On the Trade-offs between Modeling Power and Algorithmic Complexity

Chun Ye

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2016

ABSTRACT

On the Trade-offs between Modeling Power and Algorithmic Complexity

Chun Ye

Mathematical modeling is a central component of operations research. Most of the academic research in our field focuses on developing algorithmic tools for solving various mathematical problems arising from our models. However, our procedure for selecting the best model to use in any particular application is ad hoc. This dissertation seeks to rigorously quantify the trade-offs between various design criteria in model construction through a series of case studies. The hope is that a better understanding of the pros and cons of different models (for the same application) can guide and improve the model selection process.

In this dissertation, we focus on two broad types of trade-offs. The first type arises naturally in mechanism or market design, a discipline that focuses on developing optimization models for complex multi-agent systems. Such systems may require satisfying multiple objectives that are potentially in conflict with one another. Hence, finding a solution that simultaneously satisfies several design requirements is challenging. The second type addresses the dynamics between model complexity and computational tractability in the context of approximation algorithms for some discrete optimization problems. The need to study this type of trade-offs is motivated by certain industry problems where the goal is to obtain the best solution within a reasonable time frame. Hence, being able to quantify and compare the degree of sub-optimality of the solution obtained under different models is helpful. Chapters 2-5 of the dissertation focus on trade-offs of the first type and Chapters 6-7 the second type.

Table of Contents

List of Figures

List of Tables

Acknowledgments

> We are nothing without the work of others our predecessors, others our teachers, others our contemporaries. Even when, in the measure of our inadequacy and our fullness, new insight and new order are created, we are still nothing without others. Yet we are more.
>
> *J. Robert Oppenheimer*

The past five years of graduate school at Columbia has been an incredible journey, one where I was given (virtually) unlimited freedom to pursue my research interests, to read and evaluate other scholarly works through a critical lens, and to craft and polish my own works of art. I am extremely grateful for the countless intellectually simulating discussions I have had with, and the numerous sound advice that I have received from my mentors and peers of the IEOR department. I would not have gotten to where I am today without their inputs.

First and foremost, I would like to thank my advisor Prof. Jay Sethuraman. I often compare doing research to waking up amid a cave, surrounded by darkness. The task is to learn our surroundings and find our way out of the cave. The journey inside the cave is simultaneously exciting and intimidating. On one hand, we get to explore realms of the unknown. On the other hand, the rugged path that we choose to explore often leads us back to where we started. A great advisor, by analogy, serves as a constant source of light, guiding our way through the unknown. In that regard, I am blessed to have Jay as my advisor. I learned so much from the insightful discussions that we had over the past five years. He taught me to sort through the seemingly chaotic darkness of the cave and see the clarity within. His phenomenal ability to distill certain fundamental insights from complicated arguments in order to make a proof simpler is an exemplary trait that I seek to develop as a researcher. His taste for interesting problems made research a hobby rather than a chore for me. Beyond research, I also benefited tremendously from Jay's professional and personal advice. Jay is incredibly down-to-earth, kind, caring and encouraging. It is always a pleasure to watch and learn from his interactions with other students and colleagues. Thank you, Jay, for being that source of light guiding me through research and beyond.

Next, I would like to thank Prof. Daniel Bienstock and Prof. Vineet Goyal, whom I have worked with extensively. Chapters 6 and 7 of this dissertation would not have existed without their mentorship. Both Dan and Vineet took many hours out of their busy schedule to meet with me. I benefited tremendously from their vast breadth of knowledge and learned a great deal from their phenomenal work ethics. I would like to thank Dr. Haris Aziz for collaborating with me on the cake cutting project, which became Chapter 4 of the dissertation. Despite the inconvenience of not being able to work in person, Haris was always extremely quick and patient in responding to my emails, answering my various questions, and sharing with me his thoughts and ideas. Moreover, I am indebted to Haris for graciously agreeing to serve on my dissertation committee and for coming a long way to attend my defense.

I am also grateful for the opportunity to learn from many wonderful instructors, including Prof. Cliff Stein, Prof. Garud Iyengar, Prof. Adam Elmachtoub, Prof. Ciamac Moallemi, Prof. Assaf Zeevi, Prof. Rocco Servedio, and Prof. Mihalis Yannakakis. They taught with extreme clarity and often complemented the routine course material with their novel personal insights and perspective. More importantly, they conveyed their excitement for their research areas through teaching.

My sincere gratitude also goes to the IEOR departmental staff, who worked relentlessly behind the scenes to make our PhD experience seamless. In particular, I'd like to give big shout outs to Jenny, Adina, Darbi, Krupa, Carmen, Jaya, Shi Yee, Liz, Mindi and Simon, for patiently answering my questions, responding to my requests, and for giving their endless support to numerous student seminars and CU INFORMS events.

My PhD life would not have been the same without the fellow students of IEOR. From the countless white board discussions, to the random hallway chats, the planned and unplanned food excursions and fun trips, to those embarrassing times at office hours and recitations when I could not answer a question, to the delight on my students' face when they finally get it, I cherish every moment. Duncan Watts, a sociologist known for his work on small-world networks, once said that graduate school "can be a means to an end... But it can also be an end in itself." *Thank you all for making the means matter so much more important than the end!* In particular, thank you Antoine Désir, for your positive attitude and your perseverance. Thank you Itai Feigenbaum, for your sense of humor and meticulousness. I had a great time working with the both of you and learned many valuable lessons through our collaboration. Thank you Gonzalo Muñoz, for your cheery personality and for putting up with my lack of small talk as a neighbor all these years. Thank you Xingyu Zhang, for entertaining me with your random commentaries during the thesis writing process. Thank you Cun Mu, Zheng Wang, Jing Guo, Linan Yang, and Daniel Mercado, for toiling with me through my first year, from the core classes to the qualifying exams. Thank you Erez Cohen, for being my awesome first-year neighbor and for sharing with me many of your (not so) crazy start-up ideas. Thank you Tulia, Daniela, Shyam, Rodrigo, Jing, Cecilia, Chen Chen, Aya, Matthieu, Tony, Peter, Arseniy, Haowen, J.B., Andrei, Xin, Xinyun, Brian, Ningyuan, Marco, Anran, Di, Vashist, Uri, Hal, Francois, Fei, Zhipeng, Irene, Ni, Yanan, Xinshang, Mingxian, Jalaj, Kevin, Michael, Allen, Xu, Shuangyu, Ryan, Raghav, K.G. and others, for sharing your experience and stories with me, and for giving me a ton of good advice.

Finally, words cannot describe my gratitude for Chen Zhang, who fed me when I was hungry, kept me entertained me when I was bored, cheered me up when I was down, and encouraged me to push my limits and to live up to my potential. I am truly lucky to have her by my side.

New York, NY

July 2016

To my beloved parents
and Chen

Chapter 1

Introduction

Mathematical modeling is a central component in the field of operations research. A model allows us to capture the essence of a real world problem and map it to a mathematical framework. Once such a framework is established, we then adapt the existing tools or develop new ones to tackle the mathematical problem at hand. It is perhaps not an exaggeration to say that much of the academic research in our field focuses on developing algorithmic tools for solving various mathematical problems induced by our models. Consequently, we have established sophisticated methods and benchmarks for evaluating the performance of our solution given a model. On the other hand, our procedure for selecting the best model to use is more ad hoc. Models that are more powerful and accurate tend to be more complex, making analysis more difficult. As a result, the standard approach is to circumvent the intractability of a complex model by making some simplifying assumptions:

1. the simplified model still captures the essence of the problem, and
2. the assumptions made significantly simplify our analysis; it is often the case that an optimal solution can be computed efficiently for the simplified model.

Unfortunately, these justifications are more philosophical than scientific, as we lack rigorous analysis for comparing and contrasting the pros and cons of various models in order to determine which model will yield the best solution in practice. It may well be the case that a suboptimal solution of a complex model outperforms an optimal solution of a simple model in practice. This dissertation by no means proposes a general methodology for model selection. Rather, *it seeks to rigorously quantify the trade-offs between various design criteria in model construction through a series of case studies, in order to* *better understand the pros and cons of different models and help guide the model selection process.* Our case studies include problems from fair division and resource allocation, facility location, knapsack, and assortment optimization. In each case, we find the best computationally tractable algorithm between a pair of simple and more sophisticated models, compare the quality of the solution with respect to a given benchmark in each case. Since the given benchmark often becomes more demanding as model complexity increases, we seek to quantify trade-offs by measuring the degree of "optimality loss" against the benchmark when we go from a simple model to a more complex model.

With the abundance of data in the modern information age, the empirical performance of algorithms constructed under different models can be evaluated and compared. Nonetheless, a rigorous framework for model selection is not as well developed as finding a good solution to a given model. Understanding the trade-offs across different models will help guide the model selection process. Moreover, establishing a rigorous framework for model selection will help discourage the common academic practice of overloading a model with stylized assumptions so as to be able to obtain an optimal solution.

The trade-offs across various models that will be addressed in this dissertation fall into two categories. The first type of trade-offs arises naturally in mechanism or market design, a discipline that focuses on developing optimization models for complex multi-agent systems. Such systems may require optimizing multiple objectives that are potentially in conflict with one another. Hence, finding a solution that simultaneously satisfies several design requirements is challenging. The second type of trade-offs addresses the dynamics between model complexity and computational tractability in the context of approximation algorithms for some discrete optimization problems. The need to study this type of trade-offs is motivated by certain industry problems where the goal is to obtain the best solution within a reasonable time frame, rather than computing the optimal solution. Hence, being able to quantify and compare the degree of sub-optimality of our solution obtained under different models is crucial for decision making. Chapters 2-5 of the dissertation focus on trade-offs of the first type and Chapters 6-7 the second type.

1.1 Trade-offs in Multi-Agent Optimization Models

Nowadays, many models of consumer purchase behavior, competition or cooperation among different firms, and algorithms for resource allocation inherently involve human actors. The nature of such problems often gives rise to multiple objectives: not only should we optimize the system performance at the aggregate level, but we would also like to treat all participants of the system fairly. The multi-objective algorithmic challenge is further complicated by the need to elicit valuable private information from the participants of the system and use this information as inputs to our algorithm. Eliciting truthful inputs is not straightforward when the participants' objectives are not aligned with the objective of the designer. Whenever potential incentive issues are not recognized and dealt with by the designer, unintended consequences may arise.

Hence, the algorithmic challenge in these problems is the delicate balance between the need for truthful input acquisition and optimization, where optimization means finding an efficient and fair allocation. In such settings, the model complexity, in addition to operational constraints, is often captured by preferential inputs. While a richer model allows for more modeling power, attempting to satisfy a given set of efficiency, fairness, and truthfulness criteria simultaneously in such a model becomes more challenging if not impossible. Any good algorithm aims to satisfy strong equity and efficiency properties must cater to agent preferences closely, making it in turn vulnerable to manipulation. One way to avoid this impossibility is to assume an environment in which agents (and the designer) can make or receive payments; allowing for payments introduces an extra lever that the designer can exploit. There are many settings, however, in which such monetary compensations are either not possible or are undesirable. For instance, selling one's kidney is forbidden by law in many parts of the world, which motivates the need for a kidney exchange program. Consequently, Chapters 2 through 4 of the dissertation explore the boundary between what can and cannot be achieved by a mechanism without the using of monetary transfers. Moreover, we also consider a quantitative measure of how far our solution is from the ideal benchmark in Chapter 5.

One application of mechanism design without money is the division and allocation of common/shared resource(s). From rent splitting between housemates, to property disputes between property owners, work sharing between co-workers, fair division problems arise naturally in every day life. The goal of the designer here is to create an allocation protocol that treats all parties fairly, achieves an efficient outcome, and discourages agents from falsifying their preference in order to game the protocol. The typical fairness notion that we will consider is called *envy-freeness*, which means that every agent weakly prefers her allocation to every other agent's allocation. For efficiency, we will consider *Pareto optimality* and it variants. An allocation is Pareto optimal if one cannot improve the allocation for one agent without hurting another agent. Finally, in order to discourage agents from misreporting their preference, we require our mechanism to be *strategyproof*: it should be a dominant strategy for every agent to report her preference truthfully.

A well known problem in the resource allocation mechanism design literature is the assignment problem. In this problem, we have a set of agents and a set of resources. Each agent has some demand requirement and each resource has a capacity limit. Moreover, each agent expresses some form of preference over the resources.[1] In a simple model, the preference can be dichotomous: either the agent finds a resource acceptable or not. On the other hand, a richer model allows an agent to specify a utility for each resource or a preference ordering over the resources. In the standard model, each agent is assumed to have a unit demand and each resource a unit capacity (however, we will consider a general model with arbitrary demand and capacity quantities in Chapters 3 and 4). The resources in consideration can either be divisible or indivisible. For each indivisible resource, we view a fractional allocation as the probability that a given resource is allocated to a given agent.

In Chapter 2,[2] we consider a special case of the assignment problem where agents have a uniform preference over the resources,[3] which are indivisible. This problem is motivated by a single machine scheduling problem where every job has a unit processing time. Every job would like to be scheduled as early as possible and has a deadline that it needs to be scheduled by. We would like to find an algorithm that outputs an efficient allocation while treating all jobs fairly. Furthermore, since the deadline of each job is private information, we would like the algorithm to be truthful so that no job can game the allocation made by the algorithm. This problem can be used to model a patient scheduling system where agents are patients and objects are appointment time slots. In such a setting, all patients would like be to treated as early as possible. Moreover, each patient has a deadline by which she needs to receive treatment before her condition gets worse.

When preferences are strict, Bogomolnaia and Moulin [30] characterized the probabilistic serial (PS) mechanism as the only mechanism satisfying some nice equity (equal treatment of equals), efficiency (ordinal efficiency), and truthfulness (strategyproofness) properties. Nonetheless, agents may

[1] We assume that each agent has a homogeneous preference within each resource.

[2] Chapter 2 is based on the paper [111].

[3] An uniform preference domain is one in which agents rank order their acceptable resources in the same fashion, modulo indifferences.

express indifference over resources in many realistic settings. Failure to incorporate indifference into the model would allow us to forgo a potentially more efficient allocation in reality. Therefore, we aim to understand whether an extension of the PS mechanism to the weak preference domain, one in which agents are allowed to express indifference over objects, can still maintain the properties that it satisfies for the strict preference domain. We show that, in the weak preference domain, not only does the PS mechanism fails strategyproofness, but so does every other mechanism that is ordinally efficient and treats equals equally. If envy-free assignments are required, then any mechanism that guarantees an ex post efficient outcome must fail even a weak form of strategyproofness. Our impossibility results suggest that allowing agents to express their preferences more explicitly comes at a price: a mechanism with good performance could be gamed by the agents in some cases.

In Chapter 3,[4] we study the fair allocation of a single divisible good to multiple agents with heterogeneous preferences, also known as *cake cutting* in the literature. Cake cutting is often used to model the fair division of a common resource. It is particularly applicable to the allocation of server time, as the advent of cloud computing for example, has increased the need for allocation policies in environments with heterogeneous user demands. In the most general model for cake cutting, a cake is often represented by the unit interval. There is a set of agents who are the cake recipients. Every agent has an integrable valuation function over the cake. For computational reasons, we consider a restricted domain where each agent has a privately known piecewise constant valuation function (PCV) over the cake. This special domain already has enough richness to capture many applications.

The main goal of the chapter is to identify and understand the trade-offs between various desirable properties attainable by a cake cutting algorithm when agents report PCV. These properties fall into three categories: efficiency (Pareto optimality and non-wastefulness), fairness ((robust) envy-freeness and proportionality), and truthfulness (group strategyproofness, strategyproofness, and strategyproofness in expectation). For a special case of PCV called piecewise uniform valuations (PUV), Chen et al. [44] proposed an algorithm that jointly satisfies the strongest property within each category. We examine different extensions of this algorithm to the PCV setting and prove that while these algorithmic extensions maintain strong efficiency and fairness guarantees, they do not satisfy strategyproofness. Specifically, we present two algorithms: the Controlled Cake Eating Algorithm (CCEA) and the Market Equilibrium Algorithm (MEA). CCEA is inspired by the probabilistic serial mechanism of [29] and

[4]Chapter 3 is based on the paper [13].

its extension due to [76]. It is non-wasteful and robust envy-free. The Market Equilibrium Algorithm (MEA) is based on the Eisenberg-Gale convex program for computing a market equilibrium. It is Pareto optimal and envy free. To demonstrate that our algorithmic results are the best achievable, we show that no strategyproof algorithm can jointly satisfy the properties satisfied by our algorithmic extensions. Finally, if we allow for randomization and only required strategyproofness to be satisfied in expectation, then we are able to obtain an algorithm (MCSD) inspired by a constrained version of random serial dictatorship that is robust proportional and strictly dominates the uniform allocation.[5] Just as in Chapter 2, we notice here that ensuring truthfulness in conjunction with strong equity and efficiency guarantees becomes more difficult as the agent preference domain becomes more expressive.

In Chapter 4,[6] we consider a model of resource allocation with arbitrary demand and supply quantities. Moreover, each agent's demand can only be satisfied in its entirety by one single resource whose capacity quantity can accommodate the demand. This problem is a generalization of the setting of Chapter 2, where each agent has an unit demand and each resource a unit supply. It is also different from the the cake cutting problem,[7] as an agent's demand here cannot be satisfied partially nor can it be split up by multiple resources in this setting. Kurokawa et al. [81] first considered the problem in the context of classroom assignment for charter schools. They modeled the agent's preference as dichotomous, and designed a mechanism that satisfies many nice properties, including: proportionality, envy-freeness, Pareto optimality, and strategyproofness. Their mechanism always computes a leximin allocation: one that maximizes the lowest probability of any school having its demand satisfied in an acceptable facility; subject to this constraint, it maximizes the second lowest probability; and so on. In Chapter 4, we seek to understand whether the leximin mechanism still satisfies the aforementioned properties that Kurokawa et al. showed for the dichotomous setting in the general preference domain. It is known from [29] that even for the unit demand and unit supply setting, the leximin mechanism fails to be strategyproof. We show in the chapter that it fails envy-freeness as well. Nonetheless, the mechanism remains Pareto optimal and we conjecture it to be proportional as well. In terms of computing a leximin allocation for this setting, Bogomolnaia [27] gave an alternative definition of the

[5]The uniform allocation is one in which every agent is indifferent between her allocation and any other agent's allocation.

[6]This chapter is based on the working paper [112].

[7]A cake cutting problem with PCV to a corresponding instance of the assignment problem with divisible resources.

probabilistic serial mechanism in the unit supply and unit demand setting by showing that mechanism always computes a generalized leximin allocation. We propose a generalization of the probabilistic serial mechanism to the classroom setting and show that our generalization also always computes a generalized leximin allocation.

In Chapters 2-4, we quantified the trade-offs among desirable properties in a binary fashion. We can also measure the efficiency loss when we impose truthfulness as a constraint. One standard method for measuring efficiency loss is to compare the ratio of optimal values with and without the incentive constraint. We apply this method in Chapter 5 to study a facility location problem on a line where the designer needs to elicit preferences from a set of agents in order to locate a facility.[8] Each agent incurs a cost equal to her distance to the facility whereas the designer wishes to minimize the L_p norm of the vector of agent costs. Note that the L_p social cost function serves as a way to trade-off between efficiency and fairness, as minimizing the L_1 norm optimizes the aggregate individual cost, whereas minimizing the L_∞ norm reduces the cost of the agent(s) who is worst off. Since an agent's objective function does not perfectly align with that of the designer, locating the facility at an optimizer of the social cost function is not strategyproof. Instead, our goal is to design a strategyproof mechanism that approximates the optimal cost well. The design of a strategyproof mechanism is important in certain applications of the problem such as peer valuation/rating, where it is crucial for no participant to purposely exaggerate or shade her rating in order to make the aggregate rating closer towards her own. Our main result shows that the mechanism that always locates the facility at the median of agents' reports provides a $2^{1-1/p}$ approximation ratio, and that this is the optimal approximation ratio among all deterministic strategyproof mechanisms. Moreover, the approximation ratio provided by the median mechanism is also optimal over a large class of randomized mechanisms. This class of randomized mechanisms subsumes many existing mechanisms proposed in the literature, see e.g. [4, 60, 98]. We also exhibit an optimal randomized mechanism for two agents. By identifying a family of approximate optimal strategyproof mechanisms (AOSM), we observe that a L_p norm objective for p large (which corresponds to objectives that puts more emphasis on fairness than on efficiency) is more vulnerable to manipulation by the agents, resulting in a worse approximation ratio.

[8] Chapter 5 is based on the paper [57].

1.2 Model Predictive Power versus Computational Tractability Trade-offs

The remaining chapters address how model complexity affects its computational tractability. We often work with stylized models because they lead to elegant mathematical results and are often computationally tractable. However, many problems of interest to practitioners are often much more complex, and key elements of such problems cannot be captured in a stylized model. As a consequence, practitioners tend to favor more accurate but less computationally tractable models. Fast heuristics are then developed to solve the associated optimization problems approximately. With the information revolution and data explosion, there is an increasing need for a rigorous understanding of the tractability of more accurate and complex models that fit the data or match the real world problem description well.

For instance, the knapsack problem is a fundamental problem that can be used to model many applications related to resource allocation. It captures any situation where the decision maker seeks to identify an optimal set of projects to invest in. Each project incurs a cost and results in a profit. Due to a budget limitation, the decision maker must select a subset of projects to invest in that maximizes profit without violating the budget constraint. The simplicity of the knapsack problem fails to capture applications where investment decisions can be made over a time horizon and the decision maker receives additional budget in each time period to invest in additional projects. To model these applications, we consider a knapsack problem whose capacity grows with time in Chapter 6.[9] The increment in knapsack capacity from period to period represents the additional budget that the decision maker receives in each time period (which can be either discrete or continuous). We are given a set of items with weight and values at the beginning of the time horizon and need to decide the subset of items to pack into the knapsack in each time period such that:

1. the sum of weights of items packed in the knapsack does not exceed its capacity in each time period.

2. an item packed into the knapsack can never be removed from the knapsack later on.

3. the discounted sum of item values in the knapsack over time is maximized.

[9] Chapter 6 is based on the working paper [23].

We refer to this problem as the incremental knapsack problem (IK for short). A natural question emerges: how much computational tractability do we give up for implementing a richer model in the context of knapsack? By formally studying its computational complexity in Chapter 6, we show that the discrete version of IK with no discounting is strongly NP-hard, whereas the regular knapsack is only weakly NP-hard. The distinction in complexity classes between regular knapsack and incremental knapsack can be attributed to two things. First, the time horizon T introduces a curse of dimensionality on computations, as an adaptation of the standard dynamic programming approach for the knapsack problem has an exponential dependence on T. Second, there is a fundamental trade-off between packing the best subset now given the available budget versus saving up some unused budget for future investment, which is not present in the standard knapsack problem and introduces additional complexity. Our algorithmic results for the discrete IK problem are two folds:

1. We give a constant factor approximation algorithm relying on a novel reduction to a well studied problem known as generalized assignment.

2. We give a PTAS (with worse running time than the constant factor approximation algorithm) by combining linear programming and enumeration based methods.

The above approximation results holds whenever the discounting factor is non-decreasing with respect to time.

For the continuous case, we focus on the special case when the knapsack capacity grows linearly with time. We show that for a special case of discounting functions that we refer to as order inducing, it suffices to determine the *subset* of items that we will pack by the time horizon. With this observation, we develop a FPTAS to determine a near optimal subset of items to pack. We partially complement our algorithmic result with an NP-hardness result for a piecewise linear capacity function with two pieces.

Similarly, understanding the interplay between model complexity and tractability allows us to effectively quantify the trade-offs among different models of consumer purchase behavior in operations management. One key challenge in any assortment planning problem is to find the "right" choice model that can simultaneously capture the purchase behavior of customers based on historical sales data and allow a firm to decide on what products to offer to maximize revenue. Since the complexity of a choice model generally increases with its modeling power, it is critical to choose a model that strikes a balance between predictability and tractability. Many existing random-utility based choice models proposed in the literature are limited by distributional assumptions. The Markov chain model is a distribution-free model introduced by Blanchet et al. [25] to combat the model selection errors that many existing choice models tend to suffer from. In this model, each item (including the no-purchase option) corresponds to a state, and consumer substitution behavior is modeled by transitions in the Markov chain. Whenever the retailer offers an item, the corresponding state becomes absorbing. A random customer arrives to her favorite item according to an initial probability distribution and continue to transition through the Markov chain until she reaches an absorbing state, which corresponds to purchasing an item (or leaving the system if the no-purchase state is reached). The authors showed the Markov chain model is rich enough to subsume the Multinomial Logit Model (MNL), arguably one of the most popular models used in practice; it also approximates many other parametric choice models well.

In Chapter 7,[10] we study the tractability of the assortment optimization problem under the Markov chain choice model. In particular, we focus on scenarios where the retailer is faced with additional operational constraints such as budget, and shelf-space limitations. Mathematically, the problem correspond to modifying a given Marko chain by selecting a subset of its states absorbing in order to maximize revenue. In the capacity constrained problem, there is a cost of selecting a state and the total cost of our solution cannot exceed a given budget. We show that even though the Markov chain model is not as tractable as the MNL model, we can still find a good approximate solution efficiently. Unlike the MNL model, (where cardinality and TU constrained assortment optimization problem can be solved in polynomial time) we show that the same problem is APX-hard (and independent set hard respectively) under the Markov chain choice model. We then proceed to give a 1/2-approximation algorithm for the cardinality constrained problem and a 1/3-approximation algorithm for the capacity constrained problem. Our algorithm increments the assortment one at a time from a consideration set. Subsequently, prices of all remaining items are adjusted to account for their incremental revenue contribution. The price updates allow us to "linearize" a non-linear revenue function, which enables us to provide theoretical worst case guarantees on the performance of our algorithm. Moreover, we demonstrate empirically that the algorithm often returns a near optimal solution and scales gracefully as the problem size grows. Finally, our solution approach is intuitive and offers operational insights into the substitution behavior of customers captured by the Markov chain choice model.

---

[10] Chapter 7 is based on the paper [53].

Chapter 2

A Note on the Assignment Problem with Uniform Preferences

2.1 Introduction

We study the assignment problem, which is concerned with allocating objects to agents, each of whom wishes to receive at most one object. Agents have preferences over the objects, and the goal is to allocate the objects to the agents in a fair and efficient manner. Further, as each agent's preference ordering over the objects is private information, we require the mechanism to be strategyproof: it should be a dominant strategy for the agents to report their preference ordering truthfully. If the objects are divisible, we can think of a fractional assignment in which an object may be allocated in varying amounts to multiple agents so that the total amount allocated of any object is at most 1, and so that each agent receives at most one unit in all. If the objects are indivisible, one can think of a lottery over assignments, which again results in a fractional assignment matrix in which entry (i, a) represents the probability that agent i receives object a. These two views are equivalent for our purposes; while in the rest of the chapter we assume that the objects are indivisible, all of our results extend to the case of divisible objects with the obvious change in interpretation. There is now a rich literature on such models with applications to many real-life allocation problems including allocating students to schools in various cities, the design of kidney exchanges, etc. [1, 2, 39, 103]. The two prominent mechanisms that have emerged from this literature are the *Random Serial Dictatorship* (RSD) mechanism and the *Probabilistic Serial* (PS) mechanism. The PS mechanism is stronger in terms of its efficiency and equity properties, but it is only weakly strategyproof in the strict preference domain and not strategyproof in the full preference domain; whereas the RSD mechanism is strategyproof, but satisfies only a weaker version of efficiency and envy-freeness. Furthermore, Bogomolnaia and Moulin [29] show that no strategyproof mechanism can satisfy the stronger form of efficiency and equity that the PS mechanism satisfies.

This chapter is inspired by the paper of Bogomolnaia and Moulin [30], which characterizes the PS mechanism on a restricted preference domain. The PS mechanism was introduced in an earlier paper of Crés and Moulin [49] that was motivated by the problem of scheduling unit-length jobs with deadlines. Suppose there are n jobs, each requiring a unit processing time, and all jobs are available at time zero. As the jobs all have unit-length, one could think of the scheduling problem as one of assigning time-slots $1, 2, \ldots, n$ to the jobs, so that slot k represents the interval $(k-1, k]$, and a job assigned to slot k finishes at time k. Jobs have deadlines and earn a non-negative utility if they complete before their deadline. Specifically, if the deadline of job j is d_j, then the utility of assigning j to slot k is monotonically decreasing in k until the deadline, after which it drops to zero. That is, if $u_{j,k}$ denotes the utility of assigning job j to slot k, then

$$u_{j,1} > u_{j,2} \ldots > u_{j,d_j} > 0 = u_{j,d_j+1} = u_{j,d_j+2}, \ldots, u_{j,n}.$$

The goal is to use a mechanism to schedule the jobs in a fair and efficient manner based on their reported utility information without the usage of money. Crés and Moulin [49] proposed the PS mechanism and showed that it finds an ordinally efficient and envy-free allocation (all definitions appear in the next section); furthermore, they showed that the PS mechanism is strategyproof on this domain: in the event each job/agent need only report their deadline, they show that it is a weakly dominant strategy for each job to report its deadline truthfully. Bogomolnaia and Moulin [30] characterize the PS mechanism on this restricted domain in two different ways: first, they show that ordinal efficiency and envy-freeness characterize the PS outcome on this restricted domain; and second, they show that it is the only strategyproof mechanism that is ordinally efficient and treats equals equally. Taken together, their result shows that the PS mechanism is perhaps the only compelling mechanism on this restricted preference domain. (Crés and Moulin [49] showed that the PS mechanism is in fact group strategyproof, although this stronger property is not needed in their characterization result of PS.)

In this chapter we consider a slightly more general domain, again inspired by the problem of scheduling unit-length jobs. For simplicity, assume there are n agents and n objects, and suppose the objects are arranged in the order $(1, 2, \ldots, n)$ by all the agents. Each agent's preference ranking, however, is determined by a *weakly* decreasing utility function over the objects, in contrast to a strictly decreasing utility function over the objects till a deadline. (A good way to visualize this preference domain is to have each agent separate the sequence of objects into indifference classes, without disturbing the common order on the objects.) This domain is quite natural in the scheduling context, where completing a job early is always (weakly) better, but jobs may be insensitive to completion times within a certain time interval, and these intervals may change from job to job. The domain considered in the earlier papers is a special case in which, for each agent, all but the final indifference class has a single object. It is then natural to ask if the two characterizations of PS extend to this domain. It turns out that the answer is negative in each case. We show that the PS outcome (actually, a correspondence) is no longer the only outcome that is ordinally efficient and envy-free, nor is the PS mechanism strategyproof on this domain. Somewhat surprisingly, we show that:

- No weakly strategyproof mechanism can satisfy both ex post efficiency and envy freeness on this domain, when there are three or more agents; and
- No strategyproof mechanism can satisfy both ordinal efficiency and equal treatment of equals on this domain, when there are four or more agents.

2.1.1 Related Literature

The literature on random assignment problems focuses on simultaneously satisfying various notions of fairness, efficiency, and strategyproofness, and several impossibility results have been established over the last two decades [10, 29, 41, 75, 76, 131]. Our two main impossibility results are strengthened versions of similar results in the literature in which preferences are drawn from richer domains. Specifically, versions of the two impossibility results have been obtained by [76] on the full preference domain (where any weak ordering of the objects is permissible), and by [29] on the strict preference domain (where any *strict* ordering of the objects is permissible). Thus the surprising element in our result is that these difficulties persist even in domains in which the preferences are severely restricted.

Our work contributes to the rich and growing literature on matching and allocation problems in which monetary transfers are not permitted. The PS mechanism and the Random Serial Dictatorship mechanisms are central mechanisms for such allocationproblems and have been studied extensively from several points of view, see the recent survey [116] for an overview. This has also inspired other characterizations and extensions of the PS mechanism [8, 10, 28, 72, 77, 78]. There is an equally extensive literature on models where monetary transfers are allowed to restore fairness or strategyproofness in a queueing or scheduling setting [55, 87, 88, 93, 120], and we refer the reader to the survey [73] for a comprehensive overview.

2.2 Preliminaries

2.2.1 Model and Definitions

An assignment problem is given by a triple $(N, O, \succsim)$, where $N = \{1, \ldots, n\}$ is the set of agents, $O = \{o_1, \ldots o_n\}$ is the set of objects, and the preference profile $\succsim = (\succsim_1, \ldots, \succsim_n)$ specifies each agent's preference ordering over the objects. If the number of agents is not the same as the number of objects, one can always balance such a problem by adding dummy agents or dummy objects. We will assume that the preference relation of each agent is complete (every pair of objects is comparable) and transitive. By $a \succsim_i b$, we mean that agent i weakly prefers object a to object b. We write $a \succ_i b$ if i strictly prefers a to b, i.e. $a \succsim_i b$ but $b \not\succsim_i a$; and we use $a \sim_i b$ when i is indifferent between a and b, i.e. $a \succsim_i b$ and $b \succsim_i a$. Note that the indifference relation is also transitive. Thus each agent has a most-preferred subset of objects (and the agent is indifferent between all the objects within this set), followed by a most-preferred subset of objects among the remaining ones, etc.

In this chapter, we shall consider the *uniform* preference domain in which $o_1 \succsim_i o_2 \succsim_i \ldots \succsim_i o_n$ for every agent $i \in N$. Agents differ in their preference ordering only in their strict preference relation $\succ_i$ (and hence their indifference relation $\sim_i$). In the rest of the chapter, we use the following notation for the preference ordering of the agents: all the objects within an indifference class for an agent appear within braces in that agent's preference list, and these maximal indifference classes are separated by a comma; objects are always written in subscript order; and the braces are omitted for singleton indifference classes. Thus, the preference ordering

$$o_1 \succ_i o_2 \sim_i o_3 \sim_i o_4 \succ_i o_5$$

for agent i is written as

$$i : o_1, \{o_2\ o_3\ o_4\}, o_5.$$

By a mechanism, we mean a mapping from the set of all preference profiles (within this restricted domain) to a doubly stochastic matrix, which we call the assignment matrix for that profile. The assignment matrix is *deterministic* if its entries are $\{0, 1\}$ (and so the outcome is a *matching* of the agents and objects); otherwise, it is *probabilistic*. When the matrix is deterministic, the ij-th entry indicates whether agent i receives object j. When the matrix is probabilistic, then its ij-th entry represents the probability that agent i receives object j. If a mechanism maps each preference profile to a deterministic matrix, the mechanism is deterministic; otherwise the mechanism is probabilistic. (Alternatively, we could have defined a probabilistic mechanism as a lottery over deterministic mechanisms. In this view, different lotteries are regarded as different mechanisms, even if they result in the same assignment matrix for each preference profile.) As a consequence of the Birkhoff-von Neumann theorem [24], the outcome of a probabilistic mechanism can be implemented as a lottery over deterministic assignments.

Given two probabilistic assignments P and Q, we say that agent i prefers P to Q if P_i, the i-th row of P stochastically dominates Q_i according to i's preferences. Formally,

$$P_i \succsim_i Q_i \Longleftrightarrow \sum_{k:k \succsim_i j} p_{ik} \geq \sum_{k:k \succsim_i j} q_{ik}, \quad \forall j \in O.$$

We say that i strictly prefers P to Q, denoted by $P_i \succ_i Q_i$, if at least one of the inequalities in the above definition is strict. Note that this definition is only a *partial* order, as an agent may not be able to compare two probabilistic allocations. Finally, we say that P *stochastically dominates* Q, denoted by $P \succsim Q$, if $P_i \succsim_i Q_i$ for all $i \in N$, with $P_i \succ_i Q_i$ for some $i \in N$. Again, this notion of stochastic dominance defines a partial order on the set of doubly stochastic matrices.

2.2.2 Desirable Properties

We define some desirable properties of mechanisms that play an important role in the rest of the chapter.

Ordinal Efficiency An assignment matrix P is *ordinally efficient* if it is not stochastically dominated by any other random assignment matrix Q such that $Q \succsim P$. It is well known that any ordinally efficient matrix can be implemented as a lottery over deterministic Pareto efficient assignments. Furthermore, checking whether or not a given assignment matrix is ordinally efficient is computationally easy [29, 76].

Ex post Efficiency A weaker notion of efficiency that we will consider is ex post efficiency. A bistochastic matrix P is *ex post efficient* if it can be written as a convex combination of Pareto efficient assignments.

Envy-Freeness An assignment matrix P is *envy free* if the probabilistic assignment of every agent i stochastically dominates the probabilistic assignment of every other agent with respect to agent i's preference ordering. Let P_i denote the probabilistic assignment of agent i in the matrix P. Then, P is envy-free if $P_i \succsim_i P_{i'}$ for all $i, i' \in N$.

Equal Treatment of Equals An assignment matrix P satisfies *equal treatment of equals* if agents with identical preferences get equivalent allocations. Formally, P satisfies equal treatment of equals if for all $i, i' \in N$ such that $\succsim_i = \succsim_{i'} = \succsim$, we have

$$\sum_{k:k \succsim j} p_{ik} = \sum_{k:k \succsim j} p_{i'k}, \quad \forall j \in O.$$

Strategyproofness The properties defined so far pertain to the outcome on a single profile. Strategyproofness, however, is a property of the mechanism, in particular, on how the mechanism behaves on pairs of profiles in which all but one of the agents report the same preference ordering. A mechanism is *strategyproof* if it is a weakly dominant strategy for each agent to report her true preference ordering. Let $P_i(\succsim)$ be the allocation matrix when the reported preference profile is $\succsim$. Formally, a mechanism is strategyproof if

$$P_i(\succsim_i, \succsim_{-i}) \ \succsim_i \ P_i(\succsim'_i, \succsim_{-i}),$$

for all agents $i \in N$, and for all preference profiles $\succsim_{-i}$ of the other agents, and for every pair of preferences $\succsim_i, \succsim'_i$ that i could report. A random assignment mechanism is *weakly strategyproof* if for each $i \in N$, and for each preference profile $\succsim_{-i}$ of the other agents, there does not exist preference ordering $\succsim'_i$ such that $P_i(\succsim'_i, \succsim_{-i}) \succ_i P_i(\succsim_i, \succsim_{-i})$. In a strategyproof mechanism, the assignment under truthful reporting stochastically dominates the assignment under any other report; in a weakly strategyproof mechanism, however, reporting her preference ordering truthfully will not result in an assignment that is stochastically dominated by the assignment under any other report. It is clear from the definitions that strategyproofness implies weak strategyproofness, but not vice-versa.

