tag{3.34}$$

SBB payments for (f', t') require additional payments h' as follows:

$$\sum_{i\in\mathbf{N}} h_i'(v_{-i}) = -\sum_{i\in\mathbf{N}'} (\gamma_i' - 1)\, v_i\, (f'(v)) \tag{3.35}$$

Where $f' : \mathbf{N}' \rightarrow \mathbb{A}'$. There exist modifications to the individual payment functions $h_i(v_{-i})$, that can calculate $f'(v) = f(\frac{\gamma'}{\gamma}v)$, since both functions use the same set of agents, $\mathbf{N}'$. The use of any agent values in individual $h_i(v_{-i})$ functions can be scaled by constant values $\frac{\gamma_i'-1}{\gamma_i-1}$, to construct payments that satisfy Equation (3.35). Since these modifications to any $h_i(v_{-i})$ will not change the set of agents required, as values are scaled by constants, these new $h_i'(v_{-i})$ still only depend on v_{-i}. Thus, there exist SBB payments t' for f'. □

Combining Lemmas 3.2, 3.3 and 3.4, we get the following result:

Theorem 3.7. *If we have a set of SBB and DSIC payments for an affine maximising SCF f defined by $\mathbb{A}', U_0, \gamma$, then there exist SBB payments for a DSIC mechanism with affine maximising SCF f' defined by $\mathbb{A}', \hat{U}_0, \hat{\gamma}$, where $\gamma_i = 0$ if and only if $\hat{\gamma}_i = 0, \forall i \in \mathbf{N}$. That is, the two mechanisms have the same $\mathbb{A}'$ and $\mathbf{N}'$.*

Due to Theorem 3.7, for the purposes of finding SBB, DSIC mechanisms we only need to consider the different possibilities for $\mathbf{N}'$ and $\mathbb{A}'$. It is the sets $\mathbf{N}'$ and $\mathbb{A}'$ that determine whether SBB payments exist for the affine maximising SCF. For simplicity, let all non-zero γ_i be equal to $|\mathbf{N}'|$ (so reciprocals of weights sum to one), which makes payments for a SBB, DSIC mechanism take the form:

$$\sum_{i \in \mathbf{N}} h_i(v_{-i}) = (1 - |\mathbf{N}'|) \max_{a \in \mathbb{A}'} \sum_{i \in \mathbf{N}'} v_i\,(a) \tag{3.36}$$

Payments that satisfy Equation (3.36) for some $\mathbb{A}'$ and $\mathbf{N}'$ can be used to find payments to achieve SBB for any DSIC allocation mechanism with the same $\mathbb{A}'$ and $\mathbf{N}'$. In Section 3.3 we examine the possible ways to achieve strong budget balance with affine maximisers.

The right-hand side of Equation (3.36) depends on the maximisation a^*, which is affected by the values of all agents in $\mathbf{N}'$. We refer to this right hand side as the *surplus*, that is generated in order to achieve DSIC and must be absorbed by rebate functions h to achieve SBB. Each individual $h_i(v_{-i})$, however, cannot depend on all agents' reported values. To achieve strong budget balance with affine maximisers, there are several possibilities, examined in the rest of this section. First, by ignoring a subset of agents (i.e. $\mathbf{N}' \neq \mathbf{N}$), these agents with $\gamma_i = 0$ can absorb any budget surplus since the right hand side of Equation (3.36) is independent of their reported values. Achieving strong budget balance by excluding a subset of agents was examined for general social choice functions by Faltings [2005]. Alternatively, if $\mathbb{A}'$ is *partitionable*, the maximisation to achieve a^* can be performed from a series of maximisations on different subsets of agents (see Subsection 3.2.5.2). Budget balance is achieved by each subset's surplus being absorbed by all other subsets. A partitioning approach was used for the allocation of identical (homogeneous) items by Guo and Conitzer [2008]. If $\mathbf{N}' = \mathbf{N}$ and $\mathbb{A}'$ is non-partitionable, then the mechanism cannot be strongly budget balanced (see Subsection 3.2.5.3). The exception is if $|\mathbb{A}'| \leq 2$, since Theorem 3.6 only applies if there are at least 3 alternatives.

3.2.5.1 Ignore Agents

If $\mathbf{N}' \neq \mathbf{N}$ then there are agents $i \notin \mathbf{N}'$ with $\gamma_i = 0$, whose reported values have no effect on the affine maximisation to choose allocation a^*. Their values may, however, be involved in tie-breaking if the affine maximisation does not choose a unique allocation (i.e. when $m > |\mathbf{N}'| + 1$). These agents' values are independent of the RHS of Equation (3.36) so they can absorb any budget surplus without harming DSIC.

Let $\mathbf{N}_X = \mathbf{N} \backslash \mathbf{N}'$ be the set of "ignored" agents. Strong budget balance can be achieved by (for example):

$$h_x(v_{-x}) = \frac{(1 - |\mathbf{N}'|)}{|\mathbf{N}_X|} \sum_{i \in \mathbf{N}'} v_i\left(a^*\right) \qquad \forall x \in \mathbf{N}_X \tag{3.37}$$

$$h_i(v_{-i}) = 0 \qquad \forall i \in \mathbf{N}' \tag{3.38}$$

This is not a unique solution and the mechanism can have an additional set of payments $\hat{h}_i(v_{-i})$, such that $\sum_{i \in \mathbf{N}} \hat{h}_i(v_{-i}) = 0$, applied to payment functions without harming SBB or DSIC. This will have no effect on the social surplus, but can be used to make payments meet additional requirements, such as individual rationality or fairness.

3.2.5.2 Partition Agents

The alternative to having a^* completely independent of some agents' types (by having $\mathbf{N}' \subsetneq \mathbf{N}$) is to limit $\mathbb{A}'$ such that the maximisation in Equation (3.12) is performed on a series of sub-problems (partitions), where each partition depends only on a *strict* subset of $\mathbf{N}$. This allows the right-hand side of Equation (3.36) to be broken up into parts that depend on only a subset of agents. Instead of budget surpluses being absorbed by ignored agents, each partition's surplus can be absorbed by any agents outside that partition. A partitioning approach was used for the allocation of identical items in [Guo and Conitzer, 2008].

We say that $\mathbb{A}'$ is **partitionable** if the agents and items ($\mathbf{N}$ and $\mathbf{M}$) can be partitioned into $k \geq 2$ disjoint subsets, $\mathbf{N}^1, \ldots \mathbf{N}^k$, $\mathbf{M}^1, \ldots \mathbf{M}^k$, where for any allocation $a \in \mathbb{A}'$, if $a(i) = j$ for $i \in \mathbf{N}^q$, then $j \in \mathbf{M}^q$. That is, agents in partition $\mathbf{N}^q$ can only receive items from $\mathbf{M}^q$. Every agent and item is in exactly one partition. The optimal allocation is then calculated from separate affine maximisations on each partition (with tie-breaking on agents in $\mathbf{N}^q$, if necessary), finding the optimal assignment of agents $\mathbf{N}^q$ to $\mathbf{M}^q$

To achieve SBB, the surplus of each partition must be absorbed using the payment functions from agents in other partitions. If we let a^q be the optimal assignment in partition q, then dividing the right hand side of Equation (3.36), we have:

$$\sum_{i \in \mathbf{N}} h_i(v_{-i}) = (1 - |\mathbf{N}'|) \sum_{i \in \mathbf{N}'} v_i\left(a^*\right) \tag{3.39}$$

$$= \sum_{q=1}^{k} \left((1 - |\mathbf{N}'|) \sum_{i \in \mathbf{N}^q} v_i\left(a^q\right) \right) \tag{3.40}$$

$$= \sum_{q=1}^{k} T^q(v_{\mathbf{N}^q}) \tag{3.41}$$

Where the surplus from each partition:

$$T^q(v_{N^q}) = (1 - |\mathbf{N}'|) \sum_{i \in \mathbf{N}^q} v_i\left(a^q\right) \tag{3.42}$$

So now it is possible to achieve SBB using payments h_i that use T_q from partitions where $i \notin \mathbf{N}^q$ (since T_q is calculated only from values of agents in $\mathbf{N}^q$). One approach is to have an agent absorb a share of the surplus from all partitions to which it does not belong. Let $\bar{\mu}(i)$ be the set of partitions an agent does *not* belong to, i.e. $\bar{\mu}(i) = \{q \in [1, k] : i \notin \mathbf{N}^q\}$. Payments are calculated as:

$$h_i(v_{-i}) = \sum_{q \in \bar{\mu}(i)} \frac{T^q(v_{\mathbf{N}^q})}{n - |\mathbf{N}^q|} \tag{3.43}$$

For each partition q there will be $n - |\mathbf{N}^q|$ agents receiving $\frac{T^q(v_{\mathbf{N}^q})}{n-|\mathbf{N}^q|}$ as part of their payments, and so the sum of all payments will be zero. It is possible to divide a partition's surplus unevenly among other agents, or to use additional payments $\hat{h}_i(v_{-i})$ as with the ignore-agents mechanisms. Provided the additional payments themselves are SBB, the mechanism will remain DSIC and SBB.

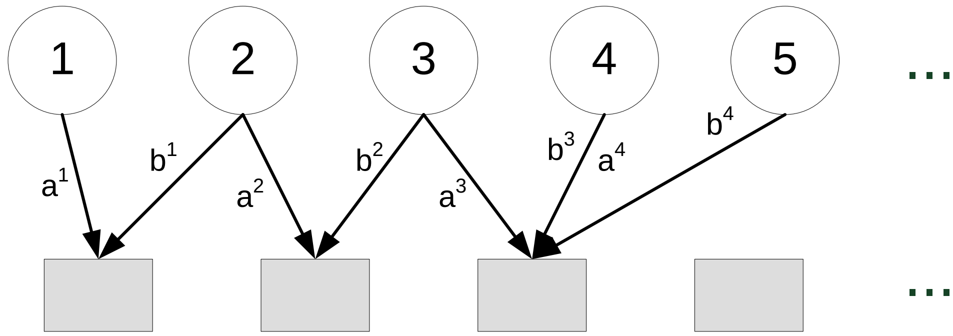


Figure 3.1: *Diagram of non-partitionable agents. Adjacently-labelled agents (circles) can take the same item (rectangles) in different allocations, so must be in the same partition.*

3.2.5.3 Single Partition and No Ignored Agents

If $\mathbf{N}' = \mathbf{N}$, $\mathbb{A}'$ is non-partitionable and $|\mathbb{A}'| > 2$ then the agents' payments cannot be SBB for DSIC mechanisms. Recall that partitionable allocation mechanisms have disjoint subsets of agents and items such that agents can only be allocated items from their own subset. If no such partition exists, then the mechanism is non-partitionable. For example, if every agent can get any item, then every agent and item must belong to the one subset.

If $\mathbb{A}'$ is non-partitionable, then for some ordering of agent labels, $\mathbb{A}'$ contains $n-1$ pairs of allocations $(a^1, b^1), \ldots, (a^{n-1}, b^{n-1})$ such that:

$$a^1(1) = b^1(2) \neq 0 \tag{3.44}$$

$$a^2(2) = b^2(3) \neq 0 \tag{3.45}$$

$$\vdots$$

$$a^{n-1}(n-1) = b^{n-1}(n) \neq 0 \tag{3.46}$$

Note that not all allocations above are necessarily unique. Each pair shows adjacently labelled agents can take the same item, so must necessarily be in the same partition (see

Figure 3.1). For example, Equation (3.45) means there is some item (not the "nothing" item 0) that agent 2 receives in allocation a^2 and agent 3 receives in allocation b^2 (this is illustrated as the second item in Figure 3.1). This is the pair of allocations that places agents 2 and 3 in the same partition. From all $n-1$ pairs of allocations, all agents are in the same partition. If such a chain of pairings for at least one permutation of agent labels did not exist, then the "break" in the chain would signify a partition.

In order to calculate a^*, the affine maximisation calculation must compare the values $v_i(a^i(i))$ to $v_{i+1}(a^i(i))$, for all $i < n$. Thus, the maximisation makes a comparison between all agents' value profiles, and so the maximisation cannot be broken up into sub-problems with sub-agents. Additional payment functions must calculate the allocation that depends on all agents' values, but using a strict subset of these values. Omitting any single agent while attempting to calculate the affine-maximising allocation will be unable to determine the actual allocation, since the omitted agent's value profile will be able to change the affine-maximising allocation. Since $\mathbf{N}' = \mathbf{N}$, the RHS of Equation (3.36) cannot be broken into parts that depends on a subset of $\mathbf{N}$ and thus no valid payment functions will be able to achieve strong budget balance.

3.2.5.4 Non-affine Maximisers

In order to have a DSIC mechanism, the requirement of an affine maximiser from Theorem 3.6 only applies if the SCF satisfies independence and there are more than 2 outcomes. In this work, we will focus on affine-maximising heterogeneous item allocation mechanisms with at least 3 outcomes, as has been done in previous work [Roberts, 1979].

Even with the independence requirement, there are additional DSIC mechanisms that are not affine maximisers but have only two outcomes. A simple example of this is a mechanism, where $\mathbb{A}' = \{a_1, a_2\}$, and a_2 is chosen if and only if all agents prefer a_2 to a_1, with no payments made by any agent. No single agent can benefit by misreporting, and the mechanism is also trivially SBB, yet it does not correspond to any affine maximiser. Since any agent can decide the outcome in situations where all agents unanimously prefer a_2, this would imply a non-zero agent weight in an affine maximiser. But, if any other agent prefers a_1, no agent can change the outcome, regardless of reported type, which would imply a zero weight for these agents. In Subsection 3.3.2, we show that mechanisms restricted to two outcomes do not outperform affine maximising SCFs in the worst case. Finally, restricting $\mathbb{A}'$ to a single possible outcome can be characterised as an extreme case of affine maximisation.

3.3 Strongly Budget Balanced, Unit Demand, Heterogeneous Allocation

Unit demand allocation mechanisms have been studied in the literature; however, the addition of the strong budget balance constraint greatly limits what can be achieved with deterministic mechanisms. In this section we discuss previous work on heterogeneous allocation under a unit demand setting. We also discuss metrics used to assess these mechanisms and worst-case results.

We then use our results from Subsection 3.2.5 to present mechanisms that achieve the optimal worst case performance for affine-maximising mechanisms when the SBB constraint is added. The poor worst-case result in Theorem 3.8 motivates examining a randomised solution to this problem.

3.3.1 Related Work

A good starting point for mechanisms for this problem is the VCG mechanism (see Subsection 3.2.1.1). While not SBB, it is guaranteed to be efficient and WBB for heterogeneous item allocation. However, its weakness due to budget imbalance is quickly exposed by examining its worst-case social surplus ratio. Under VCG payment rules, agents pay their externality or social burden. In the worst case, this payment exactly matches the utility gained by the allocation itself, giving an overall ratio of zero, as illustrated in the following example.

Example 3.1. *There is a single painting to be allocated to one of 3 agents. The agents value this painting at \$300, \$300 and \$200 respectively. The efficient allocation is to give the painting to either agent 1 or 2, and this is decided through some tie-breaking procedure to be agent 1. If either agent 2 or 3 are removed from the allocation, the allocation remains the same, so these two agents pay nothing. The presence of Agent 1, however, prevents agent 2 from receiving \$300 of utility, since agent 2 would receive the painting in agent 1's absence. Thus, agent 1 pays \$300. Calculating the social surplus ratio using Equation (3.3) gives* $\frac{300-300}{300} = 0$.

Previous work has sought to improve this ratio through redistribution of the total VCG payments. This reduces (but does not eliminate) the budget imbalance, while maintaining DSIC and efficiency. Under VCG, agent i receives a payment of $t_i(v) = \sum_{j \neq i} v_j(a^*) - \sum_{j \neq i} v_j(a^*_{-i})$, where a^* is the efficient allocation for all agents, and a^*_{-i} is the efficient allocation for agents excluding agent i. We denote the sum of all payments made to agents in a VCG mechanism, with value profile v as:

$$T^{VCG}(v) = \sum_i \left(\sum_{j \neq i} v_j(a^*) - \sum_{j \neq i} v_j(a^*_{-i}) \right) \tag{3.47}$$

This will be a negative value, since VCG is WBB. We similarly define $T^{VCG}(v_{-i})$ as the sum of VCG payments under a mechanism that only includes agents $\mathbf{N}\backslash\{i\}$.

Bailey [1997] and Cavallo [2006] developed mechanisms to redistribute part of the total VCG payment collected by the mechanism back to the agents. In this item allocation setting, these two mechanisms coincide and is known as the **Bailey/Cavallo mechanism**, which is also similar to a mechanism developed independently by Porter *et al.* [2004]. Under the Bailey/Cavallo mechanism, the agents run a VCG mechanism, then each agent i receives a discount that is an estimate of the average VCG payment. This estimate is made by calculating the total payment had the remaining $n-1$ agents run a VCG mechanism without i. That is, an agent, i, has a payment function:

$$t_i(v) = \underbrace{\sum_{j \neq i} v_j(a^*)}_{\text{Groves}} - \underbrace{\sum_{j \neq i} v_j(a^*_{-i})}_{\text{VCG}} + \underbrace{\frac{1}{n} T^{VCG}(v_{-i})}_{\text{Bailey/Cavallo}} \tag{3.48}$$

The surplus payment produced by the Bailey/Cavallo mechanism is reduced to

$$T^{VCG}(v) - \frac{1}{n} \sum_i T^{VCG}(v_{-i}) \tag{3.49}$$

Considering the agents in Example 3.1, under Bailey/Cavallo, the agents' received payments are instead:

$$t_1 = 0 - 300 + \frac{200}{3} = -233.33 \tag{3.50}$$

$$t_2 = 300 - 300 + \frac{200}{3} = 66.67 \tag{3.51}$$

$$t_3 = 300 - 300 + \frac{300}{3} = 100 \tag{3.52}$$

$$\tag{3.53}$$

Which now gives a much better social surplus ratio of $\frac{300-66.67}{300} = \frac{7}{9}$.

When maximising the social surplus ratio, the Bailey/Cavallo mechanism, as with all VCG redistribution mechanisms is limited to cases where $n > m+1$. In settings where $n \leq m+1$, no redistribution mechanism can provide a positive ratio [Gujar and Yadati, 2011; Guo, 2012], so VCG is the optimal choice. The Bailey/Cavallo mechanism, however, outperforms VCG when $n > m+1$, giving a worst case ratio of [Guo, 2012]

$$\frac{n-m-1}{n} \tag{3.54}$$

The surplus left over from the Bailey/Cavallo mechanism can also be estimated, by calculating VCG payments on $(n-2)$-sized subsets of agents, and redistributed back to further increase the worst-case ratio. This can also be repeated iteratively, considering VCG payments in smaller subsets of agents [Dufton *et al.*, 2012]. Continuing this process gives the `HETERO` [Gujar and Yadati, 2011] mechanism, which Guo [2012] proves to be individually rational, weakly budget balanced and providing the optimal worst case ratio. The payments under the `HETERO` mechanism are more complicated, and take the form:

$$t_i(v) = \sum_{j \neq i} v_j(a^*) - \sum_{j \neq i} v_j(a^*_{-i}) + \alpha_1 T^{VCG}(v_{-i}) + \sum_{k=2}^{k=n-m-1} \alpha_k T^{VCG}_{k-1}(v_{-i}) \tag{3.55}$$

Here $T^{VCG}_k(v)$ is the average sum of VCG payments taken over all subsets of agents in v, with k agents removed. The α terms are values that depend only on n and m, and are defined from 1 to L, where $L = n - m - 1$, as

$$\alpha_i = \frac{(-1)^{(i+1)}(L-i)!m!}{(n-i)!} \frac{(n-m)\binom{n-1}{m-1}}{\sum_{j=m}^{n-1}\binom{n-1}{j}} \sum_{j=0}^{L-i} \left(\binom{i+j-1}{j} \sum_{l=m+i+j}^{n-1} \binom{n-1}{l} \right) \tag{3.56}$$

The ratio given by this mechanism is

$$1 - \frac{\binom{n-1}{m}}{\sum_{j=m}^{n-1}\binom{n-1}{j}} \tag{3.57}$$

This ratio matches the worst-case ratio for the allocation of homogeneous/identical items, developed independently by Guo and Conitzer [2009] and Moulin [2009]. The mechanism achieving this optimal ratio for identical items is called the Worst Case Optimal (WCO) mechanism. Since heterogeneous mechanisms can be used with homogeneous items, they cannot exceed the ratio of WCO, in fact, when `HETERO` is used on identical objects, the payments match that of the WCO mechanism. Also, when $m = n - 2$ the `HETERO` mechanism is equivalent to Bailey/Cavallo.

Redistribution of VCG payments reduces the loss of overall utility while still maintaining an efficient allocation. However, redistribution mechanisms are inapplicable in settings where strong budget balance is a strict constraint. Since simultaneously ensuring efficiency and SBB is typically incompatible with DSIC (see Subsection 3.2.2), an SBB mechanism must sacrifice either efficiency or DSIC.

Removing DSIC altogether means an efficient allocation can always be found without the need for any payments, since we can assume agents' true types are known. DSIC is a strong requirement, and it may be reasonable to assume truthful reports from agents in a non-DSIC mechanism if we assume those agents have computational limitations on calculating a misreport, or have limited information about the true types of the other agents. Work by Parkes *et al.* [2001] examines mechanisms that are SBB but allow agents to make small misreports of their true values. These mechanisms are for a combinatorial exchange, where agents represent multiple buyers and sellers and the mechanism determines the trades of items along with payments made or received by each agent. Kothari *et al.* [2003] later applied this technique to the multi-unit auction, where a seller has a set of identical items, and buyers have non-unit demand.

An efficient allocation maximises the sum of agents' utilities for that allocation. However, given the loss of utility from budget imbalance, an inefficient allocation with budget balanced payments may actually give a greater worst case ratio. Sakurai *et al.* [2009] propose partitioning (similar to Subsection 3.2.5.2) as a technique for achieving SBB at the expense of efficiency. The technique described is for general item allocation problems so is applicable to unit demand allocation. The authors provide empirical results with average case behaviour for social surplus, but no analysis on worst-case performance or the optimal arrangement of agent partitions. Previous work has also covered mechanisms that can "destroy" items in order to achieve higher overall social surplus in the allocation of identical items [Guo and Conitzer, 2008; de Clippel *et al.*, 2009].

3.3.2 Worst-case Optimal SBB Mechanisms

From our result in Theorem 3.6, we examine the space of DSIC, SBB mechanisms for heterogeneous item allocation, focussing on affine maximisers, to find the optimal worst-case performance in this class. We examine the worst-case behaviour in a deterministic setting, focussing specifically on the social surplus ratio, which is simply the level of efficiency of the selected outcome given the hard constraint of strong budget balance, as defined in Equation (3.3). Since either $\mathbf{N}'$ or $\mathbb{A}'$ are restricted, an efficient mechanism is not possible, and this agrees with the Green-Laffont impossibility theorem [Green and Laffont, 1979] (see Subsection 3.2.2).

Theorem 3.8. *The worst-case social surplus ratio for a deterministic, SBB, DSIC mechanism, satisfying independence, for the allocation of heterogeneous items to agents of unit demand, with more than two possible allocations (i.e. $n > 2$ or $m > 2$) is zero.*

Proof. We work through the different mechanisms described in the characterisation of Subsection 3.2.5.

Firstly, consider the Ignore Agent approach. Let x be an ignored agent and let y be some non-ignored agent. Now let item $z = f_y(\mathbf{0}, v_x)$, the item agent y receives when all agents, except x report 0 (v_x will not change the allocation since it is ignored). If $v_x(z) = 1$, and all other agents' values are zero, then $\sum_{i \in \mathbf{N}} v_i(a^*(v)) = 1$ but $\sum_{i \in \mathbf{N}} v_i(f(v)) = 0$, giving a ratio of zero.

Alternatively, we can use partitions to achieve SBB. Each partition has a set of agents and items. Pick an agent $x \in \mathbf{N}^1$ and an item $z \in \mathbf{M}^2$ from different partitions and let the only non-zero valuation be $v_x(z) = 1$. Since the allocation of item z to x is not possible, then $\sum_{i \in \mathbf{N}} v_i(a^*(v)) = 1$ while $\sum_{i \in \mathbf{N}} v_i(f(v)) = 0$, giving a ratio of zero.

A mechanism that combines ignoring with partitioning can be shown to have a zero ratio using either of the two previous examples.

Finally, we can use non-affine maximising SCFs if we limit $\mathbb{A}'$ to two possible allocations. However, if $n > 2$ or $m > 2$, any selection of two allocations will leave at least one agent-item pairing impossible. If this agent's value for this item is the only non-zero value, then the mechanism cannot give an allocation with non-zero total utility, thus the ratio is zero. □

This poor performance leads us to examine randomised solutions in Chapter 4. These randomised solutions are based on two mechanism classes of interest, which we extract from the characterisation in Subsection 3.2.5. These are the `IgnoreAgents` mechanisms and `PartitionAgents` mechanisms. Individual rationality is easily achieved in both these mechanisms by using payments from the VCG mechanism. These additional VCG payments do not affect the total utility of the agents and so they can be added without affecting the ratio, SBB or DSIC.

3.3.2.1 IgnoreAgents

Under an `IgnoreAgents` mechanism, there is a fixed non-empty set of agents $\mathbf{N}_X \subseteq \mathbf{N}$, who are the ignored agents, while the remaining agents are denoted $\mathbf{N}'$. The mechanism proceeds as follows:

1. Allocate agents $\mathbf{N}'$ efficiently: $a' = \arg\max_{a \in \mathbb{A}} \sum_{i \in \mathbf{N}'} v_i(a)$.
2. While there are remaining items and agents in $\mathbf{N}_X$ left unallocated
    - Allocate first unallocated agent in $\mathbf{N}_X$ (according to some predetermined fixed ordering) its most preferred item left unallocated
3. The agents in $\mathbf{N}'$ make regular VCG payments without including the ignored agents in calculations, with the surplus transferred evenly to the ignored agents, $\mathbf{N}_X$. This gives payments

$$t_i(v) = \begin{cases} \sum_{j \in \mathbf{N}' \setminus \{i\}} v_j(a') - \sum_{j \in \mathbf{N}' \setminus \{i\}} v_j(a'_{-i}) & \forall i \in \mathbf{N}' \\ \frac{1}{|\mathbf{N}_X|} \sum_{j \in \mathbf{N}'} \left((1 - |\mathbf{N}'|) v_j(a') + \sum_{k \in \mathbf{N}' \setminus \{j\}} v_k(a'_{-j}) \right) & \forall i \in \mathbf{N}_X \end{cases} \tag{3.58}$$

   where a'_{-i} is the efficient allocation for agents $\mathbf{N}' \setminus \{i\}$.

The payments made by agents in $\mathbf{N}_X$ exactly balance out the payments made by agents in $\mathbf{N}'$, so this mechanism is SBB and DSIC. VCG payments are weakly-budget balanced and individually rational (see Subsection 3.2.1.1), so agents in $\mathbf{N}'$ will be IR, while agents in $\mathbf{N}_X$ will receive a positive payment along with either no item, or an item for which they have non-negative utility. Thus all agents have non-negative utility. However, from

Theorem 3.8 we know the mechanism is inefficient and has a worst-case social surplus ratio of zero.

3.3.2.2 PartitionAgents

In a `PartitionAgents` mechanism, the set of agents, $\mathbf{N}$, and set of items, $\mathbf{M}$, are partitioned into $k \geq 2$ disjoint subsets, $\mathbf{N}^1, \ldots \mathbf{N}^k$ and $\mathbf{M}^1, \ldots \mathbf{M}^k$. The set of possible allocations within a partition q is denoted $\mathbb{A}^q$. The mechanism proceeds as follows:

1. For each partition q, allocate agents $\mathbf{N}^q$ efficiently among items in $\mathbf{M}^q$, to get allocation $a^q = \arg\max_{a \in \mathbb{A}^q} \sum_{i \in \mathbf{N}^q} v_i(a)$.

2. Agents make regular VCG payments within their own partition then collect an even share of the surplus from all other partitions. For agent i in partition q', this gives payments

$$t_i(v) = \sum_{j \in \mathbf{N}^{q'} \setminus \{i\}} v_j(a^{q'}) - \sum_{j \in \mathbf{N}^{q'} \setminus \{i\}} v_j(a^{q'}_{-i}) + \sum_{q \neq q'} \frac{T^q(v_{\mathbf{N}^q})}{n - |\mathbf{N}^q|} \tag{3.59}$$

where

$$T^q(v_{N^q}) = \sum_{i \in \mathbf{N}^q} \left((1 - |\mathbf{N}^q|) v_i\left(a^q\right) - \sum_{j \in \mathbf{N}^q \setminus \{i\}} v_j(a^q_{-i}) \right) \tag{3.60}$$

Agents' payments within a partition ensure DSIC and IR. Additional inter-partition payments made to the agents are non-negative since VCG is WBB, so achieves SBB while maintaining IR. As with `IgnoreAgents`, Theorem 3.8 implies a worst-case social surplus ratio of zero.

3.4 Room Assignment–Rent Division

The new mechanisms we presented for heterogeneous item allocation can be adapted to the room assignment–rent division problem. In this section, we examine previous work on RARD, as well as our new impossibility result for multi-house RARD. As has been shown in previous work [Haake *et al.*, 2002; Su, 1999], no truthful, envy-free mechanism exists for the RARD problem. An envy-free allocation is an efficient allocation [Svensson, 1983; Alkan *et al.*, 1991], and the problem requires the sum of payments to be budget balanced (if $T \neq 0$, each room/agent can be given an initial, constant charge of $\frac{T}{n}$ to bring the budget to zero). As the truthful mechanism cannot guarantee efficiency when ensuring the divisible resource is entirely allocated (this follows from Theorem 3.8 and Green-Laffont [Green and Laffont, 1979]), the mechanism cannot provide an envy-free outcome for all value profiles.

3.4.1 Related Work

While truthful, deterministic solutions do not exist for the room assignment–rent division problem, there have been a number of previous solutions for finding envy-free outcomes, while assuming complete knowledge of agent types. Since there are no game theoretic issues when agents are truthful, these previous solutions deal with other issues. Su [1999] proves the existence of envy-free outcomes for this setting, along with an interactive algorithm based on Sperner's lemma [Sperner, 1928] that uses simple queries to the agents. If the price of each room is a dimension, valid pricing schemes that total to T will form a $(n-1)$-simplex in $\mathbb{R}^n$. This simplex is subdivided into a small mesh of simplices with a barycentric subdivision, where each vertex represents a possible pricing for the rooms. Each vertex is assigned to an agent, and that agent is queried as to its most preferred room at that pricing. The answers to these queries generates a labelling of these vertices. A simplex with a different agent on each vertex (which is guaranteed to exist due to Sperner's lemma) means the agent assigned to each vertex selected a different room, so a point within this simplex gives an envy-free pricing. If neither of the vertices of this simplex give an envy-free pricing, it can be further subdivided until such a pricing is found. This method can be stopped early to give an "approximately envy-free" result, where agents' envy is bounded by the size of the current simplex. Also, the nature of the queries ("Which room do you most prefer at these prices?") is beneficial for agents who cannot easily quantify their room preferences. Su's method, however, can give negative prices to some rooms.

Brams and Kilgour [2001] developed a method called the Gap Procedure. Each agent submits bids for all rooms, such that the sum of bids totals to T with an allocation and pricing chosen to maximise the sum of bids. Prices for each room are then reduced to their second highest bids and so on, until the sum of prices is equal to T. If a reduction of prices would lead to a total less than T, the prices are lowered proportionally to the gap between the current and next-lowest price, such that prices sum to T.

Abdulkadiroğlu *et al.* [2004] developed an envy-free auction method for determining the allocation and prices of rooms with any number of agents that also guarantees non-negative pricing. Rooms are initially priced at $\frac{T}{n}$ and agents select their most preferred rooms. All rooms that are over-demanded (i.e. more than 1 agent selected them) have their prices simultaneously increased while discounting other rooms. This continues until no rooms are over-demanded, at which point all agents are envy-free. This technique requires continuous price updates, and the authors present an equivalent auction that uses discrete price updates. A more general procedure, developed by Haake *et al.* [2002], finds an envy-free allocation without the restrictions that the number of objects must equal the number of agents and each agent must receive exactly one object.

For the room assignment–rent division problem, an envy-free solution relies on truthful preferences of the agents. While no deterministic RARD mechanism exists that is both envy-free and non-manipulable, such mechanisms have been developed for related problems. Sun and Yang [2003] achieved a strategy-proof and envy-free mechanism for a similar allocation problem, but has different restrictions on the allocation of the divisible resource. Instead of dividing a single quantity of some resource, each indivisible item has its own "compensation limit", which avoids the complications associated with strong budget balance. This model and proof was generalised by Andersson and Svensson [2008], and Andersson [2009] for greater flexibility on the indivisible objects, and a proof of coalitional strategy-proofness. However, the use of an item-based compensation limit instead of a single budget of divisible resource that must be entirely allocated mean that these mechanisms are incompatible with the room assignment–rent division model.

3.4.2 Multi-house Room Assignment–Rent Division

Under single-house RARD, an envy-free solution always exists, but this requires that the mechanism knows all agents' types (i.e. agents are honest). The extension to multiple houses, however, can no longer guarantee the existence of such a solution. This is true even if all agents report their type truthfully.