2.2.3 The Extended Probabilistic Serial Mechanism

We end this section with a very brief description of the EPS mechanism [76]. The EPS mechanism, like the PS mechanism, can be described as a "cake-eating" mechanism in which agents consume their best object(s) at unit rate. Roughly, each agent *simultaneously* consumes her "best set" of available objects at a unit rate at each point in time. If all the preferences are strict, this determines a unique allocation for the agents; when agents have indifferences, this mechanism is not well-defined as each agent has a choice on how her unit rate is apportioned across the objects in her best set of objects. For instance, if agent i strictly prefers a to b, whereas agent i' is indifferent between a and b, letting both agents consume a initially will result in each agent getting 1/2 of a and 1/2 of b, which is clearly inefficient in the ordinal sense; if i' consumes b at rate 1, however, the outcome is ordinally efficient. To address this issue, Katta and Sethuraman [76] proposed the EPS mechanism that:

1. Identifies a subset $S^\star$ of agents with the least collective claim over the union of their best objects $C(S^\star)$ (in terms of average claim per agent within the subset); (We will refer to $S^\star$ as the bottleneck set.)

2. Assigns each agent in $S^\star$ an amount of $\frac{|C(S^\star)|}{|S^\star|}$ of their favorite object(s);

3. Promises the rest of the agents an amount of at least $\frac{|C(S^\star)|}{|S^\star|}$ of their favorite object(s); and

4. Removes the allocated objects, and recurses on the subproblem (agents in $S^\star$ now start consuming their favorite objects(s) out of the remaining objects.)

The authors showed that the bottleneck sets can be identified by solving a sequence of parametric max flow problems. We refer the reader to their paper for a complete description of the algorithm.

Note that in the full preference domain, an agent is insensitive to different probabilistic allocations of objects within the same indifference class as long as the allocations sum up to the same quantity for every indifference class. This motivates the following equivalence relation over the set of assignment matrices. Given a preference profile $\succsim$, let $\mathcal{I}_i$ be the collection of indifference classes of objects for agent i. For every $I \in \mathcal{I}_i$, let $p_{iI} = \sum_{o_j \in I} p_{ij}$. We say that two random assignment matrices P and Q

are equivalent if and only if

$$p_{iI} = q_{iI} \quad \forall i \in N, I \in \mathcal{I}_i.$$

One can check that this defines an equivalence relation on the set of assignment matrices. An assignment matrix is an *EPS assignment* if it is equivalent to the random assignment found by the EPS mechanism.

2.3 Main Results

Bogomolnaia and Moulin [30] showed that if the preference domain is further restricted so that the acceptable set of objects for each agent i is the set $\{o_1, o_2 \ldots, o_{k_i}\}$, and if the agents have strict (and uniform) preferences over their acceptable objects, then the PS outcome is characterized by ordinal efficiency and envy-freeness, and that it is the only strategyproof mechanism that guarantees ordinal efficiency and equal treatment of equals. We show that neither one of these results holds when the agents have weak preferences.

2.3.1 Non-uniqueness of Ordinally Efficient and Envy Free Assignments

The EPS mechanism finds an equivalence class of ordinally efficient and envy free assignments for each preference profile. However, there are other assignments with these properties. For the preference profile below the following assignment is ordinally efficient and envy free.

$$\begin{array}{lllll} 1: & o_1, & \{o_2 & o_3\}, & o_4 \\ 2: & o_1, & \{o_2 & o_3\}, & o_4 \\ 3: & \{o_1 & o_2\}, & o_3, & o_4 \\ 4: & \{o_1 & o_2\}, & o_3, & o_4 \end{array} \qquad \begin{array}{ccccc} & o_1 & o_2 & o_3 & o_4 \\ 1: & \frac{1}{4} & 0 & \frac{1}{2} & \frac{1}{4} \\ 2: & \frac{1}{4} & 0 & \frac{1}{2} & \frac{1}{4} \\ 3: & \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{4} \\ 4: & \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{4} \end{array}$$

However, the EPS mechanism will not compute the above assignment since agents 1 and 2 strictly prefer o_1 to o_2 whereas agents 3 and 4 are indifferent between o_1 and o_2. Thus, in the EPS mechanism, agents 3 and 4 consume o_2 first so as to not compete with agents 1 and 2 for their unique best object.

Consequently, EPS finds the following assignment

$$\begin{array}{ccccc} & o_1 & o_2 & o_3 & o_4 \\ 1: & \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} \\ 2: & \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} \\ 3: & 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 4: & 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{array}$$

Clearly the two assignments do not belong to the same equivalence class: agents 1 and 2 strictly prefer the latter, whereas agents 3 and 4 strictly prefer the former.

2.3.2 Impossibility Results

Theorem 2.1. *For $n \geq 3$, any mechanism that is both ex-post efficient and envy-free is not weakly strategyproof in the uniform preference domain.*

Proof. We first show the impossibility result for $n = 3$. Consider Profile 1 (below). Clearly, the set of envy-free (EF) assignments at this profile is as described for some $0 \leq y \leq 1/6$.

$$\begin{array}{llll} \text{Profile 1} & & & \\ 1: & o_1, & o_2, & o_3 \\ 2: & o_1, & \{o_2 & o_3\} \\ 3: & o_1, & o_2, & o_3 \end{array} \qquad \begin{array}{cccc} & o_1 & o_2 & o_3 \\ 1: & \frac{1}{3} & \frac{1}{2} - y & \frac{1}{6} + y \\ 2: & \frac{1}{3} & 2y & \frac{2}{3} - 2y \\ 3: & \frac{1}{3} & \frac{1}{2} - y & \frac{1}{6} + y \end{array}$$

By the structure of the preferences in Profile 1, agent 2 cannot receive object o_2 in any Pareto efficient assignment, as there is always a Pareto improvement with the agent who is assigned o_3 in the same assignment. Thus $y = 0$ in any ex-post efficient (EPE) assignment.

Similarly, in Profile 2 below, the set of envy-free assignments is as described for some $0 \leq w \leq \frac{1}{6}$ and $0 \leq z \leq \frac{1}{12}$.

$$\begin{array}{l} \text{Profile 2} \\ 1: o_1, o_2, o_3 \\ 2: o_1, \{o_2\ o_3\} \\ 3: \{o_1\ o_2\}, o_3 \end{array} \qquad \begin{array}{cccc} & o_1 & o_2 & o_3 \\ 1: & \frac{1}{2} - w & \frac{1}{4} + w - z & \frac{1}{4} + z \\ 2: & \frac{1}{2} - w & w + 2z & \frac{1}{2} - 2z \\ 3: & 2w & \frac{3}{4} - 2w - z & \frac{1}{4} + z \end{array}$$

Again, agent 2 cannot be assigned o_2 in any Pareto efficient assignment, as there is always a Pareto improvement with the agent assigned o_3 in the same assignment. Hence, $w = z = 0$ in any ex-post efficient assignment.

Observe that the properties of ex-post efficiency and envy-freeness determine a unique assignment in both Profile 1 and Profile 2. Furthermore, agents 1 and 2 have the same preferences in both profiles, but agent 3's allocation in Profile 1 stochastically dominates his allocation in Profile 2, implying a failure of weak strategyproofness.

For $n \geq 4$, extend each of the profiles as follows: the first 3 agents have exactly the same preference ordering over the first 3 objects; and they have strict preferences over the objects $o_4, o_5, \ldots, o_n$; finally, agent i (for $i \geq 4$) is indifferent between the first i objects, after which he has strict preferences over the others. That is, i's preference ordering is

$$i : \{o_1 \ldots o_i\}, o_{i+1}, \ldots, o_n.$$

It is straightforward to check that agent i receives object o_i in every Pareto efficient assignment, and so the first 3 agents must be allocated the first 3 objects, leading to the same two profiles analyzed earlier. □

As the EPS mechanism is ordinally efficient (and so ex-post efficient as well) and envy-free, an immediate consequence is that the EPS mechanism is not weakly strategyproof on the uniform domain. The Random Serial Dictatorship (RSD) mechanism, which orders the agents uniform at random, and lets them successively choose a favorite object in that order, can be adapted to the setting of indifferences. RSD is both strategyproof and ex-post efficient [1, 102, 121], and so fails envy-freeness on the uniform domain. For the domain considered by Bogomolnaia and Moulin, neither of these results hold, as the PS mechanism is strategyproof and the RSD mechanism is envy-free.

Next, we show that if we relax envy freeness to equal treatment of equals, but strengthen weak strategyproofness and ex-post efficiency to strategyproofness and ordinal efficiency respectively, a similar impossibility result holds for the uniform preference domain.

Theorem 2.2. *For $n \geq 4$, any mechanism that satisfies ordinal efficiency and equal treatment of equals is not strategyproof in the uniform preference domain.*

Proof. We first show the result for $n = 4$. We will consider eight different profiles and show that in Profile 8 there is no probabilistic assignment that simultaneously satisfies ordinal efficiency (OE), equal treatment of equals (ETE), and strategyproofness (SP) in relation to the first seven profiles.

First, we compute the probability assignment for Profile 1. Notice that the only assignment that satisfies ETE is as follows:

Profile 1						o_1	o_2	o_3	o_4
1:	o_1,	o_2,	o_3,	o_4	1:	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
2:	o_1,	o_2,	o_3,	o_4	2:	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
3:	o_1,	o_2,	o_3,	o_4	3:	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
4:	o_1,	o_2,	o_3,	o_4	4:	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Consider Profile 2. Let p_{ij} be the probability that agent i is assigned the object o_j. By ordinal efficiency, $p_{41} = 0$. For otherwise $p_{42} < 1$, which means that at least one of p_{12}, p_{22}, p_{32} is strictly positive; this agent can exchange a small amount of o_2 for an equal amount of o_1 from agent 4, without altering any of the other allocations, to obtain a new allocation matrix that stochastically dominates the current one, which violates ordinal efficiency.

By strategyproofness, we must have that $p_{41} + p_{42} = \frac{1}{2}$, because if it were not the case, then there is a profitable deviation of agent 4 either from Profile 1 to Profile 2 or vice versa. Thus, we get $p_{42} = \frac{1}{2}$ since $p_{41} = 0$. Similarly, by strategyproofness, we have $p_{41} + p_{42} + p_{43} = \frac{3}{4}$, which implies $p_{43} = \frac{1}{4}$ and $p_{44} = \frac{1}{4}$.

Finally by ETE, we know the probability assignment of the first three agents must be identical, thus we get the following assignment:

Profile 2						o_1	o_2	o_3	o_4
1:	o_1,	o_2,	o_3,	o_4	1:	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{4}$
2:	o_1,	o_2,	o_3,	o_4	2:	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{4}$
3:	o_1,	o_2,	o_3,	o_4	3:	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{4}$
4:	$\{o_1$	$o_2\}$,	o_3,	o_4	4:	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

Consider Profile 3. By SP in relation to Profile 2, we must have $p_{31} + p_{32} = \frac{1}{2}$, $p_{33} = \frac{1}{4}$, and $p_{34} = \frac{1}{4}$. By ETE, the assignment for agent 4 satisfies the same constraints as that of agent 3. By OE, $p_{31} = p_{41} = 0$, because either $p_{31} > 0$ or $p_{41} > 0$ would imply that $p_{32} + p_{42} < 1$ (as $p_{31} + p_{32} + p_{41} + p_{42} = 1$) or equivalently that $p_{12} + p_{22} > 0$. Then again we have a situation where agent 1 or 2 can exchange a small amount of o_2 for an equal amount of o_1 from agent 3 or 4, which leads to a new assignment matrix that stochastically dominates the current one, violating OE. Thus, OE and SP together determine the probabilistic assignment for agents 3 and 4. Now, we can fill in the assignments for agents 1 and 2 using ETE to get:

Profile 3						o_1	o_2	o_3	o_4
1:	o_1,	o_2,	o_3,	o_4	1:	$\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{1}{4}$
2:	o_1,	o_2,	o_3,	o_4	2:	$\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{1}{4}$
3:	$\{o_1$	$o_2\}$,	o_3,	o_4	3:	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
4:	$\{o_1$	$o_2\}$,	o_3,	o_4	4:	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

Consider Profile 4. By SP in relation to Profile 3 and ETE, we must have $p_{11} + p_{12} = p_{31} + p_{32} = p_{41} + p_{42} = \frac{1}{2}$, $p_{13} = p_{33} = p_{43} = \frac{1}{4}$, and $p_{14} = p_{34} = p_{44} = \frac{1}{4}$. Since $p_{12}+ p_{32} + p_{42} \leq 1$, in order to satisfy the unit demand for agents 1, 3, and 4, we must have that at least one of p_{11}, p_{31}, p_{41} is strictly positive. Thus by OE, we must have $p_{22} = 0$ and $p_{21} = \frac{1}{2}$. Although we cannot pin down a single assignment for this profile, any feasible assignment must be of the form:

Profile 4		o_1	o_2	o_3	o_4
1:$\{o_1, o_2\}$, o_3, o_4	1:	x	$\frac{1}{2} - x$	$\frac{1}{4}$	$\frac{1}{4}$
2: o_1, o_2, o_3, o_4	2:	$\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{1}{4}$
3: $\{o_1\ o_2\}$, o_3, o_4	3:	y	$\frac{1}{2} - y$	$\frac{1}{4}$	$\frac{1}{4}$
4: $\{o_1\ o_2\}$, o_3, o_4	4:	$\frac{1}{2} - x - y$	$x + y$	$\frac{1}{4}$	$\frac{1}{4}$

for some $x, y \geq 0$ and $x + y \leq \frac{1}{2}$.

Consider Profile 5. Applying the same argument of ordinal efficiency for agent 4 in Profile 2 to agent 2 in Profile 5, we get $p_{22} = 0$. By strategyproofness in relation to Profile 1, we must have $p_{21} = \frac{1}{4}$ and $p_{21} + p_{22} + p_{23} = \frac{3}{4}$. These imply $p_{23} = \frac{1}{2}$ and $p_{24} = \frac{1}{4}$. Finally by ETE, we know the probability assignment of the agents 1, 2 and 4 must be identical, resulting in the following assignment:

Profile 5						o_1	o_2	o_3	o_4
1:	o_1,	o_2,	o_3,	o_4	1:	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$
2:	o_1,	$\{o_2$	$o_3\}$,	o_4	2:	$\frac{1}{4}$	0	$\frac{1}{2}$	$\frac{1}{4}$
3:	o_1,	o_2,	o_3,	o_4	3:	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$
4:	o_1,	o_2,	o_3,	o_4	4:	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$

Consider Profile 6. By SP in relation to Profile 2, we must have $p_{21} = \frac{1}{3}$, $p_{22} + p_{23} = \frac{5}{12}$, and $p_{24} = \frac{1}{4}$. By OE, we must have $p_{22} = 0$, which implies $p_{23} = \frac{5}{12}$. By SP in relation to Profile 5, we must have $p_{41} + p_{42} = \frac{7}{12}$, $p_{43} = \frac{1}{6}$, $p_{44} = \frac{1}{4}$. Again, by OE, we must have $p_{41} = 0$, which implies $p_{42} = \frac{7}{12}$. Subsequently, we can fill in the assignment for agents 1 and 3 using ETE to get:

Profile 6						o_1	o_2	o_3	o_4
1:	o_1,	o_2,	o_3,	o_4	1:	$\frac{1}{3}$	$\frac{5}{24}$	$\frac{5}{24}$	$\frac{1}{4}$
2:	o_1,	$\{o_2$	$o_3\}$,	o_4	2:	$\frac{1}{3}$	0	$\frac{5}{12}$	$\frac{1}{4}$
3:	o_1,	o_2,	o_3,	o_4	3:	$\frac{1}{3}$	$\frac{5}{24}$	$\frac{5}{24}$	$\frac{1}{4}$
4:	$\{o_1$	$o_2\}$,	o_3,	o_4	4:	0	$\frac{7}{12}$	$\frac{1}{6}$	$\frac{1}{4}$

Consider Profile 7. By SP in relation to Profile 3, we must have $p_{21} = \frac{1}{2}$, $p_{22} + p_{23} = \frac{1}{4}$, and $p_{24} = \frac{1}{4}$. By OE, we must have $p_{22} = 0$, which implies $p_{23} = \frac{1}{4}$. By SP in relation to Profile 6, we have $p_{31}+p_{32} = \frac{13}{24}$, $p_{33} = \frac{5}{24}$ and $p_{34} = \frac{1}{4}$. By ETE, agent 4 gets an equivalent assignment as agent 3. Notice that in this case, we must have $p_{31} > 0$ and $p_{41} > 0$ as $p_{32}+p_{42} \leq 1$ and $p_{31}+p_{32}+p_{41}+p_{42} = \frac{13}{12} > 1$, so either p_{31} or p_{41} is strictly positive. This implies $p_{12} = p_{22} = 0$ in order to satisfy OE. Thus, we get an assignment of the following form:

Profile 7		o_1	o_2	o_3	o_4
1: o_1, o_2, o_3, o_4	1:	$\frac{5}{12}$	0	$\frac{1}{3}$	$\frac{1}{4}$
2: o_1, $\{o_2\ o_3\}$, o_4	2:	$\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{1}{4}$
3: $\{o_1\ o_2\}$, o_3, o_4	3:	z	$\frac{13}{24} - z$	$\frac{5}{24}$	$\frac{1}{4}$
4: $\{o_1\ o_2\}$, o_3, o_4	4:	$\frac{1}{12} - z$	$\frac{11}{24} + z$	$\frac{5}{24}$	$\frac{1}{4}$

Finally, consider Profile 8. By SP in relation to Profile 4, we have $p_{21} = \frac{1}{2}$, $p_{22} + p_{23} = \frac{1}{4}$, $p_{24} = \frac{1}{4}$. By OE, we have $p_{22} = 0$, which implies $p_{23} = \frac{1}{4}$. By SP in relation to Profile 7 and ETE, we must have $p_{11} + p_{12} = p_{31} + x_{32} = x_{41} + x_{42} = \frac{5}{12}$, $p_{13} = p_{33} = p_{43} = \frac{1}{3}$, $p_{14} = p_{34} = p_{44} = \frac{1}{4}$. Now consider the partially filled assignment below:

Profile 8						o_1	o_2	o_3	o_4
1:	$\{o_1$	$o_2\}$,	o_3,	o_4	1:	?	?	$\frac{1}{3}$	$\frac{1}{4}$
2:	o_1,	$\{o_2$	$o_3\}$,	o_4	2:	$\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{1}{4}$
3:	$\{o_1$	$o_2\}$,	o_3,	o_4	3:	?	?	$\frac{1}{3}$	$\frac{1}{4}$
4:	$\{o_1$	$o_2\}$,	o_3,	o_4	4:	?	?	$\frac{1}{3}$	$\frac{1}{4}$

Note that this assignment allocates more than one unit of object 3. Since we used the necessary conditions induced by SP, OE, ETE to pin down all possible assignments for each of the Profiles 1-7, and none of these leads to a valid allocation for Profile 8, it is impossible to write down a random assignment for Profile 8 that simultaneously satisfies ETE, OE, and SP in relation to the other 7 profiles.

For general $n \geq 5$, we extend each of the 8 profiles as follows: agents 1 through 4 have the same preference for objects o_1 through o_4; moreover, these agents have a strict preference ordering for the rest of the objects. For every $j = 5, \ldots, n$, agent j is indifferent amongst objects o_1 through o_j and has strict preference for the rest of the objects. Similar to the argument made for general n in Theorem 2.1, we see that by OE, every agent $j = 5, \ldots, n$ receives object o_j with probability 1. Consequently, the first 4 agents must be allocated the first 4 objects, leading to the same 8 profiles analyzed earlier.

$\square$

Chapter 3

Cake Cutting Algorithms for Piecewise Constant and Piecewise Uniform Valuations

3.1 Introduction

Cake cutting is a fundamental problem that is concerned with the fair division of resources among competing agents, see e.g. [34, 101, 97]. This basic problem comes up in many applications including the division of rent among housemates, disputed land between land-owners, and work among co-workers. The framework is general enough to encapsulate the important problem of allocating a heterogeneous divisible good among multiple agents with different preferences: for example, scheduling the use of a valuable divisible resource such as server time [67].

We approach the cake cutting problem from a mechanism design perspective. The cake is modeled by the interval $[0, 1]$; and each cake recipient—who we will refer to as *an agent*— has a private value density function over the cake that is *piecewise constant*. We consider three of the most enduring goals in mechanism design and fair division: fairness, Pareto efficiency, and strategyproofness. Since many fair division algorithms may need to be deployed on a large scale, we will also aim for algorithms that are *computationally* efficient. The main research question in this chapter is as follows: *among the various definitions of fairness, Pareto efficiency, strategyproofness, and efficient computability, what* *are the maximal sets of properties that can be satisfied simultaneously?* Our main contribution is a detailed study of this question and include the design of a number of desirable cake cutting algorithms satisfying many of the properties. Our algorithms rely on transforming the cake-cutting problem to an equivalent problem of allocating objects to agents where each agent has a homogeneous preference for each object, similar to the classical assignment model. The transformation is done by pre-cutting the cake into subintervals using the union of discontinuity points of the agents' valuation functions. This transformation allows us to adapt some well-known results in the random assignment and market equilibrium literatures to the cake-cutting problem.

Drawing on the connection between cake cutting and random assignment, we present *CCEA (Controlled Cake Eating Algorithm)* for piecewise constant valuations. CCEA is a polynomial-time algorithm and satisfies robust envy-freeness and robust proportionality, which are stronger than the notions of fairness that have been considered in the traditional cake cutting literature. (Formal definitions of these properties appear in Section 3.2.1.) Informally, an allocation is robust envy-free if it remains envy-free even if an agent re-adjusts or perturbs his value density function, as long as the ordinal information of the function is unchanged.[11] CCEA uses generalizations [76, 8] of the *PS (probabilistic serial)* algorithm introduced by [29] for the random assignment problem.[12]

While CCEA satisfies some appealing properties, the allocation it finds may not be Pareto efficient. Motivated by this shortcoming of CCEA, we design an alternative algorithm called the *MEA (Market Equilibrium Algorithm)*, which relies on the solution to an Eisenberg-Gale convex programming formulation for market equilibrium. MEA is a deterministic, polynomial-time algorithm that is Pareto efficient, envy-free, and proportional for piecewise constant valuations. The algorithm developed for solving the convex program often relies on solving a sequence of max flow subroutines. The original algorithm is due to [100], who also used it for finding an α-envy-free allocation for general cake cutting valuations. Subsequently, Devanur et al. [54] developed a variant of the algorithm that runs in polynomial time. Although similar ideas using linear programs have been used explicitly to compute

[11] Although full information is a standard assumption in cake cutting, it can be argued that it is unrealistic that agents have exact Von Neumann-Morgenstern utilities for each segment of the cake. Even if they do report exact VNM utilities, they may be uncertain about these reports.

[12] The CC algorithm of [8] is a generalization of the EPS algorithm [76] which in turn is a generalization of PS algorithm of [29].

envy-free allocations in cake-cutting (see e.g. [48, 32]), they do not necessarily return a Pareto efficient allocation.

Although CCEA and MEA are desirable algorithms, they are not strategyproof for piecewise constant valuations. This motivates us to consider two questions: first, are there special cases of valuations for which these algorithms are strategyproof? And second, are there other algorithms that are strategyproof and satisfy the properties that CCEA and MEA satisfy? To answer the first question, we consider the case in which agent valuations are *piecewise uniform*—the special case of piecewise constant valuations in which each agent's value density function takes on at most one positive value. In this case, CCEA and MEA are not only strategyproof, but also group strategyproof; furthermore, these two algorithms coincide for the case of piecewise uniform valuations! Previously, Chen et al. [44] presented a deterministic, strategyproof, polynomial-time, envy-free and Pareto efficient algorithm for piecewise uniform valuations. We prove that for piecewise uniform valuations, CCEA and MEA are in fact equivalent to their algorithm. In a recent paper, Tian [123] characterized a class of strategyproof and Pareto efficient mechanisms for cake cutting when agents have piecewise uniform valuation functions. The algorithm of Tian involves maximizing the sum of concave functions over the set of feasible allocations. It is worth noting that MEA when restricted to the piecewise uniform valuation setting is a member of this family of algorithms characterized by Tian. To answer the second question, we show that no strategyproof algorithm satisfies the properties that CCEA or MEA satisfies when agent value density functions are piecewise constant. Unlike the piecewise uniform valuation setting, where each agent only cares about obtaining as much of their desired pieces of the cake as possible, an agent with a piecewise constant valuation cares about the trade off in quantities of having pieces at different levels of desirability. We lose strategyproofness when going from piecewise uniform to piecewise constant valuation function because when the agents have more flexibility in expressing their preferences, they are more likely to be able to manipulate an algorithm.

A key difficulty in obtaining a strategyproof algorithm via the transformation to an assignment problem is that the discontinuity points of each agent's valuation function is private information for the agent. Consequently, the "objects" that we obtain (under the transformation) by pre-cutting the cake can be potentially manipulated by the agents. Unlike allocating multiple homogeneous objects that are well specified in a random assignment setting, a misreporting agent in the cake cutting problem may actually have a heterogeneous preference over an "object" that he reports to have a homogeneous preference over. As a result, even though it is sufficient for a strategyproof algorithm in random assignment to output just the fractional amount as opposed to the actual piece of each object that an agent will receive, the *conversion* from fractions of intervals into an actual allocation is also a necessary step in a cake cutting algorithm. Moreover, this step needs to be done properly in order to prevent agent manipulations. To drive this point further, we describe an algorithm that is strategyproof in the random assignment setting, but is no longer strategyproof if we implement the conversion process from fractions of intervals to the union of subintervals in a deterministic fashion.

Our final algorithm, called *MCSD (Mixed Constrained Serial Dictatorship)*, addresses this difficulty: It is strategyproof in expectation, robust proportional, and satisfies unanimity. For the important case of two agents,[13] it is polynomial-time, and robust envy-free. To the best of our knowledge, it is the first cake cutting algorithm for piecewise constant valuations that satisfies strategyproofness, (ex post) proportionality, and (ex post) unanimity at the same time. MCSD requires some randomization to achieve strategyproofness in expectation. However, MCSD gives the same utility guarantee (with respect to the reported valuation functions) over all realizations of the random allocation. Although MCSD uses some essential ideas of the well-known *serial dictatorship* rule for discrete allocation, it is more involved. First, we constrain an agent's allocation by requiring that each time a dictator is chosen from a random ordering, the piece he takes has to be of maximum value $1/n$ length of the total size of the cake. Next, MCSD derandomizes the allocation obtained from all n! different permutations and aggregate them in a suitable manner.

Our main results are as follows.

Theorem 3.1. *For piecewise constant valuations, there exists a deterministic polynomial time algorithm (CCEA) that is robust envy-free and non-wasteful.*

Theorem 3.2. *For piecewise constant valuations, there exists a deterministic polynomial time algorithm (MEA) that is Pareto efficient and envy-free.*

Theorem 3.3. *For piecewise uniform valuations, there exist deterministic polynomial time algorithms (CCEA and MEA) that are group strategyproof, robust envy-free and Pareto efficient.*

---

[13] Many fair division problems involve disputes between two parties.

Theorem 3.4. *For piecewise constant valuations, there exists a randomized algorithm (MCSD) that is (ex post) robust proportional, (ex post) symmetric, and (ex post) unanimous and strategyproof in expectation. For two agents, it is also polynomial-time and robust envy-free.*

Our positive results are complemented by the following impossibility theorems. These impossibility theorems show that the properties satisfied by CCEA and MEA are maximal subsets of properties that can be satisfied by any algorithm.

Theorem 3.5. *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is strategyproof, robust proportional, and non-wasteful.*

Theorem 3.6. *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is strategyproof, Pareto efficient, and proportional.*

Theorem 3.7. *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is both Pareto efficient and robust proportional.*

As a consequence of CCEA and MEA, we generalize the positive results on piecewise uniform valuations in [44] to handle more general valuation functions, and the results in [48] for piecewise constant valuations to achieve stronger fairness and efficiency guarantees.

3.1.1 Related Work

A mathematical analysis of cake cutting started with the work of Polish mathematicians Steinhaus, Knaster, and Banach (see e.g. [117]). As applications of fair division have been identified in various multiagent settings, a topic which was once considered a mathematical curiosity has developed into a full-fledged sub-field of mathematical social sciences (see e.g. [91]). In particular, in the last few decades, the literature of cake cutting has grown considerably (see e.g. [34, 101, 91, 96]).

The cake cutting literature has been concerned with designing algorithms to allocate a cake fairly. The most important criteria of a fair allocation are *envy-freeness* and *proportionality*. In an envy-free allocation, each agent considers his allocation at least as good as any other agent's allocation. Stromquist [118] and Su [119] showed that an envy-free allocation is guaranteed to exist . In a proportional allocation, each agent gets at least $1/n$ of the value he assigns to the cake. Envy-freeness is generally a stronger notion than proportionality.[14]

Brams and Taylor [33] designed an envy-free cake cutting algorithm for an arbitrary number of players. Although their algorithm is guaranteed to eventually terminate, its running time is unbounded. Moreover, their algorithm can divide the cake into infinitely small segments, which may be unrealistic in some applications. Since the result of [33], researchers have examined restricted value density functions and proposed envy-free algorithms with efficient running time. In order to ascertain the running time of a cake cutting algorithm, it is important to formally specify the computational model and input to the problem. In some of the literature (e.g. [101]), it is assumed that the value an agent ascribes to any segment of the cake can be queried or evaluated via an oracle. While the classical literature uses this query model, recent work by computer scientists assumes agents report their value density function over the entire cake, as is common in mechanism design. We follow this approach in our work as well.

Strategyproofness has largely been ignored in cake-cutting barring a few recent exceptions [89, 84, 44, 123]. Alternative notions of strategyproofness abound in the literature on cake-cutting problems. Our definition of strategyproofness requires truthful reporting of their value density function to be a (weakly) dominant strategy for each agent. On the other hand, Bram [31] considered an algorithm to be "strategyproof" if truth-telling is a maximin strategy (maximizes the minimum payoff that an agent can get), which is a weaker notion than our requirement of dominant strategy incentive compatibility. There is a literature that studies Nash equilibria of cake-cutting algorithms, see [95, 36].

The papers most directly relevant to our chapter are [48, 32, 44]. Chen et al. [44] presented a deterministic, strategyproof, polynomial-time, envy-free and Pareto efficient algorithm for piecewise uniform valuations. Our work addresses their open problem of generalizing their algorithm to the case of piecewise constant valuations. Cohler et al. [48] and Bram et al. [32] formulated linear programs to compute envy-free allocations for piecewise constant and piecewise linear valuations. However, the algorithms are not Pareto efficient in general.

---

[14] Envy-freeness implies proportionality when every portion of the cake that is desired by at least one agent is allocated to some agent. Otherwise, the empty allocation satisfies envy-freeness, but not proportionality.

3.2 Preliminaries

3.2.1 Model

We consider the problem of dividing a "cake", represented by the interval $[0, 1]$, among the set of agents $N = \{1, 2, \ldots, n\}$. Agent i's value for different parts of the cake is represented by a *value density function* $v_i : [0, 1] \to [0, \infty)$ that is *piecewise constant* with a finite number of pieces. In other words, each agent can partition the cake into a finite number of intervals such that v_i is constant over each interval. We will also consider a special case of a piecewise constant function called *piecewise uniform* function v, where the constant is k_v or 0 (the constant may be different for different functions), for some $k_v \geq 0$. Occasionally we shall consider a family of value density functions rather than a single one. To that end, we say that two value density functions v and v' are *ordinally equivalent* if $v(x) \geq v(y) \iff v'(x) \geq v'(y) \quad \forall x, y \in [0, 1]$. For example, the valuation functions of each of the agents in two subfigures in Figure 3.1 are ordinally equivalent.

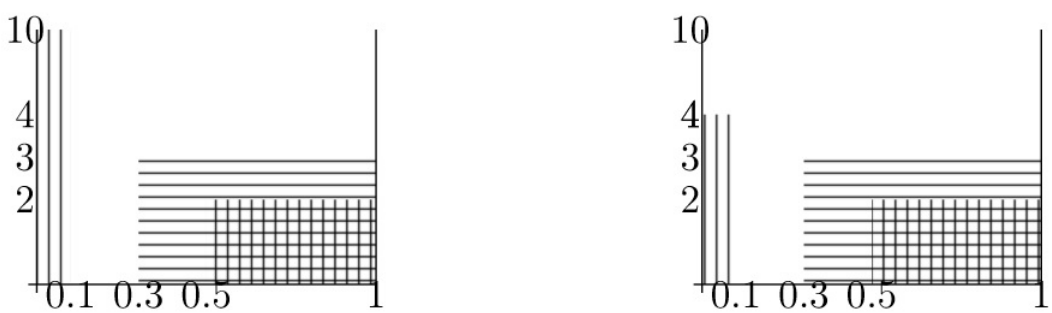

Figure 3.1: Example of a cake cutting problem with piecewise constant value density functions. The area with vertical lines is under the value density function of agent 1 and the area with horizontal lines is under the value density function of agent 2. The valuation functions of agent 1 are ordinally equivalent in the two subfigures above.

It is easily verified that the notion of ordinal equivalence partitions the class of value density functions into equivalence classes. Let $\hat{V}$ denote the class of value density functions ordinally equivalent to a given value density function v.

An *allocation* is a partition of the interval $[0, 1]$ into a set $\mathcal{A} = \{X_1, \ldots, X_n, W\}$, where each X_i is a finite union of disjoint subintervals of $[0, 1]$, and is the portion of the cake allocated to agent i; and

W is the part of the cake that is wasted, i.e. not allocated to anyone. The value of X_i to agent i is $V_i(X_i) = \int_{X_i} v_i(x)dx = \sum_{I \in X_i} \int_I v_i(x)dx$, which can be expressed as a finite sum, as each X_i is a finite union of intervals, and agent i has a piecewise constant valuation function. Note that valuations are non-atomic ($V_i([x,x]) = 0$) and additive: $V_i(X \cup Y) = V_i(X) + V_i(Y)$ where X and Y are disjoint[15].

The set of agents N and the profile of valuation functions $\{v_1, \ldots, v_n\}$ completely specify an instance of the cake-cutting problem. The goal is to find an allocation to the agents satisfying some appealing properties.

3.2.2 Properties of Allocations

The standard efficiency criterion is that of *Pareto efficiency*. An allocation is *Pareto efficient* if no agent i can get a higher value via a different allocation without some other agent j getting a lower value in that allocation. Formally, $(X_1, X_2, \ldots, X_n, W)$ is Pareto efficient if there does not exist another allocation $(Y_1, Y_2, \ldots, Y_n, W')$ such that $V_i(Y_i) \geq V_i(X_i)$ for all $i \in N$ and $V_i(Y_i) > V_i(X_i)$ for some $i \in N$. Occasionally, we shall weaken the efficiency requirement to *Non-wastefulness*: An allocation is *non-wasteful* if every portion of the cake desired by at least one agent is allocated to some agent who desires it. Formally, let Z_i represent the subintervals of $[0,1]$ for which agent i has zero value, and define $Z = \bigcap_{i \in N} Z_i$. Then, an allocation $(X_1, X_2, \ldots, X_n, W)$ is non-wasteful if and only if $X_i \cap Z_i \subseteq Z$ for all i and $W \subseteq Z$.

The two most important and commonly used criteria for an allocation to be fair are *envy-freeness* and *proportionality*. An allocation is *envy-free*, if $V_i(X_i) \geq V_i(X_j)$ for every pair of agents i and j, that is every agent considers his allocation to be at least as good as any other agent's allocation. In a *proportional* allocation, $V_i(X_i) \geq \frac{1}{n} V_i([0,1])$ for every agent i, that is, each agent gets at least $1/n$ of the value he has for the entire cake. Envy-freeness implies proportionality provided that every desirable part of the cake is allocated.

We can strengthen the fairness requirement by demanding envy-freeness or proportionality for every value density function ordinally equivalent to v_i, for each agent i. This gives rise to robust notions of these properties that we term, respectively, *robust envy-freeness* and *robust proportionality*. An allocation satisfies *robust proportionality* if for each agent i and for all $v'_i \in \hat{V}_i$, $\int_{X_i} v'_i(x)dx \geq 1/n \int_0^1 v'_i(x)dx$. (Recall that $\hat{V}_i$ contains all value density functions ordinally equivalent to v_i.) An

[15]Some papers in the literature assume $V_i[0,1] = 1$ for each agent i, but we do not make this normalization assumption.

allocation satisfies *robust envy-freeness* if for all $i, j \in N$ and for all $v'_i \in \hat{V}_i$, $\int_{X_i} v'_i(x)dx \geq \int_{X_j} v'_i(x)dx$. The motivation behind these requirements is clear: even if an agent re-adjusts or perturbs his value density function, the allocation remains envy-free or proportional as long as these perturbations do not change the relative desirability of the various parts of the cake. Thus, an agent does not have to worry too much about learning or reporting his utility accurately for various parts of the cake[16].

3.2.3 Properties of Cake Cutting Algorithms

A *deterministic cake cutting algorithm* maps each valuation profile to an allocation. A *randomized cake cutting algorithm* maps each valuation profile to a probability distribution over allocations. An algorithm (either deterministic or randomized) satisfies one of the aforementioned properties (e.g. Pareto efficiency) if it returns an allocation that satisfies the property for every valuation profile.

We assume that each agent's valuation function is private information for the agent that is not known to the algorithm designer. Therefore, the designer first asks the agents to report their value density function and then runs the algorithm on the reported input to find an allocation. As we consider piecewise constant value density functions, each agent need only report $2k + 1$ numbers if his valuation function has k breakpoints: the location of the k breakpoints, and the (constant) value rate he has for each of his $k + 1$ pieces. To incentivize the agents to report their valuations truthfully, the designer must employ a strategyproof algorithm, defined next.

A deterministic algorithm is *strategyproof* if no agent ever has an incentive to misreport in order to get a better allocation. Formally, let $X_i(v_i, v_{-i})$ be the allocation returned by an algorithm when agent i reports v_i and the other agents report v_{-i}. Then we say that an algorithm is strategyproof if

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

Similarly, a deterministic algorithm is *group-strategyproof* if it is not possible for any subset $S \subseteq N$ of agents to misreport their preferences such that each of them weakly prefers his allocation under the misreport, and such that at least one of them strictly prefers his allocation under the misreport. A deterministic algorithm is *weakly group-strategyproof* if it is not possible for any subset $S \subseteq N$ of

[16] Let us say that a cake is part chocolate and part vanilla. An agent may easily state that chocolate is more preferable than vanilla but would require much more effort to say that if the vanilla piece is exactly 1.372 times bigger than the chocolate piece then he would prefer both pieces equally.

agents to misreport their preferences such that each of them strictly prefers his allocation under the misreport.