Theorem 3.9. *Under multihouse RARD, in any setting with at least 2 houses, there exist value profiles with no envy-free solution.*

Proof. Since the total rent T_j for house j must be paid for by agents with rooms in j, a house with low cost and high room value will be envied by agents in a house with high cost and low room value. For any multi-house problem, there will be at least 2 houses, with total rent T_1 and T_2, chosen such that $T_1 \geq T_2$. Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be the sets of rooms in the two houses. We now construct a value profile such that

$$v_i(j) = \frac{T_1}{|\mathbf{M}_1|} \qquad \forall i \in \mathbf{N}, j \in \mathbf{M}_1 \tag{3.61}$$

$$v_i(j) = \frac{T_2}{|\mathbf{M}_2|} + 1 \qquad \forall i \in \mathbf{N}, j \in \mathbf{M}_2 \tag{3.62}$$

At least one room in house 1, denoted room x, has a rent of at least $\frac{T_1}{|\mathbf{M}_1|}$, giving a total utility of at most 0 for any agent occupying that room. On the other hand, at least one room in house 2, denoted room y, has a rent at most $\frac{T_2}{|\mathbf{M}_2|}$, giving the total utility of at least 1 for the agent occupying that room. Thus, whichever agent is in room x will envy the agent in room y, regardless of payments. □

Because of this result, in this problem envy-freeness cannot be achieved and randomisation will not help. A randomised mechanism can allow truthfulness, but cannot overcome the non-existence of any envy-free outcome. If this problem is relaxed to allow the rent from each house to be paid by all agents, this simply changes the problem to the single-house case. Due to the result of Theorem 3.9, we only examine single-house mechanisms for the remainder of this work.

3.5 Public Good

Mechanisms for the public good problem deal with a simpler domain. Agents' preferences are represented by a single value, and the outcome space is binary. However, as with item allocation, a truthful mechanism cannot be efficient and strongly budget balanced [Moulin and Shenker, 2001]. This result follows from general impossibility results [Green and Laffont, 1979; Roberts, 1979]. Thus, deterministic mechanisms for this problem are either manipulable, inefficient or have a budget imbalance. Generally, past work on this problem looks at either efficient mechanisms (i.e. Groves) that use redistribution, or, to a lesser extent, inefficient SBB mechanisms.

3.5.1 Related Work

As with the item allocation problem, there has been much previous work on mechanisms for the public project problem that achieve efficiency and DSIC through a VCG mechanism, then reduce the budget imbalance through payment redistribution [Moulin, 1986; Deb and Seo, 1998; Cavallo, 2006; Guo and Conitzer, 2009; Moulin, 2009]. While redistribution mechanisms for heterogeneous item allocation have a tight upper bound on worst case ratio, such a bound is currently unknown for public goods. An upper bound that is conjectured to be tight was found by [Naroditskiy *et al.*, 2012], and this upper bound approaches 1 as n increases. The ratio for VCG, on the other hand, is $\frac{1}{n}$, which approaches 0 with additional agents.

Apt *et al.* [2008] examine the public good problem under a measure they propose of *welfare dominance*. This establishes a partial ordering over mechanisms, where one mechanism welfare dominates another if it produces a smaller sum of payments for all value profiles (strictly smaller for at least one value profile). Under the public good setting we examine, no feasible, efficient and DSIC mechanism welfare dominates the VCG mechanism. Apt and Estévez-Fernández [2009] achieve a better redistribution of payments and welfare dominance of VCG by performing a sequential version of the public good problem, where agents publicly declare their bids one by one. Agents' truthfulness is not in dominant strategies, but rather in a Nash equilibrium. Since there may be multiple Nash equilibria, the sequential bid elicitation ensures all agents end up behaving according to the same Nash equilibrium.

A less-often examined solution to improving social surplus, and one that we require since we enforce SBB, is inefficiency. In this problem, this means the public good may not be acquired, even though there is enough collective benefit to outweigh its cost. Alternatively, an inefficient outcome could mean the public good is acquired where the cost outweighs its benefit, and the agents would have been better off to keep the cost of the project. While such mechanisms have been developed for other problems [Guo and Conitzer, 2008; de Clippel *et al.*, 2009], for the public project problem, recent work by Guo *et al.* [2011] used randomisation and inefficiency to achieve SBB mechanisms for the public good problem. We examine these approaches and similar mechanisms in more detail in Chapter 4.

3.6 Summary

In this chapter we presented deterministic solutions for the social choice problems we described in Chapter 2. We surveyed existing solutions from the literature for these problems and related settings. We also presented the measures that are used to assess and compare these mechanisms. These are measures for overall agent utility, measured by efficiency and social surplus ratio, and the property of envy-freeness. Our first major technical contribution in this chapter is the characterisation of deterministic DSIC mechanisms for heterogeneous item allocation when we add the constraint of independence, which we then extended to cover mechanisms that are both DSIC and SBB. Due to this result, we focus on affine-maximising social choice functions. Using these characterisations we developed mechanisms for this item allocation problem, and found a worst-case upper bound of zero for the social surplus for deterministic affine-maximising mechanisms (see Theorem 3.8). Our characterisation describes mechanisms for the RARD problem too, but without concern for envy-freeness, which past work shows is impossible for deterministic, truthful mechanisms. In addition to surveying past work on the item allocation, RARD and public good problems, we presented an impossibility result for the multi-house variant of RARD, which applies to both deterministic and stochastic mechanisms. Our results from this chapter motivates examining stochastic mechanisms in Chapter 4 to improve worst-case performance.

Chapter 4

Randomised Solutions

The constraints of our problems resulted in poor worst-case performance in deterministic, truthful mechanisms. This motivates us to examine stochastic solutions for our social choice problems. In Chapter 3, we covered deterministic solutions that achieved good performance by relaxing constraints such as strong budget balance or truthfulness. Randomised mechanisms allow these constraints to be kept, while not causing the mechanism to be vulnerable to particular sets of value profiles.

Randomised mechanisms, which we present in this chapter cannot be adequately assessed or compared using the deterministic measures presented in Section 3.1. Before presenting randomised mechanisms, we discuss measures appropriate for assessing the quality and optimality of these solutions, based on deterministic measures. These measures also directly correspond with existing deterministic measures when applied to deterministic mechanisms, which facilitates the comparison between existing and new mechanisms. We require this property of the new measures, since otherwise the new measures could be constructed to deliberately punish existing mechanisms while benefiting new mechanisms.

Using these new measures, we present and assess randomised solutions to our different problems to demonstrate the potential gain of randomisation. For heterogeneous item allocation, we provide a tight upper bound on efficiency with a randomised, strongly budget balanced mechanism. Thus, to exceed this level of performance, a mechanism designer must be able to either relax some problem constraints, or gather information about the distribution of agents' types. We also provide similar bounds on envy-freeness in the room assignment–rent division problem. Due to poor performance in stochastic DSIC mechanisms for RARD, we also examine mechanisms that are truthful in expectation. It is only knowledge of the actual random choice that can break truthfulness guarantees in a truthful in expectation mechanism. If we assume this knowledge is not available to agents prior to their reporting their types, then it is reasonable to relax our requirements to this weaker notion of truthfulness. Finally we look at the single-parameter domain of the public good problem, where the aim is again to maximise the agents' total utility.

In this section, we evaluate stochastic mechanisms $\Delta\hat{\mathcal{M}}$ which are distributions over a set of social choice functions and payment functions $\hat{\mathcal{M}}$. This is defined formally in Definition 2.2 of Chapter 2.

4.1 Measures

When examining a stochastic mechanism, which can produce many outcomes from a single value profile, it is beneficial to consider measures designed for stochastic mechanisms. To equip deterministic properties for use with stochastic mechanisms, we consider the probability of an outcome, and the expected value of the quality of the outcome. If there is no prior information about the distribution of agents' types, it is possible that a mechanism repeatedly receives "bad" value profiles. To provide guarantees on the performance of the mechanism, we look at the worst-case behaviour, which gives the lowest performance out of any possible value profile.

In order to assess the allocative efficiency of our proposed stochastic mechanisms, we present two measures that are analogous to the deterministic properties of efficiency and social surplus (see Section 3.1 for definitions of these deterministic properties). Under strong budget balance, an efficient outcome is the optimal solution, but when that isn't achieved, the social surplus ratio shows how close we are to the optimal outcome. Where budget imbalance is allowed, the ratio captures the loss of utility from both inefficient allocation and budget surplus.

Definition 4.1 (Probability of efficiency)**.** *The **probability of efficiency**, denoted p_{EF}, is the probability that a mechanism will produce an (allocatively) efficient outcome, for any possible value profile.*

Formally, given a stochastic mechanism $\Delta\hat{\mathcal{M}}$, which chooses a deterministic mechanism $\mathcal{M}^k = (f^k, t^k) \in \hat{\mathcal{M}}$ with probability p_k, the p_{EF} for a value profile v is:

$$p_{EF}(v) = \sum_{k \in \hat{\mathcal{M}}} p_k \mathbf{1}\left(v\left(f^k(v)\right) = v\left(a^*(v)\right)\right) \tag{4.1}$$

Where the function $\mathbf{1}\left(v\left(f^k(v)\right) = v\left(a^*(v)\right)\right)$ is a binary function evaluating to 1 if the value of the allocation produced by f^k is the same as the value of the efficient allocation a^*, and evaluates to 0 otherwise.

We typically consider the worst-case p_{EF}, which is the minimum for all possible value profiles. Under deterministic mechanisms, the property of allocative efficiency was boolean, in that a mechanism always produced an efficient outcome or there existed some value profile that would be allocated inefficiently. This measure is a natural extension of the deterministic measure and worst-case p_{EF} retains the information provided by measuring for efficiency. An efficient, deterministic mechanism will have a worst-case p_{EF} of one, or zero otherwise. A randomised mechanism will have $p_{EF} = 1$ if and only if it is an efficient mechanism (in the deterministic sense).

If efficiency is desirable but not an absolute requirement, then this measure allows a mechanism that is almost-always efficient in the worst case to be distinguished from one that is (almost) never efficient, whereas otherwise both mechanisms would have been labelled as inefficient.

However, this measure only considers the probability of the very best outcomes, where the sum of agents' utilities are maximised, and all other outcomes are counted as equally valueless. Where we are interested in maximising the quality of the non-optimal outcomes as well, it is appropriate to consider the expected level of efficiency for a set of agents, and this prompts an extension to the social surplus ratio.

Definition 4.2 (Worst-case expected ratio)**.** *The **worst-case expected ratio**, denoted r_{WCE}, is the probability-weighted sum of the social surplus ratio (Equation (3.3)) for each outcome of the mechanism, in the worst-case.*

Given a stochastic mechanism $\Delta\hat{\mathcal{M}}$, the worst-case expected ratio is

$$r_{WCE} = \min_{v \in V} \sum_{k \in \hat{\mathcal{M}}} p_k \left(\frac{\sum_{i \in \mathbf{N}} v_i(f^k(v)) + \sum_{i \in \mathbf{N}} t_i^k(v)}{\sum_{i \in \mathbf{N}} v_i(a^*(v))} \right) \tag{4.2}$$

Since this is a probability-weighted sum, the worst-case expected ratio for a deterministic mechanism will be calculated over a single outcome and will simply be the social surplus ratio of Equation (3.3).

These two efficiency measures look at worst-case behaviour, so for any set of agents, a mechanism is guaranteed to produce an efficient outcome with probability *at least* the worst-case p_{EF}, and have an expected social surplus ratio *at least* r_{WCE}. An alternative worst-case measure is to consider the performance of the worst possible outcome for any combination of random choice and value profile. However, this stronger measure obscures the benefits of randomisation, since the analysis forces the mechanism to *deterministically* make the worst random choice. This reduces all randomised mechanisms to deterministic mechanisms. This also reduces both randomised measures back to their deterministic counterparts and worst-case results from Chapter 3, such as Theorem 3.8, would still apply.

We similarly extend envy-freeness to apply more appropriately to stochastic mechanisms. Envy-free mechanisms always produce an outcome such that no agent is envious. As with efficiency, we can create measures that examine each of the possible outcomes separately, along with their probabilities of occurring to produce a measurement. This measures agent envy after the randomisation process. In a randomised mechanism, agent envy can also be measured before the randomisation process. A simple extension of measuring envy to a randomised mechanism is to compare each agent's lottery[1] of allocations, prior to the mechanism performing its random selection.

Definition 4.3 (*Ex ante* envy-free)**.** *A mechanism satisfying* **ex ante *envy-freeness****, has no agent strictly preferring another agent's lottery over final outcomes for any $v \in V$. That is, for an* ex ante *envy-free mechanism $\Delta\hat{\mathcal{M}}$, $\forall i, j \in \mathbf{N}$, $\forall v \in V$:*

$$\sum_{k \in \hat{\mathcal{M}}} p_k[v_i(a^k(i)) + t_i^k(v)] \geq \sum_{k \in \hat{\mathcal{M}}} p_k[v_i(a^k(j)) + t_j^k(v)] \tag{4.3}$$

where allocation $a^k = f^k(v)$.

A mechanism that is deterministically envy-free is also *ex ante* envy-free, since lotteries

[1] A **lottery** is a discrete distribution over possible states or outcomes. In this setting, an agent's outcome is an item and payment.

are over a single outcome for each agent. For the RARD problem, however, *ex ante* envy-freeness is trivial to achieve in truthful stochastic mechanisms.

Example 4.1 (Simple *ex ante* envy-free mechanism)**.** *Consider a mechanism that simply randomises over all possible allocations with equal probability and gives each agent a constant share of $\frac{T}{n}$ from the divisible resource. For item allocation, we can set all payments to zero to achieve SBB. This gives each agent an identical lottery, so agents have the same utility compared to any other agent's lottery resulting in* ex ante *envy-freeness. Since agents cannot influence the outcome, it is trivially truthful.*

The mechanism in Example 4.1 will generally provide poor final outcomes, however, with most or all agents envious in all outcomes. A randomised mechanism satisfying *ex ante* envy-freeness has no guarantee of satisfying envy-freeness in the deterministic sense.

Because *ex ante* envy-freeness is a weak requirement with few guarantees about final outcomes, we propose looking at measures of envy-freeness after the mechanism has performed the random selection. One natural measure is to look at which of these final outcomes are envy-free in the deterministic sense, and the minimum probability that the mechanism produces such an outcome.

Definition 4.4 (Probability of envy-freeness)**.** *The **probability of envy-freeness**, p_{NF}, is the (worst-case) probability that a mechanism will produce an envy-free outcome, for any set of agents. The worst-case p_{NF} is defined as*

$$p_{NF} = \min_{v \in V} \sum_{k \in \hat{\mathcal{M}}} p_k \text{EnvyFree}(v, f^k, t^k) \tag{4.4}$$

where $\text{EnvyFree}(v, f^k, t^k)$ *is 1 if* (f^k, t^k) *gives an envy-free outcome for* v*, that is, given* $a^k = f^k(v)$

$$v_i(a^k(i)) + t_i^k(v) \geq v_i(a^k(j)) + t_j^k(v) \ , \forall i, j \in \mathbf{N} \tag{4.5}$$

Otherwise EnvyFree(·) *is zero.*

The *ex ante* envy-free mechanism in Example 4.1 has a p_{NF} of zero. That is, for some sets of agents, it will never produce an envy-free outcome.

Example 4.2. *Consider two people splitting the $10 cost for a package that includes both a large steak and a small steak. They both prefer the larger steak, and the mechanism of Example 4.1 will price both steaks at $5 and assign them randomly. Whoever pays $5 for the small steak is envious of the other person, who paid the same price for the large steak.*

As with p_{EF}, there is a strong correspondence between p_{NF} and deterministic envy-freeness. Any deterministic mechanism will have a $p_{NF} = 1$ if it is envy-free and zero otherwise, while a stochastic mechanism will have a $p_{NF} = 1$ if and only if it is an envy-free mechanism (in the deterministic sense).

A limitation of p_{NF} is that it only considers outcomes where all agents are envy-free, with all other outcomes counted as equally valueless. To distinguish between different, non-envy-free outcomes, we examine the *level of envy* in each of the possible outcomes. We measure this as the number of agents who are envious of at least one other agent. This is one of multiple ways of extending the concept of envy-freeness beyond a binary property. While we consider agents to be either envious or not, one could also consider the number of other agents each agent envies (degree of envy), or the amount by which an agent is envious of another agent (magnitude of envy) [Chevaleyre *et al.*, 2009].

Definition 4.5 (Expected envy-freeness)**.** *An envy-free agent is one who does not value another agent's bundle higher than its own in a particular allocation. The* ***expected envy-freeness****, E_{NF}, is the expected fraction of envy-free agents. This is the probability-weighted sum of the fraction of envy-free agents in each outcome of the mechanism for a particular input.*

The expected envy-freeness for a particular value profile $v \in V$ in a stochastic mechanism $\Delta\hat{\mathcal{M}}$ is

$$E_{NF}(v) = \sum_{k \in \hat{\mathcal{M}}} p_k \frac{|\mathcal{E}(v, f^k, t^k)|}{|N|} \tag{4.6}$$

where $\mathcal{E}(v, f^k, t^k)$ is the set of envy-free agents under the outcome produced by (f^k, t^k). That is, given $a^k = f^k(v)$

$$\mathcal{E}(v, f^k, t^k) = \{i \in \mathbf{N} : v_i(a^k(i)) + t_i^k(v) \geq v_i(a^k(j)) + t_j^k(v) \ , \forall j \in \mathbf{N}\} \tag{4.7}$$

In Example 4.2, the basic mechanism gives $E_{NF} = \frac{1}{2}$ for these agents, since the person with the large steak is envy-free but the person with the small steak is envious.

An envy-free mechanism will always have all agents envy-free, so this gives $E_{NF} = 1$, while a mechanism that results in all agents being envious has $E_{NF} = 0$. If this measure is applied to a deterministic mechanism, it will give the worst-case fraction of envy-free agents.

4.2 Related work

Previous work on social choice problems has made use of randomisation for both computational and game theoretic benefits. Before we cover mechanisms for our specific social choice problems, we examine some related work in other settings, and general social choice settings.

Randomisation has been used in voting problems to overcome impossibility results regarding manipulation by voters. Voting rules, although widely used for a range of very important decisions, fall victim to two important theorems: Arrow's impossibility theorem [Arrow, 1950] and the related Gibbard-Satterthwaite theorem [Gibbard, 1973; Satterthwaite, 1975]. Arrow's impossibility theorem states that a social welfare function with three or more alternatives that is both Pareto efficient and independent of irrelevant alternatives must be dictatorial. Instead of looking at systems that give a complete preference ordering, the Gibbard-Satterthwaite theorem shows that systems that find a single winner that are both strategy-proof and allow any alternative to be elected, must be dictatorial. If a voting method allows tactical voting, then a voter may achieve a more desirable outcome by stating preferences other than his or her true preferences. A dictatorial voting system contains a single voter (the dictator) who can choose the winner.

The Gibbard-Satterthwaithe theorem means that reasonable voting rules are subject to manipulation. Gibbard [1977] presented a family of voting rules that used randomisation to create strategy-proof voting rules. Any strategy-proof randomised voting scheme will be a randomisation over procedures that are either dictatorial or duple (only allow two candidates to win). A simple example of a voting method that uses randomisation to be free from strategic voting, is non-dictatorial and does not prevent any candidate from winning is the random ballot. In a random ballot election, one ballot is randomly selected and the candidate ranked first on that ballot is the winner. If ballots are drawn with uniform probability, the probability of a candidate winning is equal to the fraction of votes cast for that candidate. An alternative is to randomly select two candidates, and select the most preferred of these two candidates. In either of these cases, no agent has incentive to misreport, and neither is duple or dictatorial. However, after the random choice is made, a strategy proof voting scheme will have one of these undesirable properties.

Recently, Procaccia [2010], motivated by results of Conitzer and Sandholm [2006], used randomisation and approximation to create strategy-proof, randomised voting rules. These voting rules were randomised versions of the score-based voting rules of Plurality, Borda, Copeland and Maximin. An issue discussed in this work is how to assess these randomised procedures when existing desiderata are inadequate. The approximation ratio of these randomised voting rules is defined in terms of scores given to candidates compared to their deterministic, non-SP versions. The results of Procaccia [2010] suggest the answer to the titular question "Can randomization circumvent Gibbard-Satterwaithe?" may be yes, but it is as yet unquantified just how well it can circumvent.

Other recent work on randomisation in social choice has involved the problem of "self-selection", which has n agents that must select the $k < n$ best agents from within the group. While each agent approves of a subset of other agents, the agent only cares about whether he or she is selected. Alon *et al.* [2011] first provide an impossibility result, showing that this problem has no deterministic, truthful mechanism. An agent voting to approve another agent can only lower its own chance of being selected. The authors then develop a *randomised* mechanism that is strategy-proof and approximately optimal. This works by partitioning agents such that each partition has a fixed number of places in the final k selected agents. Each agent can then only declare its approval for agents outside its own partition, thus its own preferences cannot affect its chance of being selected. As we show later in this section, random partitioning of agents is a useful technique for achieving strategy proof and strongly budget balanced mechanisms.

Faltings [2005] presented a widely applicable approach for randomised mechanisms in strongly budget balanced, social choice settings. The procedure proposed selects a subset of agents whose preferences are to be ignored while determining the optimal outcome. These agents can then absorb any surplus payments collected by the remaining agents, who run a VCG mechanism to determine the outcome. This sacrifices Pareto efficiency in order to achieve strong budget balance. Faltings performs an average-case empirical analysis to show that on randomly generated social choice problems, the total utility of the agents increases when using a budget balanced, inefficient mechanism over the efficient VCG mechanism, which can produce a very large surplus. This motivates relaxing the efficiency requirement when seeking to maximise the total utility of agents. We use this approach of randomly ignoring agents to achieve strong budget balance but examine how it performs compared to optimal worst-case behaviour.

Bogomolnaia and Moulin [2000; 2001], and later Kojima [2009] examined a randomised mechanism for a problem similar to our unit demand item allocation problem. The goal is envy-free allocation in a setting without monetary transfers (which is trivially SBB). However, these settings use ordinal, rather than cardinal, preferences for items. One setting even further restricts agents to the same ordinal ranking, where individual preferences are distinguished by a private "acceptance threshold" [Bogomolnaia and Moulin, 2000], below which an agent would rather have no item. These randomised mechanisms, which are random serial dictatorships, were shown to be efficient, envy-free and strategy proof. Although the form of preferences are quite limiting and can't be used directly for the problems we examine, the papers discuss methods of evaluating randomised allocation procedures particularly with ordinal preferences.

Randomised auctions have been discussed for commodities such as **digital goods**, which have an unlimited supply with no marginal cost. These auctions are desired to be truthful, competitive and envy-free. Goldberg and Hartline [2003] showed that for these goods, no deterministic auction exists that is truthful, envy-free and constant-competitive[2]. The authors also describe auctions that become constant-competitive at the expense of either some probability of non-envy-freeness or non-truthfulness. Mehta and Vazirani [2004] provide an equivalence result for randomised auctions of digital goods. They show that

[2]Constant-competitive auctions always produce profit (to the mechanism/auctioneer) that is within some constant factor of the optimal profit for a single item auction.

a truthful in expectation randomised auction can be expressed as a mechanism that randomly selects from a set of truthful, deterministic auctions. This equivalence result is not applicable in general allocation problems, since it relies on independence between individual agents' possible allocations. That is, if one agent is allocated a digital good from an unlimited supply, it will not affect whether or not any other particular agent can be allocated.

The other main use of randomisation in computational social choice is to allay computational or communication inefficiency. Dobzinski *et al.* [2006], for example, developed a randomised *combinatorial* auction that is dominant strategy incentive compatible. Their work looks at combinatorial auctions that are truthful and have low computational complexity and whose result is bounded by some approximation ratio. We cover communication complexity in Section 5.3 and computational complexity in Section 5.4.

4.3 Strongly Budget Balanced, Unit Demand, Heterogeneous Allocation

Other than the general techniques described in Section 4.2, there has not been previous work on randomised mechanisms for allocating heterogeneous items under our constraints. Using the results from Subsection 3.2.5, we have the space of affine-maximising DSIC and SBB mechanisms for this problem. We use these deterministic mechanisms as the basis for stochastic mechanisms that are optimal, under our constraints, according to p_{EF} or r_{WCE}.

With the aim of maximising the worst case performance on both p_{EF} and r_{WCE}, we begin by presenting two mechanisms along with bounds on their worst-case performance. The first, presented in Subsection 4.3.1, is a randomised version of the `IgnoreAgents` mechanism of Subsection 3.3.2.1, named `RandomAuctioneer`, while the second mechanism, in Subsection 4.3.2 randomises over mechanisms that partition agents from Subsection 3.3.2.2. In both these cases, these mechanisms randomise over extreme versions of the deterministic mechanisms. Only a single agent is ignored or, when agents are partitioned, there are only two partitions, one of which contains a single agent.

In Subsection 4.3.4, we show the performance achieved by these mechanisms to be a tight upper bound for the problem of heterogeneous item allocation with an affine-maximising SCF. Ignoring additional agents, or creating additional or more balanced partitions does not improve worst-case performance on p_{EF} or r_{WCE}.

4.3.1 Random Auctioneer (Ignore One Agent)

We use a mechanism that randomly ignores a single agent, while maximising efficiency among non-ignored agents. First, a mechanism ignoring more than one agent can do no better in terms of efficiency than ignoring only a single agent, since there is strictly less information for the affine maximisation to optimise over. Further, after ignoring an agent, since we are able to maximise efficiency among the non-ignored agents and our objective is to maximise p_{EF} or r_{WCE}, it does not make sense to allocate with suboptimal efficiency. Since the mechanism is always SBB, Equation (4.2) reduces to the ratio of the expected total value of the allocation compared to the total value of the efficient allocation. We use VCG payments for non-ignored agents so that the mechanism is IR, since this has no effect on DSIC, SBB or efficiency. This ignored agent can be considered the auctioneer, since it collects any surplus payments and can choose from any leftover item, thus we name this mechanism `RandomAuctioneer` .

In cases where there are fewer agents than items ($n < m$), any allocation of the surplus items to the auctioneer agent will be tied according to the affine maximisation. We can solve this using a dictatorial tie-breaking procedure, where the auctioneer agent chooses its most preferred item from those left unallocated. Any other ties, such as those when $n \geq m$ will have the same total utility regardless of which is chosen, so we simply break these ties randomly. These tie breaking procedures are trivially SBB and DSIC.

The `RandomAuctioneer` mechanism runs as follows:

1. Randomly choose agent $x \in \mathbf{N}$ to ignore, with uniform probability.

2. Find all allocations that allocate agents $\mathbf{N}' = \mathbf{N} \backslash \{x\}$ efficiently:

$$A' = \{a \in A : \sum_{i \in \mathbf{N}'} v_i(a) \geq \sum_{i \in \mathbf{N}'} v_i(a'), \forall a' \in \mathbb{A}\}$$

3. Agent x selects its most preferred allocation a' from A', with ties broken arbitrarily. If $n < m$, this means x picks its more preferred item from the surplus items.

4. Agents receive payments as described in Equation (3.58) from Subsection 3.3.2.1, where payments depend on the random choice of x. Since a single agent is ignored, we have $\mathbf{N}_X = \{x\}$ so payments are of the form:

$$t_i(v) = \sum_{j \in \mathbf{N}' \backslash \{i\}} v_j(a') - \sum_{j \in \mathbf{N}' \backslash \{i\}} v_j(a'_{-i}) \qquad \forall i \notin \mathbf{N}_X \tag{4.8}$$

$$t_x(v) = \sum_{j \in \mathbf{N}'} \left((2-n) v_j\left(a'\right) + \sum_{k \in \mathbf{N}' \backslash \{j\}} v_k(a'_{-j}) \right) \tag{4.9}$$

To demonstrate this mechanism we apply it to the situation described in Example 2.1, which we repeat here for readability.

Example 4.3. *Four colleagues, Alex, Lou, Sam and Vic, are deciding who should borrow which company car over the upcoming long weekend. Available to them are a luxury sedan, a small hatchback, two vans and an SUV. They all have different requirements and preferences for the vehicles. Alex, for example will be buying large items for a renovation, so needs the space of the SUV or van, while Lou is going on a family trip, so would like the space of the SUV or comfort of the sedan. They each work out their preferences, as summarised in Table 4.1, and then must decide who gets which vehicle.*

	Sedan	Hatch	Van 1	Van 2	SUV
Alex	\$30	\$10	\$60	\$50	\$130
Lou	\$90	\$20	\$40	\$35	\$100
Sam	\$90	\$50	\$10	\$10	\$40
Vic	\$85	\$40	\$15	\$10	\$70

Table 4.1: *Item allocation example. Value agents receive for hiring each of the five vehicles.*

They use the `RandomAuctioneer` *mechanism with VCG payments as described above. In the first step, Alex is randomly selected as the ignored participant/auctioneer, and the efficient allocation among the remaining people gives Lou the SUV, Sam the hatchback and Vic the Sedan (this is not the overall efficient allocation). Alex is then left to pick the first van. Payments received are calculated as:*

$$t_{Lou} = (50 + 85) - (90 + 70) = -25 \tag{4.10}$$

$$t_{Sam} = (100 + 85) - (100 + 85) = 0 \tag{4.11}$$

$$t_{Vic} = (100 + 50) - (100 + 90) = -40 \tag{4.12}$$

$$t_{Alex} = 25 + 40 = 65 \tag{4.13}$$

This gives a total agent utility of \$295, while the efficient allocation, where Alex and Lou swap vehicles has a total utility of \$305, giving a social surplus ratio of 0.967.

To get a lower bound on what is possible, we assess the worst case behaviour of this mechanism according to p_{EF} and r_{WCE}, defined in Section 4.1.

4.3.1.1 Probability of an Efficient Allocation

In the worst-case there is only a single efficient allocation, a^*. An example of such a value profile is

$$v = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix} \tag{4.14}$$

where $v_i(j) = 0, \forall i \neq j$, and 1 otherwise. The single efficient allocation assigns agent i to item i.

If $n > m$, then in the efficient allocation there are $(n-m)$ agents who are left unallocated in a^* (those agents with a label $i > m$). If any of these agents are excluded when calculating the efficient allocation, the mechanism will still produce an efficient allocation. Regardless of value profile, there will always be $(n - m)$ such agents. If any of the m agents who are allocated items in a^* are excluded from consideration, then they will not receive an item (as there are no surplus items) and the outcome will not be efficient. Each individual agent is excluded with probability $\frac{1}{n}$, so the worst-case probability of efficiency $p_{EF} = \frac{(n-m)}{n}$ when $n > m$.

If $n \leq m$, there will be $(m-n+1)$ surplus items for the excluded agent to choose from. For every value profile, there will always be one agent whose exclusion will not prevent the efficient allocation of the remaining agents. If removing each agent caused the allocation of the remaining agents to improve, then this would mean every agent is "pivotal", and its participation causes a net-loss to the utility of all other agents.

In the item allocation setting, each time an agent is removed, another agent takes its item, and so each agent holds an item that at least one other agent prefers to its own. Suppose when agent i is removed, agent j takes i's item. When agent j is removed, agent k takes j's item. We can continue this, and eventually we will find a cycle (since no agent retains its item), where, for example, agent k takes agent i's item. This cycle indicates that agents will exchange items to gain an overall increase in utility, meaning the original, efficient allocation could be improved, and thus not actually efficient.

In the worst case, there will only be one agent whose removal does not change the allocation. This occurs, for example, when all agents have the same preference ordering over all items, as in the value profile

$$v = \begin{pmatrix} 7 & 5 & 3 & 1 & 0 \\ 6 & 5 & 3 & 1 & 0 \\ 5 & 4 & 3 & 1 & 0 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix} \tag{4.15}$$

In this value profile, ignoring agent (row) 1, 2 or 3 will result in the non-ignored agents each taking a higher-valued item, while the allocation remains the same if agent 4 is ignored. There is a $\frac{1}{n}$ probability the mechanism removes this single agent. The ignored agent will then choose the item it values most from the surplus $(m - n + 1)$ items resulting in an efficient allocation. This gives a worst-case probability of an efficient allocation of $p_{EF} = \frac{1}{n}$.

To summarise, the worst-case probability of an efficient allocation of heterogeneous items for the Ignore Agent mechanism (and thus, a lower bound) is:

$$p_{EF} = \begin{cases} \dfrac{1}{n} & \text{, if } n \leq m \\ \dfrac{(n-m)}{n} & \text{, if } n > m \end{cases} \tag{4.16}$$

4.3.1.2 Worst-case Expected Ratio

We can scale agent values so that the value of the efficient allocation is 1, thus the worst-case expected ratio is equivalent to the worst-case expected value of the allocation for scaled value profiles.

For $n > m$, there is a potential loss of efficiency whenever one of the m agents given an item in the efficient allocation is ignored and so does not receive any item in the allocation. In the worst case, the agent that replaces these excluded efficient-allocation agents has no utility for the item it receives, while the remaining $m - 1$ agents must do *at least* as well (in total utility) as they did in the efficient allocation. But in the worst-case they will not get any increase in utility. The worst-case expected loss in efficiency is thus the value each agent has in the efficient allocation, weighted by the probability it is excluded. Only the exclusion of the m efficiently-allocated agents will cause a loss of efficiency. As all agents are ignored with equal probability ($\frac{1}{n}$), and the m allocated agents have a combined utility of 1, then the expected loss of efficiency is at most $\sum_{i \in \mathbf{N}} \frac{1}{n} v_i(a^*) = \frac{1}{n}$. This gives an overall expected ratio of $r_{WCE} = \frac{n-1}{n}$ in the worst-case.

For $n \leq m$, there is a potential loss of efficiency when ignoring any single agent, except for one. If an ignored agent's favoured item is not one of the surplus items, then the other agents must have changed allocation (compared to the efficient allocation) for some (possibly infinitesimal) increase in overall utility, or due to "tie breaking". Consider a value profile such as

$$v = \begin{pmatrix} 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.17}$$

Any excluded agent gets no utility for any of the surplus items. The loss of efficiency is the value each agent has in the efficient allocation weighted by the probability it is excluded. At least one agent will still be able to get its efficient item, but in the worst case, this agent will have zero utility for all items. So the expected loss of efficiency is still $\frac{1}{n}$.

So, in all cases, the worst-case expected ratio for this mechanism is

$$r_{WCE} = \frac{n-1}{n} \tag{4.18}$$

4.3.2 Randomly Partition Agents

Next we examine a mechanism that randomly partitions agents. Our lower-bound mechanism for the Random Partition approach consists of dividing the agents into two partitions, one with a single "solo" agent. If $n \leq m$, a single item is randomly assigned to the solo agent's partition, otherwise it is left unallocated. This solo agent is effectively ignored, since its reported type has no effect on the allocation. Note that this is distinct from ignoring one agent only when $n \leq m$, since the solo agent has a fixed item assigned to it, while an ignored agent is able to choose an item from the surplus items. When we consider the upper bound for this approach in Subsection 4.3.4, we consider more general partitioning, but show that this partitioning achieves the best efficiency within the class of partitioning mechanisms.

When $n \leq m$, the mechanism works as follows:

1. Randomly choose agent $x \in \mathbf{N}$ to place in the solo partition, along with item y. Agent x is assigned item y.

2. Allocate agents $\mathbf{N}' = \mathbf{N} \backslash \{x\}$ efficiently: $a' = \arg\max_{a \in \mathbb{A}'} \sum_{i \in \mathbf{N} \backslash \{x\}} v_i(a)$, where $\mathbb{A}' = \{a \in \mathbb{A} : a(x) = y\}$.

3. Agents receive payments as described in Subsection 3.3.2.2. Since the solo agent is effectively ignored, payment functions match those in Equations (4.8) and (4.9).

When $n > m$, the mechanism is identical to `RandomAuctioneer`.

4.3.2.1 Probability of an Efficient Allocation

When $n \leq m$, the solo agent receives a randomly selected item. In the worst case, each agent favours exactly one item, and no two agents favour the same item, as in Equation (4.14). For any choice of solo agent, there is a $\frac{1}{m}$ probability it receives its desired item. If the solo agent receives its item from the efficient allocation, since the remaining items are allocated efficiently, the overall allocation is efficient. Otherwise, the solo agent has the wrong item, so overall the allocation is inefficient. Thus, $p_{EF} = \frac{1}{m}$ in the worst-case.

In the case of $n > m$, this mechanism is the same as the `RandomAuctioneer` mechanism, as the solo agent is always left unallocated. Thus, the probability of efficiency remains the same, at $p_{EF} = \frac{(n-m)}{n}$. That is, in the worst case, the mechanism will produce an inefficient allocation whenever any of the m agents allocated in the efficient allocation are placed in the solo partition.

To summarise, the worst-case probability of an efficient allocation for heterogeneous items for the Randomly Partition Agents mechanism (and thus, a lower bound) is:

$$p_{EF} = \begin{cases} \dfrac{1}{m} & \text{, if } n \leq m \\ \dfrac{(n-m)}{n} & \text{, if } n > m \end{cases} \tag{4.19}$$

4.3.2.2 Worst-case Expected Ratio

Now we examine the worst-case ratio for this partitioning approach. In settings where $n \leq m$, a loss of efficiency will only occur when the solo agent is partitioned with the "wrong" item. In the worst case, each agent only has non-zero utility for one item, so unless the solo agent receives its favoured item, there is a loss of utility equal to the value of this item it receives in the efficient allocation. Also, the item it receives instead could be one of the $(n-1)$ items that are given to one of the remaining agents, in the efficient allocation. For a particular assignment of agents to partitions, the solo agent, x's expected utility from the allocation is $\frac{1}{m}v_x(a^*)$, since it gets nothing from the other $(m-1)$ items. A non-solo agent, i, has expected utility of $\frac{m-1}{m}v_i(a^*)$. Again, we normalise agents' utilities such the value of the efficient allocation $\sum_{i \in \mathbf{N}} v_i(a^*) = 1$. Thus, for a particular assignment of agents to partitions, the expected value of the allocation is:

$$\frac{1}{m}v_x(a^*) + \sum_{i \neq x} \frac{m-1}{m}v_i(a^*) = \frac{2-m}{m}v_x(a^*) + \sum_{i \in \mathbf{N}} \frac{m-1}{m}v_i(a^*) \tag{4.20}$$

$$= \frac{2-m}{m}v_x(a^*) + \frac{m-1}{m} \tag{4.21}$$

Recall that agents have non-negative values, and the values are normalised, so the above expected value in Equation (4.21) is between $\frac{1}{m}$ and $\frac{m-1}{m}$. Since each agent is the solo agent $x \in \mathbf{N}$ with equal probability of $\frac{1}{n}$, the overall expected value of the allocation, and thus the expected ratio in the worst-case is:

$$r_{WCE} = \sum_{x \in \mathbf{N}} \frac{1}{n} \left(\frac{2-m}{m} v_x(a^*) + \frac{m-1}{m} \right) \tag{4.22}$$

$$= \frac{2-m}{mn} + \frac{m-1}{m} \tag{4.23}$$

$$= \frac{n-1}{n} - \frac{n-2}{nm} \tag{4.24}$$

As with p_{EF}, since this mechanism is equivalent to ignoring an agent when $n > m$, the worst-case expected ratio stays at $r_{WCE} = \frac{n-1}{n}$. So overall we have, for this mechanism:

$$r_{WCE} = \begin{cases} \frac{(n-1)}{n} - \frac{(n-2)}{nm} & \text{, if } n \leq m \\ \frac{(n-1)}{n} & \text{, if } n > m \end{cases} \tag{4.25}$$

4.3.3 Randomly Partition and Ignore Agents

Partitioning and ignoring an agent will have a performance at most that of a mechanism that simply ignores an agent. When just ignoring an agent, the mechanism will maximise over the set of all possible allocations, under limited agent information. Combining the procedures will maximise over a strict subset of possible allocations with the same restrictions on observable agent values. From the perspective of efficiency, a mechanism that ignores an agent will compare the total utility of non-ignored agents for all allocations tested by the ignoring-and-partitioning mechanism, but can also consider other allocations that are precluded by partitioning. If multiple rounds of affine-maximisers are used (to break ties), then these use any previously-ignored agents (agents with zero-weight). Combining ignoring with partitioning does not expand possible tie-breaking mechanisms over simply ignoring agents, since the same set of agent have zero-weight.

Benefits to using partitioning in tandem with ignoring agents may be seen in terms of computational and communication complexity (see Sections 5.3 and 5.4 for the complexity benefits of partitioning), or with alternative objectives besides efficiency. In the problems that we examine in this work, however, the combination of partitioning and ignoring provides no useful benefit.

4.3.4 Upper Bounds

To find an upper bound on what worst-case p_{EF} and r_{WCE} is achievable under DSIC, SBB stochastic mechanisms that satisfy independence (affine maximisers), we must consider more general cases of DSIC and SBB mechanisms, rather than the specific instances we covered above. The `RandomAuctioneer` mechanism ignores a single agent, and this agent is chosen from all agents with uniform probability. To find an upper bound on what is achievable for mechanisms that ignore agents, we must consider mechanisms that can ignore any number of agents, with the probability of ignoring an agent chosen non-uniformly. We examine upper bounds for such mechanisms in Subsection 4.3.4.1.

Our approach of partitioning agents presented in Subsection 4.3.2 is an extreme case that closely resembles ignoring an agent. To find upper bounds on the performance of partitioning mechanisms, in Subsection 4.3.4.2 we examine mechanisms with any number of partitions, and with partitions of varying size. We also consider non-uniform assignment of agents and items to partitions.

Finally we can avoid the requirement of ignoring or partitioning if the mechanism is limited to only two possible outcomes. However, in Subsection 4.3.4.3 we show that limiting a mechanism to only two allocations does not improve worst-case performance.

4.3.4.1 Randomly Ignore Agents

In the Randomly Ignore Agents mechanism, ignoring more than one agent will never increase the worst-case p_{EF} or r_{WCE}. With less information about agent types, the allocation mechanism cannot find a more efficient allocation than a mechanism that ignores only a single agent. Consider the deterministic mechanisms that these mechanisms randomise over. For any affine maximisation f that ignores two or more agents, $\mathbf{N}_X$, we can simply create a mechanism that ignores only one of these agents $i \in \mathbf{N}_X$. This new mechanism can calculate $f(v)$, but can also maximise with additional agent information, so will perform at least as well.

Note here that the random serial dictatorship mechanism is an extreme case of randomly ignoring agents (with multiple tie-breaking round), where all but one agent is ignored. The first dictator, $d_1 \in \mathbf{N}$ chooses its most preferred item, equivalently to an `IgnoreAgents` mechanism with $\mathbf{N}' = \{d_1\}$, which optimises the allocation only considering agent d_1. Equations 3.13, 3.37, and 3.38 set all payments to zero:

$$t_i(v) = \begin{cases} \sum\limits_{j \in \mathbf{N}' \setminus \{i\}} v_j(a) = \sum\limits_{j \in \emptyset} v_j(a) = 0 & i \in \mathbf{N}' \\ \dfrac{(1 - |\mathbf{N}'|)}{|\mathbf{N}_X|} \sum\limits_{i \in \mathbf{N}'} v_i(a^*) = \dfrac{0}{n-1} v_{d_1} = 0 & i \in \mathbf{N}_X \end{cases} \tag{4.26}$$

Since the dictator only chooses its own item, tie-breaking between allocations is performed by iteratively running the same mechanism on the remaining agents and items. The second, randomly-chosen dictator $d_2 \in \mathbf{N}$ runs an `IgnoreAgents` mechanism with $\mathbf{N}' = \{d_2\}$, however this time the space of allocations is reduced to only include those where agent d_1 gets the item it selected.

Using an alternative tie-breaking procedure in the `RandomAuctioneer` mechanism is a variation that must be considered for finding optimal worst-case performance. Since only one agent is unallocated, the tie-breaking procedure must determine the item for this agent. Since this agent selects the item it most prefers, this is the outcome with the highest total utility of all the tied outcomes, so no other tie-breaking procedure can outperform it. While we also break ties among allocations for the agents in $\mathbf{N}'$, in the worst-case there is a unique maximum, so this choice does not affect worst-case performance.

The remaining variations are the probabilities that different agents are ignored. When $n > m$, if there is one agent, i, whose probability of being ignored is greater than $\frac{1}{n}$, then any value profile with a unique efficient allocation, such that this agent is allocated will have $p_{EF} < \frac{(n-m)}{n}$, since the agent is more likely to be left unallocated. This will also lower r_{WCE}, since this agent i may be the only agent with a non-zero item value. Similarly, when $n < m$, any non-uniform probabilities for ignoring agents will lead to lower worst-case behaviour. Using the example value profiles from Subsection 4.3.1, such as

$$v = \begin{pmatrix} 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.27}$$

we permute agents such that an agent with higher probability of being ignored is given the highest values for items that other agents desire. For example, suppose agent i has a probability of being ignored of $\frac{1}{n} + \epsilon$ for some $\epsilon > 0$, with all other agents being ignored with equal probability $(\frac{1}{n} - \frac{\epsilon}{n-1})$. We can permute the rows in Equation (4.27) such that agent i has the value profile $(0.4, 0, 0, 0, 0, 0)$. The expected loss of social surplus ratio is value each agent has in the efficient allocation weighted by the probability it is ignored.

So the expected loss of efficiency is:

$$0.4(\frac{1}{n} + \epsilon) + (0.7\frac{1}{n} - \frac{\epsilon}{n-1}) \tag{4.28}$$

$$= \frac{1}{n} + (0.4 - \frac{0.7}{n-1})\epsilon \tag{4.29}$$

$$> \frac{1}{n} \tag{4.30}$$

Since the loss is higher that $\frac{1}{n}$, the expected social surplus ratio is below the worst-case upper bound of $\frac{n-1}{n}$

Thus the bounds on the mechanism in Subsection 4.3.1 are a tight upper bound for mechanisms that ignore agents.

4.3.4.2 Randomly Partitioning Agents

For any partitioning approach, a problematic family of value profiles are those where $v_i(i) = 1$ and $v_i(j) = 0$ for all $i \neq j$ (see Equation (4.14)), along with all permutations of agent or item labels. That is, there is exactly 1 efficient allocation. If a mechanism favours placing a particular agent i with item j, this must come at the expense of at least one other pairing, (k, l). In this case, a lower probability of efficiency would happen with a value profile that had an optimal allocation a^* such that $a^*(i) = l, a^*(k) = j$, than if all pairings occurred with equal probability. So for the upper bound on worst-case performance, we are left to find the optimal number and sizes of partitions. The agents and items are partitioned into $k \geq 2$ disjoint subsets, $\mathbf{N}_1, \ldots \mathbf{N}_k$, and $\mathbf{M}_1, \ldots \mathbf{M}_k$. The sizes of these subsets is denoted as $n_q = |\mathbf{N}_q|$, and $m_q = |\mathbf{M}_q|$. If $n > m$, there can be partitions with no items (agents who are never allocated items), while if $n < m$ there can be partitions with no agents (items that are never allocated). The sizes of the partitions, and the probabilities items have of being in each partition defines a particular partitioning.

The stochastic mechanism then randomly populates each partition with items and agents according to these probabilities.

For $n \leq m$, consider one particular permutation of agents in partitions, then add the n items in the efficient allocation one-by-one. There are $\frac{m!}{(m-n)!}$ ways to place these n items (ordered) into partitions (the remaining $(m-n)$ items can go anywhere, since we can assume these are worthless in the worst case). Since order within a partition does not matter (as items can be allocated efficiently) the number of correct orderings for a partition $q = (N_q, M_q)$ is

$$\binom{m_q}{n_q} n_q! = \frac{m_q!}{(m_q - n_q)!} \tag{4.31}$$

Since all partitions need the correct items, the probability of efficiency for this value profile is:

$$p_{EF} = \frac{(m-n)!}{m!} \prod_{q=1}^{k} \frac{m_q!}{(m_q - n_q)!} \tag{4.32}$$

This expression is maximised when $k = 2$ and one partition has a single item and agent. In this case the above reduces to $\frac{1}{m}$, which agrees with Equation (4.19).

When $n > m$, we can repeat the above process, reversing the role of items and agents, to get:

$$p_{EF} = \frac{(n-m)!}{n!} \prod_{q=1}^{k} \frac{n_q!}{(n_q - m_q)!} \tag{4.33}$$

Again, this is maximised when $k = 2$, but since we can now have a partition with no items, the above is maximised when one partition has one agent and no items. Here we get $p_{EF} = \frac{(n-m)}{n}$, which also agrees with Equation (4.19), thus this is a tight bound for partition mechanisms.

For r_{WCE}, consider the family of value profiles where $v_a(b) = 1$ and $v_i(j) = 0$ for all $i \neq a$ or $j \neq b$. There is one agent-item pair that must end up in the same partition;

otherwise, it is impossible to get non-zero efficiency. Since this could be any agent-item pair, not known in advance, an upper bound on r_{WCE} is the lowest probability any specific agent will be paired with a specific item. The sum of agent utilities will be zero if this agent isn't paired to its item, and 1 if it is. If agents and items don't have exactly equal probability of being paired to each other, then select the pair with the lowest probability to find the worst-case behaviour. To maximise this minimum probability, all pairings must be equal. In this case, the probability agent a and item b end up in the same partition, and thus r_{WCE} is:

$$r_{WCE} = \sum_{q=1}^{k} \frac{n_q}{n}\frac{m_q}{m} = \frac{1}{nm}\sum_{q=1}^{k} n_q m_q \tag{4.34}$$

This is maximised when $k = 2$ and one partition has only a single agent. If $n \leq m$, this solo agent must have a single item, which gives

$$r_{WCE} = \frac{(n-1)(m-1) + 1 \cdot 1}{mn} = \frac{n-1}{n} - \frac{n-2}{nm} \tag{4.35}$$

If $n > m$, we can increase the ratio by giving the solo agent no item, so:

$$r_{WCE} = \frac{(n-1)m + 1 \cdot 0}{mn} = \frac{n-1}{n} \tag{4.36}$$

These bounds agree with Equation (4.25), thus they are tight for partition mechanisms.

4.3.4.3 Two Allocations

The requirement of an affine-maximising SCF (due to Theorem 3.6) only holds if there are at least 3 possible outcomes, so an additional class of DSIC, SBB mechanisms we consider first limit $\mathbb{A}'$ to 2 outcomes then select according to some weakly monotonic SCF. We can show that the upper bounds for this class on p_{EF} and r_{WCE} do not exceed those for the other two classes of mechanisms.

For p_{EF}, consider the value profile described above in Equation (4.14), with only a single efficient allocation. For the efficient allocation to be chosen, it must be one of the two randomly selected, which occurs with probability $\frac{2(m-n)!}{n!}$ if $n \leq m$, or $\frac{2(n-m)!}{n!}$ if $n \geq m$. This only exceeds p_{EF} for the `RandomAuctioneer` mechanism where $|\mathbb{A}| = 2$ (that is $n = 2$ and $m = 1$ or 2) and only if the mechanism is able to choose the efficient allocation from the 2 selected allocations.

Next, for r_{WCE}, consider value profiles where $v_{a,b} = 1$ and $v_{i,j} = 0$ for all $i \neq a$ or $j \neq b$. The expected ratio is thus the expected probability the mechanism randomly selects an allocation that assigns item b to agent a. The minimum number of allocations to have every possible pairing of agent to item is $\max(n, m)$, so the upper bound on r_{WCE} is $\frac{2}{\max(n,m)}$. This only exceeds the bounds for `RandomAuctioneer` when $n = 2$ and $1 \leq m \leq 3$, and again, only if the mechanism chooses the efficient allocation from the 2 selected.

To exceed the bounds of the `RandomAuctioneer` mechanism for $n = 2$, this mechanism must perform an affine maximisation on the two allocations randomly selected in order to find the efficient allocation. However, with payments of the form in Equation (3.13), with two agents and $\gamma_1 = \gamma_2 = 1$, this cannot be SBB.

4.3.4.4 Overall Tight Upper Bound

We have covered all options for stochastic DSIC, SBB mechanisms that satisfy independence for the problem of allocating heterogeneous items to unit demand agents. This covers all truthful, SBB, affine-maximising mechanisms. The bounds presented above (see Table 4.2 for a summary) are tight within this class, and the mechanisms are optimal according to the metrics of p_{EF} and r_{WCE}. We conjecture these bounds to be tight for all DSIC, SBB mechanisms. When $n > m$, the mechanisms in Subsection 4.3.1 and Sub-

	Probability of efficiency p_{EF}	Expected surplus ratio r_{WCE}
$n \leq m$	$\frac{1}{n}$	$\frac{n-1}{n}$
$n > m$	$\frac{n-m}{n}$	$\frac{n-1}{n}$

Table 4.2: *The tight bounds on the worst-case performance of the two efficiency measures for heterogeneous item allocation. These bounds are achieved by the* **`RandomAuctioneer`** *(Ignore One Agent) mechanism, which is equivalent to the Partition Agents mechanism when $n > m$.*

section 4.3.2 are equivalent, but when $n \leq m$, the optimal mechanism, according to these metrics, is `RandomAuctioneer`, which randomly ignores one agent.

Guo et al. [2011] provided a lower bound on r_{WCE} in general domains for Faltings' mechanism [Faltings, 2005], which involves ignoring an agent. This lower bound corresponds to our upper bound on r_{WCE}, which proves Faltings' mechanism to be optimal within the affine-maximising class of mechanisms when allocating heterogeneous items to unit demand agents.

Compare this worst case ratio to the deterministic `HETERO` mechanism of Gujar and Narahari [2011], which is efficient, individually rational, but weakly budget balanced (as proven by Guo [2012]). If $n \leq m + 1$, the optimal ratio is 0, but for $n > m + 1$, the tight bound on the ratio is given by:

$$r^* = 1 - \frac{\binom{n-1}{m}}{\sum_{j=m}^{n-1} \binom{n-1}{j}} \tag{4.37}$$

This is the optimal ratio for weak-budget-balanced and efficient mechanisms, and is the same worst-case optimal ratio for allocation of homogeneous items [Guo and Conitzer, 2009;

Moulin, 2009]. While the expected ratio in Equation (4.18) is clearly higher than the 0 ratio that the deterministic worst-case-optimal mechanism gives when $n \leq m+1$, it also exceeds the ratio in Equation (4.37) when

$$\frac{1}{n} < \frac{\binom{n-1}{m}}{\sum_{j=m}^{n-1} \binom{n-1}{j}} \tag{4.38}$$

This holds when $n \leq 2(m+1)$ but does not hold when $n \geq 3(m+1)$.

4.4 Room Assignment–Rent Division

The room assignment–rent division problem can be considered a more constrained variant of heterogeneous item allocation, since it requires a specific payment structure for individual agents as well as the aggregate property of SBB. In Section 3.4 we showed that no deterministic, DSIC mechanism is envy-free for RARD, and unfortunately randomisation does not provide any gain on worst-case performance of affine-maximising DSIC mechanisms, as we cover in Subsection 4.4.1. While DSIC is the stronger truthfulness constraint in stochastic mechanisms, if the random signal used by the mechanism can be reasonably assumed to be unknown to the agents, DSIC can be relaxed to truthfulness *in expectation*. This still assumes agents act truthfully in dominant strategies, regardless of other agents' types, but instead the agents act to maximise their *expected* utilities, given the random distribution of the mechanism. If agents know the random outcome prior to reporting their types, the mechanism potentially loses its guarantee of truthfulness, as it is not necessarily a distribution over deterministic, DSIC mechanisms.

4.4.1 Dominant Strategy Incentive Compatible Mechanisms

As we showed in Example 4.1 it is trivial to achieve *ex ante* envy-freeness in DSIC mechanisms. Thus, we focus on other measures of envy-freeness. For an envy-free outcome in RARD we require both efficiency (as a requirement of envy-freeness for RARD) and SBB (as a constraint of the RARD problem), so an initial mechanism to investigate is the `RandomAuctioneer` mechanism of Subsection 4.3.1, since this has the optimal p_{EF} (and when $n = m$, it is equivalent to the partitioning approach). Any mechanism must have $p_{NF} \leq p_{EF}$, so this sets the initial upper bound on the worst-case p_{NF}.

The non-ignored agents operate under a VCG mechanism, and such payments are known to be envy-free when agents only receive one item [Leonard, 1983; Cohen *et al.*, 2010]. This means the non-ignored agents will always be envy-free among themselves. However, the payment received by the ignored agent is the surplus from the VCG mechanism, which can be as much as the value of the efficient allocation among the non-ignored agents. This situation occurs when every agent is paying the exact value of its own item, so ends up with a net utility of zero, while the ignored agent receives a positive payment that all other agents will be envious of. Thus, this mechanism has a worst-case $p_{NF} = 0$, which is, in fact, the only achievable for DSIC, affine-maximising mechanisms, deterministic or stochastic.

Theorem 4.1. *A DSIC, affine-maximising mechanism for the room assignment–rent division problem has a worst-case $p_{NF} \leq \epsilon$ for some $\epsilon \approx 0$.*

Proof. There are value profiles for which only a single set of payments gives an envy-free allocation, such as when all agents have the same value profile. Consider, for example, a set of agents with the following value profile and some total payment T.

$$v = \begin{pmatrix} c_1 & c_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & 0 & \dots & 0 & 0 & 0 \\ 0 & c_2 & c_3 & c_4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & c_{n-2} & c_{n-1} & c_n \\ 0 & 0 & 0 & 0 & \dots & 0 & c_{n-1} & c_n \end{pmatrix} \tag{4.39}$$

This has a number of efficient allocations with equivalent value, such as the assignment down the main diagonal. However, to ensure envy-freeness, through comparing adjacently labelled agents, we require:

$$c_1 + t_1 \geq c_2 + t_2 \tag{4.40}$$

$$c_2 + t_2 \geq c_1 + t_1 \tag{4.41}$$

$$c_2 + t_2 \geq c_3 + t_3 \tag{4.42}$$

$$c_3 + t_3 \geq c_2 + t_2 \tag{4.43}$$

$$\dots$$

Which leads to

$$c_i + t_i = c_j + t_j \qquad \forall i, j \in \mathbf{N} \tag{4.44}$$

$$\sum_{i \in \mathbf{N}} t_i = T \tag{4.45}$$

This is a system of n linear equations, with n variables, for which there is a unique solution dependent on all c_i.

Consider an affine-maximising DSIC mechanism for the RARD problem. From the results in Subsection 3.2.5, this mechanism randomises over deterministic mechanisms that either ignore agents or partition agents. In either case, the payment function for each agent depends on the values of a strict subset of all agents. In the case of ignoring a single agent, for example, a payment function will be able to determine $n-1$ equations from the above system of equations, but will still have one degree of freedom in calculating an envy-free payment (disregarding additional constraints on these payments to maintain DSIC). A stochastic mechanism could randomise over all possible values that this final degree of freedom permits. But since the space of agents' values is continuous, there is an infinitesimal probability the correct value is chosen where the value profile is of the form above. That is $p_{EF} \leq \frac{1}{|V|} = \frac{1}{\infty} \approx 0$. □

Since, in the worst case, a DSIC mechanism from the class of affine-maximisers will always have at least one envious agent, the result in Theorem 4.1 places an upper bound on E_{NF} of $\frac{n-1}{n}$. The `RandomAuctioneer` mechanism, however, has a worst-case $E_{NF} \leq \frac{1}{n}$. Consider agents with the same value profile of

$$v_i = (1, 2, \ldots, n) \tag{4.46}$$

That is, every agent has value for item j of $v_i(j) = j$. Under VCG payments for the non-ignored agents, an agent receiving item j will have to pay a value of $j-1$, since its presence has prevented $j-1$ agents of acquiring an item of value 1 higher than what they currently hold. Under this profile, the utility provided by each of the top $n-1$ items using VCG pricing is 1. However, since all these VCG payments are transferred to the ignored agent, receiving item 1, this ignored agent has a utility of $1 + \sum_{i=1}^{n-1} i$, of which all other agents will be envious. This happens regardless of the choice of ignored agents, so there is always only a single envious agent and the $E_{NF} = \frac{1}{n}$ for this value profile.

Randomised DSIC mechanisms do not offer an improvement over deterministic mechanisms for envy-freeness in this setting. The envy-freeness constraints that clash with dominant strategy incentive compatibility remain in each of the deterministic mechanisms that we randomise over. However, truthfulness in expectation (TIE), the alternative generalisation of truthfulness to a randomised setting, allows us to develop mechanisms with improved envy-freeness while agents still act truthfully in dominant strategies.

4.4.2 Truthful in Expectation Mechanisms

Since DSIC stochastic mechanisms offer no improvement on envy-free final outcomes, we now examine RARD mechanisms that satisfy a weaker notion of truthfulness: truthful *in expectation*. Recall that a mechanism is truthful in expectation if, irrespective of the actions of other agents, an agent's expected utility cannot be increased by misreporting its type.

We begin with a simple sufficiency condition for truthfulness in expectation in stochastic mechanisms for the RARD problem.

Lemma 4.1. *An RARD mechanism is truthful in expectation if each agent's* expected *share of the divisible resource, and probability of being assigned to each indivisible item is constant (independent of reported types).*

Proof. Let $p_{i,j}(v)$ denote the probability agent $i \in \mathbf{N}$ is assigned item/room $j \in \mathbf{M}$, and $\bar{t_i}(v) = E(t_i(v))$ be agent i's expected share of the divisible resource. The expected utility of agent $i \in \mathbf{N}$ is calculated as: $E(u_i) = \sum_{j \in \mathbf{M}} p_{i,j}(v) v_i(j) + \bar{t_i}(v)$. As all $p_{i,j}$ and $\bar{t_i}(v)$ are constant with respect to the agent's bid/reported type, the agent's expected utility is constant and cannot be increased by misreporting. □

Note that these are not the necessary conditions for a truthful RARD mechanism. That is, a truthful in expectation mechanism does not necessarily have constant assignment-probabilities and expected shares of the divisible resource. We use the conditions in Lemma 4.1 to define a simple, truthful mechanism as a baseline for comparing other randomised mechanisms.

Example 4.4 (A simple stochastic RARD mechanism)**.** *From previous work [Alkan* et al.*, 1991; Haake* et al.*, 2002], given full knowledge of agents' types, we can find an envy-free allocation and division, denoted (a^*, t^*). If the mechanism meets the conditions specified by Lemma 4.1, then the mechanism will be truthful and we can calculate this envy-free outcome. Our random mechanism first calculates the envy-free solution, then randomly selects an integer $x \in [0, n-1]$. Agent i is given the item and share allocated to agent $(i + x) \pmod n$ in the envy-free allocation. Thus, $a^x(i) = a^*((i + x) \pmod n)$.*

This can be viewed as a randomisation over n deterministic mechanisms, where mechanism $x \in [0, n-1]$ deterministically allocates $a^x(i) = a^*((i + x) \pmod n)$, with payments $t_i^x = t^*_{(i+x) \pmod n}$. Note that none of these deterministic mechanisms are DSIC. However, each agent has a $\frac{1}{n}$ probability of being assigned any particular item, and an agent's expected share of the divisible resource is

$$\bar{t_i} = \sum_{j \in n} \frac{1}{n} t_j^* = \frac{1}{n} \sum_{j \in n} t_j^* = \frac{T}{n} \tag{4.47}$$

This is constant for each agent, so by Lemma 4.1 the mechanism is truthful in expectation, allowing the mechanism to correctly calculate (f^*, t^*).

The mechanism in Example 4.4 is illustrated in Figure 4.1 on a 4-agent/room problem, with agents' values presented in Table 4.3 and a total rent of $T = \$1200$. One possible envy-free assignment and division is to assign Morgan to room 1, Taylor to room 2, Sasha to room 3 and Alex to room 4, with prices set at \$400, \$200, \$350 and \$250, respectively.

Using this mechanism, whenever $x = 0$ the envy-free outcome is chosen, and this occurs with probability $\frac{1}{n}$. Apart from special cases, for all other values of x, all agents will be

	Room 1	Room 2	Room 3	Room 4
Morgan	**\$500**	\$275	\$350	\$200
Taylor	\$450	**\$300**	\$400	\$250
Sasha	\$350	\$300	**\$450**	\$300
Alex	\$300	\$250	\$300	**\$350**

Table 4.3: *Values that agents receive for each of the 4 rooms for an example RARD problem with* $T = \$1200$. *The efficient assignment is highlighted in bold.*

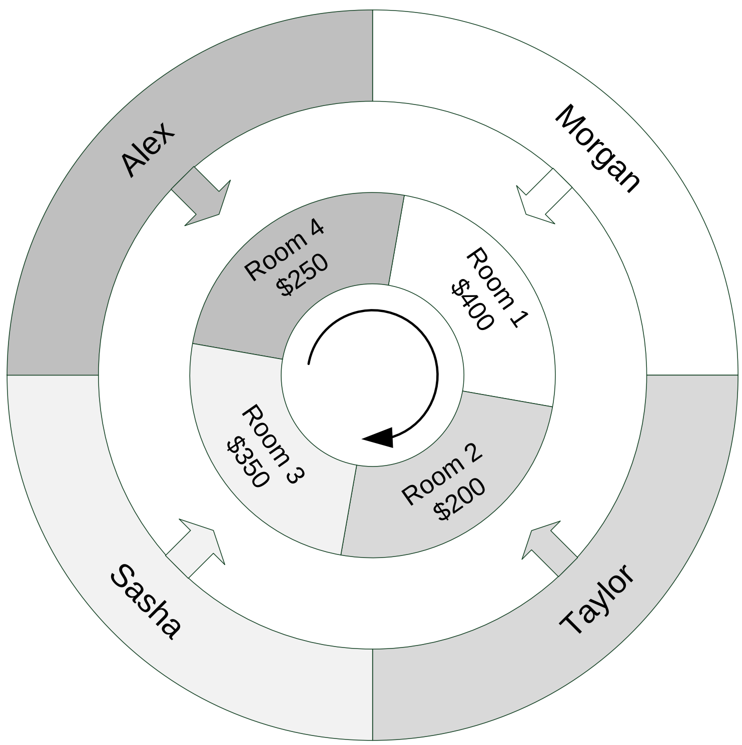


Figure 4.1: *Illustration of the simple, stochastic RARD mechanism. An envy-free room assignment and rent division is calculated (illustrated by matching shading), then all agents are shifted in the assignment by a random amount.*

envious of their bundle from the envy-free outcome, (a^*, t^*). Thus, for this mechanism the worst-case $p_{NF} = \frac{1}{n}$, which is an immediate improvement over the bound for DSIC mechanisms. When $x = 0$, there are n envy-free agents, while in the worst case, all other choices of x will have no envy-free agents. This gives a worst-case expected number of envy-free agents of $n \cdot \frac{1}{n} + 0 \cdot \frac{n-1}{n} = 1$ so the worst-case $E_{NF} = \frac{1}{n}$. In this mechanism, all agents have the same lottery over items and expected payment, so it is *ex ante* envy-free.