The definitions of strategyproofness and group-strategyproofness can be extended to randomized algorithms in different ways. In a randomized algorithm, the allocation of each agent is probabilistic, and so the definition of strategyproofness will require us to compare an agent's probabilistic allocation under his true report to his probabilistic allocation when he misreports. A natural way to compare is to compare his *expected utility* under the two allocations. This leads us to the definition of *strategyproofness in expectation*: a randomized algorithm is *strategyproof in expectation* if the expected utility to an agent from reporting truthfully is greater than or equal to his expected utility from any misreport, regardless of the reports of the other agents.

Finally, consider a special class of valuation profiles in which any part of the cake desirable to one agent is undesirable to every other agent, and in which each agent has a positive valuation for at most $1/n$ fraction of the cake. A cake cutting algorithm satisfies *unanimity*, if for any such valuation profile each agent is allocated *all* the intervals for which he has a positive valuation.

3.2.4 Relationship between the Properties

We record some important relationships among the various properties we have discussed so far. Specifically, for the cake-cutting problem:

- robust proportionality $\implies$ proportionality.
- robust envy-freeness $\implies$ envy-freeness.
- (robust) envy-freeness and non-wastefulness $\implies$ (robust) proportionality.
- group strategyproofness $\implies$ weak group strategyproofness $\implies$ strategyproofness.
- Pareto efficiency $\implies$ non-wastefulness $\implies$ unanimity.
- (robust) proportionality $\implies$ (robust) envy-freeness when there are two agents (see [44]).

3.2.5 The Free Disposal Assumption

We may assume without lost of generality that every part of the cake is desired by at least one agent. If that is not the case, we can discard the parts that are desired by no one and rescale what is left so that we get a $[0,1]$ interval representation of the cake. Notice that this procedure preserves the aforementioned properties of fairness and efficiency. We will make use of this assumption in our description of CCEA and MEA.

3.2.6 The (Random) Assignment Problem and its Relationship to Cake Cutting

An assignment problem is specified by a triple $(N, O, \succsim, cap(\cdot))$, where $N = \{1, \ldots, n\}$ is the set of agents, and $O = \{o_1, \ldots o_m\}$ is the set of objects. Each object j has a consumption capacity $cap(j)$. The preference profile $\succsim = (\succsim_1, \ldots, \succsim_n)$ specifies each agent's preference ordering over the objects. We will assume that the preference relation of each agent is complete (every pair of objects is comparable) and transitive. By $a \succsim_i b$, we mean that agent i weakly prefers object a to object b. We write $a \succ_i b$ if i strictly prefers a to b, i.e. $a \succsim_i b$ but $b \not\succsim_i a$; and we use $a \sim_i b$ when i is indifferent between a and b, i.e. $a \succsim_i b$ and $b \succsim_i a$. We assume that the indifference relation is also transitive. Thus each agent has a most-preferred subset of objects (and the agent is indifferent between all the objects within this set), followed by a most-preferred subset of objects among the remaining ones, etc. An agent may find some of the objects unacceptable, and each agent is allocated only objects that he finds acceptable. We will let A_i denote the set of acceptable objects to agent i. In the (random) assignment literature, it is also further assumed that the number of objects equals the number of agents, every object has a unit capacity, and every agent is allowed to obtain at most one unit of object(s) in total. We make no such assumptions here.

Given an instance of the cake cutting problem, we can obtain a corresponding instance of the assignment problem as follows. First, we identify the union of breakpoints of the agents' value density functions, and divide the cake up into disjoint intervals each of whose endpoints are consecutive breakpoints. We refer to these intervals as *intervals by the breakpoints*, which play the role of objects in our assignment problem. The preferences of the agents over the objects are naturally induced by their values for the corresponding subintervals. The objects that an agent finds unacceptable correspond to intervals where his value density function is zero. Moreover, the capacity of an object is exactly the length of the corresponding interval. From now on, we will refer to this transformation as the *canonical transformation*. Below is the pseudocode for the canonical transformation.

The output to an assignment problem is often captured by an allocation matrix p. If the objects are divisible, then p_{io_j} denotes the amount of object o_j allocated to agent i. If the objects are indivisible,

Algorithm 3.1 Canonical Tranformation

Input: Cake-cutting problem with piecewise constant valuations $(v_1, \dots, v_n)$.

Output: An assignment instance $(N, O, \succsim, cap(\cdot))$.

1: Identify the union of breakpoints of agents' value density functions. Let $\mathcal{J} = \{J_1, \dots, J_m\}$ be the set of intervals of $[0, 1]$ formed by the breakpoints.

2: Consider $(N, O, \succsim, cap(\cdot))$ where

- $O = \{o_1, \dots, o_m\}$ where $o_i = J_i$ for all $i \in \{1, \dots, m\}$ with $cap(i) = len(J_i)$.
- $\succsim$ is defined as follows: $o \succsim_i o'$ if and only if $v_i(x) \geq v_i(y)$ for $x \in o$ and $y \in o'$;
-

3: Discard the objects that give every agent an utility of zero from O.

then one may view $\frac{p_{io_j}}{cap(o_j)}$ as the probability that object o_j is allocated to agent i. The objects in the assignment problem obtained from the canonical transformation are assumed to be divisible. Given a cake cutting instance, let $(o_1, o_2, \dots, o_m)$ be objects in the corresponding assignment problem obtained from the canonical transformation. Let p_i be the i-th row of the allocation matrix p. We say that the allocation matrix p is *stochastically envy-free* if for every pair of i and i', and for every object o_j that agent i finds acceptable, we have $\sum_{o_k \succsim_i o_j} p_{io_k} \geq \sum_{o_k \succsim_i o_j} p_{i'o_k}$. Similarly, we say that the allocation matrix p is *stochastically proportional* if for every i, and for every object o_j that agent i finds acceptable, we have $\sum_{o_k \succsim_i o_j} p_{io_k} \geq 1/n \sum_{o_k \succsim_i o_j} cap(o_k)$. In other words, an allocation is stochastically proportional if it stochastically dominates the uniform allocation. The following two propositions show that robust envy-freeness/proportionality in a cake cutting instance is equivalent to stochastic envy-freeness/proportionality in the corresponding assignment instance.

Proposition 3.1. *For a given allocation $\mathcal{A} = \{X_1, \dots, X_n\}$ in a cake cutting instance, let p be the allocation matrix in the corresponding assignment problem. Then*

- *$\mathcal{A}$ is robustly envy-free if and only if p is stochastically envy-free.*
- *$\mathcal{A}$ is robustly proportional if and only if p is stochastically proportional.*

Both propositions follow from basic properties of first-order stochastic dominance (see [9]). Note that, as a corollary, we see that both robust envy-freeness and robust proportionality require each agent to get a piece of cake of the same length if every agent desires the entire cake.

3.3 CCEA — Controlled Cake Eating Algorithm

Given an instance of the cake cutting problem, *CCEA* first applies the canonical transformation to obtain a corresponding instance of the assignment problem and then applies the EPS algorithm of [76] to that instance. The EPS algorithm can be described as a "cake-eating" algorithm in which agents consume their best object(s) at unit rate. Roughly, each agent *simultaneously* consumes her "best set" of available objects at a unit rate at each point in time. If all the preferences are strict, this algorithm reduces to the original Probablistic Serial (PS) algorithm of [29], which determines a unique allocation for the agents. Please see Section 2.2.3 for a description of the EPS algorithm. Note that although Katta and Sethuraman considered the standard assignment problem in which each agent has unit demand and each object has unit capacity, the algorithm can be easily extended to the case in which objects have different capacities and there is no constraint on the total size of an agent's allocation.[17] It is straightforward to compute the corresponding division of the cake from the solution given by the EPS algorithm: If an agent i is given p_j units of object o_j, then in the cake allocation agent i receives a subinterval of length p_j from the interval J_j.

Algorithm 3.2 CCEA (Controlled Cake Eating Algorithm).

Input: Cake-cutting problem with piecewise constant valuations $(v_1, \ldots, v_n)$.

Output: Robust proportional, robust envy-free, and non-wasteful allocation.

1: Apply the canonical transformation to obtain $(N, O, \succsim, cap(\cdot))$.
2: $p \longleftarrow EPS(N, O', \succsim, cap(\cdot))$
3: For interval J_j be the interval correspond to object o_j, agent i is an allocated subinterval of J_j, denoted by J_j^i, which is of length p_{io_j}. For example, if $J_j = [a_j, b_j]$, then $J_j^i = [a_j + \sum_{n=1}^{i-1} p_{io_j},\ a_j + \sum_{n=1}^{i} p_{io_j}]$.
4: $X_i \longleftarrow \bigcup_{j=1}^{m} J_j^i$ for all $i \in N$ **return** $X = (X_1, \ldots, X_n)$

Example 3.1 (Illustration of CCEA). *We examine how CCEA runs on the cake cutting problem in Figure 3.1. Firstly, the set* $\mathcal{J} = \{J_1, \ldots, J_4\}$ *of subintervals of* $[0, 1]$*, formed by the consecutive points of discontinuity of the agent valuation functions, are identified:* $J_1 = [0, 0.1], J_2 = [0.1, 0.3], J_3 = [0.3, 0.5]$*, and* $J_4 = [0.5, 1]$*. The interval* J_2 *is discarded because it is desired by no agent. The corresponding assignment problem has three objects,* $\{o_1, o_3, o_4\}$*, where object* o_j *corresponds to subinterval* J_j*. The*

[17] When there is an upper bound on how much an agent can consume, EPS stops the agent from consuming beyond this limit.

capacities of the objects are given by the lengths of the corresponding subintervals: thus, $cap(o_1) = 0.1$, $cap(o_3) = 0.2$, *and* $cap(o_4) = 0.5$. *The preferences of the agents over* O *are inferred from their valuation functions in the corresponding subintervals, so that* $o_1 \succ_1 o_4 \succ_1 o_3$ *and* $o_3 \sim_2 o_4 \succ_2 o_1$. *The assignment found by the EPS algorithm on the associated assignment instance is:* $p_{1o_1} = 0.1$, $p_{1o_3} = 0$, $p_{1o_4} = 0.3$, $p_{2o_1} = 0$, $p_{2o_3} = 0.2$, *and* $p_{2o_4} = 0.2$. *The object assignment* p *can be used to divide the subintervals among the agents:* $X_1 = [0, 0.1] \cup [0.7, 1]$ *and* $X_2 = [0.3, 0.5] \cup [0.5, 0.7]$.

The main result of this section is the following.

Theorem 3.1. *For piecewise constant valuations, there exists a deterministic polynomial time algorithm (CCEA) that is robust envy-free and non-wasteful.*

Proof. Two simple observations establish that CCEA is non-wasteful: first, no agent is assigned a part of the cake for which he has zero valuation; and second, the algorithm terminates only when every portion of the cake that is desired by at least one agent is completely consumed by some agent who desires it. By Proposition 3.1, showing robust envy-freeness of CCEA is equivalent to showing stochastic envy-freeness of EPS under the canonical transformed assignment instance. This result is similar to Theorem 4 of [76]. We give the proof here for the sake of completeness. Let $M(i, o)$ be the set of objects that agent i weakly prefers to o, i.e. $M(i, o) = \{o' \mid o' \succsim_i o\}$ for every object o that agent i finds acceptable. Let $t_{i,o}$ be the time at which all objects from $M(i, o)$ are completely consumed under EPS. Note that on the time interval $[0, t_{i,o}]$, agent i has been consuming only objects from the set $M(i, o)$ under EPS. Moreover, since all other agents consume at the same rate as agent i, agent j's total consumption from objects in the set $M(i, o)$ is at most t_o. Putting it altogether, we get

$$\sum_{o' \succsim o} p_{io'} = t_{i,o} \geq \sum_{o' \succsim o} p_{jo'}$$

for all i, j and objects o that agent i finds acceptable. This completes the proof. □

Remark 3.1. *CCEA is a polynomial time algorithm. The parametric network flow problem that EPS relies on can be solved in time* $O(|V||E|\log(|V|^2/|E|))$ *due to [64], where* V *and* E *are the vertex and edge sets of the network respectively. Let* b *be the total number of breakpoints in the agents' valuation functions. Then* $|V| = O(n + b)$. *Moreover, the number of iterations of EPS is upper bounded by* b, *as at least one object corresponding to some interval is completely consumed in every iteration.*

We showed that CCEA is robust envy-free and non-wasteful. Nonetheless, CCEA is not strategyproof. We now show that this is not a flaw in our design rather the aforementioned properties are not compatible with each other in the PCV domain.

Theorem 3.5. *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is strategyproof, robust proportional, and non-wasteful.*

Proof. Consider the following three profiles:

Profile 1:

$$1 : v_1(x) = a \text{ if } x \in [0, 0.25],\ v_1(x) = b \text{ if } x \in (0.25, 0.5],\ v_1(x) = 0 \text{ if } x \in (0.5, 1]$$

$$2 : v_2(x) = a \text{ if } x \in [0, 0.25],\ v_2(x) = b \text{ if } x \in (0.25, 0.5],\ v_2(x) = 0 \text{ if } x \in (0.5, 1]$$

$$3 : v_n(x) = 0 \text{ if } x \in [0, 0.5],\ v_n(x) = a \text{ if } x \in (0.5, 1]$$

$$\dots$$

$$n : v_n(x) = 0 \text{ if } x \in [0, 0.5],\ v_n(x) = a \text{ if } x \in (0.5, 1]$$

for some $a > b > 0$.

Since the algorithm is robust proportional, it must be the case that agents 1 and 2 each receives 1/2 of $[0, 0.25]$ and 1/2 of $(0.25, 0.5]$. Denote agent 1's allocation by $A \cup B$, where $A \subset [0, 0.25]$ and $B \subset (0.25, 0.5]$. Thus, agent 2 receives $[0, 0.5] \backslash (A \cup B)$ by non-wastefulness.

Now consider profile 2:

$$1 : v_1(x) = a \text{ if } x \in A,\ v_1(x) = b \text{ if } x \in B,\ v_1(x) = 0 \text{ otherwise}$$

$$2 : v_2(x) = a \text{ if } x \in [0, 0.25],\ v_2(x) = b \text{ if } x \in (0.25, 0.5],\ v_2(x) = 0 \text{ if } x \in (0.5, 1]$$

$$3 : v_n(x) = 0 \text{ if } x \in [0, 0.5],\ v_n(x) = a \text{ if } x \in (0.5, 1]$$

$$\dots$$

$$n : v_n(x) = 0 \text{ if } x \in [0, 0.5],\ v_n(x) = a \text{ if } x \in (0.5, 1]$$

By strategyproofness, agent 1 must again receive $A \cup B$. If agent 1 receives anything less in profile 2, then he would deviate from profile 2 to profile 1. If agent 1 receives anything more in profile 2, then he would deviate from profile 1 to profile 2. Thus, agent 2 receives $[0, 0.5] \backslash (A \cup B)$ by non-wastefulness.

Now consider profile 3:

$$1 : v_1(x) = a \text{ if } x \in A,\ v_1(x) = b \text{ if } x \in B,\ v_1(x) = 0 \text{ otherwise}$$

$$2 : v_2(x) = a+\epsilon \text{ if } x \in A, v_2(x) = a \text{ if } x \in (0, 0.25]\backslash A, v_2(x) = b \text{ if } x \in (0.25, 0.5], v_2(x) = 0 \text{ if } x \in (0.5, 1]$$

$$3 : v_n(x) = 0 \text{ if } x \in [0, 0.5],\ v_n(x) = a \text{ if } x \in (0.5, 1]$$

$$\ldots$$

$$n : v_n(x) = 0 \text{ if } x \in [0, 0.5],\ v_n(x) = a \text{ if } x \in (0.5, 1]$$

By robust proportionality, both agent 1 and 2 must receive 1/2 of A. By non-wastefulness, agent 2 must receive $[0, 0.5]\backslash(A \cup B)$, since the rest of the agents all have a utility of 0 on these intervals. Hence, agent 2 in profile 2 would misreport so that he receives the allocation in profile 3, violating strategyproofness. $\square$

3.3.1 CCEA for Piecewise Uniform Valuations

We now turn to the case in which all agents have *piecewise uniform* valuations. This obtains when each agent partitions the cake into desirable and undesirable parts such that all desirable parts have equal value density, and the undesirable parts have zero value density. Clearly, this is the special case of piecewise constant valuations in which each agent i's value density function assumes only two values: a positive real number k_i or zero.

The CCEA when restricted to piecewise uniform valuation functions profile is identical to Mechanism 1 of [44]. (The one cosmetic difference is that the underlying bipartite network used for solving the parametric network flow have the supply and demand nodes swapped and arc directions reversed. Note that this change will not affect the final allocation.) We provide a description of this mechanism below as it will be needed in the subsequent parts of the chapter.

Chen et al. [44] proved their mechanism is strategyproof for when agents have piecewise uniform valuation functions.[18] We now show that their mechanism is in fact group strategyproof.

[18] The free disposal assumption is necessary to ensure the algorithm of [44] to be strategyproof for piecewise uniform valuations. Therefore, we also make use of the free disposal in the canonical transformation in the algorithmic description for CCEA. The existence of a non-free disposal algorithm that satisfies all of the desirable properties in the piecewise uniform setting remains an open question.

Algorithm 3.3 CCEA for piecewise uniform valuations.

Input: Cake-cutting problem with piecewise uniform valuations.

Output: Robust proportional, robust envy-free, and Pareto optimal allocation.

1: Apply the canonical transformation to obtain $(N, O, \succsim, size(\cdot))$.
2: Run subroutine $(N, O, \succsim, cap(\cdot))$.

Subroutine $(N', O', \succsim, cap(\cdot))$:

1: Let $C(S, O')$ be the total capacity of objects from O' that at least one agent in $S \subseteq N'$ finds acceptable. Compute bottleneck set
$$B \in \arg\min_{S} \frac{C(S, O')}{|S|},$$
break ties according to any lexicographic ordering over the power-set of N.
2: Assign $\frac{C(B,O')}{|B|}$ units of acceptable object(s) to each agent in B in the form of subinterval(s).
3: Remove the allocated objects and the bottleneck agents. Run subroutine $(N' \backslash B, O' \backslash C(B, O'), \succsim, cap(\cdot))$.

Proposition 3.2. *For piecewise uniform value functions, CCEA is group strategyproof.*

The proof is via induction on the bottleneck set. We give a proof outline here, the full proof can be found in the Appendix A. We first show that no agent in the first bottleneck set can receive a desired allocation larger in length than the one he receives when reporting truthfully. Next, we show that there is no incentive for an agent to misreport in order to receive a piece of cake that he does not desire but another agent in a subsequent bottleneck set desires.

3.4 MEA — Market Equilibrium Algorithm

In this section, we present another algorithm for cake-cutting called the Market Equilibrium Algorithm (MEA). MEA first applies the canonical transformation to turn the cake cutting instance into an assignment problem instance, and subsequently makes use of the Eisenberg-Gale convex programming formulation for finding a (Fisher) market equilibrium of that instance. The convex program can be solved in polynomial time due to recent algorithmic advances (see [54]). We show that MEA always returns an allocation that is envy-free and Pareto efficient. Somewhat surprisingly, we also show that MEA can be view as another extension of Mechanism 1 in [44]. A detailed description of the algorithm can be found below.

The connection between a fair and efficient algorithm for cake cutting and computing market equilibria was first made by [100]. Reijnierse and Potters presented an algorithm to compute an ap-

Algorithm 3.4 The Market Equilibrium Algorithm to compute a Pareto optimal, envy-free, and proportional allocation.

Input: Cake-cutting problem with piecewise constant valuations.

Output: A proportional, envy-free, and Pareto optimal allocation.

1: Let $J = \{J_1, \dots, J_k\}$ be the intervals induced by the break points of the agents' valuation functions.
2: Discard all interval(s) for which every agent has zero valuation over.
3: Let x_{ij} be the length of any subinterval of J_i that is allocated to agent j.
4: Let $l_i = len(J_i)$, $v_{ij} = v_i(x), x \in J_i$.
5: Solve the following convex program.

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} \log(u_j) \\ \text{s.t.} \quad u_j = & \sum_{i=1}^{k} v_{ij} x_{ij} \quad \forall j = 1, \dots, n \\ & \sum_{j=1}^{n} x_{ij} \leq l_i \quad \forall i = 1, \dots, k \\ & x_{ij} \geq 0 \quad \forall i, j. \end{aligned}$$

6: Let $u_j^\star$, $x_{ij}^\star$ be an optimal solution to the convex program. Partition every interval $J_i = [a_i, b_i]$ into n subintervals where the j-th subinterval $J_i^j = [a_i + \sum_{k=1}^{j-1} x_{ik}^\star, a_i + \sum_{k=1}^{j} x_{ik}^\star]$.
7: $X_j \longleftarrow \cup_{i=1}^{k} J_i^j$ be the allocation of each $j = 1, \dots, n$. **return** $X = (X_1, \dots, X_n)$.

proximately envy-free and Pareto optimal allocation for cake cutting with general valuations. However, their algorithm is not polynomial-time even for piecewise constant valuations (see [132]). MEA requires the machinery of convex programming. It remains open whether MEA can be implemented via linear programming. Cohler et al. [48] presented a linear-programming based algorithm to compute an optimal envy-free allocation. The allocation they find is Pareto efficient within the class of envy-free allocations, but need not be Pareto efficient in general.

Although MEA is not robust envy-free like CCEA, it is Pareto efficient.

Theorem 3.2. *For piecewise constant valuations, there exists a deterministic polynomial time algorithm (MEA) that is Pareto efficient and envy-free.*

Proof. Notice that the feasible region of the math program contains all feasible allocations. Pareto efficiency is immediately implied by the optimality of the solution. To see that the optimal solution of the math program is also an envy free allocation, if we instead view x_{ij} as the fractional amount of J_i

that is allocated to agent j, then scaling the v_{ij}'s appropriately (i.e. setting $v'_{ij} = v_{ij} l_i$), then solving the math program in MEA is equivalent to solving the following math program.

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} \log u_j \\
\text{s.t.} \quad & u_j = \sum_{i=1}^{k} v'_{ij} x_{ij} \quad \forall j = 1, \ldots, n \\
& \sum_{j=1}^{n} x_{ij} \leq 1 \quad \forall i = 1, \ldots, k \\
& x_{ij} \geq 0 \quad \forall i, j.
\end{aligned}
$$

Following Vazirani pages 105-107 of [126], consider a market setting of buyers (agents) and divisible goods (intervals). Each good is assumed to be desired by at least one buyer (i.e. for every good i, $v_{ij} > 0$ for some buyer j, which holds in our setting by the free disposal assumption). There is a unit of each good and each buyer is given the same amount of money say 1 dollar, for which he uses to purchases the good(s) that maximizes his utility subject to a set of given prices. The task is to find a set of equilibrium prices such that the market clears (meaning all the demands are met and no part of any good is leftover) when the buyers seek purchase good(s) to maximize their utility given the equilibrium prices. Using duality theory, one can interpret the dual variable p_i associated with the constraints $\sum_{j=1}^{n} x_{ij} \leq 1$ as the price of a unit of good i. By invoking the KKT conditions, Vazirani [126] showed the prices given by the optimal dual solution is a unique set of equilibrium prices. Moreover, the primal optimal solution for each buyer j is precisely the quantity of good(s) that the buyer ends up purchasing that maximizes his utility given the equilibrium prices.

The optimal primal solution is an envy free allocation because given the equilibrium prices and identical purchasing power, if a buyer strictly prefers another buyer's allocation, he would instead use his money to obtain the allocation of the buyer that he envies. This would result in some surplus and deficit of goods, contradicting the fact that the given prices are equilibrium prices. □

Even though MEA is envy-free and Pareto efficient, it is not strategyproof. We next observe that no envy-free and Pareto efficient algorithm can be strategyproof when agents have piecewise constant value density functions.

Theorem 3.6. *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is strategyproof, Pareto efficient, and proportional.*

Proof. For cake cutting with piecewise constant valuations and $n \geq 2$, it follows from Theorem 3 of [108] that the only type of strategyproof and Pareto optimal mechanisms are dictatorships. Consequently, there exists no strategyproof and Pareto optimal mechanism that is also proportional. $\square$

Setting strategyproofness aside, we further show that the notion of robust fairness is incompatible with Pareto efficiency. Hence, the properties satisfied by CCEA and MEA are maximal subsets of properties that can be satisfied by any algorithm for PCV.

Theorem 3.7. *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is both Pareto efficient and robust proportional.*

Consider the following n-agent profile.

$$1 : v_1(x) = v_1^1 \text{ for } x \in [0, 0.25],\ v_1(x) = v_2^1 \text{ for } x \in (0.25, 0.5],\ v_1(x) = 0 \text{ for } x \in (0.5, 1]$$

$$2 : v_2(x) = v_1^2 \text{ for } x \in [0, 0.25],\ v_2(x) = v_2^2 \text{ for } x \in (0.25, 0.5],\ v_2(x) = 0 \text{ for } x \in (0.5, 1]$$

$$3 : v_3(x) = 0 \text{ for } x \in [0, 0.5],\ v_3(x) = 1 \text{ for } x \in (0.5, 1]$$

$$\cdots$$

$$n : v_n(x) = 0 \text{ for } x \in [0, 0.5],\ v_n(x) = 1 \text{ for } x \in (0.5, 1]$$

Choose $v_1^1, v_2^1, v_1^2, v_2^2 > 0$ in such a way that $v_1^1 > v_2^1$ and $v_1^2 > v_2^2$ and $\frac{v_1^1}{v_2^1} > \frac{v_1^2}{v_2^2}$. Let x_j^i be the length of the subset of interval I_j allocated to agent i. By Pareto optimality, only agent 1 or 2 can receive allocation from $[0, 0.5]$. By either robust proportionality, the mechanism must make an allocation where $x_1^1 = x_2^1 = x_1^2 = x_2^2 = 0.25$. On the other hand, in order for the mechanism to be Pareto efficient, the allocation vector must satisfy $x_2^1 = 0$ or $x_1^2 = 0$. Hence, we have reached an impossibility.

3.4.1 MEA for Piecewise Uniform Valuations

Next, we demonstrate the equivalence between MEA and Mechanism 1 of [44] for the uniform valuations setting. By equivalence, we mean that MEA and Mechanism 1 will return two allocations that yield the same utility for every agent given any valuation profile.

Proposition 3.3. *For piecewise uniform valuations, Mechanism* 1 *of [44] is equivalent to MEA.*

Proof. Given an allocation of Mechanism 1, which corresponds to a feasible solution of the convex program, we will find a set of prices corresponding to the allocation and show that the prices are in fact the equilibrium prices. Moreover, this allocation would be an allocation that maximizes the agents' utility given the equilibrium prices.

Given a valuation profile, let N be the set of buyers or agents and G be the set of goods or intervals, where each good has capacity equaling the length of the corresponding interval. Run Mechanism 1 on the same profile. Let B_i be the i-th bottleneck set computed by Mechanism 1. Let O^i be the set of remaining goods at the start of iteration i of Mechanism 1.

Let G_i be the set of goods that are distributed amongst the buyers in B_i. In the convex program, since each buyer is endowed with 1 dollar and every buyer in B_i receives $AVG(B_i, O^i) = C(B_i, O^i)/|B_i|$ units of good(s), it is natural to define the price of a unit of each good $k \in G_i$ to be

$$p_k = \frac{|B_i|}{C(B_i, O^i)}.$$

Notice that the prices for each good is well defined. This follows from the following observations:

1. $\cup G_i = G$ or every good has at least one price. This follows from the assumption that every good is desired by at least one agent, which means that Mechanism 1 will allocate all of the goods.

2. $G_i \cap G_j = \emptyset$ for all $i \neq j$ or every good has at most one price. This follows from the fact that no fractional parts of any good is allocated to agents from two or more bottleneck sets, which is another algorithmic property of Mechanism 1.

To show that the p_k's form a set of equilibrium prices, we will show that given the p_k's, the buyers in every B_i will choose to purchase *only* goods from G_i to maximize their utility function. For uniform valuation, one can show inductively that for every i and every buyer j in B_i, buyer j's desired set of goods D_j is a subset of $\cup_{i'=1}^{i} G_i$. Moreover, Lemma 3.4 of [44] shows that $AVG(B_i, O^i)$ is an increasing function of i, which means that goods belonging to G_i are cheaper than those belonging to $G_{i'}$ for $i > i'$. Hence, all buyers in B_i will opt to buy as much of their desired goods in G_i as possible. Furthermore, each buyer has enough budget to buy up to $AVG(B_i, O^i)$ unit of goods from G_i, since the price of each good is $1/AVG(B_i, O^i)$. Finally, one can partition the goods in G_i such that every buyer in B_i receives exactly $AVG(B_i, O^i)$ unit of goods that he desires, which means that all buyers in B_i will use up their budgets to purchase all of the goods in G_i. This implies that the given prices clear the market. □

Corollary 3.1. *For piecewise uniform valuations, MEA is group-strategyproof.*

Thus if we want to generalize Mechanism 1 of [44] to piecewise constant valuations and maintain robust envy-freeness then we should opt for CCEA. On the other hand, if one still wants to achieve Pareto optimality, then MEA is the appropriate generalization. In both generalization, we lose strategyproofness.

3.5 MCSD — Mixed Constrained Serial Dictatorship Algorithm

In light of the impossibility results established in Theorems 3.5 and 3.6, it is reasonable to ask if there is a strategyproof cake-cutting algorithm satisfying some appealing properties. If Pareto efficiency is the additional property we require, it follows from Theorem 3 of [108] that every strategyproof mechanism must be a dictatorship, which in our case would reduce to giving some agent every part of the cake that he finds desirable. Chen et al. [44] asked if there is a strategyproof and proportional algorithm for piecewise constant valuations. Our next mechanism—Mixed Constrained Serial Dictatorship (MCSD)—is a partial answer to that question.

Before diving into the MCSD algorithm, we would like to draw the reader's attention to a fundamental difference between the (random) assignment problem and the cake cutting problem. In random assignment, the objects being allocated are commonly known. In the cake-cutting problem, however, the discontinuity points of each agent's valuation function is private information for that agent, so any algorithm that uses the reported discontinuity points to artificially create the objects must also incentivize the agents to report these breakpoints truthfully. Otherwise, the transformation to the random assignment instance could create an object over which some agent has non-uniform preferences. To illustrate this difficulty, consider the uniform allocation rule [41] for the assignment problem, which distributes a $1/n$ fraction of each object to each agent. It is easy to see that this rule is strategyproof (the allocation is insensitive to the reported preferences) and proportional for the assignment model. Suppose we use the same rule for the cake-cutting problem in the following manner: agents report their value density functions (breakpoints and values in each interval), and the transformation is applied to create an assignment instance with the objects being subintervals in which each agent's reported value density is a constant. For the rule to be fully specified, we need to say precisely which piece or pieces adding up to a $1/n$ fraction of each subinterval is assigned to an agent. The next proposition shows that no deterministic conversion method can make the uniform rule strategyproof.

Proposition 3.4. *No deterministic implementation of the fraction-to-subinterval conversion method for the uniform allocation rule is strategyproof.*

Chen et al. [44] proposed a randomized mechanism that is strategyproof in expectation, ex-post robust proportional and envy-free motivated by a notion of perfect partition (see their paper details). Their mechanism can be viewed as a randomized implementation of the fraction-to-subinterval conversion method for the uniform allocation rule. However, since the uniform allocation rule does not take into account the preference of the agents, it has poor efficiency guarantees. This motivates the design of MCSD which achieves better efficiency guarantees by taking agent preferences into account.

We start with a randomized algorithm that is strategyproof and robust proportional in expectation. The algorithm is a variant of random dictatorship. A random ordering of the agents is drawn, each ordering equally likely. Agents choose pieces of the cake in this order, with the additional constraint that each agent consume at most $1/n$ fraction of the cake in total. Obviously, each agent will consume his most-preferred part of the cake when it is his turn, and there may be many equally good choices for an agent. To handle this, we break ties consistently by allocating to each agent the left-most part of the cake that he prefers most; if his most preferred pieces have been completely consumed, but he has not reached his quota of $1/n$, then starts consuming the left most part of his second most preferred pieces, etc. until his quota is reached. We will call this algorithm Constrained Random Serial Dictatorship (CRSD). Notice that CRSD is strategyproof, as in every draw of lottery, it is optimal for the agents to report their valuation function truthfully. Later in Proposition 3.6, we show that CRSD is robust proportional in expectation. To summarize, CRSD is a randomized algorithm that is strategyproof and satisfies robust proportionality in expectation.

The MCSD algorithm, described next, can be thought of as a derandomized version of CRSD obtained by computing the CRSD allocation for each of the $n!$ different permutations and aggregating them appropriately. The algorithm is formally presented as Algorithm 3.5.[19]

Note that MCSD may require an exponential number of cuts of the cake in the number of agents.

[19] We do not make the free disposal assumption in the algorithmic description here.

In Example 3.2, we illustrate how MCSD works.

Example 3.2 (Illustration of MCSD)**.** *We implement MCSD on the cake cutting problem in Figure 3.1. For the permutation* (12)*, agent* 1 *first chooses the cake piece* $[0, 0.1] \cup [0.5, 0.9]$ *and agent* 2 *then takes the remaining piece* $[0.1, 0.5] \cup [0.9, 1]$*. For the permutation* (21)*, agent* 2 *first chooses the cake piece* $[0.3, 0.8]$ *and agent* 1 *then takes the remaining piece* $[0, 0.3] \cup [0.8, 1]$*.*

The set of all relevant subintervals induced by the two permutations is

$$\{[0, 0.1], [0.1, 0.3], [0.3, 0.5], [0.5, 0.8], [0.8, 0.9], [0.9, 1]\}.$$

When we we additionally consider the discontinuities in the players' valuations, the set of relevant subintervals becomes

$$\mathcal{J}' = \{[0, 0.1], [0.1, 0.3], [0.3, 0.5], [0.5, 0.6], [0.6, 0.8], [0.8, 0.9], [0.9, 1].$$

Counting the number of times each agent receives each subinterval and dividing the counts by 2*, we get:*

$$X_1 = [0, 0.1] \cup \frac{1}{2}[0.1, 0.3] \cup \frac{1}{2}[0.5, 0.6] \cup \frac{1}{2}[0.6, 0.8] \cup [0.8, 0.9] \cup \frac{1}{2}[0.9, 1]$$

and

$$X_2 = \frac{1}{2}[0.1, 0.3] \cup [0.3, 0.5] \cup \frac{1}{2}[0.5, 0.6] \cup \frac{1}{2}[0.6, 0.8] \cup \frac{1}{2}[0.9, 1].$$

where $p[a, b]$ for some $0 \leq p \leq 1$ denotes a subinterval of $[a, b]$ with length equal to p times that of $[a, b]$.

Proposition 3.5. *For piecewise constant valuations, MCSD is well-defined and returns a feasible allocation in which each agent receives a collection of intervals of total length* $1/n$*.*

The proof is deferred to Appendix A.

Proposition 3.6. *For piecewise constant valuations, MCSD satisfies robust proportionality.*

Proof. We first prove that MCSD satisfies proportionality. In the case where all agents have the same valuations as the valuation of i, i is guaranteed $1/n$ of the value of the whole cake as MCSD allocates a $1/n$ fraction of each interval to each agent. Next, fixing the value density function of agent i, agent i's utility is minimized under the MCSD allocation when all other agents have identical utility. Formally, we have for each $\pi \in \Pi^N$ and preferences V_{-i} of all agents other than i,

$$V_i(\text{MCSD}^\pi(V_i, V_{-i})) \geq V_i(\text{MCSD}^\pi(V_i, (V_i, \dots, V_i))).$$

The reason is for any fixed permutation of the agents, the predecessors of i in π leave weakly better pieces of the cake for i when their valuations are different from i compared to when their valuations are the same. Averaging over all permutations, we get

$$V_i(\text{MCSD}(V_i, V_{-i})) \geq V_i(\text{MCSD}(V_i, (V_i, \dots, V_i))) = \frac{V_i([0,1])}{n}.$$

Finally, note that when an agent selects his best possible cake piece in each permutation, the exact height of the valuation function is not relevant and only the relative height matters. Hence, MCSD in fact satisfies robust proportionality. □

Corollary 3.2. *For any valuation profile, the allocation returned by MCSD stochastically dominates that of the uniform allocation rule.*

This follows from Proposition 3.1. Hence, we have a precise measure under which MCSD is more efficient than the uniform allocation rule.

In order to implement MCSD, we need to specify how a fractional portion of the interval J_j is converted to a subinterval or collection of subintervals of J_j. As with the uniform allocation rule, the strategyproofness of MCSD depends on how this conversion is done. In fact, Proposition 3.4 also implies that no deterministic implementation of the conversion procedure can make MCSD strategyproof. This is because, when every agent has an identical valuation function, then MCSD coincides with the uniform rule due to robust proportionality.

Corollary 3.3. *MCSD is not strategyproof under any deterministic procedure that converts fraction of each interval of $\mathcal{J}'$ to a collection of subintervals of that interval.*

In light of this difficulty, we describe a randomized conversion method (see Algorithm 3.6) that makes MCSD strategyproof. The method first fixes an ordering of the agents and randomly picks a starting point inside each interval. The subintervals are then carved out in proportion to the fractional assignments. Whenever we reach the right endpoint of the interval with our cuts, we wrap around and keep going starting with the left endpoint of the interval (equivalently, identify the two end-points of the interval and turn it into a circle). This randomized implemention of MCSD is called *Constrained Mixed Serial Dictatorship*, or *CMSD* for short. We show that CMSD is strategyproof in expectation. Also note that even though the allocation given by CMSD is random, it guarantees the same ex-post utility for every agent with respect to the reported valuations.

Proposition 3.7. *The CMSD mechanism is strategyproof in expectation.*

Consider the profiles $P = (P_i, P_{-i})$ and $P' = (P'_i, P_{-i})$ that differ only in the report of agent i. Let $J_1, \ldots, J_k$ denote the intervals whose fractional allocations are specified to each agent by MCSD in profile P and $J'_1, \ldots, J'_{k'}$ denote the intervals whose fractional allocations are specified to each agent by MCSD in profile P'. Let $V_i(J)$ denote agent i's total utility derived from receiving the interval J. Let p^P_{ij} ($p^{P'}_{ij}$ respectively) be the fractional allocation of interval J_j (J'_j respectively) to agent i. Since CSRD is strategyproof, comparing the expected utility of agent i in P and P' (and treating i's true preferences as P_i), we get

$$\sum_{j=1}^{k} p^P_{ij} V_i(J_j) \geq \sum_{j=1}^{k'} p^{P'}_{ij} V_i(J'_j)$$

To show that our implementation of MCSD is strategyproof in expectation, it suffices to show that if MCSD asks for a subinterval X_{ij} of J_j with length $p_{ij} len(J_j)$ for some $0 \leq p_{ij} \leq 1$, then the output returned by Algorithm 3.6 satisfies $E[V_i(X_{ij})] = p_{ij} V_i(J_j)$. The following lemma, whose proof is in Appendix A, proves this claim.[20]

Lemma 3.1. *Let U be uniformly distributed on the interval $[a, b]$ and let $0 \leq \alpha \leq 1$. Let $A = [U, U + \alpha(b - a)]$ if $U + \alpha(b - a) \leq b$ and $A = [a, U - (1 - \alpha)(b - a)] \cup [U, b]$ if $U + \alpha(b - a) > b$, then we have that $E_U[V_i(A)] = \alpha V_i([a, b])$, where $V_i(A) = \int_A v_i(x)dx$ for any integrable function v_i.*

To apply the lemma, we take a and b to be the left and right end points of J_j, $\alpha = p_{ij}$ and $U = mod(U_j + \sum_{k=1}^{i-1} p_{ij}(b_j - a_j))$, the left endpoint of X_{ij} given in Algoirthm 3.6.