4.4.3 Maximising Probability of Envy-Freeness

A truthful mechanism that guarantees 100% probability of envy-freeness would be optimal for the three envy-free properties defined in Section 4.1. Unfortunately, this is not possible for TIE, RARD mechanisms as illustrated in Example 4.5.

Example 4.5. *Consider the two agent case and all value profiles such that $(v_1(1)-v_1(2)) > (v_2(1) - v_2(2))$. In an envy-free outcome agent 1 must always receive item 1, and agent 2 receives item 2, otherwise no envy-free payments exist. In all outcomes, to ensure envy-freeness, agent 1 requires a payment in the range $t_1 \in [(v_2(1) - v_2(2)), (v_1(1) - v_1(2))]$. This also means agent 1's expected payment, $\bar{t_1}$ also lies in this same range. Unless this expected share is constant, at least one of the two agents can increase its expected utility by misreporting to reduce this range. However, there is no constant value $t_1 \in [(v_2(1) - v_2(2)), (v_1(1) - v_1(2))]$, for all cases where $(v_1(1) - v_1(2)) > (v_2(1) - v_2(2))$. Thus, no truthful in expectation mechanism has $p_{NF} = 1$ or $E_{NF} = 1$.*

As we saw in Subsection 4.4.1, a non-zero probability of envy-freeness cannot be guaranteed for DSIC mechanisms, and we know from Example 4.4, TIE mechanisms can perform better.

Conjecture 4.1. *A truthful (in expectation) mechanism for the RARD problem with n agents has a guaranteed probability of envy-freeness of* at most $\frac{1}{n}$.

Our belief behind this conjecture is as follows. In our setting with an equal number of agents and items, an envy-free allocation is an efficient allocation [Alkan *et al.*, 1991]. So, if a mechanism were capable of envy-freeness with probability $p > \frac{1}{n}$, it would also provide an efficient allocation with probability at least p.

Using a similar technique to that used by Lavi and Swamy [2011], we can take a truthful in expectation mechanism and make a deterministic, DSIC "support" mechanism. When looking at expected utility, we don't need to consider actual deterministic outcomes, only the probabilities of being assigned different items and expected payments. Our stochastic mechanism $\Delta\hat{\mathcal{M}}$ has a distribution over allocation functions or, alternatively, a social choice function that produces a distribution over allocations $\bar{f} : V \to \Delta\mathbb{A}$. Similarly, the expected payments can be captured by an expected payment function $\bar{t} : V \to \mathbb{R}^n$. Since every deterministic payment function from $\Delta\hat{\mathcal{M}}$ is SBB (totalling exactly to T), the expected payment function $\bar{t}$ must also be SBB. So this gives a deterministic, DSIC and SBB mechanism that chooses outcomes from the space $\Delta\mathbb{A}$. An agent's utility for an outcome $\bar{a} \in \Delta\mathbb{A}$ is the agent's expected value for that distribution over item assignments. This expected value is a weighted sum of agents' values for individual allocations, so this new value function is an extension to a continuous domain of the original value function, by interpolating between values for allocations.

Since this support mechanism is DSIC, we know from Theorem 3.3 that $\bar{f}$ is weakly monotonic. While this is similar to a social choice function for heterogeneous item allocation, we cannot immediately apply Theorem 3.5 to show that $\bar{f}$ must be an affine maximiser. However, from our examination of mechanisms for this problem, we conjecture that this social choice function must also be an affine maximiser over $\Delta\mathbb{A}$ (or a distributional affine maximiser [Dobzinski and Dughmi, 2009] over $\mathbb{A}$). Thus, this support mechanism is an affine maximiser of the form Equation (3.12) with SBB expected payments as in Equation (3.13), that is:

$$\sum_{i \in \mathbf{N}} h_i(v_{-i}) = -\left(\sum_{i \in \mathbf{N}'} \frac{1}{\gamma_i}\right)\left(U_0\left(\bar{a^*}\right) + \sum_{i \in \mathbf{N}'} \gamma_i v_i\left(\bar{a^*}\right)\right) + \sum_{i \in \mathbf{N}'} v_i\left(\bar{a^*}\right) + T \tag{4.48}$$

As discussed in Subsection 3.2.5, achieving SBB through this equality requires the additional payment functions $h_i(v_{-i})$ to calculate $\bar{a^*}$. This can only be achieved for DSIC payment functions if $\bar{a^*}$ can be calculated as subproblems on strict subset of agents, i.e. ignoring or partitioning. While ignoring agents remains the same when the SCF chooses distributions over allocations, the partitioning approach is slightly different. Optimising while ignoring a single agent will give a worst-case probability of efficiency of zero, for the same reasons as Theorem 3.8, since this support mechanism is deterministic and an agent is ignored deterministically.

In Subsection 3.2.5.2, breaking the allocation up into partitions required that partitions determine an exact allocation of item to agent. Each subproblem (partition) determined the allocation of a subset of items and agents, which could then be combined to give a complete allocation. When calculating $\bar{a^*}$, each partition must operate on a strict subset of agents, but may only be responsible for a partial allocation of agents' and items' probabilities. For example, one partition may assign agent i to item j with probability 0.2, while a second partition assigns i to j with probability 0.4. In the final outcome chosen by $\bar{f}$, these would be totalled to give 0.6.

When partitioning, each partition must exclude at least one agent to allow for SBB payments. Each partition can maximise the probability of choosing the efficient allocation, but excluding all but one agent can cause this partition to miscalculate the efficient allocation (e.g. value profiles such as Equation (4.15)). Since this single, non-pivotal agent could be any agent, not known in advance, the probability of efficiency is maximised by having each agent excluded from a partition with equal probability, requiring at least n partitions. Items can have their total probabilities divided equally among partitions (that is, each partition assigns an item with a probability between 0 and $\frac{1}{n}$), as an uneven division will only harm worst-case behaviour. In the worst case, excluding all but one agent will result in a new efficient allocation among the remaining agents. In the $(n-1)$ partitions that include this single, non-pivotal agent i, this agent will be assigned a different item to the one it would receive in the efficient allocation. Since each partition assigns probabilities of each item up to $\frac{1}{n}$, agent i will only get its correct item with probability $\frac{1}{n}$, thus limiting p_{NF} to be at most $\frac{1}{n}$.

If this conjecture holds, then this is a tight bound as demonstrated by the simple randomised RARD mechanism described in Example 4.5, with $p_{NF} = \frac{1}{n}$. This places some limiting restrictions on what is possible with a strategy-proof mechanism for this problem. Envy-freeness at a low probability that asymptotically goes to zero means that most of the time (for worst-case inputs), any truthful mechanism will produce a bad result. Considering only envy-free outcomes ignores what happens in the remainder of cases. In the mechanism described above, in the $(n-1)$ non-envy-free outcomes, every single agent will be envious. This motivates measuring the quality of each outcome with more detail than a binary test of envy-freeness.

4.4.4 Maximising Expected Envy-Freeness

While having all agents envy-free is the ideal outcome, attempting to maximise the probability of such an outcome can come at the expense of the quality (in terms of level of envy) of non-envy-free outcomes. For truthful mechanisms, these non-envy-free outcomes are the most likely, so when comparing mechanisms they should not be ignored.

The above mechanism in Example 4.5, with p_{NF} of $\frac{1}{n}$, has $E_{NF} = \frac{1}{n}$ (expected number of envy-free agents is 1), as defined in Definition 4.5. This is because there is a $\frac{1}{n}$ probability of n envy-free agents, and 0 envy-free agents otherwise. By this measure alone, this is equivalent to a mechanism that always has 1 envy-free agent, such as a random (serial) dictatorship mechanism.

Example 4.6. *A random dictatorship mechanism for the RARD mechanism picks an agent at random and gives that agent its most preferred item along with the maximum non-negative share of the divisible resource (i.e.* $\max(T, 0)$*), with the remaining resources allocated to other agents independently of all agent bids. As the probability of being the dictator does not depend on reported types, and that dictator gets its most preferred item, no agent can benefit by misreporting its type. After a dictator is randomly selected, no agent has an incentive to misreport, so the mechanism is DSIC, while the payments are ensured to sum to* T*. The dictator is always envy-free while all other agents will envy the dictator in the worst case. This gives* $p_{NF} = 0$ *and* $E_{NF} = \frac{1}{n}$*.*

The maximum expected envy-freeness, E_{NF}, is 1, and this implies that every outcome is envy-free. However, as shown in Example 4.5, this is not possible for a truthful mechanism.

Conjecture 4.2. *A truthful (in expectation) mechanism for the RARD problem with* n *agents has an expected envy-freeness of* $E_{NF} \leq \frac{n-1}{n} + \frac{1}{n^2})$*. This happens with an expected number of envy-free agents of* $(n - 1 + \frac{1}{n})$*.*

From Conjecture 4.1, the maximum probability of an envy-free outcome is conjectured to be $\frac{1}{n}$, where there are n envy-free agents. The remaining outcomes, with probability $\frac{n-1}{n}$, can have at most $(n-1)$ envy-free agents (otherwise the outcome would be envy-free). This gives an expected number of envy-free agents of at most

$$n\frac{1}{n} + (n-1)\frac{n-1}{n} = n - 1 + \frac{1}{n} \tag{4.49}$$

And so the fraction of envy-free agents is

$$E_{NF} \leq \frac{n-1}{n} + \frac{1}{n^2} = 1 - \frac{n-1}{n^2} \tag{4.50}$$

The p_{NF} was maximised with a fairly simple mechanism, and in the rest of this section we present mechanisms for maximising the expected number of envy-free agents. The first is a mechanism that achieves the bound in Conjecture 4.2 for two agents, followed by a more general mechanism with expected envy-freeness of at least $\frac{n-1}{n}$, falling short of the upper bound by $\frac{1}{n^2}$.

4.4.4.1 The 2 Agent Case

For $n = 2$, this upper bound on the worst-case expected level of envy-freeness, $E_{NF} = \frac{3}{4}$, can be reached with the following mechanism. Let $\mathbf{I}_i \in \mathbb{R}^n$ denote the **point of indifference** for agent i, which is the division of the divisible resource such that all bundles have equal value according to agent i. That is, for agent $i \in \mathbf{N}$ with value profile $v_i \in V_i$, the point of indifference is the solution to the following system of equations:

$$v_i(j) + \mathbf{I}_i(j) = v_i(k) + \mathbf{I}_i(k) \qquad \forall j, k \in \mathbf{M} \tag{4.51}$$

$$\sum_{j \in \mathbf{M}} \mathbf{I}_i(j) = T \tag{4.52}$$

For two agents, this can be represented as a single value in $\mathbb{R}$, since the divisions must sum to T, and can be calculated as:

$$v_i(1) + \mathbf{I}_i = v_i(2) + (T - \mathbf{I}_i)$$
$$\Rightarrow \mathbf{I}_i = \frac{1}{2}(v_i(2) - v_i(1) + T)$$

The RARD mechanism for 2 agents proceeds as follows:

1. Randomly select agent $i \in \mathbf{N}$
2. Calculate $\mathbf{I}_i$ as in Equation (4.53) and use this to price the two rooms/items (i.e. price items such that agent i is indifferent)
3. Assign agents to rooms randomly, with equal probability

Independent of reported types, each agent has a $\frac{1}{2}$ probability of being assigned each indivisible resource, and has an constant expected payment:

$$\bar{t}_1 = \bar{t}_2 = \frac{1}{2}\left(\frac{\mathbf{I}_1 + (T - \mathbf{I}_1)}{2} + \frac{\mathbf{I}_2 + (T - \mathbf{I}_2)}{2}\right) \tag{4.53}$$
$$= \frac{T}{2} \tag{4.54}$$

So, by Lemma 4.1, this mechanism is truthful in expectation. The agent chosen to set the bundles will be envy-free with either bundle. The other agent will prefer at least one bundle, so there is a probability of $\frac{1}{2}$ this agent will be envious. Thus, with probability $\frac{1}{2}$ both agents will be envy-free, and with probability $\frac{1}{2}$ one agent (whose value was not chosen to set prices) will be envy-free. The worst-case behaviour for this mechanism is

$$p_{NF} = \frac{1}{2} \tag{4.55}$$
$$E_{NF} = \frac{1}{2}\left(\frac{1}{2}2 + \frac{1}{2}1\right) \tag{4.56}$$
$$= \frac{3}{4} \tag{4.57}$$

Thus, based on the conjectured upper bounds for both measures of envy-freeness, the worst-case behaviour cannot be improved and this mechanism is optimal for $n = 2$.

A refinement to this mechanism is possible by adding a special condition to improve non-worst-case behaviour. If both agents each prefer a different item, then both the agents can be given the items they prefer, with $\frac{T}{2}$ share of the divisible resource. In this case, the mechanism will always produce an envy-free outcome. The expected share remains as $\frac{T}{2}$ and an agent can only increase its probability of receiving a non-preferred item by misreporting. Since worst-case behaviour happens when both agents prefer different items, this refinement does not help worst-case performance.

This mechanism can be compared to the cut-and-choose mechanism from the cake cutting literature [Brams and Taylor, 1996; Brams *et al.*, 2006]. In cake cutting, there is a single, divisible resource, for which agents can prefer different segments. The cut-and-choose procedure has one agent divide the cake in such a way that it equally prefers both slices, then the other agent chooses its most preferred piece. This guarantees envy-freeness but is not a truthful mechanism. Our RARD mechanism randomly selects an agent to perform the "cut" that evenly divides the two options from that agent's perspective. While cut-and-choose is envy-free, it is not truthful, so the "choose" stage is performed randomly to ensure truthfulness in the RARD setting.

4.4.4.2 The General Case

For additional agents, we begin with the `RandomAuctioneer` mechanism of Subsection 4.3.1. Used as-is, this mechanism can cause a large degree of envy due to the payments transferred to the ignored agent, as we covered in Subsection 4.4.1. In this section, we add additional payments that are sufficient to ensure no agent envies the ignored agent, while preserving envy-freeness among the non-ignored agents.

The mechanism proceeds as follows:

1. Find the value of the efficient allocation for all agents in $\mathbf{N}$

$$\bar{C} = v(a^*) = \max_{a\in\mathbb{A}} \sum_{i\in\mathbf{N}} v_i(a) \tag{4.58}$$

2. Randomly select agent $x \in \mathbf{N}$, with uniform probability over all agents, as the agent to be ignored

3. Allocate agents $\mathbf{N}\backslash\{x\}$ efficiently $a^x = \arg\max_{a\in\mathbb{A}} \sum_{i\in\mathbf{N}\backslash\{x\}} v_i(a)$

4. Since $n = m$, there is only a single leftover item, which is then assigned to x

5. Agents make payments according to t_i^x for each agent $i \neq x$, and t_x^x for agent x, as defined in Equations (4.59) and (4.60)

$$t_i^x(v) = \sum_{j\in\mathbf{N}\backslash\{i,x\}} v_j(a^x) - \sum_{j\in\mathbf{N}\backslash\{i,x\}} v_j(a_{-i}^x) + \frac{T}{n} + \frac{\bar{C}}{n} \qquad \forall i \neq x \tag{4.59}$$

$$\begin{aligned} t_x^x(v) &= T - \sum_{i\neq x} t_i^x(v) \\ &= \sum_{j\in\mathbf{N}\backslash\{x\}} \left((2-n)v_j\left(a^x\right) + \sum_{k\in\mathbf{N}\backslash\{j,x\}} v_k(a_{-j}^x) \right) + \frac{T}{n} - \frac{(n-1)}{n}\bar{C} \end{aligned} \tag{4.60}$$

The payment for agent x is calculated based on the other agents' payments to ensure strong budget balance, i.e. the sum of all payments is equal to T. The payment $t_i^x(v)$ is made up of three parts. The first two terms (the two summations) in Equation (4.59) are the VCG payments in an allocation setting with agent x ignored. For this part of the payment function, along with the allocation function a^x, the mechanism is DSIC so agents will have no incentive to misreport. Additionally, these VCG mechanisms are known to be envy-free since agents only receive one item [Leonard, 1983; Cohen *et al.*, 2010], so there will be no envy between non-ignored agents. The term $\frac{T}{n}$ is added equally to all agents, so this will not affect envy or truthfulness. It is added to ensure payments sum to T. The final term, $\frac{\bar{C}}{n}$, is added to ensure no agents are envious of the ignored agent. It is added equally to all agents, so will not create envy between non-ignored agents.

However, this additional term breaks the incentive-compatibility of the payments, as $\bar{C}$ depends on all agents' reported values. If the choice of x were known prior to reporting values, all agents (including x) could misreport in order to increase the payment they receive. When considering *expected* utility, however, agents have a $\frac{1}{n}$ probability of being the ignored agent and paying $\frac{(n-1)}{n}\bar{C}$, while they have an $\frac{(n-1)}{n}$ probability of receiving $\frac{\bar{C}}{n}$. In expected utility this final term involving $\bar{C}$ cancels out to zero, meaning the mechanism remains truthful in expectation. Additionally, if the value of the efficient allocation is at least T, then all agents will have a non-negative expected utility.

While non-ignored agents are not envious of each other, the pricing must also ensure they are not envious of the ignored agent. For simplicity of notation, we define

$$C^x = \sum_{j \in \mathbf{N} \setminus \{x\}} v_j\left(a^x\right)$$

$$C^x_{-i} = \sum_{j \in \mathbf{N}' \setminus \{i, x\}} v_j(a^x_{-i})$$

Where C^x is the value of the efficient allocation for all agents except x, and C^x_{-i} is the value of the efficient allocation for all agents except i and x. Agent i is envious of agent x

if and only if

$$v_i(a^x(i)) + t_i^x(v) < v_i(a^x(x)) + t_x^x(v) \tag{4.61}$$

$$\Rightarrow v_i(a^x(i)) + \sum_{j \in \mathbf{N} \setminus \{i,x\}} v_j(a^x) - C_{-i}^x + \frac{T}{n} + \frac{\bar{C}}{n}$$

$$< v_i(a^x(x)) + (2-n)C^x \sum_{j \in \mathbf{N} \setminus \{x\}} C_{-j}^x + \frac{T}{n} - \frac{(n-1)}{n}\bar{C} \tag{4.62}$$

$$\Rightarrow \bar{C} < v_i(a^x(x)) + C_{-i}^x + \sum_{j \in \mathbf{N} \setminus \{x\}} C_{-j}^x - (n-1)C^x \tag{4.63}$$

Since agents have non-negative values for items, we know that $C^x \geq C_{-i}^x$. Summing out over all agents gives

$$(n-1)C^x \geq \sum_{i \in \mathbf{N} \setminus \{x\}} C_{-i}^x \tag{4.64}$$

$$\Rightarrow \bar{C} \geq \bar{C} + \sum_{i \in \mathbf{N} \setminus \{x\}} C_{-i}^x - (n-1)C^x \tag{4.65}$$

Also, for any agents $i, x \in \mathbf{N}$, we have $\bar{C} \geq C^x \geq C_{-i}^x + v_i(a^x(x))$. If this did not hold, then the value of the allocation used for C^x could have been improved by using allocation a_{-i}^x and switching agent i to item $a^x(x)$. Thus we have:

$$\bar{C} \geq v_i(a^x(x)) + C_{-i}^x + \sum_{j \in \mathbf{N} \setminus \{x\}} C_{-j}^x - (n-1)C^x$$

This directly contradicts Equation (4.63), so no agent can be envious of the ignored agent, for any choice of $x \in \mathbf{N}$. The non-ignored agents are not envious of each other so there will be at least $(n-1)$ envy-free agents in any outcome. This gives a worst-case expected level of envy-freeness of $E_{NF} = \frac{n-1}{n}$, which falls short of the upper bound by $\frac{1}{n^2}$. In the worst-case, any choice of x leads to x being envious, so $p_{NF} = 0$.

	Probability of envy-freeness p_{NF}	Expected level of envy-freeness E_{NF}
DSIC Upper Bound	0	$[\frac{1}{n}, \frac{n-1}{n}]$
TIE Upper Bound	$\frac{1}{n}$	$[\frac{n-1}{n}, \frac{n-1}{n} + \frac{1}{n^2}]$
Simple RARD	$\frac{1}{n}$	$\frac{1}{n}$
`RandomAuctioneer`-based	0	$\frac{n-1}{n}$
2-Agent RARD	$\frac{1}{n}$	$\frac{n-1}{n} + \frac{1}{n^2}$

Table 4.4: *The bounds on worst-case performance for the RARD problem. The upper bounds for E_{NF} are not tight, so this shows the range of possible values for the bound. The 2-agent mechanism is optimal on both measures, but does not apply to additional agents.*

The non-worst-case inputs can be improved by decreasing the value of $\bar{C}$ to the smallest value that always satisfies Equation (4.63). The smallest $\bar{C}$ that ensures no agent is envious of the ignored agent, for any choice of ignored agent is:

$$\bar{C} = \max_{i,x \in \mathbf{N}, i \neq x} v_i(a^x(x)) + C^x_{-i} + \sum_{j \neq x} C^x_{-j} - (n-1)C^x \tag{4.66}$$

However, in the worst-case, the ignored agent x will always be envious, while all other agents are envy-free. This does not affect the worst-case expected envy-freeness $E_{NF} = \frac{n-1}{n}$.

4.4.5 Overall Upper Bounds

Table 4.4 summarises the worst case bounds for truthful, stochastic mechanisms on the room assignment–rent division problem. These are according to the measures of probability of envy-freeness and expected level of envy-freeness. The upper bounds for E_{NF} are not tight, so we show the range of possible values for this measure in the worst-case. While we conjecture that the upper bound on p_{NF} is $\frac{1}{n}$, for TIE mechanisms, this remains as future work to prove this for all TIE mechanisms. The three mechanisms we present are all truthful in expectation, due to poor performance of DSIC mechanisms we examined for this problem. The simple RARD mechanism, presented in Example 4.4, achieves the conjectured worst-case upper bound on p_{EF}, but performs poorly on E_{NF}. The `RandomAuctioneer`-based mechanism of Subsection 4.4.4.2 has a much higher E_{NF}, but in the worst-case will never produce an envy-free outcome. For the specific setting of 2 agents, we present a mechanism in Subsection 4.4.4.1 that resembles the cut-and-choose method from cake-cutting, and is optimal according to both measures of envy-freeness.

4.5 Public Good

For the allocation-related settings we examined in this chapter, we saw that achieving the combination of SBB and DSIC required either ignoring or partitioning agents. We now examine randomised mechanisms for the single-parameter domain of public goods under strong budget balance. In the public good setting, the outcome space is binary: either the public good is purchased, or the agents receive a share of the cost of the project. Because of this, partitioning doesn't make sense, since only one partition of agents will be able to decide the outcome, choosing to purchase or not to purchase. This leaves all other partitions with no effect on the social choice function and so these agents are effectively ignored. While it is an open problem whether there exist reasonable mechanisms that are DSIC and SBB and do not ignore a subset of agents, in this section we focus on mechanisms that ignore some agents. Recall that the interpretation of the public good problem used in this thesis shifts the value function of each agent by $\frac{T}{n}$ compared to the standard definition, as has been done in previous work [Guo *et al.*, 2011]. This interpretation ensures that agents' have non-negative values and the cost of the public good, T, is exactly covered when all agents' payments sum to 0.

4.5.1 Maximising Expected Social Surplus Ratio

Recent work by Guo *et al.* [2011] characterised a family of SBB, DSIC mechanisms for the public good problem. This is based on the mechanism developed by Faltings [2005] and randomly ignores a single agent. The ignored agent, $x \in \mathbf{N}$ is given a fake bid by the mechanism. The mechanism acts as though agent x reported this fake bid, and then has all agents participate in a VCG mechanism to determine whether or not the public project is undertaken. Since the reported type of agent x was never actually used, any surplus payment is transferred to agent x, which achieves SBB without harming DSIC. Mechanisms from this family vary based on how they assign this fake bid, which is denoted $\tilde{v}_x$. The mechanism of Guo *et al.* [2011], parameterised by $\tilde{c} \in [0, T]$, proceeds as follows:

1. Randomly choose agent $x \in \mathbf{N}$ to ignore, with uniform probability.

2. Ignore agent x's reported value, replacing it with $\tilde{v}_x = T - \tilde{c}$.

3. Determine the efficient outcome according to this new profile:

$$f(v) = \begin{cases} X_0 & \tilde{v}_x + \sum\limits_{i \in \mathbf{N} \setminus \{x\}} v_i < T \\ X_1 & \text{otherwise} \end{cases} \tag{4.67}$$

4. All agents, including x, make VCG payments for the mechanism with value profile $(\tilde{v}_x, v_{-x})$, with the surplus paid to agent x.

From the definition of $\tilde{v}_x$, the good is purchased if

$$(T - \tilde{c}) + \sum_{i \in \mathbf{N} \setminus \{x\}} v_i \geq T \tag{4.68}$$

$$\Rightarrow \sum_{i \in \mathbf{N} \setminus \{x\}} v_i \geq \tilde{c} \tag{4.69}$$

Thus, an alternative interpretation of this mechanism is that agent x is ignored, and the remaining agents participate in a mechanism which has a threshold of $\tilde{c}$.

Setting $\tilde{c} = 0$ results in a mechanism that always opts to purchase the public good. The sum of non-ignored agents' values is always non-negative since all individual values must be non-negative. It is not necessary to consider mechanisms outside the range $\tilde{c} \in [0, T]$. Any mechanism with $\tilde{c} < 0$ will always purchase, so is equivalent to $\tilde{c} = 0$. If $\tilde{c} > T$, then agent x has a negative fake bid. Thus there will be circumstances where the non-ignored agents' values (and thus all agents) total more than T, so it is efficient to purchase, but the fake bid lowers the sum below T and the mechanism does not purchase. A mechanism with $\tilde{c} > T$ will never choose the efficient outcome when the mechanism with $\tilde{c} = T$ chooses inefficiently, for any value profile.

Guo *et al.* [2011] assess mechanisms from this family according to r_{WCE} and find the optimal mechanism in this family according to this measure.

Theorem 4.2. *[Guo* et al.*, 2011]*

- *Any mechanism from this family with* $\tilde{c} \in [0, \frac{n-2}{n-1}T)$ *has* $r_{WCE} < \frac{n-1}{n}$.
- *Any mechanism from this family with* $\tilde{c} \in [\frac{n-2}{n-1}T, T]$ *has* $r_{WCE} = \frac{n-1}{n}$.

This sets a tight upper bound on r_{WCE} for mechanisms from this family. However the authors also show that this can be improved by randomising over a set of mechanisms from this family. One such randomisation over the parameter $\tilde{x}$, named `OPTMIX` , they define as follows:

- With probability $\frac{1}{n+1}$, run mechanism with $\tilde{x} = 0$ (i.e. always purchase)
- With probability $\frac{n}{n+1}$, run mechanism with $\tilde{x} = T$

This mechanism has $r_{WCE} = \frac{n}{n+1}$, which is the optimal r_{WCE} of all randomisations over mechanisms of this family. Note that this family does not completely characterise all DSIC, SBB mechanisms for this problem, so this is not necessarily a tight upper bound on r_{WCE}.

While these mechanisms maximise the expected efficiency, they may not produce the efficient outcome with the maximum possible probability. We now assess public good mechanisms according to the alternative measure of efficiency in randomised mechanisms, p_{EF}. In an iterated scenario, for example, with repeated auctions for different public goods, we may wish to maximise the expected number of correct purchases, and this is achieved by maximising p_{EF}.

4.5.2 Maximising Probability of Efficiency

When maximising the probability of an efficient outcome, we are only concerned with the likelihood that the mechanism makes the right choice – i.e. purchases the public good when there is sufficient demand, and otherwise returns the cost to the agents. We examine the family of mechanisms presented in Subsection 4.5.1, which are parameterised by $\tilde{c}$. For r_{WCE}, there were benefits to randomising over this parameter, and this is also the case with p_{EF}. We show this by first examining the p_{EF} for mechanisms with a constant $\tilde{c}$.

Theorem 4.3. *Any mechanism from this family with $\tilde{c} \neq \frac{n-1}{n}T$ has a $p_{EF} = 0$.*

Proof. There are two cases to consider; the first is when $\tilde{c} < \frac{n-1}{n}T$. If we have a value profile of the form $v = (\frac{t}{n}, \frac{t}{n}, \dots \frac{t}{n})$, for some value t, the mechanism will choose to purchase the good when $\frac{n-1}{n}t > \tilde{c}$. We can set the value $t \in (\frac{n}{n-1}\tilde{c}, T)$ since $\frac{n}{n-1}\tilde{c} < T$. For this value profile, the mechanism will always choose to purchase the public good, but since the total agent value $t < T$, it is inefficient, so $p_{EF} = 0$.

The second case is where $\tilde{c} < \frac{n-1}{n}T$. This time we use a similar value profile $v = (\frac{t}{n} + \epsilon, \frac{t}{n}, \dots \frac{t}{n})$ and now the mechanism will choose to not purchase if $\frac{n-1}{n}t < \tilde{c}$. We can set the value $t \in (T, \frac{n}{n-1}\tilde{c})$ since $\frac{n}{n-1}\tilde{c} > T$. For this value profile, the mechanism will never choose to purchase the public good, but since $t > T$, it is efficient to purchase, so $p_{EF} = 0$. □

This leaves just one mechanism in this class of mechanisms with a fixed $\tilde{c}$.

Theorem 4.4. *The mechanism with* $\tilde{c} = \frac{n-1}{n}T$ *has a* $p_{EF} = \frac{1}{n}$.

Proof. First we show that $p_{EF} \geq \frac{1}{n}$. Suppose this is not true, which means there exists some value profile v such that for every choice of ignored agent, it chooses the inefficient outcome. It cannot have a $p_{EF} \in (0, \frac{1}{n})$ since there are only ever n possible random choices. Let the total utility for v be t. If $t > T$ then the mechanism should purchase, but it never does. This means that for all possible agents to ignore $x \in \mathbf{N}$ we have:

$$t - v_x < \frac{n-1}{n}T \tag{4.70}$$

Summing out over all $x \in \mathbf{N}$ gives:

$$nt - \sum_{i \in \mathbf{N}} v_i < n\frac{n-1}{n}T \tag{4.71}$$

$$\Rightarrow t < T \tag{4.72}$$

which is a contradiction. Next we suppose $t < T$, so the mechanism should not purchase, but it always does. This means that for all $x \in \mathbf{N}$ we have:

$$t - v_x > \frac{n-1}{n} T \tag{4.73}$$

Again, we can sum out over all $x \in \mathbf{N}$ to get:

$$nt - \sum_{i \in \mathbf{N}} v_i > n \frac{n-1}{n} T \tag{4.74}$$

$$\Rightarrow t > T \tag{4.75}$$

which is also a contradiction. This means no counterexample exists, so the mechanism cannot have $p_{EF} = 0$ and thus must have $p_{EF} \geq \frac{1}{n}$.

Now to show $p_{EF} \leq \frac{1}{n}$, consider the value profile $v = (\frac{n-0.5}{n} T, 0, 0, \ldots, 0)$. The total utility is less than T so the mechanism should never purchase the public good. Unless the first agent is ignored, which happens with probability $\frac{1}{n}$, the total utility of non-ignored agents is $\frac{n-0.5}{n} T > \frac{n-1}{n} T$ so the mechanism will purchase with probability $\frac{n-1}{n}$. Thus the mechanism has a $p_{EF} = \frac{1}{n}$. □

For mechanisms with constant $\tilde{c}$, the optimal p_{EF} of $\frac{1}{n}$ is achieved by setting $\tilde{c} = \frac{n-1}{n} T$. By Theorem 4.2, since $\tilde{c} > \frac{n-2}{n-1} T$ this mechanism also has an optimal expected ratio $r_{WCE} = \frac{n-1}{n}$, among mechanisms with constant $\tilde{c}$. Just as with r_{WCE}, we can improve p_{EF} by using a random $\tilde{c}$.

We now examine mechanisms that randomise over the parameter $\tilde{c}$, starting with the `OPTMIX` mechanism of Guo *et al.* [2011], which we discussed in Subsection 4.5.1. Of the family of mechanisms that ignore an agent and assign it a random bid, this mechanism achieved the optimal r_{WCE},

Theorem 4.5. *`OPTMIX` has a worst-case $p_{EF} \leq \frac{1}{n+1}$.*

Proof. We can show this with an example value profile. Let $v = (\frac{T}{n} + \epsilon, \frac{T}{n}, \dots \frac{T}{n})$, where $0 < \epsilon < \frac{T}{n}$. The total of agents' values is $T + \epsilon$, thus it is efficient to purchase the good. However, for any ignored agent, the total observed utility is at most $T\frac{n-1}{n} + \epsilon < T$, meaning the mechanism with $\tilde{x} = T$ will never purchase. Thus, the mechanism will only purchase the good when it uses $\tilde{c} = 0$, which it does with probability $\frac{1}{n+1}$. This gives a $p_{EF} = \frac{1}{n+1}$ for this value profile, and thus an upper bound for this mechanism. □

Since the public good problem only has two possible outcomes, it is trivial to achieve a $p_{EF} = \frac{1}{2}$, outperforming `OPTMIX` on this measure. A mechanism simply chooses to purchase with probability $\frac{1}{2}$ without any payment or examining agents' types. This is trivially DSIC since no agent can affect the outcome, and since there are no payments it is SBB. Except for special cases when the total of agents' values for the public good is precisely T (making both outcomes efficient), this mechanism will always pick the efficient outcome with probability $\frac{1}{2}$. Unfortunately, it has a sub-optimal $r_{WCE} = \frac{1}{2}$. When it chooses the efficient outcome, it is guaranteed to achieve a ratio of 1 since payments are budget balanced, so $r_{WCE} \geq \frac{1}{2}$ for all value profiles. However, if all agents have a value of (near) zero, when the good is purchased, their total utility is zero, which gives a ratio of zero with probability $\frac{1}{2}$ and so $r_{WCE} = \frac{1}{2}$.

We can improve this non-worst-case performance by modifying `OPTMIX` to give the following mechanism, which we name `50-50MIX` :

- With probability $\frac{1}{2}$, run mechanism with $\tilde{x} = 0$ (i.e. always purchase)
- With probability $\frac{1}{2}$, run mechanism with $\tilde{x} = T$

This still has a worst-case $p_{EF} = \frac{1}{2}$, but it is possible to purchase with probability up to and including 1 for value profiles that are sufficiently high. The `50-50MIX` mechanism has a $r_{WCE} = \frac{1}{2}$ as well. Any SBB mechanism must have a $r_{WCE} \geq p_{EF}$, since an efficient outcome gives a ratio of 1, and that happens with probability p_{EF}. Thus, `50-50MIX` has $r_{WCE} \geq \frac{1}{2}$. However, in this case where all agents' values are zero for the public good, then with probability $\frac{1}{2}$, the mechanism will purchase the public good, giving a ratio of 0, otherwise giving a ratio of 1, for an overall $r_{WCE} = \frac{1}{2}$.

While this $p_{EF} = \frac{1}{2}$ of the `50-50MIX` mechanism is not necessarily the optimal mechanism according to worst-case p_{EF}, the upper bound on worst-case p_{EF} for this family of mechanisms approaches $\frac{1}{2}$ as the number of agents increases.

Theorem 4.6. *For the public good problem any mechanism that randomises over the choice of $\tilde{c}$ has a worst-case $p_{EF} \leq \frac{1}{2} + \frac{1}{4n-2}$.*

Proof. First, we consider the value profile $v = (T - \epsilon, 0, 0, \ldots, 0)$, for some infinitesimal $\epsilon > 0$. The efficient outcome is not to purchase, however, any mechanism with $\tilde{c} < T - \epsilon$ will purchase with probability $\frac{n-1}{n}$. Thus, for any mechanism that randomises over $\tilde{c}$ we have:

$$p_{EF} \leq Pr(\tilde{c} = T) + \frac{1}{n} Pr(\tilde{c} \in (0, T)) \tag{4.76}$$

Next we have the value profile $v = (\frac{T}{n} + \epsilon, \frac{T}{n}, \ldots \frac{T}{n})$, where the efficiency outcome is to purchase the good. However, any mechanism with $\tilde{c} > \frac{n-1}{n}$ will never purchase. Thus, for any mechanism that randomises over $\tilde{c}$ we have:

$$p_{EF} \leq Pr(\tilde{c} \leq \frac{n-1}{n} T) \tag{4.77}$$

If we let $p_T = Pr(\tilde{c} = T)$, then we have:

$$p_{EF} \leq p_T + \frac{1}{n}(1 - p_T) \tag{4.78}$$

$$p_{EF} \leq (1 - p_T) \tag{4.79}$$

	Probability of efficiency p_{EF}	Expected surplus ratio r_{WCE}
Upper bound	$\frac{1}{2} + \frac{1}{4n-2}$	$\frac{n}{n+1}$
Upper bound (constant $\tilde{c}$)	$\frac{1}{n}$	$\frac{n-1}{n}$
$\tilde{c} = \frac{n-1}{n}T$	$\frac{1}{n}$	$\frac{n-1}{n}$
`OPTMIX`	$\frac{1}{n+1}$	$\frac{n}{n+1}$
`50-50MIX`	$\frac{1}{2}$	$\frac{1}{2}$

Table 4.5: *The bounds on worst-case performance for the two efficiency measures as achieved by the* ***`RandomAuctioneer`*** *mechanisms on the public good problem. Mechanisms within this family are parameterised by $\tilde{c}$, which itself can be selected according to some distribution.*

Given $p_T \in [0, 1]$, the maximum value of p_{EF} satisfying the above inequalities is

$$p_{EF} = \frac{n}{2n-1} = \frac{1}{2} + \frac{1}{4n-2} \tag{4.80}$$

□

4.5.3 Summary of Public Good Mechanisms

While we do not explore the full space of DSIC and SBB stochastic mechanisms for the public good problem, we see that randomisation offers improvements over deterministic mechanisms for this problem. There exists no deterministic DSIC and SBB mechanism for the public good problem. By using the family of `RandomAuctioneer`-based mechanisms proposed by Guo *et al.* [2011], we are able to achieve a truthful and strongly budget balanced public good mechanism. The choice of optimal mechanism from this family depends on whether the mechanism designer wishes to maximise the expected social surplus ratio, r_{WCE}, or the probability of efficiency, p_{EF}, as summarised in Table 4.5.

4.6 Summary

In Chapter 3, we saw that the constraints of our social choice problems introduced in Chapter 2 led to poor worst-case behaviour. This motivated us to examine mechanisms in this chapter that use randomisation to improve the worst-case performance of these mechanisms. The worst-case performance is summarised in Table 4.6.

The measures of efficiency and envy-freeness we used for deterministic mechanisms do not adequately compare stochastic mechanisms. In Section 4.1 we introduce and discuss the measures we use to assess the mechanisms in this chapter. These measures are motivated by measures used to evaluate stochastic algorithms. When applied to deterministic mechanisms, these measures correspond to measures we used to assess purely deterministic mechanisms. These measures guide the mechanism designer in the selection of the optimal mechanism, and allow the comparison between deterministic and stochastic solutions.

We present and assess randomised mechanisms for each of our three social choice problems. These mechanisms are based on techniques of randomly ignoring a subset of agents, or randomly partitioning agents into disjoint groups. For heterogeneous item allocation, we use the characterisation of DSIC and SBB mechanisms satisfying independence, introduced in Subsection 3.2.4, to show that these new stochastic mechanisms are optimal among DSIC and SBB mechanisms in the class of affine maximisers. We conjecture that relaxing the independence constraint will not improve worst-case performance.

We examined similar mechanisms for the room assignment–rent division problem, with the goal of envy-freeness. Since stochastic affine-maximising DSIC mechanisms for RARD still suffer from poor worst-case performance, we examine mechanisms that are truthful in expectation. TIE is an alternative generalisation of truthfulness to randomised settings, and we saw better worst-case performance from these RARD mechanisms. We presented upper bounds on our randomised envy-freeness measures for truthful mechanisms. The TIE mechanisms we presented show that this truthfulness constraint can give improved worst-case performance over DSIC mechanisms.

Finally, we applied the technique of randomly ignoring agents to the public good problem. We discussed previous work that has provided optimal mechanisms within this family of stochastic mechanisms according to expected social surplus ratio. Building on this work, we assessed mechanisms from this family according to the probability of an efficient outcome. While we did not examine the full space of stochastic, truthful mechanisms for the public good problem, we demonstrated how stochastic mechanisms, just as in the other two social choice problems, can provide improved worst-case performance over deterministic mechanisms.

Heterogeneous Item allocation		
	Probability of efficiency p_{EF}	Expected surplus ratio r_{WCE}
`RandomAuctioneer` $n \leq m$	$\frac{1}{n}$	$\frac{n-1}{n}$
`RandomAuctioneer` $n > m$	$\frac{n-m}{n}$	$\frac{n-1}{n}$
Room Assignment–Rent Division		
	Probability of envy-freeness p_{NF}	Expected level of envy-freeness E_{NF}
DSIC Upper Bound	0	$[\frac{1}{n}, \frac{n-1}{n}]$
TIE Upper Bound	$\frac{1}{n}$	$[\frac{n-1}{n}, \frac{n-1}{n} + \frac{1}{n^2}]$
Simple RARD	$\frac{1}{n}$	$\frac{1}{n}$
`RandomAuctioneer`-based	0	$\frac{n-1}{n}$
2-Agent RARD	$\frac{1}{n}$	$\frac{n-1}{n} + \frac{1}{n^2}$
Public Good		
	Probability of efficiency p_{EF}	Expected surplus ratio r_{WCE}
Upper bound	$\frac{1}{2} + \frac{1}{4n-2}$	$\frac{n}{n+1}$
Upper bound (constant $\tilde{c}$)	$\frac{1}{n}$	$\frac{n-1}{n}$
$\tilde{c} = \frac{n-1}{n}T$	$\frac{1}{n}$	$\frac{n-1}{n}$
`OPTMIX`	$\frac{1}{n+1}$	$\frac{n}{n+1}$
`50-50MIX`	$\frac{1}{2}$	$\frac{1}{2}$

Table 4.6: *Summary of worst-case bounds for stochastic mechanisms on each of the social choice problems. Note that* ***`RandomAuctioneer`*** *achieves the upper bound on both measures for heterogeneous item allocation.*

Chapter 5

Empirical Results and Complexity

In this chapter, we provide empirical results to complement the theoretical worst-case results presented in the previous chapters. Firstly, we measure the average-case performance of various mechanisms according to randomly generated value profiles. Although the mechanism designer may not know the distribution of agents' types in advance, this analysis provides insight into the rarity of the worst-case behaviour. A mechanism with poor worst-case performance may perform well on all but an infinitesimal subset of value profiles, for example. For these empirical tests we use a uniform distribution over possible value profiles to obtain a more general result for this rarity of worst-case profiles. With no prior information about agents' types, a uniform distribution assumes each value profile is equally likely, while a non-uniform distribution gives results for some specific domain.

While in previous chapters we specified that strong budget balance was a hard constraint for our social choice problems, we also show, empirically, that this constraint can yield higher overall social welfare than mechanisms that enforce an efficiency constraint but allow budget imbalance. We argue that, for the goal of maximising utility, strong budget balance should be considered even if the setting does not require it.

Next we show the trade-off between efficiency and communication or computational complexity provided by the budget-balancing techniques of `IgnoreAgents`, introduced in Subsection 3.2.5.1, and `PartitionAgents`, introduced in Subsection 3.2.5.2. While these techniques were used to enable DSIC under SBB constraints, they reduce the information required by the mechanism to determine outcomes and payments. Since knowledge of the random choice does not affect the truthfulness guarantees in DSIC mechanisms, the mechanism can request less information from the set of agents, or from a subset of agents. This also reduces the size of the optimisation problem that the mechanism needs to perform in order to determine outcomes and payments. We show that the partitioning approach offers a flexible trade-off between efficiency and both communication and computational complexity.

5.1 Average Case Performance

A question raised by Cavallo [2006] and Gujar and Yadati [2011] asks whether it is worth considering sophisticated mechanisms with optimal *worst-case* performance when simple mechanisms perform well, or even better, in the average case. Worst-case analysis shows how a single, bad value profile performs, and may not give any indication of how the mechanism usually behaves. In work such as Faltings [2005], mechanisms are assessed by their average performance on some distribution of value profiles. An average case analysis requires some distribution over agents' types, such as a uniform distribution over some subset of the space of value profiles. In Chapter 4, where we provide worst-case bounds on the measures we used to compare randomised mechanisms, we assume such a distribution is not known in advance. If prior information over the distribution of value profiles is not known, worst-case results provide a guarantee that, regardless of the distribution of agent types, we will achieve at least this level of performance. This guarantee holds even against adversarially-chosen distributions. For example, a mechanism that performs very well according to an average case analysis on uniformly distributed value profiles may perform very poorly on highly correlated value profiles.

Nevertheless, an average case analysis does provide useful information in addition to worst-case results by showing the "rarity" of value profiles that cause the mechanism to hit its worst-case bound. Average case results can reveal whether the worst-case performance occurs with most value profiles, or if it only occurs in some rare, special circumstances. Further, mechanisms that have optimal worst-case performance may under-perform other mechanisms for most value profiles. We show an example of this in the RARD problem, where the worst-case optimal mechanism always achieves its worst-case level of performance; however, another mechanism with a lower worst-case guarantee generally performs much better, even though in some rare cases it has poor performance.

To demonstrate and compare the average case performance of our mechanisms empirically, we generate random value profiles from uniform distributions over a particular subset and assess the average probability of efficiency and average expected social surplus ratio (or, for RARD, average p_{NF} and E_{NF}). Since these results are not for a specific setting, a uniform distribution is the most appropriate to explore the average behaviour of a mechanism over the full space of value profiles. To clarify how we aggregate the results of these empirical tests, for a particular mechanism and problem we generate a set of random value profiles, drawn uniformly and independently from a subset of V. For each value profile in this randomly chosen set, we calculate that value profile's p_{EF} and r_{WCE} (or p_{NF} and E_{NF}). After calculating for every value profile in the generated set, we calculate the average for each of these measures.

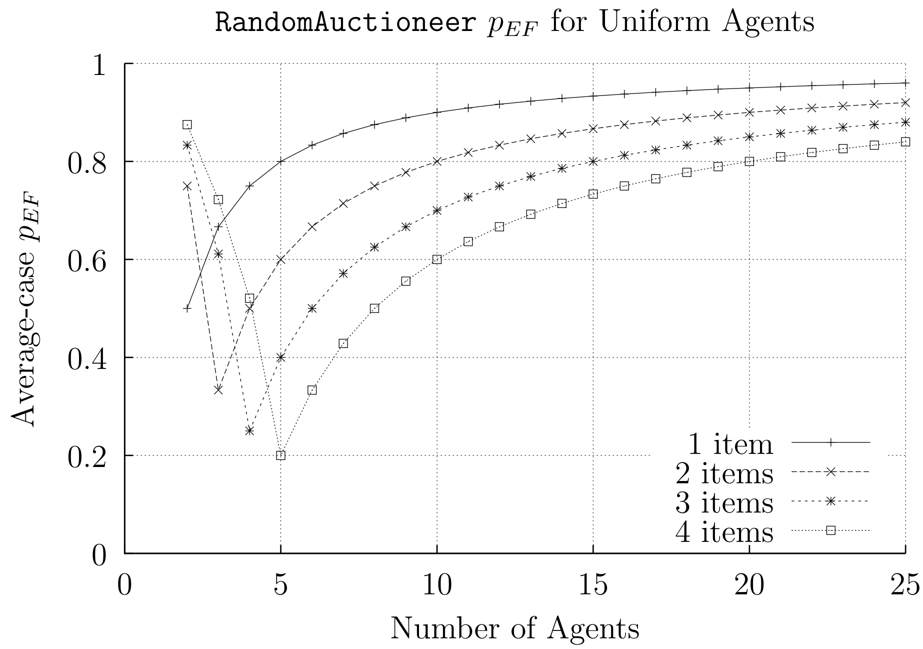
We should point out that these mechanisms were designed for optimal guaranteed behaviour with no prior knowledge over the distribution of agents' types. If such a distribution is known, we can leverage this information to improve on these guarantees. For example, consider the `RandomAuctioneer` mechanism that randomly ignores a single agent; if we know in advance that agent i typically has the lowest values for all items, by ignoring i with a higher probability we get a higher average-case expected social surplus ratio.

5.1.1 Heterogeneous Item Allocation

As shown in Section 4.3, the optimal worst-case performance for both p_{EF} and r_{WCE} under our constraints was achieved by the `RandomAuctioneer` mechanism. This randomly selects a single agent whose reported type is ignored, and this agent receives the surplus generated by a VCG mechanism conducted by the other agents. We demonstrate in this section the average case performance for increasing numbers of agents or items, where agents' types are drawn from a uniform distribution. In particular, we show that the average case p_{EF} and r_{WCE} converge to 1. This converges much faster than the worst-case bound, with the exception of p_{EF} when $n \geq m+1$. This convergence to 1 shows that, for increasing problem sizes, most value profiles will lead to a near-optimal solution.

For our empirical tests, we have a uniform distribution over all value profiles having values in the range $v_i(j) \in [0,1]$. All agents' values for items are generated independently. Given that we don't have a unit for agents' values, limiting all values to the range $[0,1]$ is arbitrary, and will give the same average case performance regardless of the (strictly positive) upper bound on the range.

The results in Figure 5.1 show the average case p_{EF} for various combinations of n and m. For each trial (i.e. fixed n and m), we averaged the performance using 10 million randomly



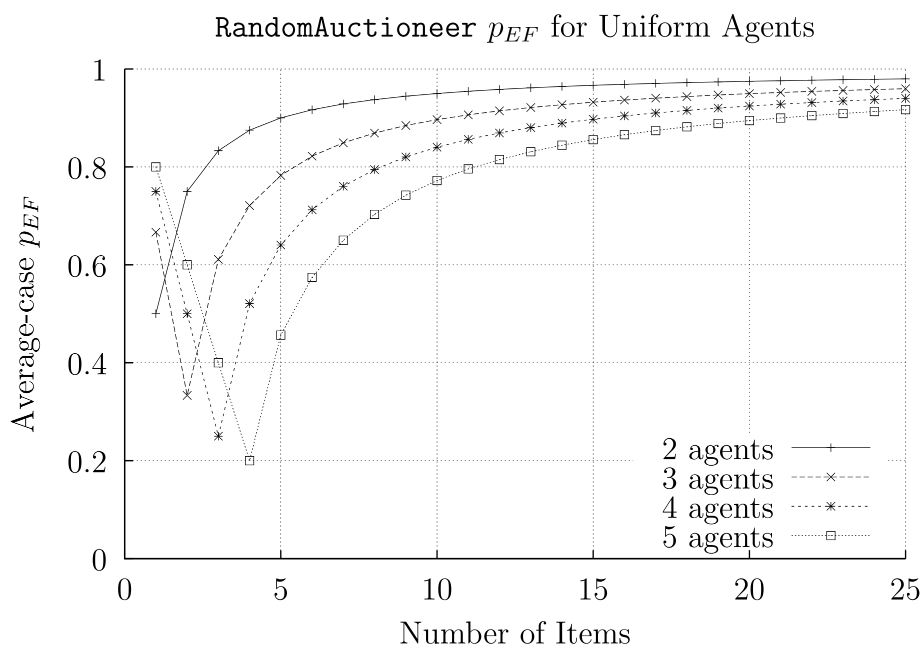


Figure 5.1: *Average case probability of efficiency for the* ***`RandomAuctioneer`*** *mechanism for heterogeneous item allocation with an increasing number of agents or items.*

generated value profiles. We observed that when $n \geq m+1$, the probability of an efficient allocation in all generated value profiles matched the bound of $p_{EF} = \frac{n-m}{n}$, as presented in Table 4.2. As is visible in Figure 5.1, when $n = m+1$ we get a minimum $p_{EF} = \frac{1}{n}$, for any fixed n or m. This is also the minimum worst-case performance, according to Equation (4.16). In these settings, where we have no surplus items for the ignored agent, the worst-case p_{EF} is what is likely to be seen for any set of agents, regardless of the distribution of agents' types. Analysing this result, we see that this is unsurprising. To improve on the worst-case bound when $n > m$, at least one agent i who is unallocated in the efficient allocation must have the exact same utility for the item that agent j receives in the efficient allocation. Otherwise, whenever an agent with an item in the efficient allocation is ignored, it will be replaced by an agent with a strictly lower value. Since we generate values from an effectively continuous domain, the probability of such a value profile is zero.

When we have $n \leq m$, on the other hand, the average p_{EF} was much higher than the bound of $\frac{1}{n}$, approaching 1 as the number of items increases. This is in direct contrast to the worst-case bound, which is constant as the number of items increases. In every set of trials with $n \leq m$, we observed value profiles that both matched the bound, as well as those that had $p_{EF} = 1$. Thus we see that while worst-case value profiles still occur, they become less likely as the number of items increases. The distribution of agents' types will have a much larger influence on average case p_{EF} when $n \leq m$ compared to when $n > m$.

Next, in Figure 5.2 we show the average case r_{WCE} for the `RandomAuctioneer` mechanism. These results are from the same set of value profiles used to generate the plots in Figure 5.1, so again we have 10 million trials for each value of n and m. Here we see that the average expected social surplus ratio rapidly approaches 1 as the number of agents or items increases. In all trials, the average expected-ratio exceeded the worst-case bound of

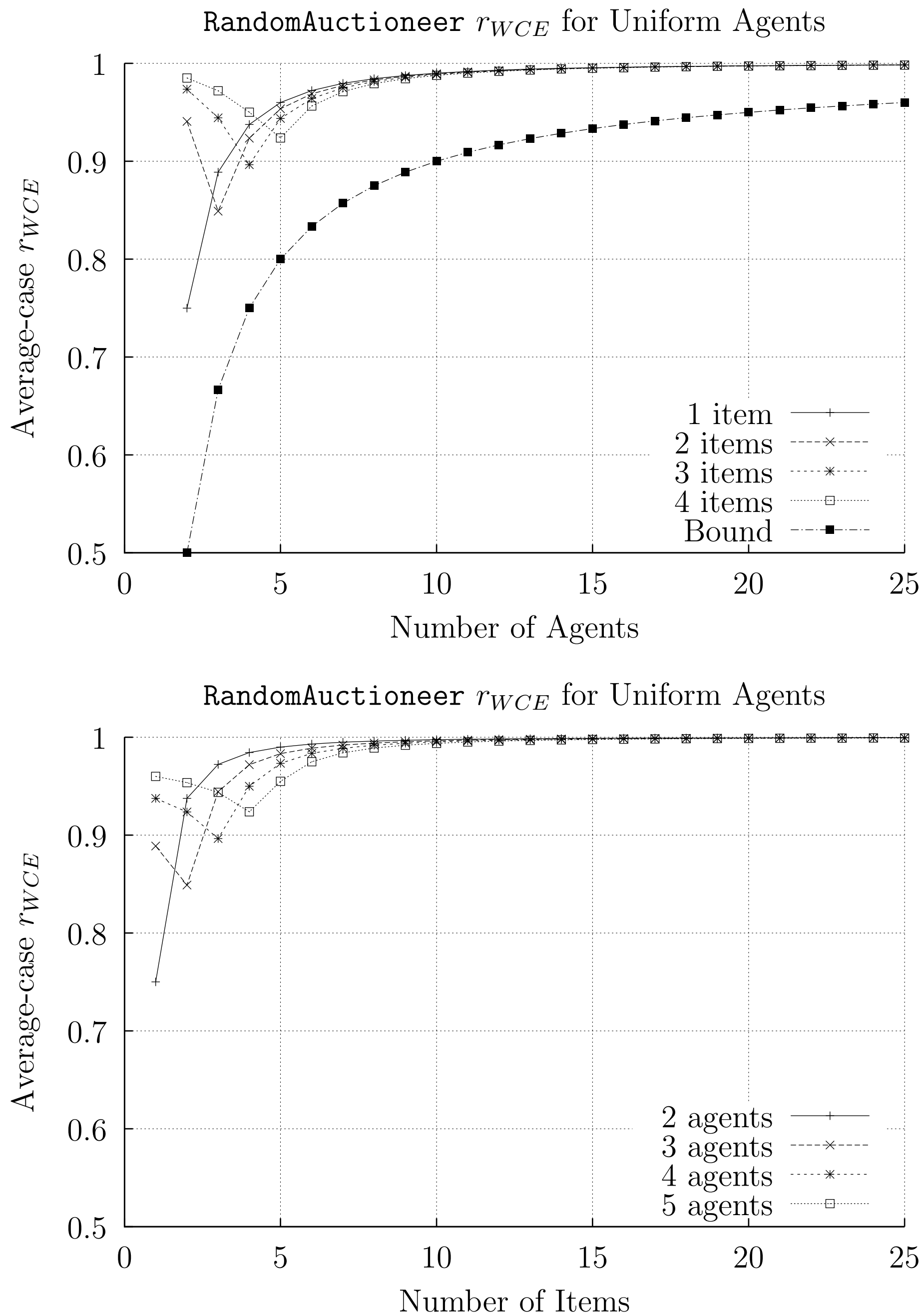


Figure 5.2: *Average case expected social surplus ratio for the* `RandomAuctioneer` *mechanism for heterogeneous item allocation with an increasing number of agents or items.*

$\frac{n-1}{n}$, as shown in Equation (4.18). However, we only observed value profiles that gave a $r_{WCE} = 1$ when $n \leq m$. As with p_{EF}, the lowest average r_{WCE} for a fixed n or fixed m occurs when $n = m + 1$, but unlike p_{EF}, this increases as n or m increase. From these results we can see that profiles giving the worst-case performance are uncommon, so the average case r_{WCE} can be much higher, depending on the distribution of agents' types.

5.1.2 Room Assignment–Rent Division

For the room assignment–rent division problem, we presented two randomised, truthful mechanisms that work with any number of agents $n \geq 2$. These mechanisms are the p_{NF} maximising mechanism in Example 4.4 of Subsection 4.4.2, and the `RandomAuctioneer`-based mechanism presented in Subsection 4.4.4.2. We tested both these mechanisms empirically to determine the average case p_{NF} and E_{NF}, using value profiles generated with uniform probability. That is, for any agent i and item j, $v_i(j)$ is drawn uniformly from the range $[0, 1]$.

In Figure 5.3, we look at the p_{NF} for RARD mechanisms. Although the mechanism based on `RandomAuctioneer` has a worst-case $p_{NF} = 0$, we see that the average-case p_{NF} stays higher than even the *optimal* worst-case bound of $p_{NF} = \frac{1}{n}$. Both this average-case and worst-case approach zero as the number of agents increase. Our trials on the p_{NF} maximising mechanism, which has the optimal worst-case p_{NF} within the class of mechanisms we explored, had an average-case that matched the worst-case bound (shown as Upper bound in Figure 5.3), since every value profile we tested had the same $p_{NF} = \frac{1}{n}$. This shows that while the p_{NF}-maximising mechanism has a higher worst-case bound, its average-case performance matches this bound, so if agents are drawn from a sufficiently diverse distribution (such as our uniform distribution), the `RandomAuctioneer` approach

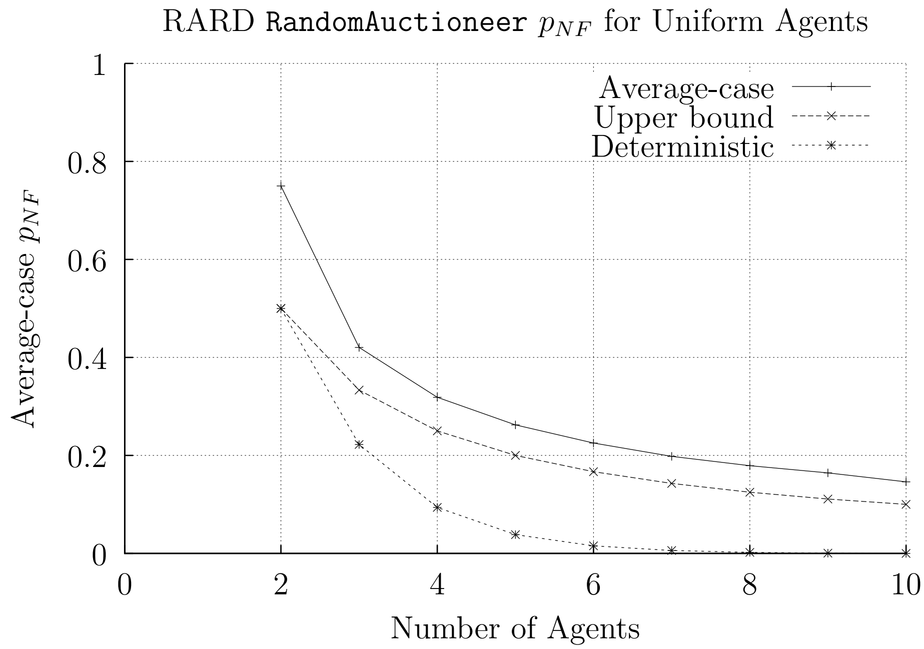


Figure 5.3: *Average case p_{NF} for a* `RandomAuctioneer` *RARD mechanism, and worst-case upper bound. Also shown is the probability of deterministic envy-freeness, where value profile gives an envy-free outcome regardless of the random choice.*

will outperform it in the average case.

Also shown in Figure 5.3 is the fraction of value profiles that give deterministic envy-freeness under `RandomAuctioneer`. That is, regardless of the random choice that determines which agent becomes the auctioneer, all agents will be envy-free. We found that this probability of deterministic envy-freeness for value profiles generated with uniform probability, while non-zero, rapidly approaches zero with additional agents.

The mechanism described in Subsection 4.4.4.2, which attempts to maximise the worst-case E_{NF}, does not meet the conjectural upper bound for either E_{NF}[1] or p_{NF}. While $(n-1)$

[1] The bound for E_{NF} is not tight, however.

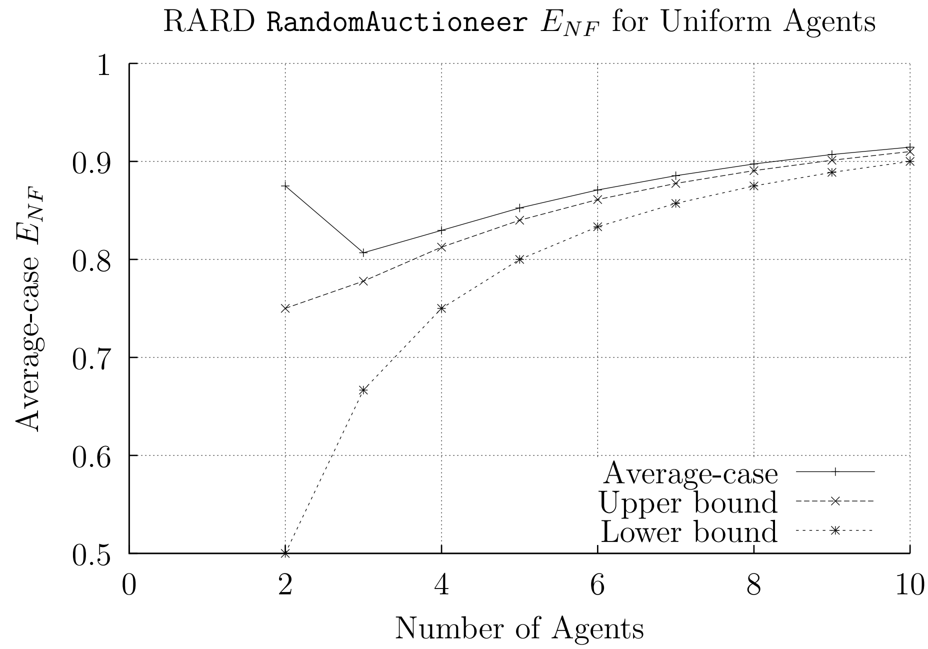


Figure 5.4: *Average case E_{NF} for a `RandomAuctioneer` RARD mechanism, along with the worst-case performance for this mechanism (lower bound) and the upper bound on E_{NF}.*

agents are guaranteed to be envy-free, the ignored agent may be envious for all choices of ignored agent. We tested both of our stochastic RARD mechanisms using the same value profiles for the p_{NF} tests. Firstly, the p_{NF}-maximising mechanism consistently gave $E_{NF} = \frac{1}{n}$ for all value profiles tests, matching its worst-case bound, so it is greatly outperformed both in worst- and average-case performance by `RandomAuctioneer`. In Figure 5.4, we compare the average-case E_{NF} of `RandomAuctioneer` to its worst-case bound (lower bound), and the upper bound on worst-case performance provided by Conjecture 4.2. All three measures approach one with increased agents, and we see that the average-case performance exceeds the upper bound on worst-case performance. Although the mechanism

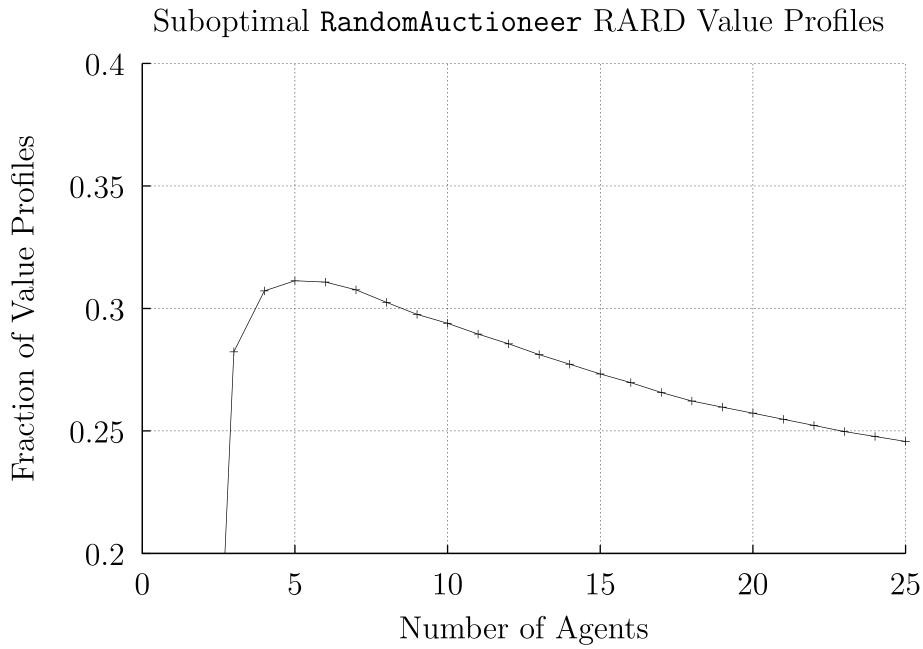


Figure 5.5: *Fraction of randomly generated value profiles that give a p_{NF} (or E_{NF}) less than the worst-case upper bound in* ***`RandomAuctioneer`****.*

performs sub-optimally in the worst-case, value profiles that lead to the suboptimal behaviour may not be common.