We end the section with some limitations of MCSD. First, while MCSD is strategyproof in expectation, it is not group strategyproof, even in the weaker sense.

Proposition 3.8. *For cake cutting with piecewise constant valuations, MCSD is not weakly group-strategyproof even for two agents.*

[20]the lemma holds for any integrable value density function v_i, not just piecewise constant value density functions.

The proof of Proposition 3.8 can be found in Appendix A.

Furthermore, as long as there are at least seven agents, MCSD is not weak group strategyproof even if the agents have piecewise uniform valuations. This follows from the fact that RSD is not weakly group-strategyproof for dichotomous preferences when there are at least seven agents [29, 26].

Second, even though MCSD satisfies both proportionality and symmetry, it is not envy-free.

Proposition 3.9. *MCSD is not envy-free for three agents even for piecewise uniform valuations.*

The proof of Proposition 3.9 can be found in Appendix A.

However for the case of two agents, it is robust envy-free and polynomial-time.

Proposition 3.10. *For two agents and piecewise constant valuations, MCSD is robust envy-free, and polynomial-time but not Pareto optimal.*

For two agents, proportionality implies envy-freeness and robust proportionality implies robust envy-freeness (see [44]). Strategyproofness follows from Proposition 3.7. Moreover, for two agents, the algorithm is polynomial time with only two permutations.

Finally, a significant drawback of MCSD is that it is not Pareto efficient when agents have piecewise constant valuations. This can be seen from the example of Proposition 3.8. Moreover, computing the MCSD allocation is non-trivial as the number of agents grows (see e.g. [105]).

3.6 Discussion

The relation between the random assignment problem and cake cutting has been noticed before [44]. However, in their discussion of related work, Chen et al. [44] argue that techniques from the random assignment literature cannot be directly applied even to piecewise uniform functions—a subclasses of piecewise constant functions. The authors attributed this difficulty to the fact that in the random assignment problem, each agents gets one object. We observed that PS can be adopted to the case when agents get multiple objects and each object has arbitrary capacity. Moreover, many of its properties of PS in the unit demand setting remain satisfied (see [39] for generalizations of PS beyond unit demand setting).

Chen et al. [44] stated that generalizing their strategyproof algorithm for piecewise uniform valuations to the case for piecewise constant valuations as an open problem. We presented two algorithms

— CCEA and MEA — that generalize Mechanism 1 of [44]. Although they both satisfy certain desirable properties, both natural generalizations are not strategyproof. Our impossibility results further rule out the existence of mechanisms satisfying the properties that CCEA or MEA satisfies along with strategproofness, which partially answers the open problem that [44] poses.

Apart from the paper of [44], we are aware of no positive results regarding discrete, strategyproof, and fair algorithms even for the restricted domain of piecewise constant valuations. In this chapter we present a proportional algorithm (MCSD) for piecewise constant valuations. If we are allowed to use randomization, then we show that MCSD can be adapted to be strategyproof in expectation. Notice that if we instead require our algorithm to be strategyproof ex post and proportional in expectation, then CRSD would satify these properties. We note that [44] showed that the uniform allocation rule is envy-free and proportional, and strategyproof in expectation. However, we argue that it is inefficient: as it does not satisfy unanimity and its allocation is always stochastically dominated by the allocation of MCSD. It remains an open question whether there exists a strategyproof algorithm that always returns a proportional allocation for the piecewise constant valuation setting. In fact, the problem is open even for the special case of piecewise constant valuation where the value density function for each agent can take up to only two different constants. Finally, note that in the piecewise uniform case, when one of constants is zero, we are able to leverage the free disposal property in order to obtain strategyproofness while not incurring fairness and efficiency losses.

One difficulty that arises in coming up with strategyproof and proportional algorithm lies in that there is no restriction on the distribution of the discontinuity points of the agents' valuation functions. To illustrate this point, suppose the algorithm designer knows that the discontinuity points of the agents' valuation functions come from a set $S = \{d_1, \dots, d_k\}$, where $0 \leq d_1 \leq \dots \leq d_k \leq 1$. Consequently, a mechanism that partitions $[0, 1]$ into intervals of the form $[d_i, d_{i+1}]$ and allocates $1/n$ of each interval to each agent would be proportional, envy-free and strategyproof. Even if the designer does not know such an S, but instead we require the minimum distance between any two consecutive discontinuity points of the agent's valuation function to be at least some $\epsilon > 0$, then we can construct a strategyproof and δ-proportional algorithm for this setting by cutting the cake into small intervals and allocating $1/n$ fraction of each interval to each agent.

Algorithm 3.5 MCSD (Mixed Constrained Serial Dictatorship)— proportional and unanimous algorithm for piecewise constant valuations

Input: Cake-cutting problem with piecewise constant valuations.

Output: A robust proportional allocation.

for each $\pi \in \Pi^N$ **do**
  $C \longleftarrow [0, 1]$ (intervals left)
  for $i = 1$ to n **do**
    $X^{\pi}_{\pi(i)} \longleftarrow$ maximum preference cake piece of size $1/n$ from C
    $C \longleftarrow C - X^{\pi}_{\pi(i)}$.
    $i \longleftarrow i + 1$.
  end for
end for
Construct a disjoint and exhaustive interval set $\mathcal{J}'$ induced by the discontinuities in agent valuations and the cake cuts in the $n!$ cake allocations.
$Y_i \longleftarrow$ empty allocation for each $i \in N$.
for each $J_j = [a_j,\ b_j] \in \mathcal{J}'$ **do**
  for each $i \in N$ **do**
    Let $p_{ij} = \frac{count(i,J_j)}{n!}$ where $count(i, J_j)$ is the number of permutations in which i gets J_j.
    Generate $A_{ij} \subseteq J_j$ that is of length $p_{ij}|J_j|$ according to a subroutine.
    $Y_i \longleftarrow Y_i \cup A_{ij}$
  end for
end for
  return $Y = (Y_1, \dots, Y_n)$

Algorithm 3.6 A subroutine that converts fractional allocation into subintervals via randomization

Input: Interval $J_j = [a_j, b_j]$ and a vector of fractional assignment $p_j = (p_{1j}, \dots, p_{nj})$, where p_{ij} is the fractional allocation of interval I_j to agent i.

Output: Random subintervals $X_{ij} \subseteq J_j$ for $i = 1, \dots, n$, where X_{ij} is the subinterval allocated to agent i

1: Generate $U_j \sim unif[a_j,\ b_j]$.

2: For $a_j \leq x \leq 2b_j - a_j$, let $\mod(x) = x$ if $a_j \leq x \leq b_j$ and $x - (b_j - a_j)$ if $x > b_j$.

3: If $\mod(U_j + \sum_{k=1}^{i-1} p_{kj}(b_j - a_j)) \leq \mod(U + \sum_{k=1}^{i} p_{kj}(b_j - a_j))$ set

$$X_{ij} = [\mod(U_j + \sum_{k=1}^{i-1} p_{kj}(b_j - a_j)),\quad \mod(U_j + \sum_{k=1}^{i} p_{kj}(b_j - a_j))]$$

4: Else set:

$$X_{ij} = [a_j,\quad \mod(U_j + \sum_{k=1}^{i} p_{kj}(b_j - a_j))] \cup [\mod(U_j + \sum_{k=1}^{i-1} p_{nj}(b_j - a_j)), b_j]$$

Chapter 4

A Generalization of the Probabilistic Serial Mechanism and its Relationship to the Leximin Allocation

4.1 Introduction

We consider a resource allocation problem on a bipartite network. There are a set of buyers or agents, each with a demand requirement, and a set of sellers or resources, each with a supply capacity. Each buyer has preferences over the set of sellers. Moreover, each buyer's demand can only be satisfied (if it is satisfied at all) in its entirety by a single seller whose capacity can accommodate the demand. The goal is to design a mechanism that assigns the buyers to the sellers in a fair and efficient manner while respecting the sellers' capacities. Moreover, as each buyer's preference ordering over the sellers is private information, we also would like our mechanism to be strategyproof. The classical random assignment problem (see e.g. [29]) can be viewed as a special case of this problem in which each buyer has unit demand and each seller has unit capacity. We will refer to this case as the UDC special case.

Kurokawa et al. [81] first considered this model in the context of classroom assignment. Those authors were contacted by a representative of one of the largest school districts in California, with the task of allocating unused classrooms in the district's public schools to the district's charter schools. Each public school has a given number of unused classrooms (its capacity), and each charter school has a number of required classrooms (its demand). Finally, an operational constraint requires that a charter school's demand must be satisfied in a single public school. Kurokawa et al. [81] modeled the agent preferences as dichotomous: every charter school lists a subset of public schools that it finds acceptable. In such a setting, they gave a mechanism that satisfies many nice properties, including: *proportionality* (for every pair of charter and public schools i and j, the probability that i's demand is satisfied by a school that i weakly prefers over j is above a threshold level), *envy-freeness* (every charter school prefers its probabilistic allocation to that of any other school), *Pareto optimality* (no other feasible probabilistic allocation is at least as good for every charter school and strictly better for some school) and *strategyproofness* (no school can benefit by misreporting its preferences). The mechanism always computes a leximin allocation: one that maximizes the lowest probability of any charter school having its demand satisfied in an acceptable facility; subject to this constraint, it maximizes the second lowest probability; and so on. We shall refer to this mechanism as the *leximin mechanism*.

The leximin mechanism was originally proposed in the seminal paper of Bogomolnaia and Moulin [26], who study the UDC special case. There they showed the equivalence between the probabilistic serial mechanism and the leximin mechanism. The PS mechanism in the UDC setting was originally proposed by Bogomolnaia and Moulin [29] as an eating procedure in which each agent consume her most preferred resource out of the available resources at each given point in time, assuming the agents have strict preferences. It was later extended by Katta and Sethuraman [76] to the general preference domain, which encompasses the dichotomous preference domain as a special case. Despite failing to be strategyproof in the general preference domain, the PS mechanism is known to be envy-free, proportional and ordinally efficient (a generalization of Pareto optimality). A natural question is whether the PS mechanism can be generalized to handle more general demands and capacities.

On a separate note, Bogomolnaia [27] recently gave an alternative definition of the PS mechanism in the spirit of a leximin allocation for the UDC setting when agents have general preferences. Given a random allocation X, let v_X be a vector such that for every agent i and her preference indifference class l, there is a entry in v_X corresponding to the probability that agent i is assigned to a seller from indifferent class l or better. Bogomolnaia [27] showed that the PS mechanism lexicographically maximizes v_X over all feasible allocations X. We will refer to this allocation as the *generalized leximin allocation* in the chapter. Note that for the dichotomous preference domain, there is only one preference indifference class for each agent. Hence, we immediately deduce from [27] that the PS mechanism always computes a leximin allocation for dichotomous preferences in the UDC setting.

Research Questions: Given the equivalence results between the PS and (generalized) leximin mechanism established by [26] and [27] for the UDC setting, it is natural to ask if there is a suitable generalization of the PS mechanism that computes a generalized leximin allocation in the arbitrary demand and supply setting. Moreover, [81] showed that the leximin mechanism is a compelling mechanism to use in the general supply and demand setting with dichotomous agent preferences as it satisfies many nice properties. It is not known what properties are satisfied by the generalized leximin mechanism in the general preference domain.

4.1.1 Our Contributions

We propose the generalized probabilistic serial (GPS) mechanism. The GPS mechanism, by definition, computes a generalized leximin allocation as defined earlier. We extend the exponential LP, proposed in [81] for the case of dichotomous preference, to compute such an allocation. Alternatively, we give another algorithmic approach for computing a generalized leximin allocation that is more closely related to the interpretation of the PS mechanism as an eating algorithm in the UDC special case. Next, we examine the properties satisfied by the GPS mechanism in the general demand and supply capacity setting with general preferences. It is known from [29] that even for the UDC setting, the PS mechanism fails to be strategyproof. We show that the generalized PS mechanism fails envy-freeness as well. Nonetheless, the mechanism remains Pareto optimal, and we conjecture that it is proportional as well.[21] Our results suggest the potential need to consider other allocation methods. A summary of the results is provided in Table 4.1.

4.1.2 Related Work

The modern literature on fair division of indivisible goods dates back at least to the seminal paper [75], which adapted the competitive equilibrium with equal incomes (CEEI) solution. The main drawbacks of this solution are its prohibitive computational and informational requirements: it requires the solution of a fixed-point problem, and a complete knowledge of the utility functions of the agents. Budish

[21] We provide a proof of proportionality for dichotomous preferences. Note that our definition of proportionality is stronger than that of [81].

Demand \ Preference	Dichotomous	General
UDC	PS = leximin ([26]) EF, Prop, PE, weak GSP ([26])	PS = generalized leximin ([27]) EF, Prop, OE ([76])
General	leximin = GPS (Prop. 4.1) EF, Prop, PE, SP ([81], Prop 4.3)	generalized leximin = GPS (Thm 4.1) Prop (?), OE (Prop. 4.4)

Table 4.1: abbreviations: EF = envy-freeness, prop = proportionality, PE = Pareto efficiency, OE = ordinal efficiency, GSP = group strategyproofness.

[38] later proposed an approximate CEEI solution, where the approximation guarantees are practical as long as the supply of each good is relatively large.

In the realm of ordinal preferences, most of the academic literature focuses on the UDC special case, also known as the random assignment problem. The two prominent mechanisms used in the random assignment literature are the random serial dictatorship (RSD) mechanism of [1], which orders the agents uniformly at random and lets them successively choose an available resource according to this (random) order, and the probabilistic serial mechanism (PS) of [29], which allows agents to "eat" (at identical rates) their shares of different resources one by one in the order in which they rank the resources. RSD is known to satisfy weaker efficiency and fairness properties than PS, but is strategyproof, whereas PS is not strategyproof in general. The probabilistic allocation for an agent-resource pair in the PS mechanism can be efficiently computed, whereas this computation is difficult for RSD [105]. The original random assignment problem involves assigning n resources to n agents. Since the work of [29], the PS mechanism has been extended to deal with the general preference domain involving indifference in preferences (see [76]), multi-unit demands (see e.g. [77, 99, 10]) and general social choice settings [12]. The Birkhoff von-Neumann theorem [24, 127] enables an algorithm to output a compact representation of a random assignment or packing by specifying its probabilistic distribution over all pairs of buyers and sellers. Budish et al. [39] recently generalized the Birkhoff von-Neumann theorem to handle many real-world combinatorial domains. Unfortunately, their results do not apply to the setting where a demand must be satisfied entirely by a single seller such as the one that we are studying.

The leximin mechanism was original proposed by [26] for the UDC setting with dichotomous agent preferences. Kurokawa et al. [81] were the first ones to propose the leximin mechanism for the setting of general supply and unsplittable demand. In addition to the nice properties mentioned in [81], Bogolmonaia and Moulin [26] also showed the Lorenz dominance of the leximin probability vector in the UDC seting, which does not hold for general supply and demand (see [81] for a counterexample). Bogomolnaia [27] later showed that the PS mechanism always returns a generalized leximin allocation. Attempting to understand the connection between the PS mechanism and the leximin allocation for the general supply and demand setting is one of the main objectives of this chapter.

4.1.3 Model and Notation

Let $\mathcal{B} = \{1, \ldots, n\}$ denote the set of *buyers* or *agents*. Let $\mathcal{S} = \{1, \ldots, m\}$ denote the set of *sellers* or *resources*. Each buyer i has a pair $(d_i, \mathcal{P}_i)$, where d_i denotes the number of units demanded by the buyer and $\mathcal{P}_i$ is a preference ordering over a subset of sellers A_i that buyer i finds acceptable. Specifically, $\mathcal{P}_i$ is a total ordering over the set $A_i \subseteq \mathcal{S}$, i.e. for every pair of sellers $j, j' \in A_i$, either $j \succeq_i j'$ (j is weakly preferred to j') or $j' \succeq_i j$ (j' is weakly preferred to j). If both of the aforementioned relations hold, then we say that i is indifferent between sellers j and j' or $j \sim_i j'$. Hence, we can partition A_i into preference indifference classes $\{A_{i1}, \ldots, A_{iL_i}\}$ indexed in decreasing preference order such that for every $l = 1, \ldots, L_i$ and every $j, j' \in A_{il}$, $j \sim_i j'$. Each seller j has a capacity c_j. Since we are not allowed to split the demand of a buyer across different sellers, buyer i finds seller j acceptable only if $d_i \leq c_j$.

A *deterministic allocation* is a mapping $X : \mathcal{B} \to \mathcal{S} \cup \{0\}$, where $X_i = X(i)$ denotes the seller that buyer i is assigned to.[22] Buyer i is not assigned to any seller if $X_i = 0$. We say that X is feasible if for all $j \in \mathcal{S}$, $\sum_{i \in \mathcal{B}: X_i = j} d_i \leq c_j$. A feasible random allocation is a distribution over feasible deterministic allocations. From hereon, when we refer to the term "allocation", we will assume that it is feasible unless stated otherwise. *A random allocation* is a probability distribution over deterministic allocations. The outcome of a random allocation can be compactly described by the following sub-stochastic matrix $P = \{p_{ij}\}$, where p_{ij} is the probability that buyer i is assigned to seller j under the random allocation.

Let $T_{il} = \cup_{l'=1}^{l} A_{il'}$. Let B_S denote the set of buyers who find some seller $s \in S$ acceptable.

In this chapter, we will focus on these desirable properties:

[22] If buyer i is assigned to seller j, then seller j satisfies all of buyer i's d_i units of demand and no larger.

1. ***Envy-freeness***: Buyer i is said to *envy* buyer i' in an allocation if i''s demand is at least that of i, and there exists some seller j such that the total probability that i is assigned to a seller that she weakly prefers to j is smaller than that of buyer i'. We say that an allocation is *envy free* if no envy exists between any pair of buyers i and i'. Mathematically, an allocation is envy-free if
$$\sum_{j' \in A_i : j' \succeq_i j} p_{ij'} \geq \sum_{j' \in A_i : j' \succeq_i j} p_{i'j'}, \quad \forall i, i' \text{ s.t. } d_{i'} \geq d_i, \quad \text{ and } \forall j$$

2. ***Proportionality***: An allocation is *proportional* if for each buyer i and for each of her preference indifference level l, the probability that i is matched with a seller in T_{il} is at least $\frac{|T_{il}|}{\max\{|T_{il}|, |B_{T_{il}}|\}}$.[23]

3. ***Ordinal Efficiency***: Let P_i and Q_i denote the probabilistic allocation vector for buyer i in random allocation matrices P and Q respectively. We say that P_i stochastically dominates Q_i according to i's preference or $P_i \succeq Q_i$ if
$$\sum_{j' : j' \succeq_i j} p_{ij'} \geq \sum_{j' : j' \succeq_i j} q_{ij'} \quad \forall i, j$$
We say that i strictly prefers P_i to Q_i, denoted by $P_i \succ_i Q_i$, if at least one of the inequalities in the above definition is strict. Finally, we say that P stochastically dominates Q, denoted by $P \succeq Q$, if $P_i \succeq_i Q_i$ for all $i \in N$, with $P_i \succ_i Q_i$ for some $i \in N$.[24] We say that a random allocation P is *ordinally efficient* if it is not stochastically dominated by any other random allocation Q.

Note that our notions of envy-freeness, proportionality, and ordinal efficiency generalize the corresponding concepts defined in [81] for dichotomous preferences. In particular, ordinal efficiency coincides with Pareto optimality for dichotomous preferences.

4.1.4 The Generalized Leximin Allocation Vector

We consider an extension of the definition of leximin allocation vector introduced by [27] for the UDC special case. Let L_i be the number of preference indifference classes in buyer i's preference ordering. Let $L = \sum_{i=1}^{n} L_i$. Given a random allocation matrix P, for every buyer i and indifference class A_{il} indexed in decreasing preference order, compute the L-dimensional vector of probability sums $(\sum_{j \in T_{il}} p_{ij})_{il}$

---

[23] The literature on fair division such as cake cutting typical uses a weaker notion of proportionality: it only requires each agent to receive an acceptable resource with probability at least $1/n$ under the allocation.

[24] Note that our notion of stochastic dominance defines a partial order on the set of sub-stochastic matrices.

and sort the components of this vector in non-decreasing order (break ties arbitrarily) to obtain V_P. We say that a random allocation is *leximin optimal* if it lexicographically maximizes the vector V_P. We will refer to such an assignment as a *generalized leximin allocation* (GLS), and the mechanism that always returns such an assignment as the *generalized leximin mechanism* (GLM). Note that our notion of the leximin allocation generalizes the notion defined by [81] for the dichotomous preference domain.

4.2 Computing a Generalized Leximin Allocation

We first extend the iterative LP algorithm introduced by [81] for finding a leximin optimal allocation for the dichotomous setting to the full preference domain. To that end, we first describe a feasibility checking subroutine:

Feasibility check subroutine:

Let v_i indicate the preference indifference class to which buyer i is assigned (we set $v_i = 0$ if i is unassigned). That is, buyer i can only be assigned to a seller in the set A_{i,v_i}. The question we wish to answer is: is there a packing satisfying the demand vector $(v_1, \ldots, v_n)$? This can be answered by solving the following integer program. Define a variable x_{ij} for every seller $j \in A_{i,v_i}$ that equals to 1 if buyer i is assigned to seller j and 0 otherwise. Then we solve the following integer program:

$$\begin{aligned} \max \quad & 0 \\ \sum_{j \in A_{i,v_i}} x_{ij} = 1 \quad & \forall i = 1, \ldots, n \text{ s.t. } v_i > 0 \\ \sum_{i: v_i \leq L_i, j \in A_{i,v_i}} d_i x_{ij} \leq c_j \quad & \forall j = 1, \ldots, m \\ x_{ij} \in \{0,1\} \quad & \forall j = 1, \ldots, m, v_i = 1, \ldots, L_i \end{aligned} \tag{4.1}$$

Given a vector $v = (v_1, \ldots, v_n)$, if the above integer program has a feasible solution for v, then we include v in the set of all feasible vectors $\mathcal{F}$. Once $\mathcal{F}$ is constructed, we define the following iterative procedure for computing a leximin allocation.

Iterative Allocation LP:

We let p^*_{il} indicate the probability that buyer i is assigned to a seller in A_{il} in the leximin allocation that we will iteratively compute. Initially set all $p^*_{il} = 0$. Let R_t be the remaining subset of buyers at the end of iteration t. Initialize $R_0 = \{1, \ldots n\}$. Let i_t be the index of the preference indifference class under consideration for buyer i in iteration t of the algorithm. Initially, set $i_1 = 1$ for all i. Now, given that we have completed iteration $t-1$ of the algorithm, we solve the following LP in iteration t with variables y_v for every $v \in \mathcal{F}$ and p_{il} for $i = 1, \ldots, n$ and $l = 1, \ldots, i_t$.

$$
\begin{aligned}
\max \quad & M && (4.2)\\
\text{s.t.} \ \sum_{l=1}^{i_t} p_{il} &\geq M \quad \forall i \in R_{t-1} \\
p_{il} &= p^*_{il} \quad \forall i = 1, \ldots, n, \ l = 1, \ldots, i_t - 1 \\
p_{il} &= \sum_{v \in \mathcal{F} \ | \ v_i = l} y_v \quad \forall i = 1, \ldots n, \ l = 1, \ldots, i_t \\
\sum_{v \in \mathcal{F}} y_v &\leq 1 \\
y_v &\geq 0 \quad \forall v.
\end{aligned}
$$

Here y_v denotes the probability that packing v is chosen and p_{il} is the probability that buyer i is assigned to a seller in A_{il}.

Let $(\tilde{p}_{il})$ be the optimal probabilities returned by LP (4.2) and $\tilde{M}$ be the optimal value of M. Let $S_{\tilde{M}}$ be the set of buyers i such that $\sum_{l=1}^{i_t} \tilde{p}_{il} = \tilde{M}$. To ensure that this constraint is actually tight for buyer i in every single optimal solution (i.e. buyer i's allocation probability cannot be further improved without hurting the allocation of another buyer), for every $b \in S_{\tilde{M}}$, we solve the following LP.

$$\begin{aligned} \max \quad & \sum_{l=1}^{b_t} p_{bl} \\ \text{s.t.} \quad & \sum_{l=1}^{i_t} p_{il} \geq \tilde{M} \quad \forall i \in R_{t-1} \\ & p_{il} = p^*_{il} \quad \forall i = 1, \dots, n, \ l = 1, \dots, i_t - 1 \\ & p_{il} = \sum_{v \in \mathcal{F} \ | \ v_i = l} y_v \quad \forall i = 1, \dots n, \ l = 1, \dots, i_t \\ & \sum_{v \in \mathcal{F}} y_v \leq 1 \\ & y_v \geq 0 \quad \forall v. \end{aligned} \tag{4.3}$$

Let $(\hat{p}_{bl})$ be the optimal probabilities returned by LP (4.3) for buyer b. If $\sum_{l=1}^{b_t} \tilde{p}_{bl} = \tilde{M}$, then set $p^*_{b,b_t} = \hat{p}_{b,b_t}$ and include it in R_t. Moreover, if $b_t < L_b$, then set $b_{t+1} = b_t + 1$. If $b_t = L_b$, then do not include buyer b in R_t. On the other hand, if we have $\sum_{l=1}^{b_t} \tilde{p}_{bl} > M$, then set $b_{t+1} = b_t$ and include b in R_t. Finally, for every $b \notin S_{\tilde{M}}$, we set $b_{t+1} = b_t$ and include b in R_t.

We keep iterating the algorithm until $R_t = \emptyset$. Note that in every iteration t, at least one of the buyers (in $S_{\tilde{M}}$) will not be included in R_t.[25] Hence, we need to solve at most $(\sum_{i=1}^{n} L_i)$ LPs before the algorithm terminates. Finally, it can be shown via induction that the vector p^* returned by the above procedure is a leximin allocation vector.

4.3 The Equivalence between the Generalized Probablistic Serial (GPS) Mechanism and the Generalized Leximin Mechanism

In this section, we propose another way of computing a generalized leximin allocation that is reminiscent of the eating procedure prescribed by the probabilistic serial mechanism in the UDC setting. The probabilistic serial (PS) mechanism in the UDC setting was originally proposed by [29] when agents (buyers) have strict preference over resources (sellers). Katta an Sethuraman [76] later extended the

[25] This is because if $\sum_{l=1}^{b_t} \tilde{p}_{bl} > \tilde{M}$ for every $b \in S_{\tilde{M}}$, then a strict convex combinations of the solutions $(\hat{p}_{bl})$ for $b \in S_{\tilde{M}}$ will result in a feasible solution of LP (4.2) with objective value strictly larger than $\tilde{M}$, a contradiction.

mechanism to accommodate indifferences in agents' preferences. At a high level, the mechanism works as follows: in every iteration, given a subset of buyers/agents S, the union of resources O_S in the preference indifference class that each agent in S is currently consuming, as well as the probabilistic assignment matrix P that has been allocated to agents in S so far, the mechanism computes a score $\mathcal{S}(S, O_S, P)$. The mechanism then finds the subset of agents S^* that minimizes the $\mathcal{S}(S, O_S, P)$ over all subsets of agents that are still allowed to consume more resources. S^* is commonly referred to as the *bottleneck set*. Each agent in S^* is allocated a probability of obtaining a resource from the preference indifference class that the agent is currently consuming so that the sum of allocated probabilities for each agent in S is $\mathcal{S}(S^*, O_{S^*}, P)$. If $\mathcal{S}(S^*, O_{S^*}, P) = 1$, then agents in S^* are removed from consideration. Otherwise, each agent in S^* starts consuming resources from her next level of preference indifference class (if any).

In this section, we seek to generalize the PS mechanism to the more general setting of arbitrary demands and supplies. The key to establishing the result is identifying the appropriate generalization of $\mathcal{S}(S, O_S, P)$. In UDC setting, the score $\mathcal{S}(S, O_S, P)$ represents the maximum probability that each agent in S can be assigned to a resource in a preference indifference class no worse than the one she is currently consuming, provided that every agent in S receives the same aggregate probability whenever possible. It turns out that this high level view of $\mathcal{S}(S, O_S, P)$ carries over to the general supply and demand setting. The difference lies in that for the UDC setting, the Birkhoff von-Neumann theorem allows us to move between a probabilistic allocation of agent-resource pairs and a distribution over matchings. Consequently, we can treat the resources as divisible goods and $\mathcal{S}(S, O_S, P)$ can be computed in a fairly straightforward manner by solving a parametric max flow problem (see e.g. [76]), which can be solved in polynomial time. Nonetheless, for general demand and supply quantities, we need to specify an explicit distribution over packings that maximizes the probabilistic assignment for a subset of buyers S. The following subproblem checks whether a distribution over packings exists for a given vector of probability assignments between buyer preference indifference class pairs.

Packing Subproblem:

Let S be a subset of buyers. Consider a subset of ordered pairs $Q = \{(i, O)\}$, where $i \in S$ is a buyer and O is a subset of sellers that buyer i can be matched to (O typically denotes the set of sellers from one of buyer i's preference indifference classes). There may be multiple ordered pairs for a given buyer

$i \in S$. Moreover, each buyer $i \in S$ participates in at least one ordered pair in Q. Let $I_Q = \cup_{O:(i,O)\in Q} O$. Finally, for every pair $(i, Q) \in Q$, we are given an integer $q_{(i,Q)}$. We also have an integer r associated with I_Q. Now, for every pair $(i, O) \in Q$, we duplicate buyer i $q_{(i,Q)}$ times. Each of the duplicated buyers finds only sellers in O acceptable. We duplicate r copies of the set of sellers I_Q. We say that the duplication is *feasible* if, there exists a packing such that

1. Every duplicated buyer is assigned to some duplicated acceptable seller.
2. At most one copy of the same buyer can be assigned to a seller in each copy of I_Q.
3. The packing respects the capacity for all duplicated sellers.

Note that if the duplication is feasible, then we know that there is a sub-distribution over packings that achieves the probabilistic assignment such that for every pair (i, O), buyer i is matched to a seller in O with probability $\frac{q_{(i,Q)}}{I_Q}$. Checking whether a duplication is feasible is a NP-hard problem, even if there is a single seller that all buyers desire. A reduction from the partition problem nearly identical to the hardness result for computing a leximin vector shown in [81] can be attained.

4.3.1 Dichotomous Preference Domain

We first present a generalization of the PS mechanism for the dichotomous preference setting as a warm up. Let p_i be the probability that buyer i is assigned to a seller that he finds acceptable. Let R denote the remaining set of buyers to be considered by the algorithm. Below is the description of the generalized PS (GPS) algorithm for the dichotomous setting:

1. Initialize $p_i = 0$ for all buyer i. Let I_i be the set of acceptable sellers for buyer i. Set $R = N$.
2. While $R \neq \emptyset$, consider the following duplication. Let l be the least common multiple of all denominators of p_i for every $i \in N \backslash R$ ($l = 1$ if $N \backslash R$ is empty). Consider a packing subproblem instance where we have $cp_i l$ copies of buyer i for every $i \in N \backslash R$ for some $c \in \mathbb{N}$, q copies of every buyer $i \in S \subseteq R$, and cl copies of each seller that buyers in $(N \backslash R) \cup S$ collectively desires. Let O_S be the set of sellers that some buyer in S finds acceptable.[26]

[26] Note that for any fractional value p_i and any l fixed, there exists a pair of integers q and c such that $\frac{q}{cl} = p_i$.

3. Let $\mathcal{S}(S, O_S, P)$ be the maximum value of $\frac{q}{cl}$ such that the duplication for the given parameters is feasible.[27]

   (a) Find $S^* = \arg\min_{S \subseteq R} \mathcal{S}(S, O_S, P)$. If there are multiple S^*'s we look for one that is a minimal subset. Also, among all minimal optimal subsets, we break ties using a lexicographical ordering.

   (b) Set $p_i = \mathcal{S}(S^*, O_{S^*}, P)$ for all $i \in S^*$. The distribution over packings is determined by the duplication subproblem that yields $\mathcal{S}(S^*, O_{S^*}, P)$.

   (c) Remove buyers in S^* from R.

4. Return the probabilistic allocation vector $p = (p_i)$.

Next, we show that the GPS mechanism indeed computes a leximin random allocation. This establishes the equivalence between the GPS mechanism and the generalized leximin mechanism for dichotomous preferences.

Lemma 4.1. *Let p^t be the probability that each of buyer is assigned to an acceptable seller in iteration t of the algorithm. Then the sequence p^t is non-decreasing with respect to time, i.e. $p^1 \leq p^2 \leq \ldots \leq p^k$, where k is the last iteration of the algorithm.*

Proof. It suffices to show that $p^t \leq p^{t+1}$. Let S_t and P_t be the bottleneck set of buyers in iteration t and probabilistic assignments matrix at the end of iteration t respectively. Since S_t minimizes the value of $\mathcal{S}(S, O_S, P_{t-1})$ as a function of S, it must be the case that $\mathcal{S}(S_t \cup S_{t+1}, O_{S_t \cup S_{t+1}}, P_{t-1}) \geq \mathcal{S}(S_t, O_{S_t}, P_{t-1})$. Consequently, there is a distribution over packings that ensures every buyer in $S_t \cup S_{t+1}$ is matched to one of her acceptable sellers with probability at least $\mathcal{S}(S_t, O_{S_t}, P_{t-1})$. This implies that we can match each buyer in S_{t+1} to be matched to one of her acceptable sellers with probability at least $\mathcal{S}(S_t, O_{S_t}, P_{t-1})$ while giving the same guarantees to buyers in S_t. Then by definition, it must be the case that

$$p^{t+1} = \mathcal{S}(S_{t+1}, O_{S_{t+1}}, P_t) \geq \mathcal{S}(S_t, O_{S_t}, P_{t-1}) = p^t,$$

as desired. □

[27] Note that due to the nature of the algorithm, the duplication with $q = 0$ is always feasible.

Proposition 4.1. *The GPS mechanism computes a leximin probabilistic allocation for the dichotomous preference domain.*

Proof. Let p be the probabilistic allocation vector computed by the PS mechanism. We will show inductively that p is lexicographically identical to the leximin vector u entry by entry. Let v_i denote the probabilistic allocation of buyer i in an allocation vector v, and $v_{(j)}$ denote the j-th smallest entry of the vector v.

Base case: the probabilistic assignment computed by PS is a feasible assignment. Hence, the value of a minimum element is at most the value of a minimum element of the leximin probability vector, or $p_{(1)} \leq u_{(1)}$. On the other hand, look at the subset of buyers S_1, the subset of buyers who received the minimum probabilistic assignment in p, note that this corresponds to the subset of buyers who received an assignment in the first iteration of PS by Lemma 4.1. The PS mechanism maximizes their average probabilistic allocation among all feasible allocations, which is no worse than the minimum probabilistic assignment received by the buyers in S_1 in any feasible allocation. Let $i \in S_1$ be an buyer who receives the minimum probabilistic allocation amongst all buyers in S_1 with respect to u, then we have that $p_{(1)} \geq u_i \geq u_{(1)}$. Hence, it is the case that $p_{(1)} = u_{(1)}$. Moreover, we can assume without lost of generality that $p_{(1)}$ and $u_{(1)}$ correspond to the utility of the same buyer, namely buyer i.

Now, conditioning on the fact that the first k minimum probabilistic entries of p and u agree, we will show that $(k+1)$-st entry also agrees. Again, since p corresponds to a feasible random assignment, and the k smallest entries of p and u agree, it must be the case that $p_{(k+1)} \leq u_{(k+1)}$. To show that $p_{(k+1)} \geq u_{(k+1)}$, we will assume from the inductive hypothesis that the same subset of buyers are assigned the k minimum probabilities in both p and u. Note that if we continue the PS mechanism on the remaining set of buyers R after fixing the allocation of the buyers who received the k smallest probabilistic assignments, then we would have the identical probabilistic allocation for all buyers as the one we get starting from scratch. Let S_1^R be the first bottleneck set computed amongst the remaining buyers R given the allocation for the k minimum probability buyers. By the previous observation and Lemma 4.1, every buyer in S_1^R is assigned to an acceptable seller with probability $p_{(k+1)}$. Note that the minimum probabilistic allocation of the buyer i in S_1^R given by u is no better than the one computed by PS, since PS maximizes their allocations over all assignments that share the k smallest entries with that of PS. Hence, we get that $p_{(k+1)} \geq u_i \geq u_{(k+1)}$. Moreover, together with the inductive hypothesis, we have shown that the same subset of buyers are assigned the $k+1$ minimum probabilities in both p

and u. □

4.3.2 Full Preference Domain

In this subsection, we describe the GPS mechanism over the full preference domain. Let p_{ij} be the probability that buyer i is assigned a seller from his j-th tier of preference indifference class A_{ij}. let C_i be the index of the preference indifference class that buyer i is currently consuming from. Let R denote the remaining set of buyers to be considered by the algorithm.

1. Initialize $p_{ij} = 0$ and $C_i = 1$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, L_i$. Set $R = N$.

2. While $R \neq \emptyset$:

   (a) Create an instance of the packing subproblem as follows. Fixed a subset of buyers S, let l be the least common multiple of all denominators of p_{ij} where $p_{ij} > 0$. ($l = 1$ if all $p_{ij} = 0$.) For some $c \in \mathbb{N}$, we create cl copies of each seller in

   $$\{s \in A_{ij} \text{ for some } i \in S, j = 1, \ldots, C_i \text{ or } i \notin S, j = 1, \ldots, C_i - 1 \text{ if } C_i > 1\},$$

   and $p_{ij}cl$ copies of buyer i each of whom finds only sellers in A_{ij} acceptable for every $j < C_i$ and $i = 1, \ldots, n$. For every $i \in S$, let $m_{i,C_i} \geq 0$ be the number of copies of buyers i each finding only sellers in A_{i,C_i} acceptable. We require m_{i,C_i} to satisfy the following condition for every pair of buyers $i, i' \in S$

   $$\sum_{j<C_i} p_{ij}cl + m_{i,C_i} = \sum_{j<C_{i'}} p_{i'j}cl + m_{i',C_{i'}} = M \tag{4.4}$$

   We define the score $\mathcal{S}(S, O_S, P) = \max_{m_{i,C_i}, i \in S} \frac{M}{cl}$, for all parameters m_{i,C_i}'s, c that satisfies (4.4) and for which the duplicated packing problem has a feasible solution.[28]

   (b) Find $S^* \in \arg\min \mathcal{S}(S, O_S, P)$ over all $S \subseteq R$. Each buyer in $i \in S^*$ is matched to a seller in A_{i,C_i} with probability $\mathcal{S}(S^*, O_{S^*}, P) - \sum_{j<C_i} p_{ij}$. The distribution over packings is determined by the duplication subproblem that yields $\mathcal{S}(S^*, O_{S^*}, P)$.

   (c) If $\mathcal{S}(S^*, O_{S^*}, P) < 1$, then for every $i \in S^*$, set $C_i = C_i + 1$ if $C_i < L_i$. Otherwise, remove i from R. If $\mathcal{S}(S^*, O_{S^*}, P) = 1$, then remove sellers S^* from R. C_i stays the same for every $i \in R \backslash S^*$.

[28] One can show that there always exists a feasible solution for $M = \max_{i \in S} \sum_{j<C_i} p_{ij}$.

3. Return the probability allocation matrix P.

Next, we show that the GPS mechanism indeed computes a leximin random allocation. This establishes the equivalence between the GPS mechanism and the generalized leximin mechanism for general preferences.

Lemma 4.2. *Let S_t and P_t be the bottleneck set of buyer preference indifference class pairs in iteration t and probabilistic assignments matrix at the end of iteration t respectively. Then the sequence $\mathcal{S}(S_t, O_{S_t}, P_{t-1})$ is non-decreasing with respect to t.*

Proof. It suffices to show that $\mathcal{S}(S_t, O_{S_t}, P_{t-1}) \leq \mathcal{S}(S_{t+1}, O_{S_{t+1}}, P_t)$. Since S_t minimizes the value of $\mathcal{S}(S, O_S, P_{t-1})$ as a function of S, it must be the case that $\mathcal{S}(S_t \cup S_{t+1}, O_{S_t \cup S_{t+1}}, P_{t-1}) \geq \mathcal{S}(S_t, O_{S_t}, P_{t-1})$. Consequently, there is a distribution over packings that ensures every buyer in $S_t \cup S_{t+1}$ is matched to one of the sellers the preference indifference class that she is currently consuming or better with probability at least $\mathcal{S}(S_t, O_{S_t}, P_{t-1})$. This implies that buyers in S_{t+1} can be matched to one of the sellers from her current preference indifference class with probability at least $\mathcal{S}(S_t, O_{S_t}, P_{t-1})$ while giving the same guarantees to buyers in S_t. Then it must be the case that

$$\mathcal{S}(S_{t+1}, O_{S_{t+1}}, P_t) \geq \mathcal{S}(S_t, O_{S_t}, P_{t-1}),$$

as desired. □

Theorem 4.1. *The GPS mechanism computes a leximin probabilistic allocation for full preference domain.*

Proof. Let v^P be the vector of probability partial sums $(\sum_{j \in T_{il}} p_{ij})_{il}$ computed by the PS mechanism. We inductively show that v^P is lexicographically identical to the leximin vector u entry by entry. Given a vector of probability partial sums v, let v_{il} be the probability that buyer i will be assigned to a seller from one of her top l preference indifference classes computed by the PS mechanism, and let $v_{(j)}$ denote the j-th smallest entry of the vector v.

Base case: the probabilistic assignment computed by PS is a feasible assignment. Hence, the value of a minimum element is at most the value of a minimum element of the leximin probability vector, or $v^P_{(1)} \leq u_{(1)}$. On the other hand, look at S_1, the subset of buyers who received the minimum probabilistic assignment in P from their current indifference class, note that this corresponds to the subset of buyer preference indifference class pairs who received an assignment in the first iteration of PS by Lemma 4.2. The PS mechanism maximizes their average probabilistic allocation for being assigned to a seller from their top preference indifference class of sellers among all feasible random allocations, which is no worse than the minimum probabilistic assignment for the top preference indifference class received by the buyers in S_1 in any feasible random allocation. Let $i \in S_1$ be an buyer who receives the minimum probabilistic allocation from her top preference indifference class amongst all buyers in S_1 with respect to u, then we have that $p_{(1)} \geq u_{i1} \geq u_{(1)}$. Hence, it is the case that $v^P_{(1)} = u_{(1)}$. Moreover, we can assume without lost of generality that $p_{(1)}$ and $u_{(1)}$ correspond to the utility of the same buyer preference indifference class pair, namely buyer $(i, 1)$.

Now, conditioning on the fact that the first k minimum probabilistic entries of p and u agree, we will show that the $(k+1)$-st entry also agrees. Again, since p corresponds to a feasible random assignment, and the k smallest entries of p and u agree, it must be the case that $p_{(k+1)} \leq u_{(k+1)}$. To show that $p_{(k+1)} \geq u_{(k+1)}$, we will assume from the inductive hypothesis that the subset of buyer preference indifference class pairs corresponding to the k minimum probabilities are the same in both p and u. Note that if we continue the PS mechanism upon fixing the allocation of the buyers who received the k smallest probabilistic assignments, then we would have the identical probabilistic allocation for all buyers as the one we get starting from scratch.[29] Let S^R_1 be the first bottleneck set computed amongst the remaining buyers R given the allocation for the k minimum probability buyer preference indifference class pairs. By the previous observation and Lemma 4.2, every buyer in S^R_1 is assigned to a seller from the preference indifference class that the buyer is currently consuming from or better with probability $p_{(k+1)}$. Note that the minimum probabilistic allocation of the buyer in S^R_1 for a seller from the preference indifference class that the buyer is currently consuming from or better given by u is no better than the one computed by PS, since PS maximizes their allocations over all assignments that share the k smallest entries with that of PS. We will refer to this buyer as i. Hence, we get that $p_{(k+1)} \geq u_{i,C_i} \geq u_{(k+1)}$. Moreover, together with the inductive hypothesis, we have shown that the same subset of buyer preference indifference class pairs are assigned the $k+1$ minimum probabilities in both p and u.

□

[29]Subject to the appropriate tie breaking rule.

Now that we have established the equivalence between the GPS mechanism and the generalized leximin mechanism, we will use the terms "GPS mechanism" and "generalized leximin mechanism" interchangeably throughout the remainder of the chapter. Bogomolnaia first established this equivalence in the UDC special case in [27]. One can view Theorem 4.1 as a generalization of her result. Note that the leximin allocation sequentially maximizes minimum entry of a probabilistic allocation vector, whereas the GPS mechanism iteratively identifies a subset of agents who maximum possible allocation is minimized. Hence, one can view our result as establishing the equivalence between a max min and a min max allocation problem.

4.4 Properties of the Generalized Leximin Mechanism

Kurokawa et al. [81] showed that the leximin mechanism satisfies satisfies envy-freeness, proportionality, Pareto optimality, and weak group strategyproofness in the dichotomous setting. The generalized leximin mechanism is not strategyproof in the general preference domain as it coincides with the PS mechanism in the UDC special case. Even for the special case, it is shown in [29] that the PS mechanism is not weakly strategyproof. Somewhat surprisingly, we show that the leximin allocation is not envy free for the general preference domain. On the other hand, we show that the leximin allocation satisfies ordinal efficiency (an extension of Pareto optimality) in the general preference domain, and proportionality in the dichotomous preference domain (note that our definition of proportionality is stronger than that of [81]. We conjecture that proportionality is satisfied by the leximin allocation in the general preference domain as well.

Proposition 4.2. *The leximin allocation is not envy-free.*

Proof. Consider the following instance instance with 4 buyers $(1, 2, 3, 4)$ and 2 sellers (a, b), with buyer demand quantities being

$$d_1 = 3, d_2 = d_3 = d_4 = 2,$$

and seller supply quantities being

$$s_a = 4,\ s_b = 3.$$

Buyer preferences are as follows:

$$1 : a$$

$$2 : a, b$$

$$3 : b, a$$

$$4 : b$$

Note that the leximin allocation is given by

	a	b
1:	1/2	0
2:	1/2	0
3:	1/4	1/2
4:	0	1/2

The distribution over packings for the above assignment is as follows.

$$1 - a, 3 - b \ \text{ w.p. } 1/4$$

$$1 - a, 4 - b \ \text{ w.p. } 1/4$$

$$2 - a, 3 - b \ \text{ w.p. } 1/4$$

$$2 - a, 3 - a, 4 - b \ \text{ w.p. } 1/4$$

Buyer 2, who is assigned to only seller a with probability 1/2, will envy the allocation of buyer 3. The main reason that buyer 3 receives a better probabilistic assignment than buyer 2 is that seller a, who is buyer 3's second most preferred choice, can serve buyers 2 and 3 simultaneously. On the contrary, seller b, who is buyer 2's second most preferred choice, cannot serve buyers 2 and 3 simultaneously.

□

Next, we show that leximin allocation is proportional in the dichotomous preference domain.

Proposition 4.3. *A leximin allocation is proportional when agents have dichotomous preferences.*

Proof. Let $\mathcal{A}$ be a leximin assignment. Let $v_{\mathcal{A}}$ denote the agent to acceptable resource assignment probability vector corresponding to $\mathcal{A}$. let $p_1 < p_2 < \ldots < p_k$ be distinct entry values of $v_{\mathcal{A}}$. Define the j-th bottleneck set B_j to be the set of agents who receives an acceptable resource with probability p_j for $j = 1, \ldots, k$. Note that the bottleneck sets form a partition of the set of agents. Consider the following lemma.

Lemma 4.3. *Let R_j denote the set of resources that are desired by an agent in one of the first j bottleneck sets. Let D_i be the set of resources that agent i finds acceptable. For any set of resources R, let A^j_R be the set of agents from the first j bottleneck sets who desires some resource from R. Then we have that for any j such that $|A^j_{D_i}| \geq 1$,*

$$p_j \geq \min\left\{\frac{|D_i \cap R_j|}{|A^j_{D_i}|}, 1\right\} \quad \forall i, j.$$

To see that this lemma implies proportionality. Let A_{D_i} be the set of agents who find some resource in D_i acceptable. Let k^i be the bottleneck set that the agent i belongs to. Applying this lemma with $j = j^i$ (note that $i \in A^{j^i}_{D_i}$, so $|A^j_{D_i}| \geq 1$), we get

$$p_{j^i} \geq \min\left\{\frac{|D_i \cap R_{j^i}|}{|A^{j^i}_{D_i}|}, 1\right\} = \min\left\{\frac{|D_i|}{|A^{j^i}_{D_i}|}, 1\right\} \geq \min\left\{\frac{|D_i|}{|A_{D_i}|}, 1\right\}.$$

The equality holds because $D_i \subseteq R_{j^i}$, and the second inequality holds because $A^{j^i}_{D_i} \subseteq A_{D_i}$. Proportionality follows immediately as agent i belongs to the bottleneck set B_{j^i}. Now we prove the lemma.

Proof. We show this via induction on the bottleneck set starting from the first j. From the inductive hypothesis, we have that

$$p_j \geq p_{j-1} \geq \min\left\{\frac{|D_i \cap R_{j-1}|}{|A^{j-1}_{D_i}|}, 1\right\}.$$

We will assume from now on that $\frac{|D_i \cap R_{j-1}|}{|A^{j-1}_{D_i}|} \leq 1$, otherwise we are done with the proof. Since we can write $\frac{|D_i \cap R_j|}{|A^j_{D_i}|}$ as a convex combination of $\frac{|D_i \cap R_{j-1}|}{|A^{j-1}_{D_i}|}$ and $\frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i} \backslash A^{j-1}_{D_i}|}$, it suffices to show that $p_j \geq \frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i} \backslash A^{j-1}_{D_i}|}$.

Note that $R_j \backslash R_{j-1}$ are resources desired only by agents in bottleneck set B_j out of all agents in the first j bottleneck sets, and $A^j_{D_i} \backslash A^{j-1}_{D_i}$ are the set of agents in B_j who desire some resource in D_i. Consequently, we deduce that $A^j_{D_i \cap (R_j \backslash R_{j-1})} \subseteq A^j_{D_i} \backslash A^{j-1}_{D_i}$. We will show that $p_j \geq \frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i \cap (R_j \backslash R_{j-1})}|} \geq$

$\frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i} \backslash A^{j-1}_{D_i}|}$. To do so, we will identify a special set of resources $\tilde{R} \subseteq D_i \cap (R_j \backslash R_{j-1})$ (to be defined later). For agents in $\cup_{j'=1}^{j} B_{j'} \backslash A^j_{\tilde{R}}$, we fix their allocation to be the one given by a leximin allocation $\mathcal{A}$. For agents in $A^j_{\tilde{R}}$, we will produce a distribution over *matchings* such that each of them receives an acceptable object with probability $\frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i \cap (R_j \backslash R_{j-1})}|}$ using only resources in $\tilde{R}$. The agents from all remaining bottleneck sets will receive no resource. Call the distribution over packings that we find $\mathcal{A}'$. Then it must be the case that $p_j \geq \frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i \cap (R_j \backslash R_{j-1})}|}$. Otherwise, $v_{\mathcal{A}'}$ entry-wise dominates $v_{\mathcal{A}}$ for every agent in the first j bottleneck sets and strictly dominates $v_{\mathcal{A}}$ for agents in $A^j_{\tilde{R}}$. Then the distribution over packing $\epsilon \mathcal{A}' + (1-\epsilon)\mathcal{A}$ would lexicographically dominate $\mathcal{A}$ for ϵ sufficiently small.

The following lemma formally defines $\tilde{R}$ and demonstrates its existence. Let $R(A, R)$ denote the set of resources some agent in A desires among the resources in R. Since we will only use the term $R(A, D_i \cap (R_j \backslash R_{j-1}))$ for some $A \subseteq A^j_{D_i \cap (R_j \backslash R_{j-1}))}$ in the lemma and its proof, we use $R(A)$ to denote $R(A, D_i \cap (R_j \backslash R_{j-1}))$ from now on for simplicity.

Lemma 4.4. *There exists $\tilde{R} \subseteq D_i \cap (R_j \backslash R_{j-1})$ such that for any $A \subseteq A^j_{\tilde{R}}$,*

$$\frac{|R(A)|}{|A|} \geq \frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i \cap (R_j \backslash R_{j-1})}|}.$$

Lemma 4.4 along with the generalized Hall's marriage theorem imply that there exists a distribution over matchings in which every agent in A^j_R is matched with a resource with probability at least $\frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i \cap (R_j \backslash R_{j-1})}|}$. The generalized Hall's marriage theorem can be shown via a direct application of the max-flow min cut theorem.

We now describe the procedure for obtaining $\tilde{R}$:

1. Initialize $R = D_i \cap (R_j \backslash R_{j-1})$.
2. If $A^j_R = \min_{A \subseteq A^j_R} \frac{|R(A)|}{|A|}$, then stop and set $\tilde{R} = R$. Otherwise, select a set $R(A^*)$, where A^* is a maximal subset in $\arg\min_{A \subseteq A^j_R} \frac{|R(A)|}{|A|}$, set $R = R \backslash R(A^*_t)$ for the next iteration and repeat.

Let R_t and A^*_t be the set R and A^* in iteration t of the above procedure respectively. Note that for any t, we can write $\frac{|R_t|}{|A^j_{R_t}|}$ as a convex combination of $\frac{|R(A^*_t)|}{|A^*_t|}$ and $\frac{|R_t \backslash R(A^*_t)|}{|A^j_{R_t} \backslash A^*_t|} = \frac{|R_{t+1}|}{|A^j_{R_t} \backslash A^*_t|}$. Since $\frac{|R(A^*_t)|}{|A^*_t|} \leq \frac{|R_t|}{|A^j_{R_t}|}$, it must be the case that $\frac{|R_{t+1}|}{|A^j_{R_t} \backslash A^*_t|} \geq \frac{|R_t|}{|A^j_{R_t}|}$. Moreover, note that $A^j_{R_{t+1}} \subseteq A^j_{R_t} \backslash A^*_t$. Putting it together, we get

$$\frac{|R_{t+1}|}{|A^j_{R_{t+1}}|} \geq \frac{|R_{t+1}|}{|A^j_{R_t} \backslash A^*_t|} \geq \frac{|R_t|}{|A^j_{R_t}|}.$$

Hence, when the procedure terminates, we have that for set $A \subseteq \tilde{R}$

$$\frac{|R(A)|}{|A|} \geq \frac{|\tilde{R}|}{|A^j_{\tilde{R}}|} \geq \frac{|R_1|}{|A^j_{R_1}|} = \frac{|D_i \cap (R_j \backslash R_{j-1})|}{|A^j_{D_i \cap (R_j \backslash R_{j-1})}|},$$

as desired. □

□

Proposition 4.4. *A leximin allocation is ordinally efficient.*

Proof. This follows directly from the definition. Given a leximin allocation and its probabilistic assignment matrix P, if there exists another feasible random allocation whose probabilistic assignment matrix Q stochastically dominates P, then the corresponding vector v_Q would also entry-wise weakly dominate v_Q and strictly dominate v_Q in at least one entry, which leads to a contradiction. □

4.5 Discussion

We showed that the positive results of [81] for the leximin mechanism in the dichotomous preference domain does not carry over to the full preference domain. In particular, the leximin mechanism is no longer envy-free or strategyproof. Numerous impossibility results (see e.g. Chapter 2 of this thesis, [29, 76, 11]) have shown incompatibility between strategyproofness and various efficiency and fairness notions on different preference domains in the UDC setting. Nonetheless, even setting strategyproofness aside, our work opens up the question whether an envy-free, proportional, and ordinally efficiency mechanism exists in this setting.

The computational complexity of the GPS mechanism is another direction that can be explored. For the UDC setting, Katta and Sethuraman [76] showed that the bottleneck set of agents in each iteration can be identified by solving a parametric network flow problem. On the other hand, Kurokawa [81] showed that computing a leximin allocation in our setting is NP-hard. Hence, we cannot hope for a polynomial time implementation of the GPS mechanism unless P = NP. Nonetheless, in the current specification of the GPS mechanism, the packing subproblem in Section 4.3 (which is already difficult) is computed for every subset of agents S. Perhaps one can compute the bottleneck agent set via a single optimization problem and reduce the computational complexity as a result.

Chapter 5

Approximately Optimal Mechanisms for Strategyproof Facility Location: Minimizing L_p Norm of Costs

Joint work with Itai Feigenbaum.

5.1 Introduction

We consider the problem of locating a single facility on the real line. This facility serves a set of n agents, each of whom is located somewhere on the line as well. Each agent cares about his distance to the facility, and incurs a disutility (equivalently, cost) that is equal to his distance to access the facility. An agent's location is assumed to be private information that is known only to him. Agents report their locations to a central planner who decides where to locate the facility based on the reports of the agents. The planner's objective is to minimize a "social" cost function that depends on the vector of distances that the agents need to travel to access the facility. It is natural for the planner to consider locating the facility at a point that minimizes her objective function, but in that case the agents may not have an incentive to report their locations truthfully. As an example, consider the case of 2 agents located at x_1 and x_2 respectively, and suppose the location that optimizes the planner's objective is the mid-point $(x_1 + x_2)/2$. Then, assuming $x_1 < x_2$, agent 1 has an incentive to report a location $x_1' < x_1$ so that the planner's decision results in the facility being located closer to his true location. The planner can address this issue by restricting herself to a *strategyproof* mechanism: by this we mean that it should be a (weakly) dominant strategy for each agent to report his location truthfully to the central planner. For instance, the planner could always locate the facility at agent 1's reported location, which is strategyproof. Even though strategyproofness is an attractive property, but it comes at a cost: based on the earlier example, it is clear that the planner cannot hope to optimize her objective. One way to avoid this difficulty is to assume an environment in which agents (and the planner) can make or receive payments; in such a case, the planner selects the location of the facility, and also a payment scheme, which specifies the amount of money an agent pays (or receives) as a function of the reported locations of the agents as well as the location of the facility. This option gives the planner the ability to support the "optimal" solution as the outcome of a strategyproof mechanism by constructing a carefully designed payment scheme in which any potential benefit for a misreporting agent from a change in the location of the facility is offset by an increase in his payment.

There are many settings, however, in which such monetary compensations are either not possible or are undesirable. This motivated Procaccia and Tennenholtz [98] to formulate the notion of *Approximate Mechanism Design without Money*. In this model the planner restricts herself to strategyproof mechanisms, but is willing to settle for one that does not necessarily optimize her objective. Instead, the planner's goal is to find a mechanism that effectively *approximates* her objective function. This is captured by the standard notion of approximation that is widely used in the CS literature: for a minimization problem, an algorithm is an α-approximation if the solution it finds is guaranteed to have cost at most α times that of the optimal cost ($\alpha \geq 1$).

Procaccia and Tennenholtz [98] apply the notion of approximate mechanism design without money to the facility location problem considered here for two different objectives: (i) *minisum*, where the goal is to minimize the sum of the costs of the agents; and (ii) *minimax*, where the goal is to minimize the maximum agent cost. They show that for the minimax objective choosing any k-th median—picking the kth largest reported location—is a strategyproof, 2-approximate mechanism. They design a randomized mechanism called LRM (Left-Right-Middle) and show that it is a strategyproof, 3/2-approximate mechanism; furthermore, they show that those mechanisms provide the optimal worst-case approximation ratio possible (among all deterministic and randomized strategyproof mechanisms, respectively). For the *minisum* objective, it is known that choosing the median reported location is optimal and strategyproof, see [90]. Feldman and Wilf [60] consider the same facility location problem on a line but with the social cost function being the L_2 norm of the agents' costs (Feldman and Wilf actually used the sum of squares of the agents' costs, however most of their results can be easily converted to the L_2 norm. Of course, the approximation ratios they report need to be adjusted as well). They show that the median is a $\sqrt{2}$-approximate strategyproof mechanism for this objective function, and provide a randomized $(1 + \sqrt{2})/2$-approximate strategyproof mechanism. Feldman and Wilf also generalize the median mechanism to maintain strategyproofness and a $\sqrt{2}$ approximation ratio on trees; furthermore, they provide a family of randomized strategyproof mechanisms for trees, and in particular show that a member of this family reduces the approximation ratio to strictly below $\sqrt{2}$. A general survey of approximate mechanism design without money for facility location problems has been written by [45].

Aside from the recent literature on approximate mechanism design, our work is loosely related to other strands in the literature with a much longer history. First is the classical work on social choice, which deals with the aggregating the preferences of a set of voters over a set of alternatives, see e.g. [92]. The location problem we consider is a special case in which the alternatives are all possible points on the real line (the location of the facility), and agents have single-peaked preferences. An important difference, however, is the following: a typical social choice problem is to find an aggregation rule satisfying a desired set of properties, whereas in our case the planner wishes to optimize or approximate a given social objective function. Nevertheless, various techniques and results from this literature are useful in our setting as well. An important result along these lines is [90]'s characterization of strategyproof mechanisms on the line. A parallel characterization result was developed by [109] for general graphs. In both these papers, much like in this chapter, generalized medians play an important role; also, despite not having a specific objective function, these characterizations assume less specific efficiency related properties, such as Pareto efficiency and onto range. Additional papers along these lines are [20, 50]. It is important to note that impossibility results abound in social choice models—our focus on the simple special case enables us to avoid impossibility results such as the Gibbard-Satterthwaite Theorem (see [68, 106]), which implies the non-existence of a *reasonable* social choice function. Second is the classical work in operations research on graphical location problems that considers locating the facility at a Condorcet point (see e.g. [69, 82, 15, 16]). (A Condorcet point is one that is preferred by a majority of agents to any other location.) This literature seeks to establish bounds on the total cost to all the agents to access the facility divided by the minimum cost, with the understanding that smaller ratios are better. However, this literature does not model individual agent incentives, and moreover does not also explore other mechanisms. Finally, there is a rich literature on facility location problems and variations (such as the k-median and k-center problems) where agent incentives are not taken into account. In such problems, there is typically a single objective function (the planner's), and agent locations are known. In this literature, one resorts to approximation algorithms for a different reason—often, these optimization problems turn out to be computationally intractable, and the focus is on developing computationally efficient heuristics for which a worst-case approximation guarantee can be proved (see [128], and chapters 25-26 of [125]). To our knowledge, most of the algorithms designed in this literature violate our (rather strong) strategyproofness requirements. In addition, some consideration has been given in literature to the circle topology, by [5, 6]. It is important to note that while the idea of using approximate mechanisms to induce strategyproofness was first proposed in 2009, the problem of finding strategyproof mechanisms has received attention beforehand. Those papers allow much more generality in the preferences of the agents, but typically do not have a specific objective function to optimize, and thus approximation is not of relevance there.

In this chapter, we follow the suggestion of Feldman and Wilf [60] and study the problem of locating a single facility on a line, but with the objective function being the L_p norm of the vector of agent-costs (for general $p \geq 1$). In the context of real world facility location problems, where the agents must drive to and from the facility, the L_p norm can represent situations where travel time or other cost increases superlinearly with the distance (as suggested in [35]). For example, when driving over larger distances, there is an increased likelihood (depending on traffic) of the need to stop and refuel, or, in the case of electric cars, stop and recharge–which is even more costly since such recharging can be done at home, without wasting the driver's time. As another example, certain hybrid cars increase their fuel consumption in longer drives— which is relevant if the cost represents fuel consumption rather than travel time. For such problems, our results provide strong lower bounds, robust to the topology of the road network (since they only require a line) and the value of p. We also hope that our results regarding the median will guide the construction of good mechanisms for more general topologies, similarly to the case of $p = 2$ in [60], where the optimality of the median on the line inspires the construction of a mechanism for tree networks using the appropriate adaptation of the median. Another use of the L_p norm is to strike a balance between efficiency and fairness. The cases of $p = 1$ and $p = \infty$, which were both studied in [98], can be viewed as representing the two extremes on the spectrum between maximizing efficiency (minimizing the total social cost) and maximizing fairness (minimizing the cost of the agent who is worst off). Thus, our definition of social cost allows for a controlled tradeoff between efficiency and fairness by varying the value of p. On the line, this interpretation of the L_p norm becomes particularly interesting in the context of voting. Public opinion on many issues is considered to be on a spectrum between political left and right, lending itself naturally to a one dimensional description. One of the common problems in democratic societies is to balance between majority rule and respecting minority rights; thus, the L_p measure allows for a quantitative exploration of this balance. Of course, this interpretation of the L_p norm can be relevant to physical facility location problems as well.

We define the problem formally in section 2. In section 3, we show that the median mechanism (which is strategyproof) provides a $2^{1-\frac{1}{p}}$ approximation ratio, and that this is the optimal approximation ratio among all deterministic strategyproof mechanisms. We move onto randomized mechanisms in section 4. First, we present a negative result: we show that for integer $\infty > p > 2$, *no* mechanism— from a rather large class of randomized mechanisms— has an approximation ratio better than that of the median mechanism, as the number of agents goes to infinity. It is worth noting that all the mechanisms proposed in literature so far— for minimax, minisum, and the L_2 social cost functions— belong to this class of mechanisms. Next, we consider the case of 2 agents, and show that the LRM mechanism provides the optimal approximation ratio among all randomized strategyproof mechanisms (that satisfy certain mild assumptions) for this special case, for every $p \geq 1$. Our result for the special case of 2 agents also gives a lower bound on the approximation ratio for all randomized mechanisms. We briefly discuss some directions for further research in section 5. In Appendix B we discuss some technical details omitted from the chapter, as well as an additional negative result for an alternative definition of the agents' cost.

5.2 Model

Let $N = \{1, 2, \ldots, n\}$, $n \geq 2$, be the set of agents. Each agent $i \in N$ reports a location $x_i \in \mathbb{R}$. A *deterministic* mechanism is a collection of functions $f = \{f_n | \, n \in \mathbb{N}, n \geq 2\}$ such that each $f_n : \mathbb{R}^n \to \mathbb{R}$ maps each location profile $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ to the location of a facility. We will abuse notation and let $f(\mathbf{x})$ denote $f_n(\mathbf{x})$. Under a similar notational abuse, a *randomized* mechanism is a collection of functions f that maps each location profile to a probability distribution over $\mathbb{R}$: if $f(x_1, x_2, \ldots, x_n)$ is the distribution π, then the facility is located by drawing a single sample from π.

Our focus will be on deterministic and randomized mechanisms for the problem of locating a single facility when the location of any agent is *private* information to that agent and cannot be observed or otherwise verified. It is therefore critical that the mechanism be *strategyproof*—it should be optimal for each agent i to report his *true* location x_i rather than something else. To that end we assume that if the facility is located at y, an agent's disutility, equivalently cost, is simply his distance to y. Thus, an agent whose true location is x_i incurs a cost $C(x_i, y) = |x_i - y|$. If the location of the facility is random and according to a distribution π, then the cost of agent i is simply $C(x_i, \pi) = \mathbb{E}_{y\sim\pi}|x_i - y|$, where y is a random variable with distribution π. The formal definition of strategyproofness is now:[30]

Definition 5.1. *A mechanism f is strategyproof if for each $i \in N$, each $x_i, x_i' \in \mathbb{R}$, and for each $\mathbf{x_{-i}} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots x_n) \in \mathbb{R}^{n-1}$,*

$$C(x_i, f(x_i, \mathbf{x_{-i}})) \leq C(x_i, f(x_i', \mathbf{x_{-i}})),$$

where $(\alpha, \mathbf{x_{-i}})$ denotes a vector with the i-th component being α and the j-th component being x_j for all $j \neq i$.

The class of strategyproof mechanisms is quite large: for example, locating the facility at agent 1's reported location is strategyproof, but is not particularly appealing because it fails almost every reasonable notion of fairness and could also be highly "inefficient". To address these issues, and to winnow down the class of acceptable mechanisms, we impose additional requirements that stem from efficiency or fairness considerations. In this chapter we assume that locating a facility at y when the location profile is $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ incurs the *social cost*

$$sc(\mathbf{x}, y) \;=\; \left(\sum_{i\in N} |x_i - y|^p\right)^{1/p}, \quad p \geq 1,$$

which one can viewed as the L_p norm of the individual cost vector. For a randomized mechanism f that maps $\mathbf{x}$ to a distribution π, we define the social cost to be[31]

$$sc(\mathbf{x}, \pi) \;=\; \mathbb{E}_{y\sim\pi}\left[\left(\sum_{i\in N} |x_i - y|^p\right)^{1/p}\right].$$

[30] Note that for randomized mechanisms, we require strategyproofness in expectation, rather than ex-post.

[31] For this definition of social cost, an alternative option is to let the agents' costs increase non-linearly with their distance from the facility, in particular $C(x_i, y) = |x_i - y|^p$. In Appendix B we provide an interesting result for this case.

For this definition of social cost, our goal now is to find a strategyproof mechanism that does well with respect to minimizing the social cost. A natural mechanism (and this is the approach taken in the classical literature on facility location) is the "optimal" mechanism: each location profile $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is mapped to $OPT(\mathbf{x})$, defined as $OPT(\mathbf{x}) \in \arg\min_{y \in \mathbb{R}} sc(\mathbf{x}, y)$.[32] This optimal mechanism is not strategyproof as shown in the following example.

Example. Suppose there are two agents located at the points 0 and 1 respectively on the real line. If they report their locations truthfully, the optimal mechanism will locate the facility at $y = 0.5$, for any $p > 1$. Assuming agent 2 reports $x_2 = 1$, if agent 1 reports $x_1' = -1$ instead, the facility will be located at 0, which is best for agent 1.

Given that strategyproofness and optimality cannot be achieved simultaneously, it is necessary to find a tradeoff. In this chapter we shall restrict ourselves to strategyproof mechanisms that approximate the optimal social cost as best as possible. The notion of approximation that we use is standard in computer science: an α-approximation algorithm is one that is guaranteed to have cost no more than α times the optimal social cost. Formally, the approximation ratio of an algorithm A is $\sup_I\{A(I)/OPT(I)\}$, where the supremum is taken over all possible instances I of the problem, and $A(I)$ and $OPT(I)$ are, respectively, the costs incurred by algorithm A and the optimal algorithm on the instance I.[33] Our goal then is to design strategyproof (deterministic or randomized) mechanisms whose approximation ratio is as close to 1 as possible.

5.3 The Median Mechanism

For the location profile $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the median mechanism is a deterministic mechanism that locates the facility at the "median" of the reported locations. The median is unique if n is odd, but not when n is even, so we need to be more specific in describing the mechanism. For odd n, say $n = 2k-1$ for some $k \geq 1$, the facility is located at $x_{[k]}$, where $x_{[k]}$ is the kth largest component of the location

[32] Strictly speaking, the mechanism is not well defined in cases where the social cost at $\mathbf{x}$ is minimized by multiple locations, but we could pick an exogenous tie-braking rule to deal with such cases.

[33] For the case of randomized mechanisms, it should be noted that this is the approximation ratio is in expectation rather than with high probability.

profile. For even n, say $n = 2k$, the "median" can be any point in the interval $[x_{[k]}, x_{[k+1]}]$; to ensure strategyproofness, we need to pick either $x_{[k]}$ or $x_{[k+1]}$, and as a matter of convention we take the median to be $x_{[k]}$. It is well known that the median mechanism is strategyproof.[34] Furthermore, the median mechanism is *anonymous*.[35] Thus we may assume, without loss of generality, that each agent reports her location truthfully.

Our main result in this section is that, for any $p \geq 1$, the median mechanism uniformly achieves the best possible approximation ratio among all deterministic strategyproof mechanisms. We start with two simple observations, which will be used in the proof of this main result.

Lemma 5.1. *For any real numbers a, b, c with $a \leq b \leq c$, and any $p \geq 1$,*

$$(c-a)^p \;\leq\; 2^{p-1}[(c-b)^p + (b-a)^p].$$

Proof. For any $p \geq 1$, $f(x) = x^p$ is a convex function on $[0, \infty)$, and so for any $\lambda \in [0,1]$ and $x, y \geq 0$,

$$f(\lambda x + (1-\lambda)y)) \;\leq\; \lambda f(x) + (1-\lambda) f(y). \tag{5.1}$$

Setting $\lambda = 1/2$, $x = c - b$, and $y = b - a$, we get:

$$\frac{1}{2^p}(c-a)^p \;\leq\; \frac{1}{2}[(c-b)^p + (b-a)^p]. \tag{5.2}$$

Multiplying both sides of the inequality by 2^p gives the result. $\square$

Lemma 5.2. *For any non-negative real numbers a and b, and any $p \geq 1$,*

$$(a+b)^p \;\geq\; a^p + b^p.$$

Proof. For integer p, the result is a direct consequence of the binomial theorem; the same argument covers the case of rational p as well. Continuity implies the result for all p. $\square$

Theorem 5.1. *Suppose there are n agents with the location profile $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Define the social cost of locating a facility at y as $(\sum_{i=1}^n |y - x_i|^p)^{\frac{1}{p}}$ for $p \geq 1$. The social cost incurred by the median mechanism is at most $2^{1-\frac{1}{p}}$ times the optimal social cost.*[36]

[34] A classical paper of [90] for a closely related model shows that all deterministic strategyproof mechanisms are essentially generalized median mechanisms.

[35] In an anonymous mechanism, the facility location is the same for two location profiles that are permutations of each other.

[36] This is a generalization of the results for $p = 2$ [60], $p = 1$ and $p = \infty$ [98] (when $p = \infty$, the median mechanism provides a 2-approximation).

Proof. We may assume that $x_1 \leq ... \leq x_n$. Let OPT be a facility location that minimizes the social cost, and let m be the median. The inequality we need to prove is

$$\sum_{i=1}^{n} |m - x_i|^p \;\leq\; 2^{p-1} \sum_{i=1}^{n} |OPT - x_i|^p.$$

We do this by pairing each location x_i with its "symmetric" location x_{n+1-i} and arguing that the total cost of these two locations in the median mechanism is within the required bound of their total cost in an optimal solution. For even n, this completes the argument; for odd n the only location without such a pair is the median itself, which incurs *zero* cost in the median mechanism, and so the argument is complete. Formally, the result follows if we can show

$$|m - x_i|^p + |x_{n+1-i} - m|^p \leq 2^{p-1}(|OPT - x_i|^p + |OPT - x_{n+1-i}|^p), \quad \forall\ i \leq \lfloor n/2 \rfloor.$$

We consider two cases, depending on whether OPT is in the interval $[x_i, x_{n+1-i}]$ or not. In each of these cases, OPT may be above the median or below, but the proof remains identical in each subcase, so we give only one.

1. $x_i \leq m \leq OPT \leq x_{n+1-i}$ or $x_i \leq OPT \leq m \leq x_{n+1-i}$. We will prove the first of these subcases; the proof of the second is identical. Applying Lemma 5.1 by setting $a = m$, $b = OPT$, and $c = x_{n+1-i}$, we get

$$|x_{n+1-i} - m|^p \leq 2^{p-1}(|x_{n+1-i} - OPT|^p + |OPT - m|^p).$$

   Thus,

$$\begin{aligned} |m - x_i|^p + |x_{n+1-i} - m|^p &\leq |m - x_i|^p + 2^{p-1}(|x_{n+1-i} - OPT|^p + |OPT - m|^p) \\ &\leq 2^{p-1}(|m - x_i|^p + |x_{n+1-i} - OPT|^p + |OPT - m|^p) \\ &\leq 2^{p-1}(|x_{n+1-i} - OPT|^p + |OPT - x_i|^p), \end{aligned}$$

   where the last inequality is obtained by applying Lemma 5.2 to the terms $|m - x_i|^p$ and $|OPT - m|^p$.

2. $OPT \leq x_i \leq m \leq x_{n+1-i}$ or $x_i \leq m \leq x_{n+1-i} \leq OPT$. Again, we prove only the first subcase. Note that

$$\begin{aligned} |x_{n+1-i} - m|^p + |m - x_i|^p &\leq |x_{n+1-i} - x_i|^p \\ &\leq |OPT - x_{n+1-i}|^p \\ &\leq 2^{p-1}(|OPT - x_i|^p + |OPT - x_{n+1-i}|^p) \end{aligned}$$

where the first inequality follows from Lemma 5.2. (Note that Lemma 5.1 is not used in the proof of this case.)

We end this section by showing that *no* deterministic and strategyproof mechanism can give a better approximation to the social cost.