Since, for values of $n > 2$, the `RandomAuctioneer` mechanism has a worst-case bound below the conjectured upper bounds for both p_{NF} and E_{NF}, we examined the fraction of value profiles such that $p_{NF} < \frac{1}{n}$ and $E_{NF} < \frac{n-1}{n} + \frac{1}{n^2}$. Since the only agent that can be envious is the auctioneer agent, the only value profiles that satisfy these conditions (and thus fall below the upper bound) are those where any random auctioneer will be envious. When $n = 2$, all value profiles meet or exceeded the upper bound for both p_{NF} and E_{NF}. When $n > 2$, the fraction of randomly generated profiles falling below the upper bound (having the auctioneer always envious) is consistently below $\frac{1}{3}$, and decreases as n increases, which we show in Figure 5.5. This suggests that the majority of value profiles lead to a performance that falls within the optimal bound for these two measures. The fraction of value profiles that do achieve the bound, increases with additional agents. However, there is still a significant fraction of profiles for which this mechanism falls short of this bound. Thus, these worst-case profiles are not rare, special cases so there is potential improvement if the upper bound is tight.

5.1.3 Public Good

In our average-case tests of public good mechanisms, we examined average p_{EF} and r_{WCE} for mechanisms with fixed $\tilde{c} \in \{0, \frac{n-1}{n}T, T\}$, along with the `OPTMIX` and `50-50MIX` mechanisms. We found that while randomising over the value of $\tilde{c}$ helps the worst-case performance (see Section 4.5), average case performance is maximised with a constant $\tilde{c}$. For these tests, we use value profiles drawn from a uniform distribution with one of two possible ranges. First, the range $v_i \in [0, \frac{2T}{n}]$, which means that with probability 0.5, the efficient outcome will be to purchase the good[2]. Next, to sample a set of agents that are more likely than not to want the public good, we draw agents' values uniformly from the range $v_i \in [0, \frac{2T}{n-1}]$. Since the mechanisms are independent of the scale of agents' values, for simplicity and without loss of generality we set $T = 1$. For each value of n we generated 10 million random value profiles to calculate the average performance.

When we measure p_{EF}, the only mechanism with a non-zero worst case p_{EF} for constant $\tilde{c}$ has $\tilde{c} = \frac{n-1}{n}T$, while the `50-50MIX` mechanism had the highest worst-case p_{EF} of the mechanisms we considered. As shown in Figure 5.6, when looking at average case per-

[2]The sum of n independent, random values, each with a uniform distribution between 0 and c, follows an Irwin-Hall distribution[Irwin, 1927; Hall, 1927], which has a median value of $\frac{n}{2}c$.

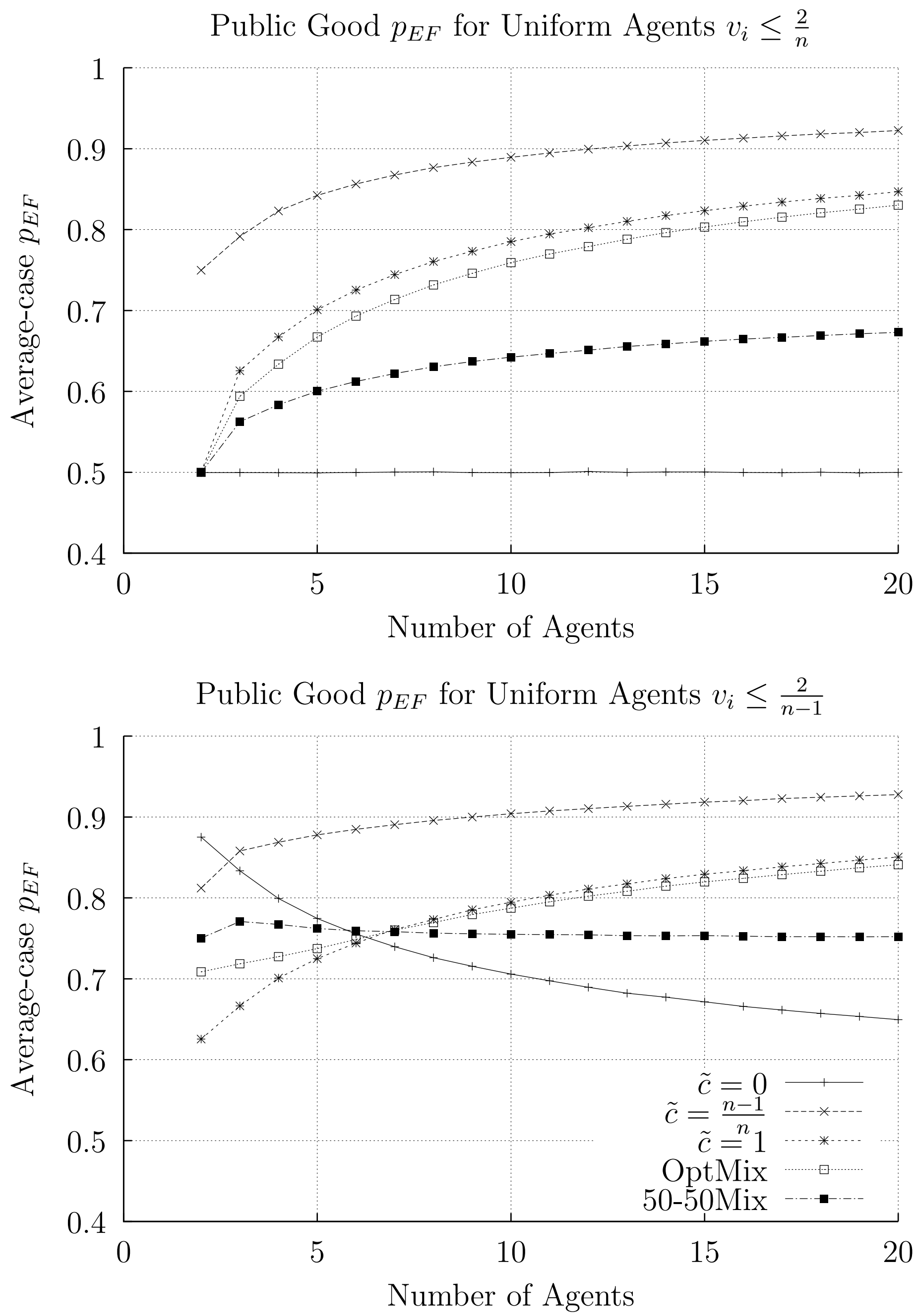


Figure 5.6: *Average-case probability of efficiency for the public good problem, with* $T = 1$.

formance, the mechanisms `OPTMIX` and `50-50MIX` have a performance that lies between the performance of the two mechanisms they randomise over, namely $\tilde{c} = 0$ and $\tilde{c} = 1$. Indeed, any mechanism that randomises over the value of $\tilde{c}$ will have an average-case performance that is a probability-weighted sum of the average performance of the underlying constant-$\tilde{c}$ mechanisms. Since the benefits of randomising over $\tilde{c}$ disappear in the average-case, the best average-case performance comes from a mechanism with constant $\tilde{c}$. Figure 5.6 shows that this is typically $\tilde{c} = \frac{n-1}{n}T$, unless there is a high probability of the efficient outcome being to purchase. Note that since $\tilde{c} = 0$ always builds, the p_{EF} for this mechanism reveals the probability that the particular distribution of agents has an efficient outcome where the good is purchased.

For average-case r_{WCE}, shown in Figure 5.7, again we see that those mechanisms with a constant $\tilde{c}$ outperform those that randomise over $\tilde{c}$. The ratio quickly approaches 1 as the number of agents increases, so with a large enough set of agents, there is a very minor loss of efficiency due to randomisation. As we show in Section 5.2, this loss of efficiency is often more than compensated by the lack of budget imbalance.

5.2 Efficiency versus Strong Budget Balance

In this work, we have focussed on mechanisms which have strong budget balance as a constraint. This constraint may be due to actual restrictions where surplus payments cannot be collected or destroyed. Alternatively, strong budget balance can be enforced to prevent utility loss from the set of agents. As we showed in Chapter 3, DSIC mechanisms cannot be simultaneously efficient and SBB, so a mechanism designer must choose between these properties. While an efficient allocation maximises the collective utility of the agents for the allocation alone, the payments made by the agents to the mechanism subtract from

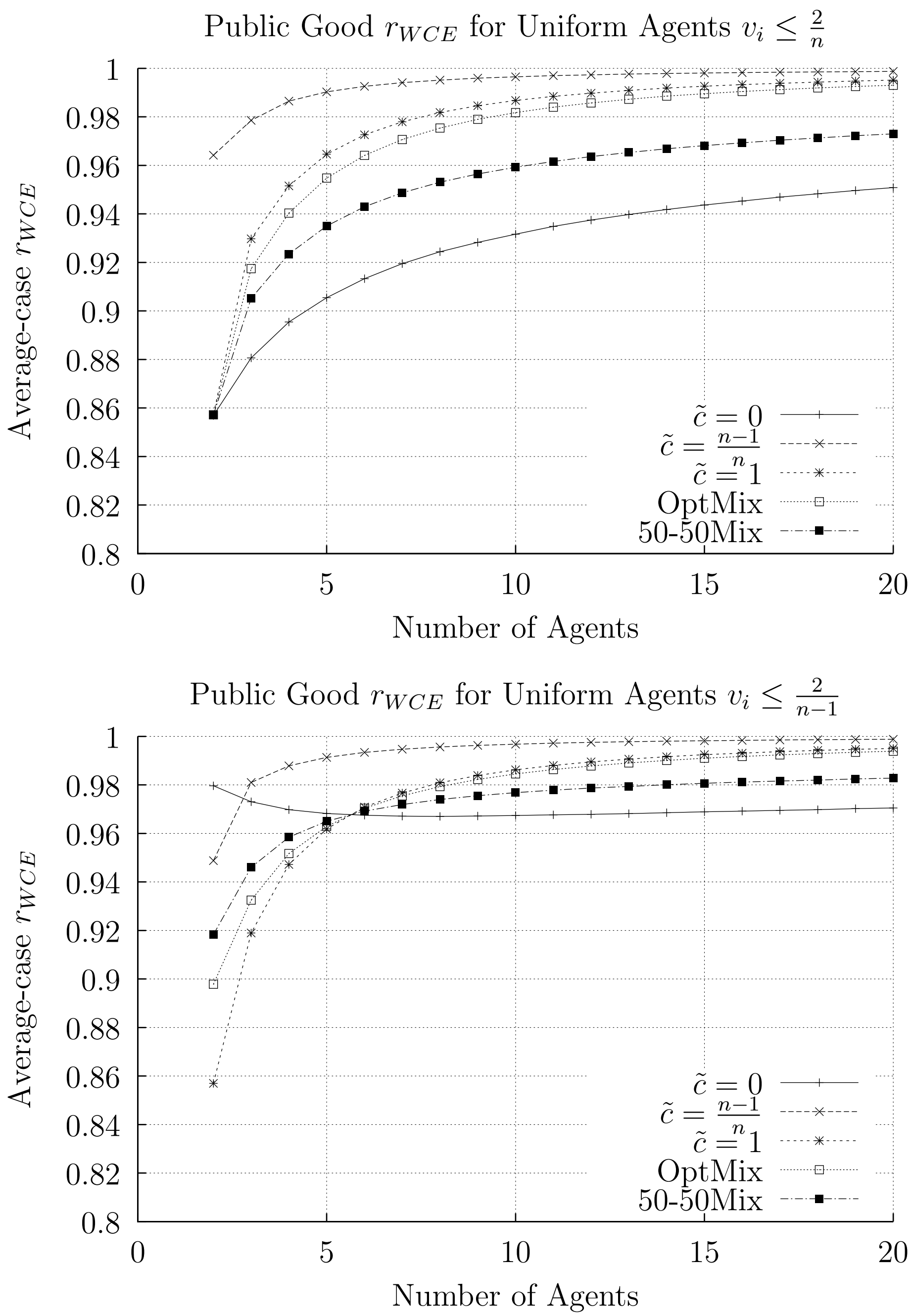


Figure 5.7: *Average-case expected ratio for the public good problem, with $T = 1$.*

this collective utility. In this section, we present empirical results comparing the expected ratio of our SBB mechanisms to the VCG mechanism, which is perhaps the most well known efficient and DSIC mechanism. For heterogeneous item allocation with $n \leq m+1$, VCG provides the optimal worst-case ratio (zero) for DSIC, efficient mechanisms [Gujar and Yadati, 2011; Guo, 2012], while redistribution mechanisms offer improvements when $n > m+1$.

Figure 5.8 illustrates the comparison between `RandomAuctioneer` and VCG on the heterogeneous item allocation problem with a fixed number of items ($m = 4$) or fixed number of agents ($n = 4$). Since VCG is always efficient ($p_{EF} = 1$) we compare based on r_{WCE}. We found that in all settings, `RandomAuctioneer` provided a higher average r_{WCE} compared to VCG. When applied to a deterministic mechanism, r_{WCE} is the normal social surplus ratio. However, this difference becomes much more pronounced as the number of agents increases relative to the number of items. With additional agents, the payments made under the VCG approach the value of the allocated items, on average, which lowers the overall ratio. Under `RandomAuctioneer`, with additional agents, those agents who should be allocated in the efficient allocation are less likely to be ignored, and when they are, it is more likely that the chosen allocation has a close value to the efficient allocation.

We found a similar, but less pronounced result for the public good problem. In the top plot of Figure 5.9, we use a uniform distribution, as above, and this shows the average ratio converges to 1 much faster using a randomised SBB mechanism compared to VCG. While in all other tests we use a uniform distribution of agents' types, we perform a test with the public good problem on a non-uniform distribution to demonstrate the higher worst-case bound of the stochastic mechanisms. If agents' values come from a distribution where a single agent tends to have most (in our tests, 95%) of the total utility, then the ratio of VCG decreases with additional agents while, even in worst-case performance, the

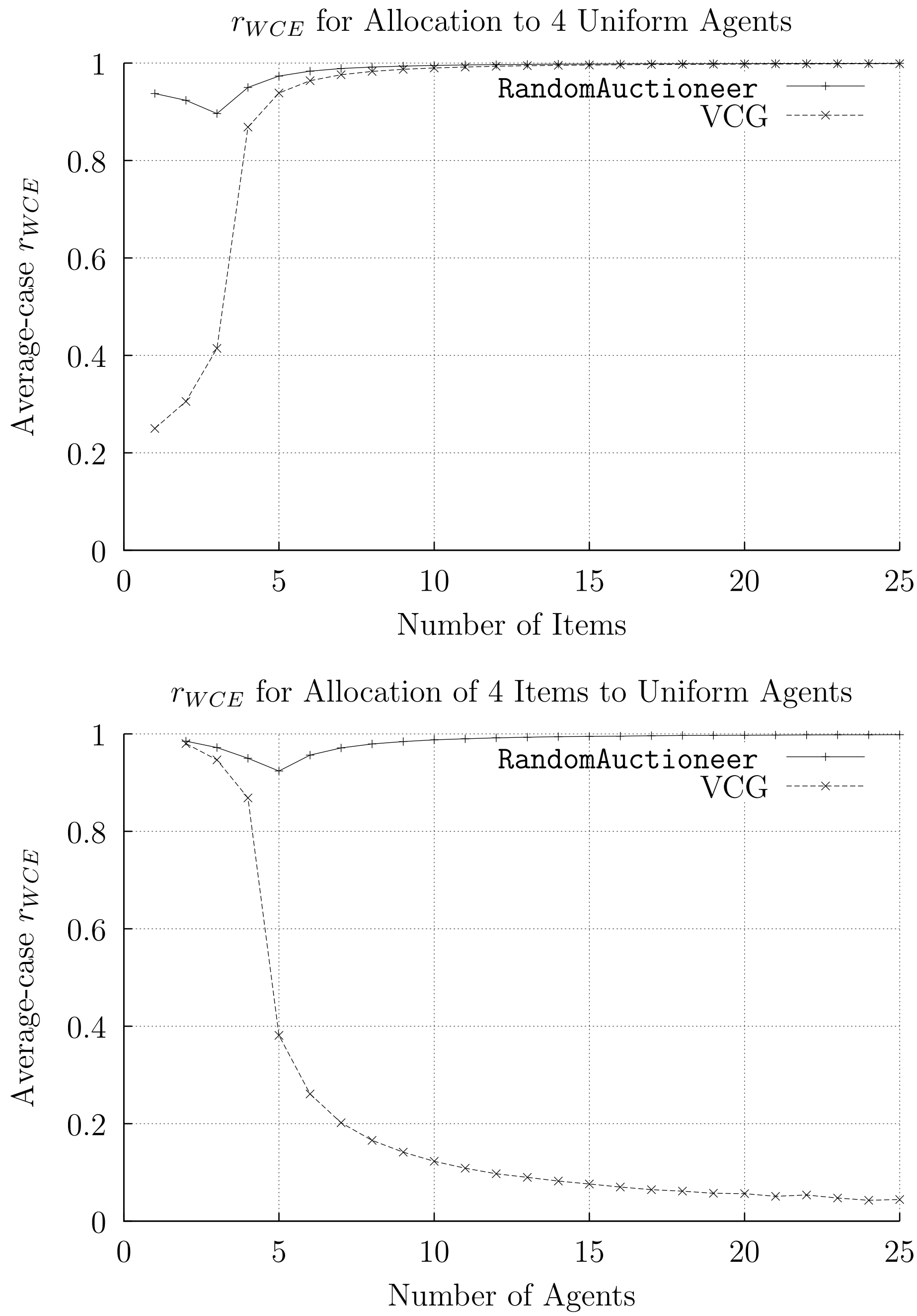


Figure 5.8: *Comparison of average-case r_{WCE} between the strongly budget balanced, but inefficient* ***`RandomAuctioneer`*** *mechanism and the efficient VCG mechanism.*

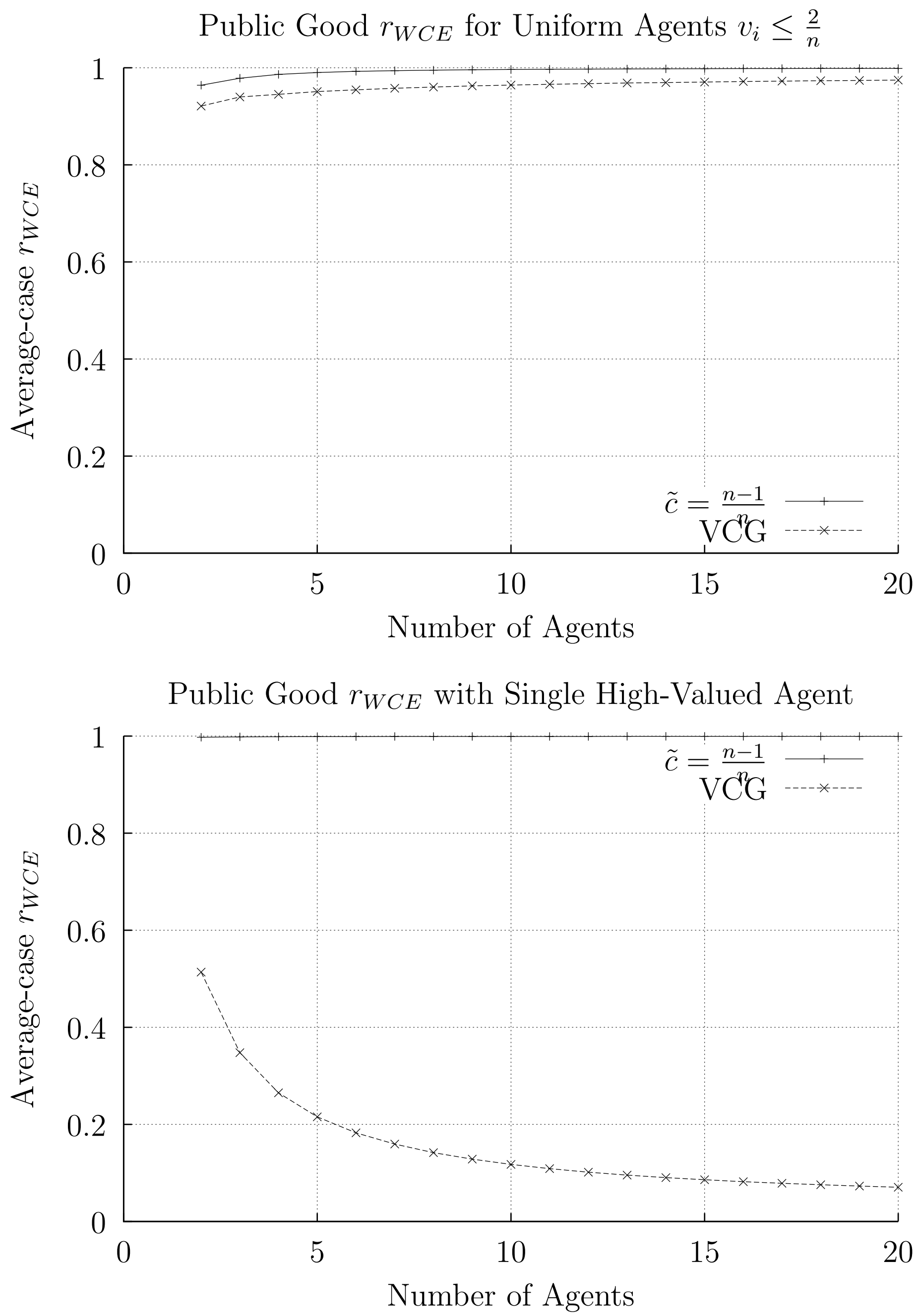


Figure 5.9: *Comparison of average-case r_{WCE} between the strongly budget balanced Public Good mechanism with $\tilde{c} = \frac{n-1}{n}$, and the efficient VCG mechanism.*

randomised mechanism with $\tilde{c} = \frac{n-1}{n}$ has a ratio approaching 1. For these tests, shown in bottom plot of Figure 5.9, a value profile is generated such that all agents' values are from a uniformly distribution as before, then agents' values are scaled by positive factors such that 95% of the total utility is attributed to a single agent. The median total utility for this distribution remains at T.

The results in this section demonstrate that sacrificing a guarantee of allocative efficiency does not go against the goal of maximising overall agent utility. Intuitively, it may seem that the best way to achieve the highest overall agent utility is to choose a mechanism that always picks the outcome giving the highest overall agent utility (i.e. a Groves mechanism). However, the payments required for both DSIC and allocative efficiency can result in a large utility loss for the group of agents. Thus, an inefficient mechanism, perhaps surprisingly, can give a better overall utility than an efficient mechanism.

5.3 Communication Complexity

A potentially beneficial side effect of randomisation is a reduction in the communication and computational complexity of the mechanism. If an agent is ignored, the mechanism does not need this agent to report its type in order to determine the allocation. Under item allocation with surplus items, the ignored agent need only report its value for the remaining items, which will be strictly less than the m values it would have to report in a Groves mechanism. Similarly, with a partitioning approach, the mechanism only requires an agent to report its value for items in its own partition. Agents may be able to determine which random choice was made (i.e. which agents were ignored or which partitions were chosen) by observing the queries the mechanism makes. However, since we use DSIC mechanisms, revealing the random choice will not harm the truthfulness of the mechanism. In this section, to demonstrate this potential benefit, we show the trade-off between high r_{WCE} and low communication complexity that the partitioning technique facilitates.

Consider a mechanism that allocates m heterogeneous items to n unit-demand agents. Each agent's type consists of m values, one for each item. Thus, a Groves or VCG mechanism would require nm values be communicated in order to calculate the allocation and payments. Ignoring k agents can linearly reduce the number of values that need to be communicated to the centre, since only a subset of the n agents need to report their m values for each item. If $n - k \geq m$, the mechanism only needs $(n - k)m$ values to be reported, since there are no surplus items left over for the ignored agents to choose from.

Partitioning agents, on the other hand, requires each agent to report its value only for items within its own partition. For a partitioning mechanism with $k \geq 2$ partitions, where partition q contains n_q agents and m_q items, the number of values that need to be communicated to the mechanism in order to determine the allocation and payments is:

$$\prod_{q=1}^{k}(n_q m_q) \tag{5.1}$$

Given that a Groves mechanism requires the full value profile, consisting of nm values, the fraction of values that need to be communicated in a partitioning mechanism is:

$$\frac{1}{nm}\sum_{q=1}^{k} n_q m_q \tag{5.2}$$

Observe that this precisely matches the upper bound on r_{WCE} for partitioning mechanisms presented in Equation (4.34) of Subsection 4.3.4.2. Thus, using partitioning to reduce the communication complexity by a factor of α places an upper bound on the worst-case r_{WCE} of α.

If we have n and m both divisible by k, then having k equal-sized partitions means the number of values the mechanism requires, according to Equation (5.1), is:

$$\prod_{q=1}^{k} \frac{n}{k}\frac{m}{k} = \frac{nm}{k} \tag{5.3}$$

That is, the mechanism requires $\frac{1}{k}$ of the complete value profile, as needed by a Groves mechanism. While the number of values is slightly higher when n or m is not divisible by k, the difference decreases as n, m or k increase.

From Equations (4.3.4.2) and (5.1) we know that increasing the number of partitions decrease communication complexity but at the cost of reduced worst-case efficiency. To demonstrate the trade-off between communication complexity and average-case efficiency, we performed empirical tests on heterogeneous item allocation under increasing numbers of partitions. For these tests we examined the case where $n = m$ and partitioned agents as evenly as possible. That is, each partition contains between $\lfloor \frac{n}{k} \rfloor$ and $\lceil \frac{n}{k} \rceil$ agents and items. By way of comparison, we also generated the average-case r_{WCE} for both the `RandomAuctioneer` and VCG mechanisms.

In Figure 5.10 we show the average-case r_{WCE} using between 2 and 5 partitions, compared to the `RandomAuctioneer` and VCG mechanisms. Unsurprisingly, with additional partitions, the average-case ratio decreases, but not as dramatically as the worst-case bound would suggest. Using 4 partitions, for example, does not have half the average ratio as compared to using 2 partitions. The loss of r_{WCE} with additional partitions also decreases as the problem size increases, while larger problem sizes would see a larger benefit from reduced communication complexity. With large enough problem size, the partitioning approaches even exceed the average performance of the VCG mechanism due to the budget imbalance of VCG. However, due to the inefficiency of partitioning, it is always outperformed by the SBB mechanism `RandomAuctioneer`. This shows that if communication is costly, and a concern of the mechanism designer, than partitioning offers a flexible trade-off

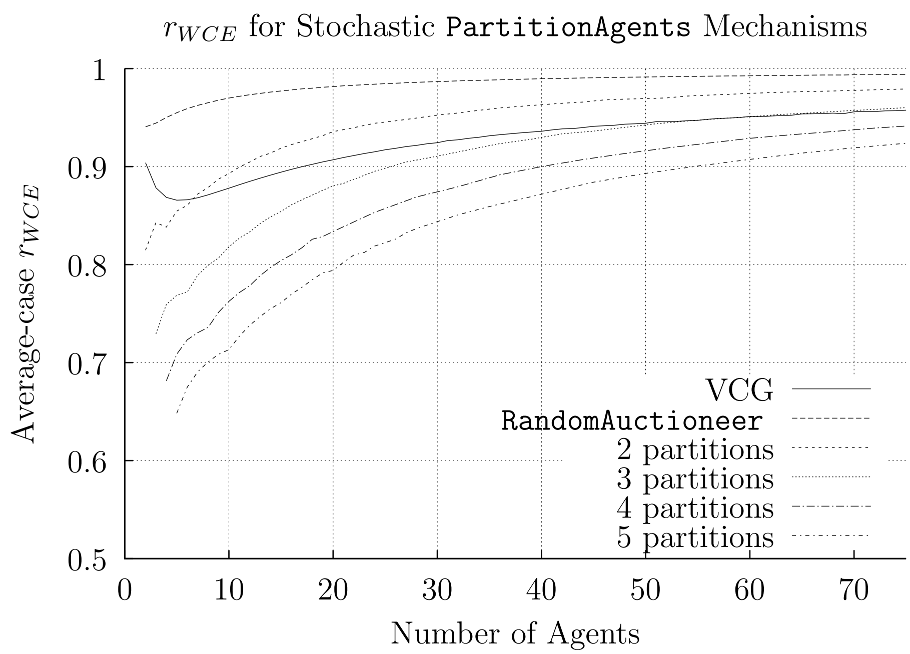


Figure 5.10: *Average-case r_{WCE} for partitioning mechanisms, with different numbers of partitions, for the heterogeneous item allocation problem. These are compared to the `RandomAuctioneer` and VCG mechanisms.*

whereby partitions can lower the communication cost at the expense of social surplus ratio. Also, if communication is slow or unreliable, a partitioning approach allows a mechanism to iteratively use fewer and fewer partitions as it receives more values. With many partitions initially, the mechanism does not require many (or any if $k = \min(m+1, n)$) reported values. This allows a preliminary solution to be generated before the complete value profile is received. However, as more values are received, the mechanism can use fewer partitions to achieve a higher r_{WCE}.

This tradeoff is related to work on expressiveness in mechanisms presented by Benisch *et al.* [2008] and more recent work [Dütting *et al.*, 2011a; Dütting *et al.*, 2011b; Caragiannis *et al.*, 2011; Conitzer and Sandholm, 2012]. The expressiveness of a mechanism captures how well an agent's reported type can influence the outcome chosen by a mechanism that seeks to find a efficient outcome. In settings such as combinatorial auctions, an agent's type assigns a value for all possible combinations of items. The information requested by a mechanism that is not fully expressive may prevent an agent from being able to report a difference in utility between two possible combinations. For example, if our heterogeneous item allocation mechanisms were used for a combinatorial auction setting, since agents can only report values for individual items, they are unable to express any synergistic effects of receiving multiple items. Work on expressiveness defines formal measures of expressiveness in various settings. Similar to the work we presented in this section, previous work has shown a tradeoff between the expressiveness of a mechanism and its efficiency. Unlike (expected) social surplus ratio, this efficiency does not consider payments made by agents, so loss of efficiency is measured by the loss of utility due to inefficient allocation and not budget imbalance. Also, we measure the communication requirement in terms of the number of values requested by the mechanism. As noted by Benisch *et al.* [2008], this measurement may not be appropriate for a purely theoretical analysis, since Cantor [1874] showed there exists a one-to-one mapping from $\mathbb{R}^n$ to $\mathbb{R}$, and so, theoretically, all multi-dimensional types can be expressed as a single real number. From a practical viewpoint, our measure captures the number of values that need to be transmitted to the mechanism. The amount of information transmitted will be directly proportional to the number of values, and the information used to encode each value.

5.4 Computational Complexity

Just as partitioning can lower communication complexity with an increased number of partitions (see Section 5.3), the same scheme can be used to flexibly improve the computational complexity of a mechanism. For the allocation of heterogeneous items, a DSIC mechanism must, in general, solve an affine maximisation to determine the appropriate assignment of items to agents. The payment functions may require additional assignment calculations, such as the VCG mechanism and our stochastic mechanisms. These payment functions calculate the value of allocations on subsets of agents.

Finding the assignment of m items to n agents can be solved in polynomial time using algorithms such as the Hungarian method [Kuhn, 1955; Edmonds and Karp, 1972; Bertsekas, 1981]. The Hungarian method can solve the required affine maximisation to determine the allocation in $O(\max(n,m)^3)$. While more efficient techniques may exist, the complexity will be at least $\Theta(nm)$, since this is the number of agents' values.

By partitioning (or ignoring) agents, we reduce the size of the allocations that need to be calculated, and thus reduce the overall complexity. We consider the heterogeneous item allocation setting with $n = m$, as we examined in Section 5.3. Using the Hungarian algorithm to find an allocation for this setting has a run-time of $O(n^3)$. However, if we divide agents up into k equal-sized partitions, each partition has $\frac{n}{k}$ agents and $\frac{m}{k}$ items.