Lemma 5.3. *Consider the case of two agents and suppose the location profile is (x_1, x_2) with $x_1 < x_2$. For $p \geq 1$, suppose the social cost of locating a facility at y is $(|x_1 - y|^p + |x_2 - y|^p)^{1/p}$. Any deterministic mechanism whose approximation ratio is better than $2^{1-\frac{1}{p}}$ for $p > 1$ must locate the facility at y for some $y \in (x_1, x_2)$.*[37]

Proof. The function $sc(\mathbf{x}, y)$ is strictly convex in y, and its unique minimizer is $y^* = (x_1 + x_2)/2$, with the corresponding value $sc(\mathbf{x}, y^*) = |x_2 - x_1|/2^{1-\frac{1}{p}}$. Moreover $sc(\mathbf{x}, x_1) = sc(\mathbf{x}, x_2) = |x_2 - x_1| = 2^{1-\frac{1}{p}} sc(\mathbf{x}, y^*)$. It follows that for the deterministic mechanism to do strictly better than the stated ratio, the facility cannot be located at the reported locations; locating the facility to the left of x_1 or to the right of x_2 only increases the cost of the mechanism, so the only option left for a mechanism to do better is to locate the facility in the interior, i.e., in (x_1, x_2). □

Theorem 5.2. *Any strategyproof deterministic mechanism has an approximation ratio of at least $2^{1-\frac{1}{p}}$ for the L_p social cost function for any $p \geq 1$.*[38]

Proof. Using Lemma 5.3, we can now argue similarly to the case of $p = \infty$ (Theorem 3.2 in [98]).[39] Suppose $p > 1$ (the bound holds trivially for $p = 1$), and suppose a deterministic strategyproof mechanism yields an approximation ratio strictly better than $2^{1-\frac{1}{p}}$ for the L_p social cost. For the two-agent location profile $x_1 = 0, x_2 = 1$, Lemma 5.3 implies the facility is located at some $y \in (0, 1)$. Now consider the location profile $x_1 = 0, x_2 = y$. Again, by Lemma 5.3, the mechanism must locate the facility at $y' \in (0, y)$ to guarantee the improved approximation. But if agent 2 is located at $y < 1$, he can misreport his location as 1, forcing the mechanism to locate the facility at y, his true location; this violates strategyproofness. □

---

[37]Ex-post Pareto efficiency (as defined in section 4.2) requires the facility to be located in $[x_1, x_2]$; thus, this property is stronger.

[38]The lower bound of 2 on the approximation ratio holds when $p = \infty$, see [98].

[39]Another argument along this line can be found in the proof of Theorem 4.4 in [60].

5.4 Randomized Mechanisms

Recall that when the social cost is measured by the L_2 norm or the L_∞ norm, randomization provably improves the approximation ratio. In the former case, Feldman and Wilf [60] describe an algorithm whose approximation ratio is $(\sqrt{2}+1)/2$; for the latter, Procaccia and Tennenholtz [98] design an algorithm with an approximation ratio of $3/2$. The mechanisms in both cases are simple and somewhat similar, placing non-negative probabilities *only* on the optimal location and generalized medians (defined shortly), where these probabilities are independent of the reported location profile. In this section we show that this is not enough in general; namely, randomizing over generalized medians and the optimal location does not improve the approximation ratio of the median mechanism for *any* integer $p \in (2, \infty)$. For the case of 2 agents we show that the best approximation ratio is given by the LRM mechanism among all strategyproof mechanisms. Extending this analysis even to the case of 3 agents appears to be non-trivial.

5.4.1 Mixing Dictatorships and Generalized Medians with the Optimal Location

We begin with a definition of generalized medians.

Definition 5.2. *Let $\mathbf{x} \in \mathbb{R}^n$, $S \subseteq N$, and $m \in \{1, \ldots, |S|\}$. Let $S = \{s_1, \ldots, s_{|S|}\}$, where $x_{s_i} \leq x_{s_{i+1}}$. Then, the mth generalized median of subset S in location profile $\mathbf{x}$ is $x_{[m,S]} = x_{s_m}$.*[40] *If $S = N$, we allow for the shorthand $x_{[m]} = x_{[m,N]}$.*

Next, we define the class of mechanisms currently used in literature:

Definition 5.3. *Let f be a mechanism which satisfied the following. For every $n \in \mathbb{N}$, $S \subseteq N$, $m \in \{1, \ldots, |S|\}$, there exist non-negative numbers ${v^{S^n}}_m$, and v^n_{OPT} with $v^n_{OPT} + \sum_{S \subseteq N, m \in \{1,\ldots,|S|\}} {v^{S^n}}_m = 1$, such that for every profile $(x_1, x_2, \ldots, x_n)$, f locates the facility at OPT with probability v^n_{OPT} and at $x_{[m,S]}$ with probability ${v^{S^n}}_m$ (where OPT is the optimal location for the profile $(x_1, x_2, \ldots, x_n)$).*[41] *If f satisfies these properties, we say that f is a Mixed Generalized Medians Optimal (MGMO) mechanism.*

We now show that for integer $p > 2$, MGMO mechanisms cannot beat the median.

[40] That is, $x_{[m,S]}$ is the mth largest location among the locations of the agents in S, allowing for repetition.

[41] When a location appears more than once in OPT and $x_{[m,S]}$ for $S \subseteq N$ and $m \in \{1, \ldots, |S|\}$, the probabilities add up.

Theorem 5.3. *Let f be a strategyproof MGMO mechanism. Then, for any finite integer $p > 2$, the approximation ratio of f is at least $2^{1-\frac{1}{p}}$.*

Proof. Fix $n = 2k$, with $k \in \mathbb{N}$. In all profiles in our proof, the relative order of agents locations remains the same: specifically, $i < j$ implies $x_i \leq x_j$ for all of our profiles $\mathbf{x}$. For every $S \subseteq N$, and every $j \in S$ let $S(j)$ be the number of agents with index weakly smaller than j in S (for example, if $S = \{2, 4, 9\}$, then $S(2) = 1$, $S(4) = 2$, and $S(9) = 3$). On our profiles, the probability that the location of agent $j \in N$ is chosen as a generalized median therefore is $v_j^n = \sum_{S \subseteq N : j \in S} v^{Sn}_{S(j)}$.

For $j = 1, \ldots, k$, define the profile $\mathbf{x}^j$ as follows (where a_j is a parameter to be defined shortly): agents 1 through j are located at $-a_j$; agents $j+1$ through k are located at 0; agents $k+1$ through $2k-j+1$ are located at 1; and agents $2k-j+2$ through $2k$ are located at $1+a_j$ (note the *slight* asymmetry in the location of the agents: while k agents are at or below zero, and k agents are at or above 1, there is an additional agent at 1 compared to zero and so one less agent at $1+a_j$ compared to $-a_j$). Now, a_j is chosen to be the smallest positive root of the function $g_j(\alpha) = j\alpha^{p-1} - (k-j+1) - (j-1)(1+\alpha)^{p-1}$; such an a_j must exist by the intermediate value theorem, as $g_j(0) < 0$ and $g_j(\alpha)$ is a continuous function of α with $g_j(\alpha) \to \infty$ as $\alpha \to \infty$.

We show that the optimal mechanism locates the facility at zero for the profile $\mathbf{x}^j$, i.e., $OPT = 0$. Note that the social cost for this profile, when locating the facility at $z \in [0, 1]$, is $j(z + a_j)^p + (k - j)z^p + (k - j + 1)(1 - z)^p + (j - 1)(1 + a_j - z)^p$, and when $z \in (-a_j, 0)$ the social cost becomes $j(z + a_j)^p + (k - j)(-z)^p + (k - j + 1)(1 - z)^p + (j - 1)(1 + a_j - z)^p$. Note that the social cost function is differentiable for $z \in (0, 1)$ and for $z \in (-a_j, 0)$. The left and right derivatives at 0 are both $pja_j^{p-1} - p(k-j+1) - p(j-1)(1+a_j)^{p-1}$, and thus the social cost function is differentiable on $(-a_j, 1)$ with its derivative at $z = 0$ equal to zero (by our choice of a_j). The fact that this is a global minimum now follows from strict convexity of the social cost function $||\mathbf{x}^j - z(1, \ldots, 1)||_p$ (for all $z \in \mathbb{R}$). Thus, indeed, $OPT = 0$.

We now attempt to bound v_{OPT}. For each profile $\mathbf{x}^j$, consider the profile $\mathbf{x}'^j$ that differs only in the location of agent j: namely, $x'^j_j = 0$ instead of $-a_j$. Note that on this profile, $OPT = 0.5$ by symmetry. Strategyproofness implies that a deviation from profile $\mathbf{x}'^j$ to profile $\mathbf{x}^j$ should not be beneficial for agent j, namely $a_j v_j^n - \frac{1}{2} v_{OPT}^n \geq 0$ (where a_j is the increase in agent j's cost caused by that deviation when the facility is built in his reported location, and $\frac{1}{2}$ is the decrease in his cost caused by that deviation when the facility is located at OPT), which implies $v_j^n \geq \frac{v_{OPT}^n}{2a_j}$. Defining a_j

for $j = k+1, \ldots, 2k$ in a symmetric fashion, it follows that the same inequality holds for j in that range, and that $a_j = a_{2k-j+1}$. Summing those inequalities up, we get:

$$1 - v_{OPT}^n = \sum_{j=1}^{2k} v_j^n \geq \sum_{j=1}^{2k} \frac{v_{OPT}^n}{2a_j} = 2\sum_{j=1}^{k} \frac{v_{OPT}^n}{2a_j} = \sum_{j=1}^{k} \frac{v_{OPT}^n}{a_j}$$

$$v_{OPT}^n \leq \frac{1}{1 + \sum_{j=1}^{k} \frac{1}{a_j}}$$

Now, we claim it is enough to show that as $n \to \infty$ (or equivalently, as $k \to \infty$), $\sum_{j=1}^{k} \frac{1}{a_j} \to \infty$. The inequality then implies that $v_{OPT}^n \to 0$. Consider the profile which locates k agents at 0 and k agents at 1. The social cost of locating the facility at OPT on this profile is $\sqrt[p]{n}/2$, while the social cost of locating the facility at an agent's location is $\sqrt[p]{n}2^{-\frac{1}{p}}$; thus, the approximation ratio of f on this profile is $\frac{v_{OPT}^n \sqrt[p]{n}/2 + (1-v_{OPT}^n)\sqrt[p]{n}2^{-\frac{1}{p}}}{\sqrt[p]{n}/2} = 2^{1-\frac{1}{p}} - (2^{1-\frac{1}{p}} - 1)v_{OPT}^n$. Thus, as $n \to \infty$, the approximation ratio on these profiles approaches $2^{1-\frac{1}{p}}$, completing the proof.

We are left with the task of showing that $\lim_{k\to\infty} \sum_{j=1}^{k} \frac{1}{a_j} = \infty$. To do so, we first show that for $j \geq k^{\frac{1}{p-1}} + 1$, $2^{p-1}(j-1) > a_j$. Recall that a_j was defined as the smallest positive root of $g_j(\alpha)$, and that $g_j(0) < 0$. Thus, it is enough to show that for j in the appropriate range, $g_j(2^{p-1}(j-1)) > 0$. For notational convenience, we denote $Q = 2^{p-1}$.

$$\begin{aligned}
g_j(Q(j-1)) &= jQ^{p-1}(j-1)^{p-1} - (k-j+1) - (j-1)(1+Q(j-1))^{p-1} \\
&= Q^{p-1}(j-1)^{p-1} - k - (j-1)\sum_{i=1}^{p-2}\binom{p-1}{i}(Q(j-1))^{p-1-i} \\
&\geq Q^{p-1}(j-1)^{p-1} - (j-1)^{p-1} - (j-1)\sum_{i=1}^{p-2}\binom{p-1}{i}(Q(j-1))^{p-1-i} \\
&\geq Q^{p-1}(j-1)^{p-1} - (j-1)^{p-1} - (j-1)\sum_{i=1}^{p-2}\binom{p-1}{i}(Q(j-1))^{p-2} \\
&> Q^{p-1}(j-1)^{p-1} - (j-1)\sum_{i=1}^{p-1}\binom{p-1}{i}(Q(j-1))^{p-2} \\
&= Q^{p-1}(j-1)^{p-1} - (j-1)(Q(j-1))^{p-2}\sum_{i=1}^{p-1}\binom{p-1}{i} \\
&> Q^{p-1}(j-1)^{p-1} - (j-1)(Q(j-1))^{p-2}2^{p-1} = 0.
\end{aligned}$$

Now,

$$\lim_{k\to\infty}\sum_{j=1}^{k}\frac{1}{a_j} > \lim_{k\to\infty}\sum_{j=\lceil k^{\frac{1}{p-1}}+1\rceil}^{k}\frac{1}{2^{p-1}j}$$
$$= \frac{1}{2^{p-1}}\lim_{k\to\infty}\sum_{j=\lceil k^{\frac{1}{p-1}}+1\rceil}^{k}\frac{1}{j}$$
$$\geq \frac{1}{2^{p-1}}\lim_{k\to\infty}\int_{k^{\frac{1}{p-1}}+2}^{k}\frac{1}{t}dt$$
$$= \frac{1}{2^{p-1}}(\lim_{k\to\infty}\int_{k^{\frac{1}{p-1}}}^{k}\frac{1}{t}dt - \lim_{k\to\infty}\int_{k^{\frac{1}{p-1}}}^{k^{\frac{1}{p-1}}+2}\frac{1}{t}dt)$$
$$= \frac{1}{2^{p-1}}((\lim_{k\to\infty}(1-\frac{1}{p-1})\ln k) - 0) = \infty$$

which completes our proof. □

5.4.2 Optimality of the LRM Mechanism for 2 Agents

Procaccia and Tennenholtz [98] defined the mechanism Left-Right-Middle (LRM) as follows: place the facility with probability $\frac{1}{2}$ at OPT, and with probability $\frac{1}{4}$ at each of $x_{[1]}$ and $x_{[n]}$. They have shown that it is strategyproof, and that it provides a best-possible approximation ratio of $\frac{3}{2}$ when $p = \infty$. Our next result shows that the LRM mechanism provides the best possible approximation ratio among all shift and scale invariant (defined below) strategyproof mechanisms for the case of 2 agents for all L_p social cost functions for $p \geq 1$.

We begin with some definitions: we say that a mechanism f is *shift and scale invariant* if for every location profile $\mathbf{x} = (x_1, x_2)$ and every $c \in \mathbb{R}$, the following two properties are satisfied:[42]

1. Shift Invariance: the random variables $Y' \sim f(x_1 + c, x_2 + c)$ and $Y + c$ s.t. $Y \sim f(\mathbf{x})$ are equal in distribution.

[42] While these two properties are natural and reasonable to expect, it should be noted that they are not implied by strategyproofness- one example is the constant mechanism, which always locates the facility at the same point regardless of the reports. Requiring unanimity in addition to strategyproofness is also not sufficient to guarantee these properties; for example, the mechanism that runs LRM if $x_{[1]} = 0$, and otherwise locates the facility at $x_{[1]}$ and $x_{[2]}$ with probability 1/2 each, is easily seen to be strategyproof and unanimous but neither shift nor scale invariant.

2. Scale Invariance: the random variables $Y' \sim f(cx_1, cx_2)$ and cY s.t. $Y \sim f(\mathbf{x})$ are equal in distribution.[43]

A convenient notation for a given location profile $\mathbf{x}$ is to denote its midpoint as $m_{\mathbf{x}} = \frac{x_1+x_2}{2}$. We say that a mechanism f is *symmetric* if for any location profile $\mathbf{x}$ and for any $y \in \mathbb{R}$, $\mathbb{P}(f(\mathbf{x}) \geq m_{\mathbf{x}} + y) = \mathbb{P}(f(\mathbf{x}) \leq m_{\mathbf{x}} - y)$.

The structure of the proof is as follows. Our goal is to show that within the class of strategyproof, shift invariant and scale invariant mechanisms, we can further limit ourselves to symmetric mechanism that locate the facility always at the agents' locations or the midpoint; within this further restricted class, it becomes easy to prove that LRM is optimal. We achieve this goal gradually. First we show that we may restrict ourselves to symmetric (and anonymous) mechanisms. We then provide a characterization of strategyproofness for such mechanisms, and use it to show that we can further restrict ourselves to mechanisms which, for each profile $\mathbf{x}$, do not locate the facility both at $(\min\{x_1, x_2\}, \max\{x_1, x_2\})$ and at $(-\infty, \min\{x_1, x_2\}) \cup (\max\{x_1, x_2\}, \infty)$ with positive probability. We then show that we can restrict ourselves to mechanisms that locate the facility always at the agents' locations or the midpoint.

The following lemma allows us to focus on symmetric mechanisms.

Lemma 5.4. *Given any strategyproof, shift and scale invariant mechanism, there exists a symmetric, strategyproof, shift and scale invariant mechanism with the same worst-case approximation ratio.*

Proof. Given a mechanism f, we define the *mirror mechanism* of f, f_{mirror}, to be such that for every profile $\mathbf{x}$, we have that $\mathbb{P}(f_{mirror}(\mathbf{x}) \geq m_{\mathbf{x}} + b) = \mathbb{P}(f(\mathbf{x}) \leq m_{\mathbf{x}} - b)$ for all $b \in \mathbb{R}$.[44]

We will need the following notation: For each profile $\mathbf{x} = (x_1, x_2)$, let $Y_{x_1,x_2} \sim f(\mathbf{x})$, and $Y'_{x_1,x_2} \sim f_{mirror}(\mathbf{x})$. We claim that f_{mirror} is shift invariant, scale invariant and strategyproof (all of the equalities below are in distribution):

1. Shift invariance: let $c \in \mathbb{R}$. Then $Y'_{x_1+c,x_2+c} = 2m_{x_1+c,x_2+c} - Y_{x_1+c,x_2+c} = 2m_{\mathbf{x}} + 2c - Y_{x_1,x_2} - c = Y'_{x_1,x_2} + c$.

[43] It is possible to replace shift invariance with symmetry in our assumptions, and preserve our results; see Appendix B.

[44] Equivalently, the mirror mechanism can be thought of as follows: whenever f locates the facility at $y \in \mathbb{R}$ (that is, the single sampling of $f(\mathbf{x})$ yields y), f_{mirror} "mirrors" that location about $m_{\mathbf{x}}$, meaning it locates the facility at $2m_{\mathbf{x}} - y$.

2. Scale invariance: let $c \in \mathbb{R}$. Then $Y'_{cx_1,cx_2} = 2cm_{x_1,x_2} - Y_{cx_1,cx_2} = c(2m_{x_1,x_2} - Y_{x_1,x_2}) = cY'_{x_1,x_2}$.

3. Strategyproofness: assume f_{mirror} is not strategyproof, and assume without loss of generality that agent 2 has a profitable misreport: there exist profiles (w_1, w_2) and $(w_1, w_2 + \alpha)$ for some $\alpha \in \mathbb{R}$ such that $\mathbb{E}[|w_2 - Y'_{w_1,w_2}|] > \mathbb{E}[|w_2 - Y'_{w_1,w_2+\alpha}|]$. However, note that $w_2 - Y'_{w_1,w_2+\alpha} = -w_1 - \alpha + Y_{w_1,w_2+\alpha} = Y_{w_1-\alpha,w_2} - w_1$ (the second equality follows from shift invariance), and that $w_2 - Y'_{w_1,w_2} = Y_{w_1,w_2} - w_1$. Thus, it follows that $\mathbb{E}[|w_1 - Y_{w_1,w_2}|] > \mathbb{E}[|Y_{w_1-\alpha,w_2} - w_1|]$, violating strategyproofness for f. Thus f_{mirror} must be strategyproof.

Therefore, the mechanism g that picks f with probability 1/2 and f_{mirror} with probability 1/2 is a strategyproof mechanism that is also symmetric; g trivially satisfies shift and scale invariance. Finally, note that g has the same approximation ratio as f for all location profiles, since f_{mirror} has the same approximation ratio as f. □

Mechanisms which satisfy shift and scale invariance as well as symmetry also satisfy anonymity:

Lemma 5.5. *If a mechanism f is shift invariant, scale invariant and symmetric, it is also anonymous.*

Proof. Again, all equalities are in distribution. Let $\mathbf{x}$ be a location profile. We need to prove $Y_{x_1,x_2} = Y_{x_2,x_1}$. Shift and scale invariance gives $Y_{x_2,x_1} = -Y_{x_1,x_2} + x_1 + x_2$; thus, $\mathbb{P}(Y_{x_2,x_1} \leq b) = \mathbb{P}(x_1 + x_2 - b \leq Y_{x_1,x_2})$. But $\mathbb{P}(x_1 + x_2 - b \leq Y_{x_1,x_2}) = \mathbb{P}(Y_{x_1,x_2} \leq b)$ by symmetry about $m_{\mathbf{x}}$, thus $Y_{x_2,x_1} = Y_{x_1,x_2}$. □

The next lemma deals with an equivalent condition for strategyproofness for symmetric, shift and scale invariant mechanisms.

Lemma 5.6. *A symmetric, shift and scale invariant mechanism f is strategyproof if and only if for any profile $\mathbf{x} \in \mathbb{R}^2$ with $x_1 = 0 < x_2$, the following conditions hold:*

1. $-\int_{(-\infty,x_2)} y dF(y) + \int_{(x_2,\infty)} y dF(y) + x_2 \mathbb{P}(Y = x_2) \geq 0$

2. $\int_{(-\infty,x_2)} y dF(y) - \int_{(x_2,\infty)} y dF(y) + x_2 \mathbb{P}(Y = x_2) \geq 0$

where $Y \sim f(\mathbf{x})$ with c.d.f. F.

Proof. The proof is in Appendix B. □

Given a strategyproof, shift invariant, scale invariant and symmetric mechanism, the upcoming results demonstrate how to find another strategyproof, shift invariant, scale invariant and symmetric mechanism that restricts the probability assignment to x_1, x_2, and $m_{\mathbf{x}}$ for every profile $\mathbf{x}$ and simultaneously gives a weakly better approximation than the original mechanism.

Lemma 5.7. *Let f be a strategyproof, shift invariant, scale invariant and symmetric mechanism. There exists another strategyproof, shift invariant, scale invariant and symmetric mechanism g with a weakly smaller expected social cost on every profile, such that at least one of the following two properties holds:*

(1) For every two-agent profile $\mathbf{x}$, $\mathbb{P}(g(\mathbf{x}) \in (x_1, x_2)) = 0$ for every two-agent profile $\mathbf{x}$. (Doesn't utilize interior)[45]

(2) For every two-agent profile $\mathbf{x}$, $\mathbb{P}(g(\mathbf{x}) \in (-\infty, x_1) \cup (x_2, \infty)) = 0$ for every two-agent profile $\mathbf{x}$. (Ex-post Pareto efficiency)

Proof. The proof is in Appendix B. □

Lemma 5.8. *Let f be a strategyproof, shift invariant, scale invariant, symmetric mechanism. Assume that f is either ex-post Pareto efficient or doesn't utilize interior. Then there exists another strategyproof mechanism g with a weakly smaller expected social cost on every profile, such that $\mathbb{P}(g(\mathbf{x}) \in \{x_1, x_2, m_{\mathbf{x}}\}) = 1$ for every location profile $\mathbf{x}$. Furthermore, g satisfies shift invariance, scale invariance and symmetry.*

Proof. We break the proof into two cases.

1. Assume f is ex-post Pareto efficient. Let g be the mechanism that satisfies $\mathbb{P}(g(\mathbf{x}) = x_1) = \mathbb{P}(f(\mathbf{x}) = x_1)$, $\mathbb{P}(g(\mathbf{x}) = x_2) = \mathbb{P}(f(x) = x_2)$, $\mathbb{P}(g(\mathbf{x}) = m_{\mathbf{x}}) = 1 - \mathbb{P}(g(\mathbf{x}) = x_1) - \mathbb{P}(g(\mathbf{x}) = x_2)$. Note that since $m_{\mathbf{x}}$ minimizes the social cost function for the profile $\mathbf{x}$, g certainly provides a weakly better approximation ratio than f. Furthermore, symmetry, shift and scale invariance are preserved.

   Let us prove that condition 1 in Lemma 6 holds for g; the proof for condition 2 is similar. Since f is a strategyproof mechanism, the condition implies that for any profile $\mathbf{x} = (x_1, x_2)$ with

[45] Note that it is possible for such a mechanism to still be ex-post Pareto efficient, if $\mathbb{P}(g(\mathbf{x}) \in \{x_1, x_2\})$.

$x_1 = 0 < x_2$,

$$\begin{aligned} 0 &\leq -\int_{[0,x_2)} ydF(y) + x_2\mathbb{P}(f(\mathbf{x}) = x_2) \\ &= -\int_{(0,x_2)} ydF(y) + x_2\mathbb{P}(f(\mathbf{x}) = x_2) \\ &= -\mathbb{E}[f(\mathbf{x})\mathbb{1}\big(f(\mathbf{x}) \in (x_1, x_2)\big)] + x_2\mathbb{P}(f(\mathbf{x}) = x_2) \\ &= -m_{\mathbf{x}}\mathbb{P}(f(\mathbf{x}) \in (x_1, x_2)) + x_2\mathbb{P}(f(\mathbf{x}) = x_2) \\ &= -m_{\mathbf{x}}\mathbb{P}(g(\mathbf{x}) = m_{\mathbf{x}}) + x_2\mathbb{P}(g(\mathbf{x}) = x_2) \\ &= -\int_{(0,x_2)} ydG(y) + x_2\mathbb{P}(g(\mathbf{x}) = x_2). \end{aligned}$$

The third equality holds because the distribution is symmetric around $m_{\mathbf{x}}$. Hence, the condition is satisfied for the mechanism g.

2. Assume f doesn't utilize interior. Let g be the mechanism which, for every profile $\mathbf{x}$, locates $\mathbb{P}(g(\mathbf{x}) = x_1) = \mathbb{P}(g(\mathbf{x}) = x_2) = 0.5$, which is clearly strategyproof, shift invariant, scale invariant, and symmetric. $sc(\mathbf{x}, x_2)$ minimizes $sc(\mathbf{x}, y)$ among $y \geq x_2$ and $sc(\mathbf{x}, x_1)$ minimizes $sc(\mathbf{x}, y)$ among $y \leq x_1$. Hence, $\mathbb{E}[sc(\mathbf{x}, g(\mathbf{x}))] \leq \mathbb{E}[sc(\mathbf{x}, f(\mathbf{x}))]$.

□

Now we are ready to prove the main theorem.

Theorem 5.4. *The LRM mechanism gives the best approximation ratio among all strategyproof mechanisms that are shift invariant, scale invariant and ex-post Pareto efficient.*

Proof. By the previous lemma, it suffices to search among the class of strategyproof shift invariant, scale invariant and symmetric mechanisms where any element f of the class satisfies the property that $\mathbb{P}(f(\mathbf{x}) \in \{x_1, x_2, m_{\mathbf{x}}\}) = 1$ for every location profile $\mathbf{x}$. Clearly, for such mechanisms, the approximation ratio increases as $\mathbb{P}(f(\mathbf{x}) \in \{x_1, x_2\})$ increases. Assume $\mathbb{P}(f(\mathbf{x}) \in \{x_1, x_2\}) < 0.5$. Then $\mathbb{P}(f(\mathbf{x}) = m_{\mathbf{x}}) > 0.5$, and by symmetry, $\mathbb{P}(f(\mathbf{x}) = x_2) < 0.25$. But this gives, when $x_1 = 0$ and $x_2 > 0$, that $-m_{\mathbf{x}}\mathbb{P}(f(\mathbf{x}) = m_{\mathbf{x}}) + x_2\mathbb{P}(f(\mathbf{x}) = x_2) = -\frac{x_2}{2}\mathbb{P}(f(\mathbf{x}) = m_{\mathbf{x}}) + x_2\mathbb{P}(f(\mathbf{x}) = x_2) < 0$, violating strategyproofness by Lemma 6. Thus we must have that $\mathbb{P}(f(\mathbf{x}) \in \{x_1, x_2\}) \geq 0.5$, which implies that among all such mechanisms, LRM provides the best approximation ratio of $0.5(2^{1-\frac{1}{p}} + 1)$. □

An immediate consequence of Theorem 5.4 is the following corollary.

Corollary 5.1. *Any strategyproof shift and scale invariant mechanism has an approximation of at least* $0.5(2^{1-\frac{1}{p}} + 1)$ *in the worst case.*

5.5 Discussion

The most important open question in our view is whether or not randomization can help improve the worst-case approximation ratio for general L_p norm cost functions. The case of $p = 1$ is uninteresting because there is an optimal deterministic mechanism; for $p = 2$ and $p = \infty$ we already saw that randomization improves the worst-case approximation ratio, but we do not know if this is simply a happy coincidence, or if one can obtain similar results for all $p > 2$. Our negative result in Section 4 implies that any improvement by randomization would require a different approach than the existing mechanisms.

There are many other natural questions as well: for instance, what happens for more general topologies such as trees or cycles? Is it possible to characterize all randomized strategyproof mechanisms on specific topologies?

Finally, we believe it is of interest to consider more general cost functions for the individual agents. The properties established for LRM and many other randomized mechanisms depend on the assumption that agents incur costs that are *exactly equal* to the distance to access the facility. Clearly, this is a very restrictive assumption, and working with more general individual agent costs is an interesting direction to broaden the applicability of this class of models (see Appendix B.1 for a result regarding this direction).[46]

[46] For deterministic mechanisms, our result continues to hold for arbitrary single peaked cost functions, as long as the social cost remains an L_p measure of the distances.

Chapter 6

Approximation Algorithms for the Incremental Knapsack Problem

6.1 Introduction

Traditional optimization problems often deal with a setting where the input parameters are static. However, the static solution that we obtain from such a problem may be inadequate for a system whose parameters change over time. We consider one special case of this dynamic environment in which we have a maximization problem subject to certain capacity constraints. All of the inputs to the optimization problem are static except the capacities, which increase weakly over time. The goal is to find a sequence of compatible feasible solutions over time that maximizes a certain aggregate objective function. We will call such an optimization problem *an incremental optimization problem.* Unlike online and stochastic optimization problems, all input parameters are known with certainty from the outset.

In this chapter we consider the *incremental knapsack problem*, a special case of the incremental optimization problem. In the *discrete* incremental knapsack problem, we are given a knapsack whose capacity grows as a function of time. There is a time horizon of T periods and the capacity of the knapsack is B_t in period t for $t = 1, \ldots, T$. We are also given a set of n items to be placed in the knapsack. Item i has a weight $w_i > 0$ that is independent of the time period, and a value at time t of the form $v_i \Delta_t$ where $v_i > 0$ and $\Delta_t > 0$ (this particular functional form will allow us to model discounting). At any time period t, we require that the sum of the weights of the items in the knapsack cannot exceed the knapsack capacity B_t. Moreover, once an item is placed in the knapsack, it cannot be removed from the knapsack at a later time period. Finally, we are interested in maximizing the total discounted knapsack values[47] over time.

To put it formally, for any $X \subseteq S$, define $V(X)$ to be $\sum_{i \in X} v_i$ and $W(X)$ to be $\sum_{i \in X} w_i$. Then we are interested in finding a feasible solution $F = \{S_1, S_2, \ldots, S_T\}$ with $S_1 \subseteq S_2 \ldots \subseteq S_T \subseteq S$, where S_t represents the subset of items in the knapsack in period t, that maximizes the quantity $\sum_{t=1}^{T} V(S_t)\Delta_t$ subject to the constraints $W(S_t) \leq B_t$ for $t = 1, \ldots, T$. The special case where $\Delta_t = 1$ for all t will be called *time-invariant*. For brevity, in what follows we will denote the incremental knapsack problem as $\boldsymbol{\mathcal{DIK}}$, and its time-invariant version as $\boldsymbol{\mathcal{DIIK}}$.

One can also consider a continuous version of the problem. Here we assume that there is a continuous time parameter $s \in [0, S]$ for some $S > 0$. We are given a knapsack capacity function $B(s)$, weakly increasing in s, and a set K of n items to be placed in the knapsack. Item i has a value of v_i and a weight of w_i, both time-independent. At any time s, the sum of the weights of the items in the knapsack cannot exceed the knapsack capacity $B(s)$. Moreover, once an item is placed in the knapsack, it cannot be removed from the knapsack at a later time. We are interested in finding a feasible solution $F = \{K(s)\}_{s \in [0,S]}$ that maximizes the quantity $\int_1^S \Delta(s)V(K(s))ds$, where $V(K(s))$ is the total value of the items found in the knapsack at time s, under F and Δ is a discounting function. If one allows for any arbitrary capacity function B, then one can embed any instance of the discrete problem as a corresponding instance of the continuous problem with time horizon $S = T + 1$ by keeping the same item size and value while setting $B(s) = B_i$ for $i - 1 \leq s < i$ for $i = 1, \ldots, T$ (with $B_0 = 0$). Since the continuous version of the problem is more general, we will denote it as $\boldsymbol{\mathcal{IK}}$ and its time invariant case as $\boldsymbol{\mathcal{IIK}}$.

$\mathcal{IK}$ can be used to model an investment problem over a time horizon where the set of project costs and rewards are known in advance. We get additional funds in each time period and unused funds roll over from one period to the next. Consequently, $B(s)$ represents the budget at time s. Projects that we have invested in generate a reward (think of it as a benefit to society or to the investor) at each time point and we assume that the total reward is additive across projects and time (taking discounting into account). Like the standard knapsack problem, having a limited budget prevents us from investing in all of the projects. The additional complication lies in the fact that we may not

[47] By knapsack value at time t, we mean the sum of item values that are packed into the knapsack by time t.

have enough budget to invest in some projects (that generate high reward) initially. Hence, there is trade-off between investing in the affordable projects now versus saving budget in order to invest in more valuable projects later.

Since the single period knapsack problem is already known to be NP-hard, we look for polynomial time approximation algorithms for different special cases of $\mathcal{IK}$. For a maximization problem, a k-*approximation algorithm* (for some $k \leq 1$) is a polynomial time algorithm that guarantees, for all instances of the problem, a solution whose value is within k times the value of an optimal solution. Moreover, we say that the maximization problem has a (fully) polynomial time approximation scheme, or a PTAS (FPTAS respectively), if for every $0 \leq \epsilon < 1$, the algorithm guarantees, for all instances of the problem, a solution whose value is within $1 - \epsilon$ times the value of an optimal solution. Moreover, the algorithm should run in time that is polynomial in the size of the inputs and ϵ.

6.1.1 Scheduling Interpretation

The incremental knapsack problem can also be interpreted as a special case of a single machine scheduling problem with the objective $\max \sum_{i=1}^{n} v_i g(C_i)$, for some non-increasing function g. We treat the weight of an item as its processing time in the corresponding scheduling problem. Given a sequence σ for which the items are packed into the knapsack, let C_i denote the completion time of job i under σ. Let $B_\sigma^{-1}(C_i)$ be the first time t in which $B(t) \geq C_i$. Then item i contributes a reward $v_i \int_{B_\sigma^{-1}(C_i)}^{T} \Delta(s)ds$ under σ. Our objective is to identify a sequence σ that maximizes the quantity

$$\sum_{i=1}^{n} v_i \int_{B_\sigma^{-1}(C_i)}^{S} \Delta(s)ds.$$

Contrary to the well studied problem of minimizing $\sum_{j=1}^{n} v_j f(C_j)$, where f is an arbitrary non-decreasing function f, the maximization version of the problem has not been widely explored. Finally, exact algorithms that maximize $\sum_{i=1}^{n} v_i(T - B_\sigma^{-1}(C_i))$ also minimize $\sum_{i=1}^{n} v_i B_\sigma^{-1}(C_i)$. Nonetheless, this equivalence does not preserve approximation guarantees, and we are not aware of any way to utilize the existing literature (on the scheduling problem) to construct a good approximation algorithm for the incremental knapsack problem.

6.1.2 Related Work

The special case of $\mathcal{DIIK}$ where $v_i = w_i$ for all i has been examined in the literature. This problem is known as the *incremental subset sum problem*. Hartline [71] gave a 1/2-approximation algorithm for the incremental subset sum problem via dynamic programming. Sharp [113] gave a PTAS for the incremental subset sum problem for a fixed T. This algorithm uses a variant of the dynamic programming algorithm for the standard (i.e., 1-period) knapsack problem, and runs in time $O((\frac{Vn}{\epsilon})^T)$, where $V = \max_i\{v_i\}$. In Section 6.2, we will show that this problem is in fact strongly NP-hard when T is taken to be an input. Consequently, the classic result of [66] rules out an FPTAS for the incremental subset sum problem (and its generalizations) unless $P = NP$. Hartline [70] gave a $O(1/\log T)$-approximation algorithm for $\mathcal{DIIK}$.

A well-studied problem related to $\mathcal{DIK}$ is the *generalized assignment problem* (GAP). In the generalized assignment problem, we are given a set of m knapsacks and n items, with knapsack j having a capacity b_j. Further, placing item i in knapsack j consumes w_{ij} units of capacity of knapsack j, and generates a value of v_{ij}. Notice that a variant of $\mathcal{DIK}$ where one is only allowed to pack an additional $B_{t+1} - B_t$ units at each time t, is a special case of the generalized assignment problem: here, we would set $b_t = B_{t+1} - B_t$ and $w_{it} = w_i$ for all i and $v_{it} = v_i \sum_t^T \Delta_t$ for all i and t. However, $\mathcal{DIK}$ is not a special case of GAP because in $\mathcal{DIK}$ we are allowed to pack more than $B_{t+1} - B_t$ units at time t, assuming the knapsack has spare capacity from earlier time periods. Approximation algorithms for GAP have been studied by [114, 43, 56, 61], starting with the work of Shmoys and Tardos [114]. They presented an LP-based algorithm for the minimization version of the problem. Chekuri and Khanna [43] later observed that their algorithm can be modified into a 1/2-approximation algorithm for the maximization version of the problem. The authors also identified a few APX-hard special cases of generalized assignment. Fleischer et al. [61] gave an algorithm with approximation ratio of $1 - 1/e$. The best known constant factor algorithm is due to [56], who improved the approximation factor of [61] by a small ϵ. Chekuri and Khanna [43] presented a PTAS for the special case of GAP where item weights and values do not depend on the knapsack in which they are placed. Unfortunately, these results are not directly applicable to $\mathcal{DIK}$, because the knapsack capacities cannot be decomposed over time.