Finding the allocation for all k partitions using the Hungarian method has a run-time of:

$$kO((\frac{n}{k})^3) = O(\frac{n^3}{k^2}) \tag{5.4}$$

This can potentially reduce the run-time of the allocation to a factor of $\frac{1}{k^2}$ of the run-time for finding an efficient allocation. Even if an allocation algorithm had run time of $\Theta(nm)$, the run-time would be reduced to $\frac{1}{k}$ of the time required to find an efficient allocation.

While additional partitions result in lower r_{WCE}, the average-case, as illustrated in Figure 5.10, suffers a smaller loss as the problem size increases. Just as with communication complexity, if computational complexity is a significant concern to the mechanism designer, especially for large problem sizes, then partitioning offers a flexible trade-off between high allocative efficiency and low computational complexity. It also allows for solutions with lower r_{WCE} to be calculated quickly, before the mechanism finds an allocation with fewer partitions. This allows a mechanism to produce a DSIC, SBB allocation at any time, but with improved r_{WCE} the more time it is allowed to run. Such an "anytime mechanism" is closely related to **anytime algorithms** [Zilberstein and Russell, 1995; Zilberstein, 1996] from the algorithm design literature. An anytime algorithm can be stopped at any time and provide a solution, however the solution improves as the algorithm is allowed more time to run. This is in contrast to a traditional, contract algorithm, which runs until a complete solution is found, and may not provide any useful solution if stopped early.

5.5 Summary

We showed, through empirical results, the average case performance of various mechanisms on our three social choice problems. By using a uniform distribution of agents' value profiles, we demonstrated the rarity of value profiles that lead to worst-case behaviour.

Typically, our stochastic mechanisms outperformed their worst-case bounds, with performance on all measures approaching the optimal value of one as problem sizes increased. The main exception was the p_{EF} in heterogeneous item allocation with $n \geq m+1$, where the performance of all generated value profiles matched the worst-case bound of $p_{EF} = \frac{n-m}{n}$.

Our three settings required the mechanism to strongly budget balance all payments. Our empirical results showed that, even if SBB is optional, a mechanism designer can potentially achieve higher overall agent utility by enforcing SBB rather than enforcing allocative efficiency.

Next, we demonstrated that the `IgnoreAgents` and `PartitionAgents` techniques can be used to improve communication and computational complexity of a mechanism. Randomly partitioning provides a flexible trade-off between complexity and outcome quality. With additional partitions, less information is required from the agents, and the mechanism can calculate an outcome in less time. This does not harm the DSIC or SBB guarantees of the mechanism. We showed that increasing the number of partitions greatly lowers the worst-case bound of the mechanism for r_{WCE} in the heterogeneous item allocation problem, but found that losses were much lower in the average case. As the problem size increases, the average-case loss of r_{WCE} due to additional partitions decreases. In these larger problems, a mechanism would have a greater need for improvements in computational and communication complexity.

Chapter 6

Conclusions

In this work, we examined randomised approaches to the design of mechanisms for problems in computational social choice in order to overcome game theoretic and computational issues. In addition to novel mechanisms and characterisations, we discussed measurements appropriate for randomised mechanisms that facilitate the comparison between such mechanisms, and the comparison to deterministic mechanisms. Our techniques were developed primarily to improve the quality of the outcome produced by the mechanism, but we also showed how these techniques can flexibly lower the computational and communication demands of a mechanism. We provided empirical results of average-case performance to complement our theoretical worst-case analysis.

While computational social choice spans a wide domain of problems from voting on a single winner, to fair division of a cake [Chevaleyre *et al.*, 2007; Brandt *et al.*, 2012], the main focus of this work is on problems of multiagent resource allocation. The first problem is the allocation of heterogeneous items to a group of agents, such that each agent receives (or desires to receive) at most one item. The goal of this allocation procedure is to maximise the aggregate utility of the participants. We also examine the room assignment–rent division problem, which also deals with allocating heterogeneous resources, but we are instead concerned with minimising envy between agents. Finally we look at the public good problem, where agents make a collective decision on whether to pay for some publicly owned resource, the benefits of which can be enjoyed by all participants.

Since agents can misreport their true preferences to manipulate the outcome of the decision making process, we enforce truthfulness, through either dominant strategy incentive compatibility, or the weaker requirement of truthfulness in expectation. As these problems occur without an explicit "auctioneer" or external funding source, the group of agents cannot expect to receive arbitrary money to participate in the auction, neither do they have a residual claimant to receive any surplus payments generated by the auction. Thus, the allocation procedures are required to satisfy strong budget balance.

In this work, we described the set of possible mechanisms that are DSIC for the problem of allocating heterogeneous items to agents with unit demand, under an independence constraint. This focussed our attention on (multi-round) affine-maximising social choice functions. We further characterised mechanisms that are DSIC, strongly budget balanced and satisfy independence. From these results, we arrived at two techniques to achieve truthfulness in a strongly budget balanced setting. The first technique is to ignore the reported types of some subset of agents, having these agents absorb any budget imbalance. With the ignored agent chosen randomly, we name this the `RandomAuctioneer` mechanism, since an ignored agent effectively becomes the auctioneer, collecting payments from the agents and being given the choice of any left-over item. Alternatively, the agents can be partitioned into disjoint groups, each solving a sub-problem of the overall social choice problem. Any surplus generated by a particular partition can be absorbed by one or more other partitions. We found upper bounds that showed poor worst-case performance of these techniques on deterministic mechanisms, which motivated the study of stochastic solutions. These stochastic mechanisms randomise over the choice of ignored agents or the choice of partitions.

We demonstrated two measures for comparing stochastic allocation mechanisms that are DSIC and SBB, namely the probability of efficiency, and the expected social surplus ratio. Where the mechanism designer's goals include maximising the overall utility of the group of agents, mechanisms can be compared based on their expected social surplus. Alternatively, agents' utilities may be of secondary importance, and the designer is most concerned with achieving an efficient allocation, in which case the probability of achieving such an outcome will appropriately compare mechanisms. While we apply these measures to the specific problem of allocation under unit demand, these natural extensions to previous properties for deterministic allocation mechanisms are appropriate in broader settings. They provide extra detail required to adequately compare stochastic mechanisms, while maintaining their original, deterministic meaning when applied to deterministic mechanisms.

Using our two measures and looking at worst-case bounds, we found the optimal stochastic mechanisms for our allocation problem under our constraints. These mechanisms work by either ignoring a single agent's reported value entirely, or by partitioning agents and items into separate allocation groups. Our optimal mechanisms are particular instantiations of the general technique developed by Faltings [2005] but with proof of optimality for our particular setting. They are also similar to the partition mechanisms used to allocate identical (homogeneous) items proposed by Guo and Conitzer [2008].

We applied these strategies to our other social choice problems, with randomised mechanisms for achieving envy-freeness in the room assignment–rent division problem. A deterministic mechanism is unable to provide an envy-free outcome while ensuring agents have no incentive to misreport their preferences. As with heterogeneous item allocation, we are able to obtain truthful mechanisms for this problem through the use of ignoring or partitioning. For a randomised mechanism, there are several possible outcomes, so evaluating and comparing these mechanisms by purely deterministic measures is not always suitable. We presented measures of envy-freeness appropriate for comparing randomised mechanisms.

Calculating envy between agents' lotteries of outcomes is not an effective measure in the RARD problem, as we show it is trivial to achieve this in mechanisms, and it does not consider the quality of final outcomes. Instead we focused on measures related to those measures we used for efficiency. The probability of envy-freeness shows, in the worst case, what probability the mechanism will achieve the ideal outcome of envy-freeness in all agents. We also assessed mechanisms based on the expected fraction of envious agents, which can give an expected level of quality where the ideal outcome is unlikely.

The nature of the public good problem precludes the use of partitioning, but the `RandomAuctioneer` approach allows truthful and strongly budget balanced mechanisms. Since allocative efficiency is a concern of the public good problem, we use the same measures we applied to the heterogeneous item allocation problem.

We assessed the theoretical worst-case performance of these new, stochastic mechanisms on our three problems and showed that they outperform deterministic approaches. For the item allocation and RARD problems, due to our characterisations we were able to provide optimal mechanisms, according to our measures of efficiency and envy-freeness.

In addition to assessing the solution quality in terms of collective utility or envy-freeness, we were also concerned with the communication and computational complexity of these mechanisms. The `RandomAuctioneer` and partitioning approaches were used to allow budget balanced, truthful mechanisms. However, these two techniques also facilitate improvements to the communication and computational requirements of a mechanism. Ignoring or partitioning agents lowers the amount of information the mechanism needs to collect from agents, while also reducing the time required to calculate a solution to the social choice problem. Both techniques over a flexible trade-off between the final solution quality and the communication and computational demands.

This work included empirical results in addition to our theoretical worst-case analyses. For our stochastic mechanisms, we performed an average-case analysis using randomly generated value profiles. These results showed that, especially with larger problem sizes, the worst-case performance of these mechanisms occurs in a small fraction of possible value profiles, making the average case typically much higher.

Our three social choice problems required strong budget balance due to constraints of the problem definition. However, we showed that this constraint can improve the overall utility of agents. In our empirical results we compared the VCG mechanism, which maximises the total utility when choosing an outcome, to our strongly budget balanced mechanisms. The loss of utility due to budget imbalance in VCG was greater than the loss due to an inefficient outcome chosen by our stochastic mechanisms. While the budget imbalance can be reduced in some settings, our results show that inefficient allocation should be considered when attempting to maximise the overall utility of a set of agents.

When assessing the techniques to improve communication and computational complexity, we also performed empirical average-case tests. The partitioning approach reduces the complexity of a mechanism as the number of partitions increases, but with increasing losses to solution quality (e.g. efficiency). We showed that in the item allocation problem, as the problem size increases, the loss due to additional partitions reduces. This is a promising result, since it is larger problems that would benefit more from such improvements to communication and computational demands. This partitioning approach also allows the creation of a mechanism that iteratively uses fewer partitions. Much like an anytime algorithm, the mechanism can progressively improve on the allocation as more agent types are received or as more time is allowed to calculate an allocation, while being able to present a solution at any time, if stopped early.

In all, randomisation is a powerful tool for the design of mechanisms in computational social choice. It can improve on game theoretic worst-case bounds of deterministic mechanisms as well as communication and computational requirements. Moving to stochastic solutions, however, necessitates new measures of solution quality, which we discussed in this thesis.

6.1 Future Work

The techniques we presented in this work can be extended to additional problems. Most direct would be to agents with non-unit demand, and more general auction settings such as combinatorial auctions. This use can be to allow strong budget balance and truthfulness, to improve overall agent utility, or to improve computational demands. Combinatorial auctions in particular can suffer from large computational problems, so a partitioning approach may be especially beneficial, along with using iteratively fewer partitions for "anytime mechanism design".

The stochastic measures introduced in this work can be directly applied to other settings examining stochastic mechanisms with the objective of efficiency or envy-freeness. For example, the work of Feldman and Lai [2012] extended the impossibility results of Cohen *et al.* [2010] for truthful, envy-free mechanisms in allocations with bounded capacity (a generalisation of unit-demand). Our stochastic measures and techniques can be applied to find truthful, envy-free stochastic mechanisms in these settings. They can also be adapted to other settings for properties such as profit maximisation or egalitarian social welfare.

We showed that sacrificing efficiency to achieve strong budget balance can result in higher overall agent utility compared to mechanisms that enforce efficiency but allow budget imbalance. Using our characterisation of DSIC, item allocation mechanisms, future work can find a mechanism that enforces neither efficiency nor strong budget balance, yet provides the optimal worst-case (expected) social surplus ratio.

For the item allocation and public good problems, we focussed on DSIC mechanisms. If this is relaxed to include mechanisms that are truthful in expectation, then more mechanisms are possible, which may improve on the worst-case bounds for item allocation. However, it may be the case that no improvements are possible with this relaxation, as Mehta and Vazirani [2004] showed in the digital goods setting.

The stochastic mechanisms presented used uniform randomisations to determine ignored agents or partitions, since no prior distribution is assumed for agent types. If such prior knowledge existed, then the stochastic mechanisms can be tuned to achieve better expected performance. The gains from using this prior knowledge, or from relaxing to truthfulness in expectation, can be quantified using the measures we present in Section 4.1.

A core assumption in this work is rational agents with full information on other agents' types and unlimited computing power. If agents do not have complete information, then DSIC may be a needlessly strong requirement, which can be relaxed to enable additional mechanisms. Agents may also act irrationally due to computational limitations (bounded rationality). Alternatively, they may act according to some other behavioural model, as studied by behavioural game theory and economics. Future work may be required to adapt the mechanisms we presented to such agents.

Finally, there has been recent work on "derandomising" stochastic auctions [Aggarwal *et al.*, 2005; Ben-Zwi *et al.*, 2009; Aggarwal *et al.*, 2011], which aim to find a deterministic auction with approximately the same revenue for a given stochastic auction. The auctions examined, however, have unlimited supply, and thus the allocation of any two agents is independent, so such techniques cannot be used in our settings. Future work using the two randomisation techniques we presented for other social choice problems may be able to derandomise these mechanisms. Since we used randomisation to gain truthfulness under rational agents, some form of derandomisation may be possible with assumptions on agents having bounded rationality.

References

[Abdulkadiroğlu *et al.*, 2004] Atila Abdulkadiroğlu, Tayfun Sönmez, and M. Utku Ünver. Room assignment-rent division: A market approach. *Social Choice and Welfare*, 22(3):515–538, 2004.

[Aggarwal *et al.*, 2005] Gagan Aggarwal, Amos Fiat, Andrew V. Goldberg, Jason D. Hartline, Nicole Immorlica, and Madhu Sudan. Derandomization of auctions. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 619–625, New York, NY, USA, 2005. ACM.

[Aggarwal *et al.*, 2011] Gagan Aggarwal, Amos Fiat, Andrew V. Goldberg, Jason D. Hartline, Nicole Immorlica, and Madhu Sudan. Derandomization of auctions. *Games and Economic Behavior*, 72(1):1 – 11, 2011.

[Alkan *et al.*, 1991] Ahmet Alkan, Gabrielle Demange, and David Gale. Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59(4):1023–39, 1991.

[Alon *et al.*, 2011] Noga Alon, Felix Fischer, Ariel D. Procaccia, and Moshe Tennenholtz. Sum of us: Strategyproof selection from the selectors. In *Proceedings of Thirteenth conference on Theoretical Aspects of Rationality and Knowledge (TARK XIII)*, March 2011.

[Andersson and Svensson, 2008] Tommy Andersson and Lars-Gunnar Svensson. Non-manipulable assignment of individuals to positions revisited. *Mathematical Social Sciences*, 56(3):350–354, November 2008.

[Andersson, 2009] Tommy Andersson. A general strategy-proof fair allocation mechanism revisited. *Economics Bulletin*, 29(3):1717–1722, 2009.

[Apt and Estévez-Fernández, 2009] Krzysztof R. Apt and Arantza Estévez-Fernández. Sequential pivotal mechanisms for public project problems. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, SAGT '09, pages 85–96, Berlin, Heidelberg, 2009. Springer-Verlag.

[Apt *et al.*, 2008] Krzysztof R. Apt, Vincent Conitzer, Mingyu Guo, and Evangelos Markakis. Welfare undominated Groves mechanisms. *Internet and Network Economics*, pages 426–437, 2008.

[Aragones, 1995] Enriqueta Aragones. A derivation of the money Rawlsian solution. *Social Choice and Welfare*, 12(3):267–276, 06 1995.

[Ariely, 2009] Dan Ariely. *Predictably Irrational*. HarperCollins Publishers, New York, NY, USA, revised and expanded edition edition, 2009.

[Arrow, 1950] Kenneth J. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58:328, 1950.

[Aspnes, 2003] James Aspnes. Randomized protocols for asynchronous consensus. *Distributed Computing*, 16(2):165–175, 09 2003.

[Ausubel, 2006] Lawrence M. Ausubel. An efficient dynamic auction for heterogeneous commodities. *The American economic review*, 96(3):602–629, 2006.

[Babaioff and Walsh, 2005] Moshe Babaioff and William E. Walsh. Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation. *Decision Support Systems*, 39(1):123–149, 2005.

[Bailey, 1997] Martin J. Bailey. The demand revealing process: To distribute the surplus. *Public Choice*, 91(2):107–26, April 1997.

[Bansal and Garg, 2000] Vipul Bansal and Rahul Garg. Efficiency and price discovery in multi-item auctions. *SIGecom Exch.*, 2(1):26–32, December 2000.

[Bartholdi *et al.*, 1989] John Bartholdi, Craig Tovey, and Michael Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[Ben-Zwi *et al.*, 2009] Oren Ben-Zwi, Ilan Newman, and Guy Wolfovitz. A new derandomization of auctions. In Marios Mavronicolas and Vicky Papadopoulou, editors, *Algorithmic Game Theory*, volume 5814 of *Lecture Notes in Computer Science*, pages 233–237. Springer Berlin / Heidelberg, 2009.

[Benisch *et al.*, 2008] Michael Benisch, Norman M. Sadeh, and Tuomas Sandholm. A theory of expressiveness in mechanisms. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, pages 17–23, Chicago, Illinois, USA, July 2008. AAA.

[Bertsekas, 1981] Dimitri P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, December 1981.

[Bhattacharya *et al.*, 2010] Sayan Bhattacharya, Gagan Goel, Sreenivas Gollapudi, and Kamesh Munagala. Budget constrained auctions with heterogeneous items. In *Proceed-*

ings of the 42nd ACM symposium on Theory of computing, STOC '10, pages 379–388, New York, NY, USA, 2010. ACM.

[Bikhchandani *et al.*, 2006] Sushil Bikhchandani, Shurojit Chatterji, Ron Lavi, Ahuva Mu'alem, Noam Nisan, and Arunava Sen. Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, 74(4):1109–1132, 2006.

[Bogomolnaia and Moulin, 2000] Anna Bogomolnaia and Hervé Moulin. A simple random assignment problem with a unique solution. *Journal of Economic Theory*, 19(3):623–636, 2000.

[Bogomolnaia and Moulin, 2001] Anna Bogomolnaia and Hervé Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295 – 328, 2001.

[Brams and Kilgour, 2001] Steven J. Brams and D. Marc Kilgour. Competitive fair division. *Journal of Political Economy*, 109(2):418–443, 2001.

[Brams and Taylor, 1996] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, February 1996.

[Brams *et al.*, 2006] Steven J. Brams, Michael A. Jones, and Christian Klamler. Better ways to cut a cake. *Notices of the AMS*, 53(11):1314 – 1321, December 2006.

[Brandt *et al.*, 2012] Felix Brandt, Vincent Conitzer, and Ulle Endriss. Computational social choice. In G. Weiss, editor, *Multiagent Systems*. MIT Press, 2012.

[Buchfuhrer *et al.*, 2010] Dave Buchfuhrer, Michael Schapira, and Yaron Singer. Computation and incentives in combinatorial public projects. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, pages 33–42, New York, NY, USA, 2010. ACM.

[Cantor, 1874] Georg Cantor. Über eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen. *Journal für die reine und angwandte Mathematik*, 77:258–262, 1874.

[Caragiannis *et al.*, 2011] Ioannis Caragiannis, John K. Lai, and Ariel D. Procaccia. Towards more expressive cake cutting. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume One*, IJCAI'11, pages 127–132. AAAI Press, 2011.

[Carbajal *et al.*, 2013] Juan Carlos Carbajal, Andrew McLennan, and Rabee Tourky. Truthful implementation and preference aggregation in restricted domains. *Journal of Economic Theory*, (0):–, 2013.

[Cavallo, 2006] Ruggiero Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. In *AAMAS'06*, Hakodate, Japan, 2006.

[Chan, 1996] Timothy M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16:361–368, 1996.

[Chen, 2008] Yan Chen. Incentive-compatible mechanisms for pure public goods: A survey of experimental research. *Handbook of Experimental Economics Results*, 1:625–643, 2008.

[Chevaleyre *et al.*, 2006] Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A. Rodríguez-Aguilar, and Paulo Sousa. Issues in multiagent resource allocation. *Informatica*, 30(1):3 – 31, 2006.

[Chevaleyre *et al.*, 2007] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In *SOFSEM '07: Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science*, pages 51–69, Berlin, Heidelberg, 2007. Springer-Verlag.

[Chevaleyre *et al.*, 2009] Yann Chevaleyre, Ulle Endriss, and Nicolas Maudet. Distributed fair allocation of indivisible goods. Working Paper, ILLC, University of Amsterdam, October 2009.

[Clarke, 1971] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[Cloutier *et al.*, 2010] John Cloutier, Kathryn L. Nyman, and Francis Edward Su. Two-player envy-free multi-cake division. *Mathematical Social Sciences*, 59(1):26 – 37, 2010.

[Cohen *et al.*, 2010] Edith Cohen, Michal Feldman, Amos Fiat, Haim Kaplan, and Svetlana Olonetsky. Truth and envy in capacitated allocation games. Discussion Paper Series dp540, Center for Rationality and Interactive Decision Theory, Hebrew University, Jerusalem, February 2010.

[Conitzer and Sandholm, 2002] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: complexity and strategy-proofness. In *AAAI'02: Proceedings of the 18th National Conference on Artificial intelligence*, pages 392–397, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[Conitzer and Sandholm, 2004] Vincent Conitzer and Tuomas Sandholm. Computational criticisms of the revelation principle. In *Proceedings of the 5th ACM conference on Electronic commerce*, EC '04, pages 262–263, New York, NY, USA, 2004. ACM.

[Conitzer and Sandholm, 2006] Vincent Conitzer and Tuomas Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *AAAI'06: Proceedings of the 21st National Conference on Artificial intelligence*, pages 627–634. AAAI Press, 2006.

[Conitzer and Sandholm, 2012] Vincent Conitzer and Tuomas Sandholm. Computing optimal outcomes under an expressive representation of settings with externalities. *Journal* *of Computer and System Sciences*, 78(1):2 – 14, 2012. ¡ce:title¿JCSS Knowledge Representation and Reasoning¡/ce:title¿.

[Conitzer *et al.*, 2005] Vincent Conitzer, Tuomas Sandholm, and Paolo Santi. Combinatorial auctions with k-wise dependent valuations. In *AAAI'05: Proceedings of the 20th National Conference on Artificial intelligence*, pages 248–254. AAAI Press, 2005.

[Conitzer *et al.*, 2007] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):14, 2007.

[Conitzer, 2005] Vincent Conitzer. Computational aspects of mechanism design. In *AAAI'05: Proceedings of the 20th National Conference on Artificial intelligence*, pages 1642–1643. AAAI Press, 2005.

[de Clippel *et al.*, 2009] Geoffroy de Clippel, Victor Naroditskiy, and Amy Greenwald. Destroy to save. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 207–214, New York, NY, USA, 2009. ACM.

[Deb and Seo, 1998] Rajat Deb and Tae Kun Seo. Maximal surplus from the pivotal mechanism: a closed form solution. *Review of Economic Design*, 3(4):347–357, 1998.

[Deb *et al.*, 2002] Rajat Deb, Indranil K. Ghosh, and Tae Kun Seo. Welfare asymptotics of the pivotal mechanism for excludable public goods. *Mathematical Social Sciences*, 43(2):209–224, 2002.

[Demange *et al.*, 1986] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[Demko and Hill, 1988] Stephen Demko and Theodore P. Hill. Equitable distribution of indivisible objects. *Mathematical Social Sciences*, 16(2):145 – 158, 1988.

[Dobzinski and Dughmi, 2009] Shahar Dobzinski and Shaddin Dughmi. On the power of randomization in algorithmic mechanism design. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 505–514. IEEE, 2009.

[Dobzinski *et al.*, 2006] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In *STOC '06: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, pages 644–652, New York, NY, USA, 2006. ACM.

[Dufton and Larson, 2011] Lachlan Dufton and Kate Larson. Randomised room assignment-rent division. In *IJCAI-2011 Workshop on Social Choice and Artificial Intelligence (WSCAI-2011)*, Barcelona, Spain, July 2011.

[Dufton *et al.*, 2012] Lachlan Dufton, Victor Naroditskiy, Maria Polukarov, and Nicholas R. Jennings. Optimizing payments in dominant-strategy mechanisms for multi-parameter domains. In *AAAI-12: Twenty-Sixth Conference on Artificial Intelligence*, 2012.

[Dütting *et al.*, 2011a] Paul Dütting, Felix Fischer, and David C. Parkes. Simplicity-expressiveness tradeoffs in mechanism design. In *Proceedings of the 12th ACM conference on Electronic commerce*, EC '11, pages 341–350, New York, NY, USA, 2011. ACM.

[Dütting *et al.*, 2011b] Paul Dütting, Monika Henzinger, and Ingmar Weber. An expressive mechanism for auctions on the web. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 127–136, New York, NY, USA, 2011. ACM.

[Edmonds and Karp, 1972] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, April 1972.

[Faltings, 2005] Boi Faltings. A budget-balanced, incentive-compatible scheme for social choice. In Peyman Faratin and Juan Rodríguez-Aguilar, editors, *Agent-Mediated Electronic Commerce VI*, volume 3435 of *Lecture Notes in Computer Science*, pages 30–43. Springer Berlin / Heidelberg, 2005.

[Feldman and Lai, 2012] Michal Feldman and John K. Lai. Mechanisms and impossibilities for truthful, envy-free allocations. In Maria Serna, editor, *Proceedings of the 5th International Symposium on Algorithmic Game Theory (SAGT)*, volume 7615 of *Lecture Notes in Computer Science*, pages 120–131, Barcelona, Spain, October 2012. Springer.

[Foley, 1967] Duncan K. Foley. Resource allocation and the public sector. *Yale Economic Essays*, 7:45–98, 1967.

[Garg and Mishra, 2002] Rahul Garg and Debasis Mishra. Auction of multiple heterogeneous items among multiple buyers and sellers using software agents linked via a communication network, May 3 2002. US Patent App. 10/139,090.

[Gershkov and Moldovanu, 2009] Alex Gershkov and Benny Moldovanu. Dynamic revenue maximization with heterogeneous objects: A mechanism design approach. *American Economic Journal: Microeconomics*, 1(2):168–198, 2009.

[Gibbard, 1973] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, July 1973.

[Gibbard, 1977] Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–681, April 1977.

[Goldberg and Hartline, 2003] Andrew V. Goldberg and Jason D. Hartline. Envy-free auctions for digital goods. In *EC '03: Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 29–35, New York, NY, USA, 2003. ACM.

[Goldberg and Hartline, 2005] Andrew V. Goldberg and Jason D. Hartline. Collusion-resistant mechanisms for single-parameter agents. In *SODA '05: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 620–629, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[Green and Laffont, 1977] Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45(2):427–38, March 1977.

[Green and Laffont, 1979] Jerry R. Green and Jean-Jacques Laffont. *Incentives in public decision-making*. North-Holland Publishing Company, Amsterdam, 1979.

[Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[Gujar and Yadati, 2008] Sujit Gujar and Narahari Yadati. Redistribution of VCG payments in assignment of heterogeneous objects. In Christos Papadimitriou and Shuzhong Zhang, editors, *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*, volume 5385 of *Lecture Notes in Computer Science*, pages 438–445. Springer Berlin / Heidelberg, 2008.

[Gujar and Yadati, 2011] Sujit Gujar and Narahari Yadati. Redistribution mechanisms for assignment of heterogeneous objects. *Journal of Artificial Intelligence Research*, 41:152–229, 2011.

[Guo and Conitzer, 2007] Mingyu Guo and Vincent Conitzer. Worst-case optimal redistribution of VCG payments. In *Proceedings of the 8th ACM conference on Electronic commerce*, EC '07, pages 30–39, New York, NY, USA, 2007. ACM.

[Guo and Conitzer, 2008] Mingyu Guo and Vincent Conitzer. Better redistribution with inefficient allocation in multi-unit auctions with unit demand. In *Proceedings of the 9th*

ACM conference on Electronic commerce, EC '08, pages 210–219, New York, NY, USA, 2008. ACM.

[Guo and Conitzer, 2009] Mingyu Guo and Vincent Conitzer. Worst-case optimal redistribution of VCG payments in multi-unit auctions. *Games and Economic Behavior*, 67(1):69 – 98, 2009.

[Guo and Conitzer, 2010] Mingyu Guo and Vincent Conitzer. Optimal-in-expectation redistribution mechanisms. *Artificial Intelligence*, 174(5-6):363–381, 2010.

[Guo *et al.*, 2011] Mingyu Guo, Victor Naroditskiy, Vincent Conitzer, Amy Greenwald, and Nicholas R. Jennings. Budget-balanced and nearly efficient randomized mechanisms: Public goods and beyond. In *The 7th Workshop on Internet and Network Economics*, pages 158–169, September 2011.

[Guo, 2011] Mingyu Guo. VCG redistribution with gross substitutes. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[Guo, 2012] Mingyu Guo. Worst-case optimal redistribution of VCG payments in heterogeneous-item auctions with unit demand. In *AAMAS*, 2012.

[Guruswami *et al.*, 2005] Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On profit-maximizing envy-free pricing. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1164–1173, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[Haake *et al.*, 2002] Claus-Jochen Haake, Matthias G. Raith, and Francis E. Su. Bidding for envy-freeness: A procedural approach to n-player fair-division problems. *Social Choice and Welfare*, 19(4):723–749, October 2002.

[Hall, 1927] Philip Hall. The distribution of means for samples of size n drawn from a population in which the variate takes values between 0 and 1, all such values being equally probable. *Biometrika*, 19(3-4):240–244, 1927.

[Hartline and Roughgarden, 2008] Jason D. Hartline and Tim Roughgarden. Optimal mechanism design and money burning. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 75–84, New York, NY, USA, 2008. ACM.

[Irwin, 1927] Joseph Oscar Irwin. On the frequency distribution of the means of samples from a population having any law of frequency with finite moments, with special reference to pearson's type ii. *Biometrika*, 19(3-4):225–239, 1927.

[Jain and Varaiya, 2005] Rahul Jain and Pravin Varaiya. Efficient market mechanisms for network resource allocation. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 1056–1061. IEEE, 2005.

[Kojima, 2009] Fuhito Kojima. Random assignment of multiple indivisible objects. *Mathematical Social Sciences*, 57(1):134–142, 2009.

[Kothari *et al.*, 2003] Anshul Kothari, David C. Parkes, and Subhash Suri. Approximately-strategyproof and tractable multi-unit auctions. In *Proceedings of the 4th ACM conference on Electronic commerce*, EC '03, pages 166–175, New York, NY, USA, 2003. ACM.

[Kuhn, 1955] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[Lavi and Swamy, 2011] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. *J. ACM*, 58(6):25:1–25:24, December 2011.

[Lavi *et al.*, 2003] Ron Lavi, Ahuva Mu'alem, and Noam Nisan. Towards a characterization of truthful combinatorial auctions. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03)*, 2003.

[Ledyard *et al.*, 1997] John O. Ledyard, David Porter, and Antonio Rangel. Experiments testing multiobject allocation mechanisms. *Journal of Economics & Management Strategy*, 6(3):639–675, 1997.

[Leonard, 1983] Herman B. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–79, June 1983.

[Maskin, 1986] Eric Maskin. *On the fair allocation of indivisible goods*. Harvard Institute of Economic Research, 1986.

[Meertens *et al.*, 2002] Marc Meertens, Jos Potters, and Hans Reijnierse. Envy-free and pareto efficient allocations in economies with indivisible goods and money. *Mathematical Social Sciences*, 44(3):223–233, December 2002.

[Mehta and Vazirani, 2004] Aranyak Mehta and Vijay V. Vazirani. Randomized truthful auctions of digital goods are randomizations over truthful auctions. In *EC '04: Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 120–124, New York, NY, USA, 2004. ACM.

[Miller, 1976] Gary L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13(3):300–317, December 1976. invited publication.

[Moore and Repullo, 1990] John Moore and Rafael Repullo. Nash implementation: A full characterization. *Econometrica*, 58(5):1083–99, 1990.

[Motwani and Raghavan, 1996] Rajeev Motwani and Prabhakar Raghavan. Randomized algorithms. *ACM Computing Surveys (CSUR)*, 28(1):33–37, 1996.

[Moulin and Shenker, 2001] Hervé Moulin and Scott Shenker. Strategyproof sharing of submodular costs:budget balance versus efficiency. *Economic Theory*, 18:511–533, 2001. 10.1007/PL00004200.

[Moulin, 1986] Hervé Moulin. Characterizations of the pivotal mechanism. *Journal of Public Economics*, 31(1):53–78, 1986.

[Moulin, 2007] Hervé Moulin. Efficient, strategy-proof and almost budget-balanced assignment. Technical report, Citeseer, 2007.

[Moulin, 2009] Hervé Moulin. Almost budget-balanced VCG mechanisms to assign multiple objects. *Journal of Economic theory*, 144(1):96–119, 2009.

[Myerson, 1979] Roger B. Myerson. Incentive compatibility and the bargaining problem. *Econometrica*, 47(1):pp. 61–73, 1979.

[Myerson, 1981] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[Naroditskiy *et al.*, 2012] Victor Naroditskiy, Mingyu Guo, Lachlan Dufton, Maria Polukarov, and Nicholas R. Jennings. Redistribution of VCG payments in public project problems. In *Proceedings of the 8th International Workshop on Internet and Network Economics (WINE)*, 2012.

[Nisan and Segal, 2006] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, July 2006.

[Othman and Sandholm, 2009] Abraham Othman and Tuomas Sandholm. Better with byzantine: Manipulation-optimal mechanisms. In Marios Mavronicolas and Vicky Papadopoulou, editors, *Algorithmic Game Theory*, volume 5814 of *Lecture Notes in Computer Science*, pages 60–71. Springer Berlin Heidelberg, 2009.

[Papadimitriou *et al.*, 2008] Christos Papadimitriou, Michael Schapira, and Yaron Singer. On the hardness of being truthful. In *IEEE 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08)*, pages 250–259. IEEE, 2008.

[Parkes *et al.*, 2001] David C. Parkes, Jayant Kalagnanam, and Marta Eso. Achieving budget-balance with vickrey-based payment schemes in exchanges. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, IJCAI'01, pages 1161–1168, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[Parkes, 2001] David C. Parkes. *Iterative combinatorial auctions: Achieving economic and computational efficiency*. PhD thesis, University of Pennsylvania, 2001.

[Pease *et al.*, 1980] Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, April 1980.

[Porter *et al.*, 2004] Ryan Porter, Yoav Shoham, and Moshe Tennenholtz. Fair imposition. *Journal of Economic Theory*, 118(2):209–228, 2004.

[Procaccia, 2010] Ariel D. Procaccia. Can approximation circumvent Gibbard-Satterthwaite? In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, 2010.

[Rabin, 1980] Michael O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128–138, 1980.

[Roberts, 1979] Kevin Roberts. The characterization of implementable choice rules. In Jean-Jacques Laffont, editor, *Aggregation and Revelation of Preferences. Papers presented at the first European Summer Workshop of the Econometric Society*, pages 321–349, North-Holland, 1979.

[Sakai, 2007] T. Sakai. Fairness and implementability in allocation of indivisible objects with monetary compensations. *Journal of Mathematical Economics*, 43(5):549–563, 2007.

[Sakurai *et al.*, 2009] Yuko Sakurai, Yasumasa Saito, Atsushi Iwasaki, and Makoto Yokoo. Sequential partition mechanism for strongly budget-balanced redistribution. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1285–1286, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[Satterthwaite, 1975] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187 – 217, 1975.

[Shehory and Kraus, 1998] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165 – 200, 1998.

[Sperner, 1928] Emanuel Sperner. Neuer beweis für die invarianz der dimensionszahl und des gebietes. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 6:265–272, 1928.

[Steinhaus, 1948] Hugo Steinhaus. The problem of fair division. *Econometrica*, 14(1):101–104, January 1948.

[Su, 1999] Francis E. Su. Rental harmony: Sperner's lemma in fair division. *The American Mathematical Monthly*, 106(10):930–942, 1999.

[Sun and Yang, 2003] Ning Sun and Zaifu Yang. A general strategy proof fair allocation mechanism. *Economics Letters*, 81(1):73–79, October 2003.

[Svensson, 1983] Lars-Gunnar Svensson. Large indivisibles: An analysis with respect to price equilibrium and fairness. *Econometrica*, 51(4):939–954, July 1983.

[Tadenuma and Thomson, 1991] Koichi Tadenuma and William Thomson. No-envy and consistency in economies with indivisible goods. *Econometrica*, 59(6):1755–67, November 1991.

[Tadenuma and Thomson, 1995] Koichi Tadenuma and William Thomson. Games of fair division. *Games and Economic Behavior*, 9(2):191 – 204, 1995.

[Taylor, 2002] Alan D. Taylor. The manipulability of voting systems. *The American Mathematical Monthly*, 109(4):321–337, April 2002.

[Varian, 1992] Hal R. Varian. *Microeconomic Analysis*. W. W. Norton & Company, Inc., New York, NY, USA, 3rd edition, 1992.

[Vickrey, 1961] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[Von Neumann and Morgenstern, 1953] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press (Princeton), 1953.

[Willson, 2003] Stephen J. Willson. Money-egalitarian-equivalent and gain-maximin allocations of indivisible items with monetary compensation. *Social Choice and Welfare*, 20(2):247–259, 2003.

[Yenmez, 2012] Mehmet Bumin Yenmez. Dissolving multi-partnerships efficiently. *Journal of Mathematical Economics*, 48:77–82, 2012.

[Zilberstein and Russell, 1995] Shlomo Zilberstein and Stuart J. Russell. Approximate reasoning using anytime algorithms. In S. Natarajan, editor, *Imprecise and Approximate Computation*. Kluwer Academic Publishers, 1995.

[Zilberstein, 1996] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.

APPENDICES

Appendix A

Proof of Theorem 3.5

The proof closely follows the proof of Theorem A by Roberts [Roberts, 1979].

For this section we define $V = \mathbb{R}^{|\mathbf{N}| \times |\mathbb{A}'|}$ as a set of agents' values over all possible *allocations* $\mathbb{A}' \subseteq \mathbb{A}$, so we need $v_i(x) = v_i(y)$ whenever $x(i) = y(i)$. Also, in the following lemmas and definitions, we use $v(x) \ll^{xy} v(y)$ between two vectors where $v \in V$ to denote $v_i(x) < v_i(y)$ where $x(i) \neq y(i)$ (i.e., agent i changes item), while the two values are necessarily equal when $x(i) = y(i)$. This holds for all $i \in \mathbf{N}$. The relation $\gg^{xy}$ is defined similarly. Note that $\ll^{xy} \equiv \ll^{yx}$. For simplicity of notation we write $\ll$ where the two allocations are clear from the context.

We begin our proof of affine-maximising by redefining the binary relation $T(v)$ between two allocations $x, y \in \mathbb{A}$, defined as follows: $xT(v)y$ if, for some $\epsilon \in V$, $\epsilon \gg^{xy} 0$, $L \in \mathbb{R}$, we have that, for any $v' \in V$ satisfying

$$v'(x) \gg v(x) - \epsilon \tag{A.1}$$

$$v'(y) \ll v(y) + \epsilon \tag{A.2}$$

$$v'_i(w) < L \text{ , } \forall i \in \mathbf{N} \text{ , } \forall w \neq x, y \tag{A.3}$$

then $f(v') = x$. This gives a notion of x being a preferred allocation to y under value profile v. In particular we will use it to show when certain allocations can *not* be chosen. The value L ensures other outcomes are valued sufficiently low such that $f(v') \neq w$. Note that when agents have the same item in both x and y, in which case $v_i(x) = v_i(y)$, the first two inequalities are satisfied by any $\epsilon_i \geq 0$.

Next, the modification to Lemma A.1 of [Roberts, 1979].

Lemma A.1. *$\forall x, y,\ x \neq y : xT(v)y \rightarrow y \neq f(v)$.*

Proof. Assume this is not true, so we have $xT(v)y$ and $y = f(v)$. From $xT(v)y$ we have $\epsilon \gg^{xy} 0$ and L. This means $f(v') = x$ for some v' that satisfies:

$$v'(x) = v(x) - \epsilon \tag{A.4}$$

$$v'(y) = v(y) + \epsilon \tag{A.5}$$

$$v'_i(w) \leq v_i(w) \text{ , } \forall i \in \mathbf{N} \text{ , } \forall w \neq x, y, w_i \neq y_i \tag{A.6}$$

Where $\epsilon_i = 0$ if $x_i = y_i$. Values $v'_i(w)$ are chosen to be sufficiently low to satisfy the above inequality as well as less than L. When $w_i = y_i$ then $v'_i(w) = v'_i(y) = v_i(y) + \epsilon_i = v_i(w) + \epsilon_i$.

This gives us:

$$v'(y) - v'(x) = v(y) - v(x) + 2\epsilon \gg v(y) - v(x) \tag{A.7}$$

and

$$v'_i(y) - v'_i(w) > v_i(y) + \epsilon - v_i(w) > v_i(y) - v_i(w) \ , \forall w \neq x, y, w_i \neq y_i \tag{A.8}$$

$$v'_i(y) - v'_i(w) = v_i(y) - v_i(w) = 0 \ , \forall w \neq x, y, w_i = y_i \tag{A.9}$$

$$\Rightarrow v'(y) - v'(w) \gg v(y) - v(w) \ , \forall w \neq x, y \tag{A.10}$$

By Lemma 3.1 (given $f(v) = y$) this implies $f(v') = y$, but $f(v') = x \neq y$, a contradiction.

□

Next, we can show that $T(v)$ retains transitivity, and a dependence on relative differences in v by modifying Lemmas A.2 and A.3 of [Roberts, 1979].

Lemma A.2. *$xT(v)y$ and $yT(v)z \rightarrow xT(v)z$.*

Proof. If x, y, z are non-distinct, this is trivial. Let $xT(v)y$ and $yT(v)z$. From the definition of $xT(v)y$ we have $\epsilon \gg^{xy} 0$ and L such that:

$$v'(x) \gg v(x) - \epsilon \tag{A.11}$$

$$v'(y) \ll v(y) + \epsilon \tag{A.12}$$

$$v'_i(w) < L \ , \forall i \in \mathbf{N} \ , \forall w \neq x, y \tag{A.13}$$

implies $x = f(v')$. Let V' be the set of v' satisfying the above inequalities. Similarly, from $yT(v)z$, we have $\epsilon' \gg^{yz} 0$ and L' such that:

$$v''(y) \gg v(y) - \epsilon' \tag{A.14}$$

$$v''(z) \ll v(z) + \epsilon' \tag{A.15}$$

$$v''_i(w) < L' \ , \forall i \in \mathbf{N} \ , \forall w \neq x, y \tag{A.16}$$

implies $y = f(v'')$. Let V'' be the set of v'' satisfying the above inequalities. Now consider v''' defined as:

$$v'''(x) = v(x) - \frac{\epsilon}{2} \ , \ x_i \neq z_i \tag{A.17}$$

$$v'''(y) = v(y) \tag{A.18}$$

$$v'''(z) = v(z) + \frac{\epsilon'}{2} \ , \ x_i \neq z_i \tag{A.19}$$

$$v_i'''(w) = L'' \ , \ \forall w \text{ s.t. } w_i \neq x_i, y_i \tag{A.20}$$

Where $L'' < \min(L, L')$. Note that this may not be a valid value profile as it must ensure $v_i(a) = v_i(b)$ whenever $a_i = b_i$. As $v'''(y) = v(y)$ then ϵ_i must be zero when $x_i = y_i$ and ϵ'_i must be zero when $z_i = y_i$, this is the case from the definition of T and $\gg$ above. When $x_i = z_i$, then $v_i'''(x) = v_i(x) = v_i(z) = v_i'''(z)$. Where $w_i = x_i$ (or y_i, z_i), then $v_i'''(w)$ must necessarily equal $v(x)$ (or $v(y)$, $v(z)$).

We can now show that $f(v''') = x$ by showing that it cannot be w, y, z. First, assume that $f(v''') = w$, for $w \neq x, y, z$. But, we can find $v'' \in V''$ such that:

$$v''(w) \gg^{wu} v'''(w) \tag{A.21}$$

$$v''(u) \ll^{wu} v'''(u) \ , \ \forall u \neq w \tag{A.22}$$

Coupling this with $f(v''') = w$ means that, by Lemma 3.1, $f(v'') = w \neq y$, which is a contradiction. A similar v'' can be found, replacing w with z to show that $f(v''') = z \rightarrow f(v'') = z \neq y$, also a contradiction.

If, instead, we let $f(v''') = y$ then we have some $v' \in V''$ such that:

$$v'(y) \gg^{yu} v'''(y) \tag{A.23}$$

$$v'(u) \ll^{yu} v'''(u) \ , \ \forall u \neq y \tag{A.24}$$

Which, from Lemma 3.1 gives us $f(v') = y \neq x$, another contradiction. Thus, we must have $f(v''') = x$. If we then define $\epsilon''_i = \min(\epsilon_i, \epsilon'_i)$, for $x_i \neq z_i$ and $L''' \leq L''$ then, from Lemma 3.1 and $f(v''') = x$ we have:

$$v^*(x) \gg v(x) - \frac{\epsilon''}{2} \tag{A.25}$$

$$v^*(z) \ll v(z) + \frac{\epsilon''}{2} \tag{A.26}$$

$$v^*(w) < L'' \ , \forall w \neq x, z \tag{A.27}$$

$$\Rightarrow v^*(x) - v^*(u) \gg v(x) - v(u) \ , \forall u \neq x \tag{A.28}$$

and this implies $f(v^*) = x$. This, by definition of T, means that $xT(v)z$. □

Lemma A.3. *For any pair of value profiles, $v, v' \in V$, if $v(x) - v(y) = v'(x) - v'(y)$ then:*

$$xT(v)y \leftrightarrow xT(v')y, \text{ and} \tag{A.29}$$

$$yT(v)x \leftrightarrow yT(v')x \tag{A.30}$$

Proof. Due to symmetry, we only need to prove that if

$$v'(x) = v(x) + \alpha, \tag{A.31}$$

$$v'(y) = v(y) + \alpha, \text{ and} \tag{A.32}$$

$$xT(v)y \tag{A.33}$$

then $xT(v')y$. From $xT(v)y$ we have some $\epsilon \geq 0$, and L such that

$$v''(x) \gg v(x) - \epsilon \tag{A.34}$$

$$v''(y) \ll v(y) + \epsilon \tag{A.35}$$

$$v''_i(w) < L \ , \forall i \in \mathbf{N} \ , \forall w \neq x, y \tag{A.36}$$

implies $x = f(v'')$. Pick one such v'' defined as

$$v''(x) = v(x) - \frac{\epsilon}{2} \tag{A.37}$$

$$v''(y) = v(y) + \frac{\epsilon}{2} \tag{A.38}$$

$$v''_i(w) = (L - \delta) \ , \ \delta > 0 \ , \ \forall w \neq x, y, \ w_i \neq x_i, y_i \tag{A.39}$$

Now consider any v''' that satisfies

$$v'''(x) \gg v'(x) - \frac{\epsilon}{2} = v''(x) + \alpha \tag{A.40}$$

$$v'''(y) \gg v'(y) + \frac{\epsilon}{2} = v''(y) + \alpha \tag{A.41}$$

$$v'''_i(w) < L' \ , \ \forall w \neq x, y, \ w_i \neq x_i, y_i \tag{A.42}$$

where $L' < L - \delta \min_i \alpha_i$. Since we have $f(v'') = x$, by Lemma 3.1 we have $f(v''') = x$, and thus this defines $xT(v')y$. □

As in the general case, Lemma A.3 shows the relationship $T(v)$ between two allocations x, y depends only on $v(x) - v(y)$. We can thus define a set $P(x, y)$ that captures all the relative values between x and y, where x is preferred, given y is normalised to zero. We must ensure valid value profiles, however. Specifically define

$$P(x, y) = \{\alpha \in \mathbb{R}^n : xT(v)y, \text{ for } v_i(x) = \alpha_i \text{ if } x_i \neq y_i, \text{ 0 otherwise. } v_i(y) = 0\} \tag{A.43}$$

We have that if $\alpha \in P(x, y)$ and $\beta \gg^{xy} \alpha$, then $\beta \in P(x, y)$.

The symmetric set, $Q(x, y)$ is defined as

$$Q(x, y) = \{\alpha \in \mathbb{R}^n : yT(v)x, \text{ for } v_i(x) = \alpha_i \text{ if } x_i \neq y_i, \text{ 0 otherwise. } v_i(y) = 0\} \tag{A.44}$$

Due to Lemma 3.1, if $f(v) = x$, then for any $\epsilon \gg^{xy} 0, y \neq x$, we have $(v(x) - v(y) + \epsilon) \in P(x, y)$, and $(v(y) - v(x) - \epsilon) \in Q(x, y)$. Note that zero values for α_i when $x_i = y_i$ are necessary, since $v_i(x) = v_i(y) \Rightarrow v_i(x) - v_i(y) = 0$. Further, for all $x, y \in \mathbb{A}'$, both $P(x, y)$ and $Q(x, y)$ are non-empty.

As with Lemma A.4 in [Roberts, 1979], we have $P(x, y), Q(x, y)$ disjoint.

Lemma A.4. *For all x, y such that $x \neq y$, $P(x, y)$ and $Q(x, y)$ are disjoint.*

Proof. Proof by contradiction, so assume there is some α such that $\alpha \in P(x, y)$ and $Q(x, y)$. This means $xT(v)y$ and $yT(v)x$ for $v(x) - v(y) = \alpha$. As $xT(v)y$, we have some v' where

$$v'(x) \ll v(x) \tag{A.45}$$

$$v'(y) \gg v(y) \tag{A.46}$$

$$v'(z) < L, \forall z \neq x, y \tag{A.47}$$

$$f(v') = x \tag{A.48}$$

Similarly, since we also have $yT(v)x$ then there exists some v'' where

$$v''(x) \gg v(x) \tag{A.49}$$

$$v''(y) \ll v(y) \tag{A.50}$$

$$v''_i(z) = K \ , \forall z \text{ s.t. } z_i \neq x_i, y_i \tag{A.51}$$

$$f(v'') = y \tag{A.52}$$

Let $L < K$. This gives us

$$v'(x) \ll^{xy} v''(x) \tag{A.53}$$

$$v'(y) \gg^{xy} v''(y) \tag{A.54}$$

$$v'_i(z) < v''_i(z) \ , \forall z \neq x, y \ , z_i \neq x_i, y_i \tag{A.55}$$

Which, combined with $f(v'') = y$ and Lemma 3.1 means that $f(v') = y$. But $f(v') = x \neq y$, which is a contradiction. □

Next we can show that if a value α is not in $Q(x, y)$ then it can be placed in $P(x, y)$ with any infinitesimal increase (in the values of agents who change items).

Lemma A.5. *For all* $x, y, x \neq y$*,* $\alpha \notin Q(x, y) \rightarrow \alpha + \epsilon \in P(x, y)$*,* $\forall \epsilon \gg^{xy} 0$.

Proof. Assume this is not true, that is, $\alpha \notin Q(x, y)$ and $\alpha + \epsilon \notin P(x, y)$, for some $\epsilon \gg^{xy} 0$. This means for any v where $v(x) - v(y) = \alpha + \frac{\epsilon}{2}$, then we have $\alpha \notin Q(x, y) \rightarrow y \neq f(v)$ and $\alpha + \epsilon \notin P(x, y) \rightarrow x \neq f(v)$. As $x, y \in \mathbb{A}'$, they come from the set of possible allocations, and as such $\exists v'$ such that $f(v') = x$. Choose a v that satisfies

$$v(x) - v(y) = \alpha + \frac{\epsilon}{2} \tag{A.56}$$

$$v(x) \geq v'(x) \tag{A.57}$$

$$v(y) \geq v'(y) \tag{A.58}$$

$$v(w) \leq v'(w) \ , \forall w \neq x, y \tag{A.59}$$

Let $z = f(v)$, since we know that $z \neq x, y$. We can then choose some v'' such that

$$v'(x) \leq v''(x) \leq v(x) \tag{A.60}$$

$$v'(z) \geq v''(z) \geq v(z) \tag{A.61}$$

$$v(y) \geq v'(y) \geq v''(y) \tag{A.62}$$

$$v'(w) \geq v(w) \geq v''(w) \ , \forall w \neq x, y \tag{A.63}$$

Since $f(v) = z$, if we compare v and v'', by Lemma 3.1 we have $f(v'') = z$. Also, since $f(v') = x$, if we compare v' and v'', by Lemma 3.1 we have $f(v'') = x$. This is a contradiction. □

This provides some insight into the boundary between $P(x, y)$ and $Q(x, y)$. By the definition of T, the sets $P(x, y)$ and $Q(x, y)$ are open, in the space $\mathbb{R}^n$, with $P(x, y)$ bounded below and $Q(x, y)$ bounded above. The lower bound

$$\gamma(x, y) = \inf\{\alpha : \alpha \in P(x, y)\} \tag{A.64}$$

is not contained in $P(x, y)$. From Lemma A.5, we have $\alpha \gg \gamma(x, y) \to \alpha \in P(x, y)$ and $\alpha \ll \gamma(x, y) \to \alpha \in Q(x, y)$.

Lemma A.6. *For all distinct x, y, z, $\gamma(x, z) =^{xz} \gamma(x, y) + \gamma(y, z)$. Here we use $a =^{xy} b$ to denote $a_i = b_i$ when $x_i \neq y_i$, with no specific relation otherwise.*

Proof. Let $\gamma(x, z) =^{xz} \gamma(x, y) + \gamma(y, z) - \delta$, and define v such that

$$v(x) =^{xy} -\frac{\delta}{4} + \gamma(x, y) \tag{A.65}$$

$$v(y) = 0 \tag{A.66}$$

$$v(z) =^{yz} \frac{\delta}{4} - \gamma(y, z) \tag{A.67}$$

We need to ensure that $v \in V$. If an agent has a different item in x, y, z then there is no problem. Also, if $x_i = z_i$ we have no additional constraints from the above equations. If $x_i = y_i$ (similarly if $y_i = z_i$) then $\gamma(x, y) = 0$ and $v_i(x) = 0$, but this doesn't force $\delta = 0$ because the first constraint only requires equality when $x_i \neq y_i$. If $z_i = x_i \neq y_i$, then the above constraints require $\delta_i = 0$, but since we don't define δ_i for $x_i = z_i$, this is not a problem.

Start by assuming $\delta > 0$, so we have:

$$v(x) - v(y) = -\frac{\delta}{4} + \gamma(x, y) \ll \gamma(x, y) \to yT(v)x \tag{A.68}$$

$$v(y) - v(z) = -\frac{\delta}{4} + \gamma(y, z) \ll \gamma(y, z) \to zT(v)y \tag{A.69}$$

So, by Lemma A.2 we have $zT(v)x$, but comparing x and z we have:

$$v(x) - v(z) = -\frac{\delta}{2} + \gamma(x, y) + \gamma(y, z) \tag{A.70}$$

$$= -\frac{\delta}{2} + \gamma(x, z) + \delta \gg \gamma(x, z) \tag{A.71}$$

$$\rightarrow xT(v)z \tag{A.72}$$

But this contradicts Lemma A.4. By reversing inequalities, we get a symmetric contradiction when $\delta < 0$. Thus $\delta = 0$. □

Next, in order to show that, after normalisation, sets $P(x, y)$ are "equivalent" for all choices of x, y we choose some "base" allocation, x, to which all allocations are normalised. Define $\delta(x) = 0$ and, for all $y \neq x$, $\delta(y) = -\gamma(x, y)$. From this definition, and Lemma A.6, we have $\gamma(y, z) = \delta(y) - \delta(z) = -\gamma(z, y)$. We then normalise our sets P by defining new sets $R(x, y)$, defined as:

$$R(x, y) = \{\alpha : \alpha + \gamma(x, y) \in P(x, y)\} \tag{A.73}$$

We can now show that R is equivalent for any pair of allocations, by modifying Lemma A.7 of [Roberts, 1979]. Note that this equivalence is somewhat different to the equality in the general setting of [Roberts, 1979]. We require equality only for agents that change items between the two allocations in both sets. When $x_i = y_i$ then for all $\alpha \in R(x, y)$ then $\alpha_i = 0$, so when comparing $R(x, y)$ to $R(w, z)$, we only compare in dimensions where $x_i \neq y_i \wedge w_i \neq z_i$ (agent i changes items in both comparisons) OR $x_i = y_i \wedge w_i = z_i$ (agent i doesn't change items, note that this doesn't require $x_i = w_i$). That is, if $R(x, y) \equiv R(w, z)$ then $\forall \alpha \in R(x, y)$, $\exists \beta \in R(w, z)$ such that if $x_i \neq y_i \wedge w_i \neq z_i$ OR $x_i = y_i \wedge w_i = z_i$ then $\alpha_i = \beta_i$.

For any α Let $\alpha_i^{xy} = \alpha_i$ if $x_i \neq y_i$ and 0 otherwise.

Lemma A.7. *For all w, x, y, z, $R(x, y) \equiv R(w, z) \equiv R$.*

Proof. It is sufficient to show that $R(x, y) \equiv R(w, y)$, as a symmetric argument will show that $R(x, y) \equiv R(x, z)$, and thus it immediately follows that $R(x, y) \equiv R(w, y) \equiv R(w, z)$. To prove by contradiction, we assume $R(x, y) \not\equiv R(w, y)$, that is, there is some α such that $\alpha^{xy} \in R(x, y)$, $\alpha^{wy} \notin R(w, y)$. Due to the openness of P and thus R, we have some $\epsilon \gg^{xw} 0$ such that:

$$\alpha^{xy} - \epsilon^{xy} + \gamma(x, y) \in P(x, y) \tag{A.74}$$

and, since $\alpha^{wy} \notin R(w, y)$, due to Lemma A.5

$$\alpha^{wy} - \frac{\epsilon^{wy}}{2} + \gamma(w, y) \in Q(w, y) \tag{A.75}$$

Define v such that

$$v(x) = \alpha^{xy} - \epsilon^{xy} + \gamma(x, y) \tag{A.76}$$

$$v(y) = 0 \tag{A.77}$$

$$v(w) = \alpha^{wy} - \frac{\epsilon^{wy}}{2} + \gamma(w, y) \tag{A.78}$$

Again, we need to ensure $v \in V$. If $x_i = y_i$ then $\alpha_i = \gamma_i(x, y) = 0$, also we have $\epsilon_i = 0$. Similarly when $y_i = w_i$. Finally, when $x_i = w_i \neq y_i$, to set $v(x) = v(w)$ we need $\epsilon_i = 0$, regardless of α_i, which we have. This value profile gives us $xT(v)y$ and $yT(v)w$ so, by Lemma A.2, $xT(v)w$. However, from Lemma A.6 we have:

$$v(w) - v(x) = \frac{\epsilon}{2} + \gamma(w, y) - \gamma(x, y) = \frac{\epsilon}{2} + \gamma(w, x) \tag{A.79}$$

As $\epsilon \gg^{xw} 0$ this means we have $xT(v)w$, which contradicts Lemma A.4. □

So the structure of f is captured by the structure of R, which, just as in the general case, remains convex.

Lemma A.8. *For all $y \neq z$, $R(y,z)$ is convex.*

Proof. Suppose not, this means we have some $\alpha \in R(y,z)$, $\beta \in R(y,z)$, but $\omega = \frac{1}{2}(\alpha+\beta) \notin R(y,z)$. Due to the openness of R, and Lemma A.5 (for the case of ω) we have some $\epsilon > 0$ such that

$$\alpha^{xy} - \epsilon^{xy} + \gamma(x,y) \in P(x,y) \tag{A.80}$$

$$\beta^{yz} - \epsilon^{yz} + \gamma(y,z) \in P(y,z) \tag{A.81}$$

$$\omega^{xz} - \epsilon^{xz} + \gamma(x,z) \in Q(x,z) \tag{A.82}$$

As $\delta \in Q(x,z) \to -\delta \in P(z,x)$ and $\gamma(z,x) = -\gamma(x,z)$, we have:

$$-\omega^{xz} + \epsilon^{xz} + \gamma(z,x) \in P(z,x) \tag{A.83}$$

Define v such that

$$v(x) = \alpha^{xy} - \epsilon^{xy} + \gamma(x,y) \tag{A.84}$$

$$v(y) = 0 \tag{A.85}$$

$$v(z) =^{yz} v(x) - \omega^{xz} + \epsilon^{xz} + \gamma(z,x) \tag{A.86}$$

This ensures $v \in V$. Whenever $x_i = y_i$ or $z_i = y_i$, then values for these allocations are forced to 0. When $x_i = z_i \neq y_i$, then $\omega^{xz}, \epsilon^{xz}, \gamma(z,x) = 0$, so $v_i(z) = v_i(x)$.

From v we have $v(x) - v(y) \in P(x,y) \to xT(v)y$ and $v(z) - v(x) = -\omega^{xz} + \epsilon^{x,z} + \gamma(z,x) \in P(z,x) \to zT(v)x$ (since when $x_i \neq z_i = y_i$, $\alpha_i = \omega_i$). By Lemma A.2 we have $zT(v)y$, which means

$$
\begin{aligned}
& v(z) - v(y) \in P(z,y) && \text{(A.87)}\\
\Rightarrow & v(x) - \omega + \epsilon + \gamma(z,x) \in P(z,y) && \text{(A.88)}\\
\Rightarrow & \alpha - \omega + \gamma(x,y) + \gamma(z,x) \in P(z,y) && \text{(A.89)}\\
\Rightarrow & \alpha - \omega + \gamma(z,y) \in P(z,y) \qquad \text{by Lemma A.6} && \text{(A.90)}\\
\Rightarrow & \frac{1}{2}(\alpha - \beta) + \gamma(z,y) \in P(z,y) && \text{(A.91)}\\
\Rightarrow & \frac{1}{2}(\alpha - \beta) \in R(y,z) && \text{(A.92)}
\end{aligned}
$$

We can repeat the above steps (from the definition of v), swapping α and β to get $\frac{1}{2}(\beta - \alpha) \in R(y,z)$.

Now define v' as

$$v'(y) = \frac{1}{2}(\alpha - \beta) + \gamma(y,z) \tag{A.93}$$

$$v'(z) = 0 \tag{A.94}$$

This is a valid value profile since $\alpha_i = \beta_i = \gamma_i(y,z) = 0$ when $y_i = z_i$. So we have:

$$v'(y) - v'(z) = \frac{1}{2}(\alpha - \beta) + \gamma(y,z) \in R(y,z) \Rightarrow yT(v')z \tag{A.95}$$

$$v'(z) - v'(y) = \frac{1}{2}(\beta - \alpha) + \gamma(z,y) \in R(y,z) \Rightarrow zT(v')y \tag{A.96}$$

But this contradicts Lemma A.4, so must have $\omega \in R(y,z)$.

□

From the definition of $R(x,y)$, if $\alpha \in R(x,y)$ then $-\alpha \notin R(x,y)$, so we have $-R(x,y) = \{\alpha : -\alpha \in R(x,y)\}$. These two sets are both disjoint and convex. We also have:

$$\alpha^{xy} \notin -R(x,y) \Rightarrow -\alpha^{xy} \notin R(x,y) \Rightarrow -\alpha^{xy} + \gamma(x,y) \notin P(x,t) \tag{A.97}$$

$$\Rightarrow \alpha^{xy} - \gamma(x,y) \notin Q(x,y) \Rightarrow \alpha^{xy} + \gamma(y,x) \notin Q(y,x) \tag{A.98}$$

So, by Lemma A.5, $\alpha^{xy} \notin -R(x,y) \Rightarrow \alpha^{xy} + \epsilon \in R(x,y)$, for all $\epsilon \gg^{xy} 0$. Since $R(x,y)$ and $-R(x,y)$ are both disjoint, convex and non-empty, there is a separating hyperplane, and thus some constant $k^{xy} \in \mathbb{R}^n$, $k^{xy} \neq 0$ such that

$$\alpha \in R(x,y) \Rightarrow k^{xy}\alpha \geq 0 \tag{A.99}$$

$$\alpha \in -R(x,y) \Rightarrow k^{xy}\alpha \leq 0 \tag{A.100}$$

Suppose we have α such that $k^{xy}\alpha > 0$, and some $\epsilon \ll^{xy} 0$ such that $k^{xy}(\alpha - \epsilon)0$. Then $\alpha - \epsilon \notin -R(x,y) \Rightarrow \alpha \in R(x,y)$. Thus,

$$k^{xy}\alpha > 0 \Rightarrow \alpha^{xy} \in R(x,y) \tag{A.101}$$

Also, since if $\epsilon \gg^{xy} 0$ then $\epsilon \in R(x,y)$, then $k^{xy} \gg^{xy} 0$.

It follows from Lemma A.7 that, for all $w, x, y, z \in \mathbb{A}'$, if $x_i \neq y_i, w_i \neq z_i$, then $k_i^{xy} = k_i^{yz} = k_i$.

Now consider $\alpha = v(x) - v(y) - \gamma(x,y)$. Since $\alpha_i = 0$ when $x_i = y_i$, we have:

$$\sum_i k_i^{xy}\alpha_i = \sum_i k_i\alpha_i \tag{A.102}$$

So

$$\sum_i k_i\alpha_i > 0 \Rightarrow xT(v)y \Rightarrow y \neq f(v) \tag{A.103}$$

$$\sum_i k_i\alpha_i < 0 \Rightarrow yT(v)x \Rightarrow x \neq f(v) \tag{A.104}$$

$$\Rightarrow \sum_i k_i(v_i(x) - \delta_i(x)) > \sum_i k_i(v_i(y) - \delta_i(y)) \Rightarrow y \neq f(v) \tag{A.105}$$

And so if we have $f(v) = x$ then

$$\Rightarrow \sum_i k_i(v_i(x) - \delta_i(x)) \geq \sum_i k_i(v_i(y) - \delta_i(y)) \ , \forall y \in \mathbb{A}' \tag{A.106}$$

Thus, setting $U_0(a) = -\sum_i k_i\delta_i(a)$, and $\gamma_i = k_i$, we have

$$f(v) = x \Rightarrow U_0(x) + \sum_i \gamma_i v_i(x) \geq U_0(y) + \sum_i \gamma_i v_i(y) \ , \forall y \in \mathbb{A}' \tag{A.107}$$

$$\Rightarrow f(v) \in \arg\max a \in \mathbb{A}' \left(U_0(a) + \sum_i \gamma_i v_i(a) \right) \tag{A.108}$$

Which defines f as in Equation (3.12), that is, f is an affine maximiser.

Index