The objective of minimizing the sum of some function of the job-competition times on a single machine is a well studied problem in machine scheduling. A comprehensive literature review of the subject can be found in surveys such as [42, 74]. Two relevant works are the paper of Cheung and Shmoys [46], who gave a 2-approximation algorithm for the problem $1\|\sum f_j(C_j)$ for any non-negative, non-decreasing function f_j, and the work of Megow and Verschae [86], who developed a PTAS for the problem $1\|\sum v_j f(C_j)$, where f is any non-decreasing function. As we observed earlier, the scheduling interpretation of $\mathcal{IK}$ is a maximization version of sum of non-increasing function of completion times. Even though an exact algorithm for solving the problem $1\|\sum v_j f(C_j)$ can also solve a transformed version of $\mathcal{IK}$, the transformation does not preserve approximation guarantees. Hence, the existing algorithms cannot be applied directly to solve our problem. Recently, Gamzu and Segev [65] proposed a PTAS for the problem $1\|\max \sum v_j/C_j$, a special case of $\mathcal{IK}$ with $S=\infty$, $B(s)=s$ and $\Delta(s)=\frac{1}{s^2}$. Their approach does not immediately generalize to other important special cases of $\mathcal{IK}$ such as $\mathcal{DIK}$ because the capacity function in the discrete setting is a step function rather than a simple linear function. One notable difference is that the special case of $\mathcal{DIK}$ where item weight/processing time equals its value is strongly NP-hard (see Proposition 6.1), whereas the corresponding special case of $1\|\max \sum v_j/C_j$ can be solved via a simple index rule (see [65]).

6.1.3 Our Contributions

We give a $(\frac{1}{2}(1-\frac{1}{e})-O(\epsilon))$-approximation algorithm for the special case of $\mathcal{DIK}$ when the discount factors are weakly increasing with respect to time (which $\mathcal{DIIK}$ is a special case of). With the assumption on the discount factors, we only lose a factor of ϵ in the approximation even if we pack items in at most $\log_\epsilon T$ time periods. This enables us to enumerate all possible time sequences of when items are packed into the knapsack (there are $O(T^{1/\epsilon})$ such time sequences). For each time sequence, our algorithm makes use of a novel reduction to the general assignment problem.

Our second result provides a PTAS for the special case of $\mathcal{DIK}$ when the discount factors are non-decreasing with respect to time. This LP-based approximation scheme involves a disjunctive formulation (background and details, below) that can be rounded to obtain the desired approximation. Specifically, we construct a disjunction over $O(N((\log T/\epsilon)^{O(\log(\log T/\epsilon)/\epsilon^2)}))$ LPs (which is polynomial in T and n), each with nT variables and $O(nT)$ constraints. This improves on the result of [113]. This PTAS also extends the earlier work of [21] and [22] on the disjunctive approach for the single period knapsack problem.

Our second result relies on the classical approach of disjunctive programming [14]. Suppose we want to find an approximate solution to $\max\{w^T x : x \in P\}$ ($P \subseteq \mathbb{R}^n$, possibly non-convex), with approximation factor α. Moreover, suppose no good convex relaxation of P is known. In this case, we may still be able to leverage the idea of disjunctive programming to give us a good approximation guarantee. The idea is to find a set of polyhedra $Q^1, Q^2, \ldots Q^L$ in $\mathbb{R}^n$ such that $P \subseteq \cup_{i=1}^L Q^i$ and for each i we can compute, in polynomial time, $x^i \in P$ with $w^T x^i \geq \alpha \max\{w^T x : x \in Q^i\}$. Taking $x^\star = \mathrm{argmax}_i\{w^T x^i\}$ yields a factor α approximate solution to the original optimization problem. As stated, this approach simply constitutes a case of enumeration (polynomially-bounded if L is polynomial in n and T). Further, $w^T x^\star \geq \alpha \max\{w^T x : x \in \mathrm{conv}(\cup_i Q^i)\}$, and this last maximization problem can be formulated as a single linear program (polynomial-sized if L is), and, as will be the case below, we obtain an approximation algorithm based on rounding.

For the continuous incremental knapsack problem $\mathcal{IK}$, we focus on the special case where the knapsack capacity grows linearly with time, or $\mathcal{LIK}$. Moreover, we consider discounting functions that are *order inducing*, meaning that once we have decided the subset of items to pack into the knapsack by the time horizon S, an optimal order in which to pack the items can be computed via a simple index rule. We observe that common discounting functions such as $\Delta(s) = 1$, $\Delta(s) = e^{-rs}$, and $\Delta(s) = (1+r)^{-s}$ for some constant $r > 0$ are all order inducing discounting functions. We then show that $\mathcal{LIK}$ with any order inducing discounting function admits an FPTAS.[48] We also show that the continuous incremental subset sum problem with linear knapsack capacity and no discounting, denoted by $\mathcal{LIIS}$, can be solved easily via a simple greedy algorithm. Our result also implies that the special case of $\mathcal{LIK}$ with no discounting and a constant number of value to weight ratio item-classes can also be solved in polynomial time. Finally, we present an NP-hardness result for the piecewise linear incremental knapsack problem ($\mathcal{PLIK}$ for short) with two pieces.

The rest of the chapter is organized as follows. The first part of the chapter focuses on $\mathcal{DIK}$. We show that the problem is strongly NP-hard and demonstrate that some common approaches will not yield a good approximation algorithm. We then move on to our algorithmic results for various special cases of $\mathcal{DIK}$. We first discuss the constant factor algorithm via a reduction to GAP, followed by our PTAS via disjunctive programming. Finally, we consider $\mathcal{LIK}$, where we present an FPTAS and a linear time greedy algorithm for $\mathcal{LIIS}$. We end with the NP-hardness result for $\mathcal{PLIK}$.

---

[48] Since the linear functional form can be compactly specified, we require the running time of our FPTAS to be polynomial in n and $\log T$.

6.2 Hardness Results for $\mathcal{DIK}$

We first show that the incremental subset sum problem, a special case of $\mathcal{DIIK}$, is already strongly NP-hard. Further, we demonstrate that some common approach will not provide a good approximation ratio.

Proposition 6.1. *The incremental subset sum problem is strongly NP hard.*

Proof. We show that 3-partition can be reduced to incremental subset sum. In the 3-partition problem, we are given a set S of $3m$ integers $a_1, \ldots a_{3m}$, and we want to decide whether S can be partitioned into m triples that all have the same sum B, where $B = (1/m)\sum_{i=1}^{3m} a_i$. It is known that 3-partition is strongly NP-hard even if all the integers take values between $B/4$ and $B/2$. Given a set S of $3m$ integers $a_1, \ldots, a_{3m}$, let a_i be the weight and value of item i. Suppose the knapsack capacity is $B_t = tB$ in period t for $t = 1, \ldots, T = m$. Lastly, we ask whether there exists a packing that achieves an objective value of $BT(T+1)/2$. Notice that since every item's weight equals to its value, a feasible solution can achieve an objective value of $BT(T+1)/2$ if and only if it saturates the knapsack capacity in every time period. It is clear that if a 3-partition exists, then such a packing exists. Conversely, if such a packing exists and since the value of the items are strictly between $B/4$ and $B/2$, three additional items must be packed in every time period in order for the knapsack to be at full capacity. Hence, if such a packing exists, then a 3-partition exists. □

We now show that the optimal solution to a $\mathcal{DIIK}$ instance may not exhibit a "nested structure", that is, constructing a feasible solution to a T period problem from an optimal solution to its $T-1$ period subproblem may be very sub-optimal. To see this, suppose there are $n = T$ items. Items 1 through $T-1$ each have unit weight and value $1/T^2$, whereas item T has weight T and value T. The knapsack capacity is t in period t. It is easy to see that the optimal solution for time periods 1 through $T-1$ is to pack one additional unit weight item in every period, giving us an objective value of $O(1)$, whereas the optimal solution for time periods 1 through T is to wait till period T to pack the weight T item, giving us an objective value of T. An alternative idea is to solve first for an optimal packing S^* for the knapsack with capacity B_T and restrict ourselves to pack items from S^* in periods 1 through $T-1$. We show that this approach may be very sub-optimal as well. Consider T items. Items 1 through $T-1$ each have unit weight and value, whereas item T has weight T and value $T+\epsilon$. The knapsack capacity is again t in period t. The optimal solution for the knapsack with capacity T is to pack the weight T item, giving us an objective value of $T+\epsilon$. Nonetheless, doing so will not enable to us to pack anything in periods 1 through $T-1$, whereas the overall optimal solution is to pack one additional unit weight item each of the periods 1 through $T-1$, giving us an objective value of $O(T^2)$.

To contrast $\mathcal{DIK}$ with the standard knapsack problem further, recall that the LP relaxation of the standard knapsack problem has an integrality gap of two when every individual item can fit into the knapsack. This is because the greedy algorithm that packs items in value-to-weight ratio order yields a 1/2-approximate solution. The approximation result of the greedy algorithm extends when all items can fit into the knapsack initially.

Proposition 6.2. *If every item can fit into the knapsack initially, then the greedy algorithm is a 1/2-approximation algorithm for $\mathcal{DIK}$.*

The proof of Proposition 6.2 can be found in Appendix C.1.

Unfortunately, the story is more complicated when some items cannot fit into the knapsack initially. To illustrate the underlying difficulty, we consider a natural generalization of the knapsack IP here. Let $x_{i,t}=1$ if item i is placed in the knapsack at time t and 0 otherwise. In order to prevent an item those size is larger than the knapsack capacity B_t to be fractionally packed into the knapsack by an LP solution, we set $x_{it}=0$ if item i does not fit into the knapsack at time t.

$$\begin{aligned}
\text{IP} = \max \quad & \sum_{t=1}^{T} \Delta_t \sum_{i=1}^{n} v_i x_{i,t} \\
\text{s.t.} \quad & \sum_{i=1}^{n} w_i x_{i,t} \le B_t \quad \forall t \\
& x_{i,t-1} \le x_{i,t} \quad \forall i, \text{and } t = 2,3,\dots,T \\
& x_{i,t} = 0 \quad \text{for any } i,\ t \text{ such that } \ w_i > B_t \\
& x_{i,t} \in \{0,1\} \quad \forall i,t.
\end{aligned} \tag{6.1}$$

Proposition 6.3. *The LP relaxation of (6.1) has an unbounded integrality gap, even when there is no discounting and has item have the same value as its weight.*

Proof. Fix a $k \geq 2$ and let $T = n^k$. Consider a set of n items, where $v_i = w_i = k^i$ for $i = 1, \ldots, \log_k(T) = n$. The knapsack capacities follow the following pattern:

$$B_t = k^i \text{ if } T(1 - \frac{1}{k^{i-1}}) + 1 \leq t \leq T(1 - \frac{1}{k^i}) \text{ for } i = 1, \ldots, \log_k(T) \text{ and } B_T = B_{T-1}.$$

Since the LP can fractional pack the items, the knapsack capacity is saturated in every time period. Moreover, since all items have weight equal to value, the optimal value of the LP solution is the sum of the knapsack capacities over all time periods:

$$T + T \sum_{i=1}^{\log_k(T)} k^i T\Big((1 - 1/k^i) - (1 - 1/k^{i-1})\Big) = T(k-1)\log_k(T) + T = O(Tk\log_k(T)).$$

Let $t_i = T(1 - 1/k^{i-1}) + 1$ denote the first time when the knapsack capacity increases to k^i. Notice that any integer feasible solution to the IP only packs at times t_i. The only items that fit in the knapsack at time t_i are items 0 through i. If we decide to pack item i in period t_i, then the total revenue we get for packing i over times $t_i \leq t \leq t_{i+1} - 1$ is $T(1/k^i - 1/k^{i+1})k^{i+1} = T(k-1)$. Since we cannot pack any item before time t_i if we pack item i in time t_i, the total revenue we get up to time $t_{i+1} - 1$ would be $T(k-1)$. For every $i > 1$, this is clearly suboptimal since we would get more revenue up to time $t_{i+1} - 1$ had we just packed item 1 in period 1 (since $kT(1 - 1/k^{i+1}) > kT(1 - 1/k) = T(k-1)$ for $i > 1$). Hence, no integer optimal solution would pack item i at time t_i for every $i > 1$.

If we do not pack item i at time t_i, then the optimal packing of the knapsack in period i is items 1 through $i-1$ for every i. This is feasible as we can pack items j at time t_{j+1} for every $j = 1, \ldots, i-1$. Hence, this is an optimal integer packing the sub-problem over time periods 1 through t_i for every $i > 1$. We evaluate this integer optimal solution by looking at how long each item has been placed in the knapsack:

$$\begin{aligned} kT + \sum_{i=2}^{n-1} k^i(T - t_{i+1} + 1) &= kT + \sum_{i=2}^{n-1} k^i(T - T(1 - 1/k^i)) \simeq kT + T(\log_k T - 1) \\ &\leq 2T\max(k, \log_k(T)). \end{aligned}$$

Hence, the integrality gap is at least $0.5\min(\log_k(T), k)$. For every k, we can choose $T = k^k$ so that $0.5\min(\log_k(T), k) = k/2$. Letting k go to infinity and we have the desired result. □

This result implies that any constant factor approximation algorithm must do something more clever than simply solving the above LP relaxation and rounding the fractional solutions to a feasible integral solution. It also suggests that our LP relaxations needs to be tightened.

6.3 A Constant Factor Approximation Algorithm

In the introduction section, we drew a connection between $\mathcal{DIK}$ and the generalized assignment problem (GAP). We mentioned that a variant of $\mathcal{DIK}$ where unused capacity from the previous period cannot roll over to the next period is equivalent to a special case of GAP. Nonetheless, in the event that unused capacity does roll over from one period to the next, we need to balance the use of current capacity to pack more items versus saving additional capacity to pack more valuable items later. Suppose we know the exact capacity required for an optimal packing in each time period, then we can easily obtain a corresponding instance of GAP and make use one of the known approximation algorithms. The main ideas behind our approximation algorithm lies in

1. making a polynomial number of guesses on the set of time periods when a near optimal solution packs additional item(s).

2. identifying a relationship between the total capacity of items packed after the k-th packing step and the knapsack capacity during the $(k-1)$-st packing step that enables the reduction to a GAP instance.

These two observations enable us to come up with a good solution after making a polynomial number of calls to a constant factor approximation algorithm, such as that of [56], for some transformed instance of GAP.

We now discuss the first idea in detail. Fix $0 < \epsilon \leq 1$. Define t_j for $j = 1, \ldots, K$, where K is the largest value such that $\lceil (1+\epsilon)^K \rceil \leq T$ as follows. Note that $K = O(\frac{\log T}{\epsilon})$. The set of time periods that we will consider is

$$\mathcal{T}_\epsilon = \{t_0^\epsilon, t_1^\epsilon, \ldots, t_{K+1}^\epsilon\}, \tag{6.2}$$

where $t_0^\epsilon = T$, and $t_j^\epsilon = T - \lceil (1+\epsilon)^{j-1} \rceil$ for $j = 1, \ldots, K+1$. Now take any feasible solution x of an instance of $\mathcal{DIK}$, we will construct a corresponding feasible solution x_ϵ that only packs items in periods from $\mathcal{T}_\epsilon$ with a small loss in objective value, provided that the discounted factors are non-decreasing with respect to time. Whenever an item i is first packed in the knapsack in time period t by x, where $t_j < t \leq t_{j-1}$, then it will be first packed in period t_{j-1} by x_ϵ. Note that x_ϵ is a feasible solution as knapsack capacity is non-decreasing over time.

Proposition 6.4. *The objective value of x_ϵ is within a factor of $1/(1+\epsilon)$ of the optimal value if the discounting factors are non-decreasing over time.*

Proof. Suppose item i is packed by x in time $t_j < t \leq t_{j-1}$ for some j. Then the contribution of item i to the objective in x is $v_i \sum_{t'=t}^{T} \Delta_{t'}$ compared to $v_i \sum_{t'=t_{j-1}}^{T} \Delta_{t'}$ in x_ϵ. Whenever $\Delta_{t'}$ is non-decreasing,

$$\frac{\sum_{t'=t}^{T} \Delta_{t'}}{\sum_{t'=t_{j-1}}^{T} \Delta_{t'}} \leq \frac{\sum_{t'=t_j}^{T} \Delta_{t'}}{\sum_{t'=t_{j-1}}^{T} \Delta_{t'}} \leq 1+\epsilon.$$

Hence, the objective value of x_ϵ is at least $1/(1+\epsilon)$ fraction of the objective value of x. □

For all feasible solutions that only pack items in time periods from $\mathcal{T}_\epsilon$, we focus on the one that achieves the highest objective value x_ϵ^*. Let $s_1 < \ldots < s_k \in \mathcal{T}_\epsilon$ be consecutive times when at least one additional item is packed into the knapsack. Then:

Lemma 6.1. *For $1 < j \leq k$, the total size of items packed into the knapsack by x_ϵ^* up to period s_j is at least $B_{s_{j-1}}$.*

Proof. If this claim does not hold for some j, then the items that are packed into the knapsack in period s_j can all fit into the knapsack in period s_{j-1}, giving us higher objective value in doing so, which contradicts the optimality of x_ϵ^*. □

Now, let's divide the packing periods dictated by x_ϵ^* into two sets: those with even index I_e versus those with odd index I_o, i.e. $I_e = \{s_2, s_4, ..., s_{2\lfloor k/2 \rfloor}\}$ and $I_o = \{s_1, s_3, ..., s_{2\lceil k/2 \rceil - 1}\}$. Clearly, items packed into the knapsack by x_ϵ^* during one of the aforementioned set of times will achieve at least $1/2$ fraction of the objective value of x_ϵ^*. Finally, by Lemma 6.1, for every $s_j \in I_e$, the total size of items packed during period s_j by x_ϵ^* is upper bounded by $B_{s_j} - B_{s_{j-2}}$, as the total size of items packed up to period s_{j-1} is at least $B_{s_{j-2}}$. The same holds for every $s_j \in I_o$. Now we are ready to describe the algorithm:

1. Fix $0 < \epsilon < 1$. For every subset S of $\mathcal{T}_\epsilon$:
   (a) Index the time periods $s_1, \ldots, s_k$ of S in increasing order and divide them into I_o and I_e. Construct a GAP instance w.r.t the packing in time periods I_o (I_e respectively).

(b) For every period $s_j \in I_o$ and (I_e respectively), create a knapsack with capacity $B_{s_j} - B_{s_{j-2}}$ (with $B_0 = 0$). Items in the GAP instance have the same data as the incremental knapsack instance. If an item i is packed into this knapsack, then we receive a value of $v_i \sum_{t=s_j}^{T} \Delta_t$.

(c) Apply the algorithm of [56] on the two GAP instances and return the better of the two solutions.

2. Take the best solution over all subset S.

Note that since $\mathcal{T}_\epsilon$ has $O(\frac{\log T}{\epsilon})$ elements, the size of the enumeration is $O(T^{1/\epsilon})$. Since our guess contains the packing of x_ϵ^* with respect to both subsets of periods I_o and I_e, our algorithm is guaranteed to return a $\frac{1}{2}(1-\frac{1}{e})-O(\epsilon)$ approximate solution, where 1/2 comes from taking the better of the solutions produced by packing using I_o versus I_e, $1-1/e$ comes from the performance of the algorithm of Feige and Vondrák, and the $O(\epsilon)$ comes from using a subset of periods from $\mathcal{T}_\epsilon$ for packing.

6.3.1 Improving the Approximation Guarantee

There are two potential methods for improving the constant following our algorithmic approach. First is to obtain more accurate guesses of the size of additional items packed into the knapsack in each time period for the reduction to GAP. Working with time periods from $\mathcal{T}_\epsilon$ and observing that at most one item will straddle two or more consecutive time periods (i.e. its packing requires the saving of residual capacity over one or more periods), the number of such guesses can be polynomially bounded in T. However, to ensure that a feasible packing of additional items in a given time period translates to a feasible packing of the corresponding knapsack in the GAP instance, we may need to blow up the size of each knapsack in the GAP instance by a factor of $1+\epsilon$. The difficulty lies in arguing that the reverse mapping from a GAP feasible solution to the corresponding feasible packing of incremental knapsack incurs an objective loss of $O(\epsilon)$, since the increment in knapsack capacity is non-homogeneous across time and an ϵ fractional increment of a knapsack in an earlier time period may have a large effect compared to the same incremental in a later time period.

Another direction through which we can potentially improve the approximation ratio is to come up with a better approximation algorithm for the special case of GAP that we manage reduce to. Chekuri and Khanna [43] showed that even certain special cases of GAP are APX-hard, including a special case where all items have the same weight across knapsacks but different values. Nonetheless, the instance of GAP that enables them to perform the reduction requires the existence of items i and i' and knapsack k and k', where item i is more valuable when placed in knapsack k versus k' and vice versa for item i'. On the contrary, in the reduced GAP instance, an item will give us higher value when it is placed in a knapsack corresponding to an earlier time period. Chekuri and Khanna gave a PTAS for the special case of GAP where all items weight and values are identical across knapsacks. Whether the GAP instances that we obtained from $\mathcal{DIK}$ admits a PTAS remains an open question.

6.4 A PTAS for $\mathcal{DIIK}$

Now we are ready to present the PTAS for $\mathcal{DIIK}$. This algorithm can be extended to the case of $\mathcal{DIK}$ with monotonically nondecreasing Δ_t. Consider an instance of $\mathcal{DIIK}$ and let $\epsilon \in (0,1)$. Without loss of generality, we can assume that the v_i's are integral. Moreover, we will only pack in periods $\mathcal{T}_\epsilon = \{s_1, \ldots, s_{|\mathcal{T}_\epsilon|}\}$ and lose a factor of $O(\epsilon)$ in the process (see Proposition 6.4).[49] Let $T' = |\mathcal{T}_\epsilon|$. To ease the notation, from hereon we will simply refer to period s_t as period t for $t = 1, \ldots, T'$. Fixing an optimal solution OPT, and let h be a the maximum valued item that is ever placed in the knapsack by OPT. Then it suffices to optimize over the set of items $S^h = \{i \in S | v_i \leq v_h\}$. We partition S^h into $K+1$ subsets $X = \{S^{1,h}, S^{2,h}, \ldots, S^{K,h}, T^h\}$, where

$$S^{k,h} = \{j \in S, j \neq h: \ (1-\epsilon)^{k-1} v_h \geq v_j > (1-\epsilon)^k v_h\} \quad \text{for } k = 1, \ldots, K,$$

and

$$T^h = \{j \in S : (1-\epsilon)^K v_h \geq v_j\}.$$

In order to attain the approximation ratio, we will choose K large enough so that $(1-\epsilon)^K < \epsilon/T'$ or equivalently, $K > \frac{\log(T'/\epsilon)}{\epsilon}$.

Consider a modified instance of the problem where items have identical weights as the original instance and item i has a modified value of $v'_i = (1-\epsilon)^{k-1} v_h$ if $i \in S^{k,h}$ and $v'_i = v_i$ otherwise. Let OPT_m denote an optimal solution to the modified instance of the problem. Let $V(SOL)$ and $V_m(SOL)$ be the objective value with respect to a solution SOL of the original instance and the modified instance respectively. As we did not change the item weights, OPT_m is a feasible solution to

[49] $\mathcal{T}_\epsilon$ here is defined in the same way as in expression (6.2)

the original instance. Moreover,

$$V(OPT_m) \geq (1-\epsilon)V_m(OPT_m) \geq (1-\epsilon)V_m(OPT) \geq (1-\epsilon)V(OPT),$$

where the first inequality follows from the fact that $v_i \geq (1-\epsilon)v'_i$ for every item i, the second inequality follows from the fact that OPT_m is an optimal solution to the modified instance, and the third inequality follows from the fact that $v_i \leq v'_i$ for every item i.

Now, since all items within each $S^{k,h}$ have equal value in the modified instance, it is clear that conditioning on the number of items chosen by OPT within each $S^{k,h}$, OPT would choose the items within the same value class in the order of non-decreasing weight (breaking ties arbitrarily). Thus, it suffices to enumerate feasible solutions that can be described by a collection of vectors $\{\sigma^1, \ldots, \sigma^K\}$, where $\sigma_t^k \in \{0, 1, \ldots, |S^{k,h}|\}$ denotes the number of items chosen from $S^{k,h}$ in time period $t \in \mathcal{T}_\epsilon$, in order to find an optimal solution. Nonetheless, the number of potential solutions that we have to enumerate would be exponential in n if we attempt to enumerate all possible configurations of $\{\sigma^1, \ldots, \sigma^K\}$. Consequently, we will only explicitly enumerate σ_t^k taking values from $\{0, 1, \ldots, \min(\lceil 1/\epsilon \rceil, |S^{k,h}|)\}$. For σ_t^k taking values larger than $J = \lceil 1/\epsilon \rceil$, we will instead let the feasible region of an LP capture these feasible points and let the LP choose the optimal value for us and subsequently round this value down to an integer. Lastly, since we don't know the most valuable item h taken by OPT in the original instance of the problem, we will have to guess such an item by enumeration.

Our disjunctive procedure is as follows. First, we guess the most valuable item $h \in S$ packed by an optimal solution. Subsequently, we only consider choosing items from S^h and round the values of the items in S^h to obtain the modified instance of the problem. We will then focus on solving the modified instance of the problem. Let k_i, $i = 1, 2, \ldots, |S^{k,h}|$, be the i-th lightest weight item in $S^{k,h}$ (break ties arbitrarily). Let $x_{k_i,t}$ be the variable indicating whether item k_i is placed in the knapsack in time period t. Let $\sigma = \{\sigma^1, \ldots, \sigma^K\} \in \{0, \ldots, J\}^{T'K}$ and define the following polyhedron:

$$Q^{\sigma,h} = \{x \in [0,1]^{T'n} : x_{i,t} = 0 \quad \forall (i,t) \text{ s.t. } v_i > v_h \tag{6.3}$$

$$x_{h,T'} = 1 \tag{6.4}$$

$$x_{k_1,t} = x_{k_2,t} = \ldots = x_{k_{|S^{k,h}|},t} = 0 \quad \forall (k,t) \text{ s.t. } \sigma_t^k = 0 \tag{6.5}$$

$$x_{k_1,t} = x_{k_2,t} = \ldots = x_{k_{\sigma_t^k},t} = 1, \; x_{k_{\sigma_t^k+1},t} = \ldots = x_{k_{|S^{k,h}|},t} = 0 \tag{6.6}$$

$$\forall (k,t) \text{ s.t. } 1 \leq \sigma_t^k < J \text{ and } \sigma_t^k < |S^{k,h}| \tag{6.7}$$

$$x_{k_1,t} = x_{k_2,t} = \ldots = x_{k_{\sigma_t^k},t} = 1 \quad \forall (k,t) \text{ s.t. } \sigma_t^k = J \text{ and } \sigma_t^k < |S^{k,h}| \tag{6.8}$$

$$x_{k_1,t} = x_{k_2,t} = \ldots = x_{k_{|S^{k,h}|},t} = 1 \quad \forall (k,t) \text{ s.t. } \sigma_t^k \geq |S^{k,h}| \tag{6.9}$$

$$w_h x_{h,t} + \sum_{k=1}^{K} \sum_{i=1}^{|S^{k,h}|} w_{k_i} x_{k_i,t} + \sum_{i \in T^h} w_i x_{i,t} \leq B_t \quad \forall t \tag{6.10}$$

$$x_{k_i,t-1} \leq x_{k_i,t} \quad \forall (k,i), \text{and } t = 2,3,\ldots,T' \tag{6.11}$$

$$x_{i,t-1} \leq x_{i,t} \quad \forall i \in T^h, \text{and } t = 2,3,\ldots,T'\}. \tag{6.12}$$

Equation (6.3) ensures that all items more valuable than h are never packed in the knapsack. Equation (6.4) ensures that item h is packed into the knapsack at some point over the time horizon. Equations (6.5) - (6.9) encodes our guesses on how many items from each value class to pack at each time period. Inequality (6.10) is the knapsack capacity constraint and inequalities (6.11) and (6.12) are precedence constraints. Note that the optimal solution x^* to the $\mathcal{DIIK}$ IP (6.1) with T' and the time horizon and v'_i as the item values is contained in some Q^{σ^*,h^*}, where h^* is the most valuable item ever packed by x^* and σ^* denotes the number of items packed by x^* from each value class in each time period. Hence, as long as we can solve the LP $\{\max \sum_{t=1}^{T'} \Delta_t \sum_{i=1}^{n} v'_i x_{i,t} : \; x \in Q^{\sigma,h}\}$ for every (σ,h), round the optimal LP solution to a feasible integer solution with an objective loss bounded by $(1-\epsilon)$, and take the best solution over all values of (σ,h), we would obtain a $(1-O(\epsilon))$ for this special case of $\mathcal{DIK}$. Before diving into the details of how to convert an LP optimal solution to a feasible integer solution with small loss in objective value, we first upper bound the number of LPs that need to be solved.

Lemma 6.2. *There are a total of* $O\Big(N(1/\epsilon + T')^{O(\log(T'/\epsilon)/\epsilon^2)}\Big)$ *LPs in our disjunctive procedure.*

Proof. Let us first count the number of LPs for a fixed guess of h. For a fixed $k \in \{1, \ldots, K\}$, we have $\sigma_1^k \leq \sigma_2^k \leq \ldots \leq \sigma_{T'}^k$. If $\sigma_{T'}^k = m$, then there are at most $\binom{m+T'-1}{m}$ feasible T'-tuples $(\sigma_1^k, \sigma_2^k, \ldots, \sigma_{T'}^k)$ since the vector values are integers. Since $0 \leq m \leq J$, we have that $\binom{m+T'-1}{m} \leq (J + T')^J$. Consequently, there are at most $\sum_{m=1}^{J} \binom{m+T'-1}{m} \leq J(J+T')^J$ feasible T'-tuples $(\sigma_1^k, \sigma_2^k, \ldots, \sigma_{T'}^k)$. Thus, there are at most $(J(J+T')^J)^K = O\Big((1/\epsilon + T')^{O(\log(T'/\epsilon)/\epsilon^2)}\Big)$ in the disjunctive procedure for a fixed h, giving us $O\Big(N(1/\epsilon + T')^{O(\log(T'/\epsilon)/\epsilon^2)}\Big)$ LPs in total. $\square$

Since $T' = O(\frac{\log T}{\epsilon})$, we have $O\Big(N(1/\epsilon + T')^{O(\log(T'/\epsilon)/\epsilon^2)}\Big) = O(N(\frac{\log T}{\epsilon})^{O(\log(\log T/\epsilon^2)/\epsilon^2)})$, which is polynomial in T and n.

Now, we are ready to present our rounding procedure.

Theorem 6.1. *For every non-empty polyhedron $Q^{\sigma,h}$, there exists a polynomially computable point $x^{\sigma,h}$ feasible for $\mathcal{IIK}$, such that*

$$\sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v_i' x_{i,t}^{\sigma,h} \geq (1-\epsilon) \max\{\sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v_i' x_{i,t} : \ x \in Q^{\sigma,h}\}.$$

Proof. Let $\bar{x}$ be an optimal solution of $\max\{\sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v_i' x_{i,t} : \ x \in Q^{\sigma,h}\}$. We will decompose the objective into value classes and show that our rounding procedure gives a small loss in objective for each value class. More precisely, we will show the validity of the inequality

$$\sum_{t=1}^{T'} \Delta_t \sum_{i=1}^{|S^{k,h}|} v_{k_i}' x_{k_i,t}^{\sigma,h} \geq (1-\epsilon) \sum_{t=1}^{T'} \Delta_t \sum_{i=1}^{|S^{k,h}|} v_{k_i}' \bar{x}_{k_i,t}, \tag{6.13}$$

for every $S^{k,h}$ and that of

$$\sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' x_{i,t}^{\sigma,h} \geq \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' \bar{x}_{i,t} - \Delta_{T'} \epsilon v_h, \tag{6.14}$$

in Lemma C.2.

Finally $x_{h,t}^{\sigma,h} = \bar{x}_{h,t}$ for every t. The two inequalities imply:

$$\begin{aligned}
\sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v_i' x_{i,t}^{\sigma,h} &= \sum_{t=1}^{T'} \Delta_t \bar{x}_{h,t} + \sum_{t=1}^{T'} \Delta_t \sum_{k=1}^{K} \sum_{i=1}^{|S^{k,h}|} v_{k_i}' x_{k_i,t}^{\sigma,h} + \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' x_{i,t}^{\sigma,h} \\
&\geq \sum_{t=1}^{T'} \Delta_t \bar{x}_{h,t} + (1-\epsilon) \sum_{t=1}^{T'} \Delta_t \sum_{k=1}^{K} \sum_{i=1}^{|S^{k,h}|} v_{k_i}' \bar{x}_{k_i,t} + \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' \bar{x}_{i,t} - \Delta_{T'} \epsilon v_h \\
&\geq (1-\epsilon) \sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v_i' \bar{x}_{i,t}.
\end{aligned}$$

Recall that $\bar{x}_{h,T} = 1$. Proofs of Lemma **??** and **??** can be found in the Appendix C.2. $\square$

Putting everything together, we have our approximation theorem.

Theorem 6.2. *Let* $y \in \arg\max_{Q^{\sigma,h} \neq \emptyset} \sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v'_i x^{\sigma,h}_{i,t}$, *where* $x^{\sigma,h}$ *is a feasible point for* $I\,I\,K$, *then we have that*

$$\sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v'_i y_{i,t} \geq (1-\epsilon) V_m(OPT_m) \geq (1-\epsilon)^2 V(OPT).$$

6.4.1 Discussion

Proposition 6.4 buys us a lot of leverage in both of the constant factor algorithm and the PTAS that we presented, as it allows us to consider and enumerate through just logarithmically many time periods and only lose an approximation factor of ϵ in the process. Nonetheless, when the discounting factors are decreasing with respect to time, then the proposition does not hold, as one may need to keep a constant fraction of the T time periods in order to ensure a $1 - O(\epsilon)$-approximate solution. New algorithmic ideas that balance two competing forces are needed: items packed in later time periods may not contribute much to the objective when discounting factors decrease with time, but there could be large increases in knapsack capacities during later time periods, which enables us to pack very valuable items. Finding an algorithm that results in a good approximation ratio for this case is an intriguing open problem.

6.5 Continuous Knapsack with Linear Capacity

In the remainder of this chapter, we will consider the incremental knapsack problem when the knapsack capacity grows continuously with time, or $\mathcal{IK}$ for short. It suffices to think of a solution of an instance of $\mathcal{IK}$ as an ordering of the items $\sigma_1, \ldots, \sigma_n$. Once the ordering is given, then we will pack item σ_i at time $t_{\sigma_i} = \inf\{t \mid \sum_{j \leq i} w_{\sigma_j} \leq B(t)\}$, i.e. the earliest time that the item can fit into the knapsack, given the order. Item σ_i is not packed into the knapsack by the end of the time horizon if $\sum_{j \leq i} w_{\sigma_j} > B(T)$.

We start with $\mathcal{LIK}$, the case where the capacity of the knapsack grows linearly with time, i.e. $B(t) = ct$ for some constant $c > 0$. Moreover, we will consider *order inducing* discounting functions where it suffices to decide which maximal subset of the n items to pack by time T. We say that S is a maximal subset to pack by time T if $\sum_{i \in S} w_i \leq T$ and $\sum_{i \in S \cup j} w_i > T$ for any $j \in \mathcal{N} \backslash S$. It is clear that any optimal packing will pack a maximal subset of items by time T. Once the subset is decided, then the optimal ordering of the items can be decided via a simple index rule. We will show that the following common discounting functions $\Delta(s) = 1$, $\Delta(s) = e^{-rs}$, and $\Delta(s) = (1+r)^{-s}$ for some value $r > 0$ are all order inducing discounting functions.

Proposition 6.5. *Let r_i be an index asssociated with job i. Given any maximal subset S to pack by time T, then any optimal packs the items in decreasing order of their indices, where*

- $r_i = \frac{v_{\sigma_i}}{w_{\sigma_i}}$ *if* $\Delta(s) = 1$;
- $r_i = \frac{v_{\sigma_i} e^{-rw_{\sigma_i}}}{1-e^{-rw_{\sigma_i}}}$ *if* $\Delta(s) = e^{-rs}$, *for any* $r > 0$;
- $r_i = \frac{v_{\sigma_i}(1+r)^{-w_{\sigma_i}}}{1-(1+r)^{-w_{\sigma_i}}}$ *for any* $r > 0$.

Proof of the proposition can be found in Appendix C.3.

Hence, given an order inducing discounting function, the problem boils down to choosing an optimal subset $S^\star$ of items to pack by time T. This problem can be formulated as the following math program. We will assume from here on that the items are ordered according to the index rule.

$$\begin{aligned} \max_x \quad & \sum_{i=1}^{n} \Big(\int_{\frac{1}{c}\sum_{j=1}^{i} w_j x_j}^{T} \Delta(s) ds \Big) v_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_i \leq cT \\ & x_i \in \{0,1\} \quad \forall i = 1, \ldots, n. \end{aligned} \tag{6.15}$$

Here $x_i = 1$ if and only if i belongs to the subset chosen. Moreover, we will assume without lost of generality from here on that $c = 1$, as we can scale the item weights appropriately.

Now we develop an FPTAS for solving the math program in (6.15) using an idea inspired by Chapter 5 of [110]. Let S_k denote a set of vectors in 2 dimensions, where each vector v corresponds to a feasible solution to the subproblem of (6.15) where one is only allowed to pack items 1 through k, for $k = 1, \ldots, N$. In particular, v stores the objective value and w stores the sum of the weights of the feasible solution that it corresponds to. Notice that one can enumerate through the elements of S_k via the following recursion.

1. For $k = 1$, we have that $S_1 = \{(0,0), (v_1 \int_{w_1}^{T} \Delta(s)ds, w_1)\}$ if $w_1 \leq T$, else $S_1 = \{(0,0)\}$.
2. For $k = 2, \ldots, n$, we construct S_k from S_{k-1} as follows: for every (v, w) in S_{k-1}, include (v, w) in S_k. Also include $(v + v_k \int_{w+w_k}^{T} \Delta(s)ds,\ w + w_k)$ in S_k if $w + w_k \leq T$.

Upon scaling, for every k, the vectors in S_k (except $(0,0)$) lies in a rectangle $[1, VT] \times [1, W]$ in $\mathbb{R}^2$, where $V = \sum_i v_i$ and $W = \sum_i w_i$. Given a $\epsilon > 0$, set $\delta = 1 + \frac{\epsilon}{2n}$. Define the following set of rectangles that covers $[1, VT] \times [1, W]$

$$R = \{[\delta^{i-1}, \delta^i] \times [\delta^{j-1}, \delta^j] | i = 1, \ldots, L_1, j = 1, \ldots, L_2\},$$

where $L_1 = \lceil \ln(TV)/\ln(\delta) \rceil \leq \lceil \frac{2n}{\epsilon}(\ln T + \ln V) \rceil$ and $L_2 = \lceil \ln(W)/\ln(\delta) \rceil \leq \lceil \frac{2n}{\epsilon}(\ln W) \rceil$. Hence, the number of rectangles is bounded by $O(\frac{n^2}{\epsilon^2}(\log T + \log V)(\log W))$, which is polynomial in the size of the inputs.[50] To get an FPTAS, we would like to define S'_k for $k = 1, \ldots, n$ recursively as we did for S_k such that

1. $S'_k \subseteq S_k$ for all $k = 1, \ldots, n$.
2. For every k fixed, every rectangle in R contains at most one element of S'_k.
3. For every k and every element $(v, w) \in S_k$, there exists an element $(v', w') \in S'_k$ that is "close" to (v, w). The precise notion of "closeness" will be specified later.

S'_k is constructed as follows.

1. For $k = 1$, set $S'_1 = S_1$. For every rectangle containing two or more elements of S'_1, we keep the one with the smallest w coordinate and delete the rest from S'_1.
2. For $k = 2, \ldots, n$, we construct S'_k from S'_{k-1} as follows: for every (v, w) in S'_{k-1}, include (v, w) in S'_k. Also include $(v + v_k \int_{w+w_k}^{T} \Delta(s)ds,\ w + w_k)$ in S'_k if $w + w_k \leq T$. For every rectangle containing two or more elements of S'_k, we keep the one with the smallest w coordinate and delete the rest from S'_k.

[50] Note that since we are no longer explicitly specifying the capacity of the knapsack in every time period in the continuous case, a polynomial time algorithm needs to have a running time that is polynomial in $\log T$.

This construction of S'_k satisfies criteria 1 and 2 above by definition. Moreover, the construction can be done in $O(nL_1L_2)$ time, which is polynomial in the size of the inputs. The following lemma specifies what we meant by the points in S'_k being "close to" the points in S_k.

Lemma 6.3. *For every k and for every vector $(v, w) \in S_k$, there exists a vector $(v', w') \in S'_k$ such that*

$$\delta^k v' \geq v \quad \text{and} \quad w' \leq w$$

Proof. We prove this via induction on k. The base case $k = 1$ holds by the definition of S'_1. Now for the inductive step. Take any $(v, w) \in S_k$, if $(v, w) \in S_{k-1}$, then we are done by the inductive hypothesis. Otherwise, by definition, there exists $(\hat{v}, \hat{w}) \in S_{k-1}$ such that $(v, w) = (\hat{v} + v_k \int_{\hat{w}+w_k}^{T} \Delta(s)ds,\ \hat{w} + w_k)$. Moreover, by the inductive hypothesis, there exists $(\tilde{v}, \tilde{w}) \in S'_{k-1}$ such that $\delta^{k-1}\tilde{v} \geq \hat{v}$ and $\tilde{w} \leq \hat{w}$. Finally, let $(v', w') \in S'_k$ be the point that lies in the same rectangle as $(\tilde{v} + v_k \int_{\tilde{w}+w_k}^{T} \Delta(s)ds,\ \tilde{w} + w_k)$ (which is guaranteed to exist by the way S'_k is constructed). Then we have that

$$\begin{aligned}\delta^k v' &\geq \delta^{k-1}\big(\tilde{v} + v_k \int_{\tilde{w}+w_k}^{T} \Delta(s)ds\big) \\ &\geq \hat{v} + v_k \int_{\hat{w}+w_k}^{T} \Delta(s)ds = v.\end{aligned}$$

The first inequality follows from the fact that $\delta v' \geq \tilde{v} + v_k \int_{\tilde{w}+w_k}^{T} \Delta(s)ds$ because the two vectors lie in the same rectangle. The second inequality follows from the fact that $\delta^{k-1}\tilde{v} \geq \hat{v}$ and that $\tilde{w} \leq \hat{w}$. Moreover, we have that

$$w' \leq \tilde{w} + w_k \leq \hat{w} + w_k = w.$$

□

Now, take the vector $(v^\star, w^\star) \in S_n$ corresponding an optimal solution of (6.15), then there exists a solution $(v', w') \in S'_n$ such that $(1+\epsilon)v' \geq \delta^n v' \geq v^\star$, which is the optimal objective value. Hence, if we search through S'_n for the vector with the highest v component, then we would obtain an $1 - O(\epsilon)$ approximate solution to the optimal solution of (6.15), as desired. The running time of the algorithm is $O(nL_1L_2)$ as we need to compute S'_1 through S'_k for $k = 1, \ldots, n$ and the computation of S'_k from S'_{k-1} takes $O(L_1L_2)$ time. Since $L_1L_2 = O(\frac{n^2}{\epsilon^2}(\log T + \log V)(\log W))$, our algorithm runs in polynomial of the size of the inputs and $1/\epsilon$. Hence, we have a bona fide FPTAS. The existence of an FPTAS also implies that the problem is at most weakly NP hard.

Theorem 6.3. *There exists a FPTAS for $\mathcal{LIK}$ with order inducing discounting functions.*

6.5.1 Incremental Subset Sum with Linear Capacity

Now we consider a special case of incremental knapsack with a linear capacity growth function, no discounting, and where the weight of each item equals to its value, which we denote by $\mathcal{LIIS}$ for short. We will show that the problem can be solved in linear time via a greedy algorithm. This result contrasts that of the discrete incremental subset sum problem, which is shown to be stronly NP-hard in Proposition 6.1 (even the standard subset sum problem is weakly NP-hard). Let $B(t) = ct$ being the capacity function of the knapsack for some constant $c > 0$. Since the discounting function $\Delta(s) = 1$ is order inducing, once we have decided on a subset S of items to pack into the knapsack on $[0, T]$. The optimal ordering is to pack items in non-increasing order of value-to-weight ratio. However, since every item's weight equals its value, every ordering of the items in S yield the same revenue. If we compare the loss in revenue of this solution to that of the LP relaxation of the problem, where items can be fractionally packed and thus the knapsack is full in every time period, then we incur a *loss in revenue* of

$$\sum_{i \in S} \frac{w_i^2}{2c} + \frac{1}{2}(T - \frac{1}{c}\sum_{i \in S} w_i)(cT - \sum_{i \in S} w_i)$$

compared to the total revenue $\frac{cT^2}{2}$ of an optimal fractional packing. From here on we will assume that $c = 1$ by scaling the item weights appropriately.

Now we are ready for the main theorem.

Theorem 6.4. *Sorting the items in non-decreasing order of item weights and greedily packing them until packing any more item would exceed the knapsack capacity of T is optimal for $\mathcal{LIIS}$.*

Proof. Let $S^\star$ and S_g denote an optimal set of items to pack and the set of items packed by greedy respectively. Let $c(S)$ denote the revenue loss of S. We would like to show that $c(S_g) \leq c(S^\star)$. We know that

$$\sum_{i \in S^\star \cap S_g} w_i^2 = \sum_{i \in S^\star \cap S_g} w_i^2.$$

Hence, it suffices to show that

$$\sum_{i \in S_g \setminus S^\star} w_i^2 + (T - \sum_{i \in S_g} w_i)^2 \leq \sum_{i \in S^\star \setminus S_g} w_i^2 + (T - \sum_{i \in S^\star} w_i)^2. \tag{6.16}$$

If $S^\star \backslash S_g = \emptyset$, then we are done, since greedy packs everything OPT packs and potentially more. Otherwise, note that for every $k \in S^\star \backslash S_g$, we have that

$$w_k \geq \max_{i \in S_g \backslash S^\star} w_i,$$

otherwise greedy would have packed item k before an item in $\arg\max_{i \in S_g \backslash S^\star} w_i$. Moreover, we also have that for every $k \in S^\star \backslash S_g$,

$$w_k > T - \sum_{i \in S_g} w_i,$$

because if it weren't the case, then item k would have been packed by greedy. Now consider the following lemma.

Lemma 6.4. *Let $a_1, \ldots a_p$ and $b_1, \ldots, b_q$ be nonnegative numbers such that $\sum_{i=1}^p a_i = \sum_{i=1}^q b_i$. Moreover, if $p \geq 2$, suppose also that every $k = 1, \ldots, p-1$, $a_k \geq \max_{i=1,\ldots,q} b_i$. Then*

$$\sum_{i=1}^p a_i^2 \geq \sum_{i=1}^q b_i^2.$$

Now suppose the lemma holds, then letting $p = |S^\star \backslash S_g| + 1$, $q = |S_g \backslash S^\star| + 1$, $\{a_i\}_{i=1}^p = (S^\star \backslash S_g) \cup \{T - \sum_{i \in S^\star} w_i\}$ with $a_p = T - \sum_{i \in S^\star} w_i$, and $\{b_i\}_{i=1}^q = (S_g \backslash S^\star) \cup \{T - \sum_{i \in S_g} w_i\}$. Then one can check that $\{a_i\}_{i=1}^p$ and $\{b_i\}_{i=1}^q$ satisfies the conditions of Lemma 6.4. In particular, note that

$$\sum_{i=1}^p a_i = \sum_{i=1}^q b_i = T - \sum_{i \in S^\star \cap S_g} w_i.$$

Hence, equation (6.16) follows directly from an application of Lemma 6.4. The proof of Lemma 6.4 can be found in Appendix C.4. □

Note that Theorem 6.4 does not immediately generalize to other discounting functions because the cumulative loss in revenue while we are waiting to save up enough capacity to pack item i does not just depend on its weight but also the time at which we pack the item.

Corollary 6.1. *Any incremental knapsack instance with linear capacity function and no discounting in which there are at most k value-to-weight ratio classes can be solved exactly in $O(n^k)$ time.*

Proof. We first guess the number of items from each of the value-to-weight ratio classes packed by the optimal solution. Then we pack items in non-increasing order of the ratio and within each ratio class, we pack the items with the smallest weight first. □

6.5.2 Discussion

Whether there exists a polynomial time algorithm for $\mathcal{LIK}$ with no discounting remains an open question. Theorem 6.3 gives an FPTAS for solving a slight generalization of the problem (for any *order inducing* discounting function). Nonetheless, we are unable to derive an NP-hardness result to complement our FPTAS. We conjecture that some reduction from the partition problem or its variants exists. By Corollary 6.1, such a reduction (if one exists) will require us to construct an instance of the incremental knapsack problem where there are a non-constant number of value-to-weight ratio classes. The ratio classes should be constructed in a delicate fashion so that neither a non-decreasing weight ordering greedy solution dominates (which would happen if the ratios are nearly identical) nor does a non-increasing value-to-weight ratio ordering solution (which would happen if the ratios are far apart from each other) dominate.

6.6 Piecewise Linear Capacity function

We say that the capacity function is *piecewise linear* if the time horizon can be partitioned into p subintervals such that the capacity function is linear within each subinterval. We first show that $\mathcal{IK}$ is NP-hard when we have a monotone piecewise capacity function with two pieces. The rough intuition behind the hardness reduction is that the problem is very similar to the standard knapsack problem if the slope of the first linear piece is large while the slope of the second linear piece is close to zero.

Theorem 6.5. *$\mathcal{PLIK}$ with a two linear pieces and no discounting is NP-hard.*

Proof. The reduction is from partition. Given a set of n positive integers $\{a_1, \ldots, a_n\}$, let $a = \frac{1}{2}\sum_{i=1}^{n} a_i$, the partition problem asks if there exists a subset that sums to a. Given an instance of the partition problem $\{a_1, \ldots, a_n\}$, we construct an instance of incremental knapsack as follows. There are n items with $v_i = w_i = a_i$. The capacity function is the following 2-piecewise linear function: $B(t) = t$ for $0 \leq t \leq a$, and $B(t) = a$ for $T \geq t \geq a$. There is a time horizon T to be specified.

Suppose there exists a partition S. Then we can pack the items in S into the knapsack by time a. Moreover, the order in which the items are packed will not change the objective value since item weight equals to item value. Hence, let us assume that the items in S are packed in increasing order of their subscript. Consequently, we get an objective value of

$$\sum_{i \in S}(T - \sum_{j \in S \mid j \leq i} a_j)a_i \geq \sum_{i \in S}(T - a)a_i = a(T - a).$$

Hence, if there exists a partition S. Then the optimal value of the correspondingly incremental knapsack instance is at least $a(T - a)$. On the other hand, suppose there does not exist a partition. Let $S^\star$ be an optimal subset to pack by time T in the corresponding incremental knapsack instance. Then, assuming that we pack the items in increasing order of subscript, we have that

$$\sum_{i \in S^\star}(T - \sum_{j \in S^\star \mid j \leq i} a_j)a_i = T \sum_{i \in S^\star} a_i - \sum_{i \in S^\star} \sum_{j \in S^\star \mid j \leq i} a_i a_j \leq T(a - 1),$$

since $\sum_{i \in S^\star} a_i < a$ and a_i's are integers. We choose T so that $a(T - a) > T(a - 1)$. One candidate would be $T = a^2 + 1$. Consequently, we have a valid polynomial size reduction: given any set of n positive integers $\{a_1, \ldots, a_n\}$, there exists a partition if and only if one can attain an objective value of at least $a(T - a) = a(a^2 - a + 1)$ in the corresponding incremental knapsack problem. □

6.6.1 Discussion

As is the case for linear capacity growth, it is clear that once an ordering for the items to be packed by time T is determined, one would pack the items as early as possible with respect to that ordering. However, it is no longer the case that once a subset to pack by time T is chosen, the optimal packing follows the non-increasing value-to-weight ratio. Nonetheless, it remains true via Proposition 6.5 that the items are packed in decreasing value-to-weight ratio ordering within each linear piece.

It seems plausible to extend the FPTAS for $\mathcal{LIK}$ to obtain an FPTAS for $\mathcal{PLIK}$ with a constant number of linear pieces, if the discounting function is order inducing. This is because, with these pseudo-polynomial number of guesses on the time at which the last (straddling) item is packed into each linear capacity segment, the problem essentially decouples into an instance of p knapsacks each with linear capacity and an interval during which items can be packed into the knapsack. Since items packed within each knapsack will be arranged in value-to-weight ratio order, the problem again boils down to guessing the subset that will be completely packed within each knapsack. Nonetheless, Lemma 6.3 does not easily generalize for this case. Generalizing Lemma 6.3 and turning the pseudo-polynomial number of guesses of the packing epoch of straddling items into an approximation scheme are the two challenges that we are currently trying to overcome so as to obtain an FPTAS for this setting.

Chapter 7

Capacity Constrained Assortment Optimization under the Markov Chain based Choice Model

Joint work with Antoine Désir.

7.1 Introduction

Assortment optimization problems arise widely in many practical applications such as retailing and online advertising. In these problems, the goal is to select a subset from a universe of substitutable items to offer to customers in order to maximize the expected revenue. The demand of any item depends on the substitution behavior of the customers that is captured mathematically by a choice model. The choice model specifies the probability that a random consumer selects a particular item from any given offer set. The objective of the decision maker is to identify an offer set that maximizes expected revenue.

Many parametric choice models have extensively been studied in the literature in diverse areas including marketing, transportation, economics, and operations management. The *Multinomial logit (MNL) model* is by far the most popular model in practice due to its tractability [122]. However, some of the simplifying assumptions behind this model, such as the Independence of Irrelevant Alternatives property, make it inadequate for many applications. Consequently, more complex choice models have been developed to capture a richer class of substitution behaviors. Such models include the nested logit model [129] and the mixture of Multinomial logit model [85]. Nonetheless, the increase in model complexity makes their estimation and assortment optimization problems significantly more difficult. Hence, one of the key challenges in assortment planing is choosing a model that strikes a good balance between its predictability and tractability, as there is a fundamental tradeoff between these desirable properties.

In a recent paper, Blanchet et al. [25] consider a Markov chain based choice model. Here, customer substitution is captured by a Markov chain, where each item (including the no-purchase option) corresponds to a state, and substitutions are modeled using transitions in the Markov chain. The authors show that this model provides a good approximation in choice probabilities to a large class of existing choice models, allowing it to circumvent the model selection problem. In particular, the Markov chain choice model is a generalization of several known choice models in the literature including MNL, Generalized Attraction Model (GAM) ([62]), and the exogenous demand model ([79]). Furthermore, Blanchet et al. [25] show that the unconstrained assortment optimization problem under the Markov chain model is polynomial time solvable. Zhang and Cooper [130] also consider the Markov chain model in the context of airline revenue management, and present a simulation study. In a recent paper, Feldman and Topaloglu [58] study the network revenue management problem under the Markov chain model and give a linear programming based algorithm.

In this chapter, we consider the capacity constrained assortment problem under the Markov chain model. In this problem, every item i is associated with a weight w_i, and the decision maker is restricted to selecting an assortment whose total weight is at most a given bound, W. Therefore, we can formulate the capacity constrained assortment optimization problem as

$$\max_{S\subseteq\mathcal{N}} \left\{ R(S) : \sum_{i\in S} w_i \leq W \right\}, \tag{Capacity-Assort}$$

where $\mathcal{N}$ denotes the universe of substitutable items and $R(S)$ denotes the expected revenue for any assortment $S \subseteq \mathcal{N}$ under the Markov chain model. For the special case of uniform item weights (i.e. $w_i = 1$ for all i), the capacity constraint reduces to a constraint on the number of items in the assortment. We refer to this setting as the cardinality constrained assortment optimization problem:

$$\max_{S\subseteq\mathcal{N}} \left\{ R(S) : |S| \leq k \right\}. \tag{Cardinality-Assort}$$

The cardinality and capacity constraints on assortments arise naturally in many applications, allowing one to model practical scenarios, such as a shelf space constraint or budget limitations. Capacity constrained assortment optimization has been studied in the literature for many parametric choice models. Davis et al. [51] give an exact algorithm for MNL under cardinality constraint, and more generally, under totally-unimodular constraints. Gallego and Topaloglu [63] propose an exact algorithm for the cardinality constrained problem for a special case of the nested logit model. More recently, Feldman and Topaloglu [59] present an exact algorithm for the latter model when the cardinality constraint is across different nests. Rusmevichientong et al. [104] devise a polynomial-time approximation scheme (PTAS) for the cardinality constrained assortment problem under a mixture of MNL choice model. Désir and Goyal [52] propose a fully polynomial-time approximation scheme (FPTAS) for the capacity constrained assortment problem under both the nested logit and the mixture of MNL models.

7.1.1 Our Contributions

Hardness of Approximation. We show that the capacity constrained assortment optimization problem under the Markov chain model is NP-hard to approximate within a factor better than some given constant, even when all items have uniform prices and unit weights. In this case, the capacity constraint reduces to a bound on the number of items, i.e. to a cardinality constraint. It is interesting to note that, while the unconstrained assortment optimization problem under the Markov chain choice model can be solved optimally in polynomial time, the cardinality constrained problem is APX-hard. In contrast, in both the MNL and Nested logit models, the unconstrained assortment optimization and the cardinality constrained assortment problems have the same complexity.

We also consider the case of totally-unimodular (TU) constraints on the assortment. Note that a cardinality constraint is a special case of TU constraints. These capture a wide range of practical constraints such as precedence, display locations, and quality consistent pricing constraints ([51]). We show that the assortment optimization problem under general totally-unimodular (TU) constraints for the Markov chain choice model is hard to approximate within a factor of $O(n^{1/2-\epsilon})$ for any fixed $\epsilon > 0$, where n is the number of items. This result drastically contrasts that of Davis et al. [51], who prove that the assortment optimization problem with TU constraints for the MNL model can be solved in polynomial time.

Approximation Algorithms: Uniform Prices. The above hardness results motivate us to consider approximation algorithms for the capacity constrained assortment optimization problem under the Markov chain choice model. For the special case, when all item prices are equal, we show that the revenue function is submodular and monotone. Therefore, we can obtain a $(1 - 1/e)$-approximation for the cardinality constrained problem using a greedy algorithm ([94]). In fact, for this special case of uniform prices, we can get a $(1 - 1/e)$-approximation for more general constraints such as a constant number of capacity constraints ([80]) and matroid constraint ([40]).

It is worth mentioning that, from a practical point of view, the uniform-price setting turns the objective function into that of maximizing sales probability. This scenario is very common when products are horizontally-differentiated, i.e., differ by characteristics that do not affect quality or price, such as iPads coming in a variety of colors, or yogurt with different amounts of fat-content.

Approximation Algorithms: General Prices. For the general case of non-uniform item prices, the revenue function is neither submodular nor monotone. Moreover, the performance of the greedy algorithm can be arbitrarily bad even for the cardinality constrained problem. Our main contribution in this chapter is to present a "local-ratio" based algorithm to obtain a $(1/2 - \epsilon)$-approximation for the cardinality constrained assortment optimization problem under the Markov chain model. The running time of our algorithm is polynomial in the input size and $1/\epsilon$. The algorithm is based on a "local-ratio" paradigm that builds the solution iteratively. In each iteration, the algorithm makes an appropriate greedy choice and then constructs a modified instance such that the final objective value is the sum of the objective value of the current solution and the objective value of the solution in the modified instance. Therefore, the local-ratio paradigm allows us to capture the externality of our action in each iteration on the remaining instance by constructing an appropriate modified instance; thereby, linearizing the revenue function even though the original objective function is non-linear. This technique may be of independent interest. We also obtain a $(1/3 - \epsilon)$-approximation for the general capacity constrained assortment optimization problem using the local-ratio paradigm. Our approach also provides an alternative strongly-polynomial exact algorithm for the unconstrained assortment optimization problem under the Markov chain model.

Computational Results. We conduct a computational study to compare the numerical performance of our algorithm. We focus on two particular issues: performance and computational efficiency. We present an exact mixed-integer programming (MIP) formulation of the problem to compute the exact optimal solution for comparison. In the numerical experiments, we observe that the practical performance of our algorithm is significantly better than its worst-case theoretical guarantee. Specifically, although the theoretical guarantee is $(1/2 - \epsilon)$ for the cardinality constrained problem, we observe that the approximation ratio is 0.97 on average and at least 0.77 across all instances considered in our experiments. With respect to computational efficiency, our algorithm is scalable and terminates in a few seconds, and in fact, within one minute in the worst case over all large instances tested. On the other hand, the MIP does not terminate even within a time limit of 2 hours on most of these large instances ($n = 200$).

7.1.2 The Markov Chain Model and Notations

We denote the universe of n products by the set $\mathcal{N} = \{1, 2, \ldots, n\}$ and the no-purchase option by 0, with the convention that $\mathcal{N}_+ = \mathcal{N} \cup \{0\}$. We consider a Markov chain $\mathcal{M}$ with states $\mathcal{N}_+$ to model the substitution behavior of customers. This model is completely specified by initial arrival probabilities λ_i for all states $i \in \mathcal{N}_+$ and the transition probabilities ρ_{ij} for all $i \in \mathcal{N}_+, j \in \mathcal{N}_+$. If a retailer chooses to offer a subset of products S to consumers, then the corresponding states in S of the Markov chain become absorbing states. A customer arrives in state i with probability λ_i if the state is absorbing. Otherwise, the customer transitions to a different state $j \neq i$ and the process continues until the customer reaches an absorbing state. In other words, the probability of a random customer purchasing product i with S being the offer set of products is the probability that the customer reaches state i before any other absorbing states in the underlying Markov chain.

Following [25], we assume that for each state $j \in \mathcal{N}$, there is a path to state 0 with non-zero probability. For a given offer set $S \subseteq \mathcal{N}$, let $\pi(i, S)$ be the choice probability that item i is chosen when the assortment S is offered. Let p_i denote the price of item i. For any assortment S, the expected revenue can be written as

$$R(S) = \sum_{i \in S} \pi(i, S) \cdot p_i.$$

For any (possibly empty) pairwise-disjoint subsets $U, V, W \subseteq \mathcal{N}_+$, let $\mathbb{P}_j(U \prec V \prec W)$ denote the probability that starting from j, we first visit some state in U before visiting any state in $V \cup W$, and subsequently visit some state in V before visiting any state in W, with respect to the transition probabilities of $\mathcal{M}$. Let $\mathbb{P}(U \prec V \prec W) = \sum_{j=1}^{n} \lambda_j \mathbb{P}_j(U \prec V \prec W)$. Note that with this notation, we can write $\pi(i, S) = \mathbb{P}(i \prec S_+ \backslash \{i\})$ where $S_+ = S \cup \{0\}$ for all $S \subseteq \mathcal{N}$ (in this case, $W = \emptyset$).

7.1.3 Outline

The remainder of this chapter is organized as follows. In Section 7.2, we present the hardness results for the constrained assortment optimization problem under the Markov chain model. We present the special case of uniform price items in Section 7.3. We also illustrate why several greedy algorithms, including the one that is provably good for uniform prices, do not provide good approximations for arbitrary prices. In Sections 7.4 and 7.5, we present the local-ratio paradigm and our algorithm for the cardinality constrained problem. We present the generalization to the capacity constrained problem in Section 7.6. Finally, the computational study is presented in Section 7.7.

7.2 Hardness of Approximation

In this section, we present our hardness of approximation results for the constrained assortment optimization problem under the Markov chain choice model.

7.2.1 APX-hardness for Cardinality Constraint with Uniform Prices

We show that Cardinality-Assort is APX-hard, i.e., it is NP-hard to approximate within a given constant. In particular, we prove this result even when all items have uniform prices.

Theorem 7.1. *Cardinality-Assort is APX-hard, even when all items have equal prices.*

Our proof is based on gap preserving reduction from the minimum vertex cover problem on 3-regular (or cubic) graphs, to which we refer to as VCC. This problem is known to be APX-hard (see [3]). In other words, for some constant $\alpha > 0$, it is NP-hard to distinguish whether the minimum-cardinality vertex cover is of size at most k or at least $(1 + \alpha)k$ for cubic graphs. Given any instance $\mathcal{I}$ of the VCC problem with a cubic graph $G = (V, E)$ and $k > |E|/3$, we construct an instance $\mathcal{M}(\mathcal{I})$ of Cardinality-Assort as follows. We consider a Markov chain with states corresponding to each vertex in G and an additional state 0 corresponding to the no-purchase item 0. Each state has a transition to state 0 with probability 1/4. In addition, each state has transitions to the states corresponding to their neighbors in G with probability 1/4 each (since G is a 3-regular graph, the sum of transition probabilities out of any state is one).

To prove the hardness result, we establish the following two properties: i) if the minimum vertex cover of instance $\mathcal{I}$ has size at most k, then the optimal expected revenue for instance $\mathcal{M}(\mathcal{I})$ is at least $\left(\frac{3}{4}+\frac{k}{4n}\right)$, and ii) if the minimum vertex cover of instance $\mathcal{I}$ has size at least $(1+\alpha)k$, then the optimal expected revenue for instance $\mathcal{M}(\mathcal{I})$ is at most $\left(\frac{3}{4}+\frac{k}{4n}-\frac{\alpha}{16}\right)$. Therefore, there is a constant gap between the optimal objective value for instance $\mathcal{M}(\mathcal{I})$ of Cardinality-Assort for the two cases. Since it is NP-hard to distinguish between the two cases for instance $\mathcal{I}$, this implies that it is NP-hard to approximate Cardinality-Assort better than some constant (strictly smaller than 1); thereby, proving the APX-hardness of Cardinality-Assort. We would like to note that Cardinality-Assort is APX-hard even for the special case of uniform item prices. Furthermore, our hardness reduction provides interesting insights towards the structure of difficult instances of the problem.

We present a detailed proof of Theorem 7.1 in Appendix D.1.

7.2.2 Totally-Unimodular Constraints

We consider the assortment optimization under the Markov chain model for the more general case of totally-unimodular constraints. Let $x^S \in \{0,1\}^{|\mathcal{N}|}$ denote the incidence vector for any assortment $S \subseteq \mathcal{N}$ where $x_i^S = 1$ if $i \in S$ and $x_i^S = 0$ otherwise. The assortment optimization problem subject to a totally-unimodular constraint can be formulated as follows:

$$\max_{S \subseteq \mathcal{N}} \left\{ R(S) : Ax^S \leq b \right\}. \tag{TU-Assort}$$

Here, A is a totally-unimodular matrix, and b is an integer vector. Note that the cardinality constraint in Cardinality-Assort is a special case of TU-Assort. We show that TU-Assort is NP-hard to approximate within factor $O(n^{1/2-\epsilon})$, for any fixed $\epsilon > 0$ for the Markov chain model. This result drastically contrasts that of [51], who proved that the assortment optimization problem with totally-unimodular constraints can be solved in polynomial time when consumers choose according to the MNL model.

To establish our inapproximability results for TU-Assort, we demonstrate that totally-unimodular constraints in the Markov chain model capture the distribution over permutations model as a special case. Aouad et al. [7] show that even unconstrained assortment optimization under a general distribution over permutations (or rankings) model is hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$ (n is the number of substitutable items). In an instance of the assortment optimization problem over the distribution over permutations model, we are given a collection of items $\mathcal{N} = \{1, \ldots, n\}$ with prices $p_1 \leq \cdots \leq p_n$, respectively. In addition, we are given an arbitrary (known) distribution on K preference lists, $L_1, \ldots, L_K$, each of which specifies a subset of the items listed in decreasing order of preference. A customer with a given preference list selects the most preferred item that is offered (possibly the no-purchase item) according to his/her list. The goal is to find an assortment such that the expected revenue is maximized.

Theorem 7.2. *TU-Assort cannot be approximated in polynomial-time within a factor $O(n^{1/2-\epsilon})$, for any fixed $\epsilon > 0$, unless $P = NP$.*

We present the proof in Appendix D.1.

7.3 Special Case: Uniform Price Items

In this section, we consider a special case of Cardinality-Assort when item prices are uniform, and prove that this setting can be efficiently approximated within factor $1 - 1/e$.

7.3.1 Constant Factor Approximation Algorithm

When all prices are equal, we show that the revenue function is submodular and monotone. Using the classical result of [94], we have that a greedy algorithm guarantees a $(1 - 1/e)$-approximation for Cardinality-Assort for this special case of uniform prices. We start with a few definitions.

Definition 7.1. *A revenue function $R : 2^{\mathcal{N}} \to \mathbb{R}_+$ is monotone when for all $S \subseteq \mathcal{N}$ and $i \in \mathcal{N}$, we have $R(S \cup \{i\}) \geq R(S)$.*

Definition 7.2. *A revenue function $R : 2^{\mathcal{N}} \to \mathbb{R}_+$ is submodular when for all $S \subseteq T \subseteq \mathcal{N}$ and $i \in \mathcal{N} \backslash T$, we have $R(S \cup \{i\}) - R(S) \geq R(T \cup \{i\}) - R(T)$.*

Theorem 7.3. *When all items have uniform prices, the revenue function $R(\cdot)$ is submodular and monotone.*

Proof. Let p be the price of every item in $\mathcal{N}$. Since item prices are identical, for every subset S and item $i \in \mathcal{N} \backslash S$, we have

$$R(S \cup \{i\}) = R(S) + p \cdot \mathbb{P}(i \prec 0 \prec S).$$

Recall that $\mathbb{P}(i \prec 0 \prec S)$ is the probability that the Markov chain visits state i and then visits state 0 without visiting any state in S. When all prices are equal, the marginal increase in revenue by adding item i is only due to the additional demand that item i is able to capture. Consequently, $R(\cdot)$ is monotone as the quantity $p \cdot \mathbb{P}(i \prec 0 \prec S)$ is non-negative. Moreover, the submodularity of R follows from the fact that for all $S \subseteq T$, we have

$$R(S \cup \{i\}) - R(S) = p \cdot \mathbb{P}(i \prec 0 \prec S) \geq p \cdot \mathbb{P}(i \prec 0 \prec T) = R(T \cup \{i\}) - R(T).$$

$\square$

Therefore, from the classical result of [94] for maximizing a monotone submodular function subject to a cardinality constraint, we know that the greedy algorithm gives a $(1 - 1/e)$-approximation bound for Cardinality-Assort with uniform prices. Algorithm 7.1 describes this procedure in detail. Note that

Algorithm 7.1 Greedy Algorithm

1: Let S be the set of states picked so far, starting with $S = \emptyset$.

2: While $|S| < k$ and there exists $i \in \mathcal{N} \backslash S$ such that $R(S \cup \{i\}) - R(S) \geq 0$,

(a) Let i^* be the item for which $R(S \cup \{i\}) - R(S)$ is maximized, breaking ties arbitrarily.

(b) Add i^* to S.

3: Return S.

for uniform prices, when $|S| < k < n$, the condition in Step 2 that there exist $i \in \mathcal{N} \backslash S$ such that $R(S \cup \{i\}) - R(S) \geq 0$ is redundant as the revenue function is monotone, which is not necessarily true for the case of arbitrary prices. Therefore, we include this condition to describe the greedy algorithm for the general case to discuss implications for arbitrary prices.

More General Constraints for Uniform Prices. For the special case of uniform prices, since the revenue function is monotone and submodular, we can exploit the existing machinery for approximately maximizing submodular monotone functions subject to a wide range of constraints (see, for instance, [83, 37, 80, 40]). This way, constant-factor approximations can be obtained for the assortment optimization under the Markov chain model for more general constraints. For instance, Kulik et al. [80] give a $(1-1/e)$-approximation algorithm for maximizing a monotone submodular function under a fixed number of knapsack (capacity) constraints, and Calinescu et al. [40] give a $(1 - 1/e)$-approximation for maximizing a monotone submodular function under a matroid constraint.

7.3.2 Bad Examples for Arbitrary Prices

The approximation guarantees we establish for uniform prices do not extend to the more general setting with arbitrary prices, even for Cardinality-Assort. In what follows, we point out the drawbacks of the natural greedy heuristics, including Algorithm 7.1, in approximating Cardinality-Assort for arbitrary prices. Intuitively, the performance of Algorithm 7.1 for general prices can be bad since it can make a low-price item absorbing that subsequently blocks all probabilistic transitions going into high price items. We formalize this intuition in the following lemma.

Lemma 7.1. *For arbitrary instances of Cardinality-Assort with a cardinality constraint of k, Algorithm 7.1 can compute solutions whose expected revenue is only $O(1/k)$ times the optimum.*

Proof. Consider the following instance of Cardinality-Assort with $n = k+1$ items, where k is the upper bound specified by the cardinality constraint. We have a state s and states $i = 0, \ldots, k$. The arrival rates are all equal to 0, except for λ_s which is equal to 1. Moreover

$$p_i = \begin{cases} (1/k) + \epsilon & \text{if } i = s \\ 1 & \text{if } i = 1, \ldots, k, \end{cases} \qquad \rho_{ij} = \begin{cases} 1/k & \text{if } i = s \text{ and } j = 1, \ldots, k \\ 1 & \text{if } i = 1, \ldots, k \text{ and } j = 0 \\ 0 & \text{otherwise,} \end{cases}$$

where $\epsilon \leq 1/(2k)$. Figure 7.1 provides a graphical representation of this instance. Algorithm 7.1 first

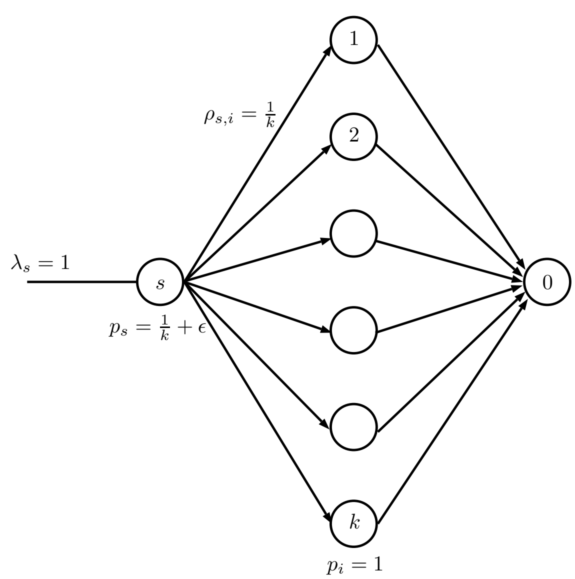

Figure 7.1: A bad example for Algorithm 7.1.

picks item s as $R(\{s\}) = (1/k) + \epsilon$ while $R(\{i\}) = (1/k)$, for $i = 1, \ldots k$. Once s is selected, adding any other state cannot increase the revenue. Therefore, the greedy algorithm gives a revenue of $(1/k) + \epsilon$. However, the optimal solution is to offer items 1 to k, which gives a revenue of 1 in total. When ϵ tends to 0, the approximation ratio goes to $1/k$. □

In fact, we can show that the above example is the worst possible and Algorithm 7.1 gives a $1/k$-approximation for Cardinality-Assort.

Lemma 7.2. *Algorithm 7.1 guarantees a $1/k$-approximation for Cardinality-Assort.*

We present the proof of the above lemma in Appendix D.2.

Modified Greedy Algorithm. The bad instance for Algorithm 7.1 shows that the algorithm may focus too much on local improvements in each iteration, without taking into account the information of the entire network induced by the probability transition matrix or the number of remaining iterations. Therefore, we consider a modified greedy algorithm that accounts for the Markov chain structure by using the optimal solution to the unconstrained assortment problem, where there is no restriction on the number of items picked. This solution can be computed via an algorithm proposed by Blanchet et al. [25] (we also give an alternative strongly-polynomial algorithm for the unconstrained problem in Section 7.4.4). Intuitively, the items picked by the unconstrained optimal assortment should not block each other's demand too much. Let U^* be the optimal unconstrained assortment whose associated revenue can be written as

$$R(U^*) = \sum_{i \in U^*} \mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i. \tag{7.1}$$

A natural candidate algorithm takes the k states with the largest $\mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i$ value within an unconstrained optimal solution, and sets these states to be absorbing. Algorithm 7.2 describes this procedure.

Algorithm 7.2 Greedy Algorithm on Optimal Unconstrained Assortment

1: Let U^* be an optimal solution to the unconstrained problem.

2: Sort items of U^* in decreasing order of $\mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i$.

3: Return $S = \{\text{top } k \text{ items in the sorted order}\}$.

We show in the following lemma that even Algorithm 7.2 performs poorly in the worst case. In fact, we present an example where every subset of k items of the optimal solution U^* has revenue a factor k away from the optimal.

Lemma 7.3. *There are instances where the revenue obtained by Algorithm 7.2 is far from optimal by a factor of $k/|U^*|$ where k is the upper bound in the cardinality constraint.*

Proof. Consider the following instance of the problem with $n+2$ items (or states). We have a state s and states $i = 1, \ldots, n$ and state 0 corresponding to the no-purchase option. The arrival rates are all equal to 0, except for λ_s which is equal to 1. Moreover

$$p_i = \begin{cases} 1-\epsilon & \text{if } i = s \\ 1 & \text{if } i = 1, \ldots, n, \end{cases} \qquad \rho_{ij} = \begin{cases} 1/n & \text{if } i = s \text{ and } j = 1, \ldots, n \\ 1 & \text{if } i = 1, \ldots, n \text{ and } j = 0 \\ 0 & \text{otherwise,} \end{cases}$$

where $\epsilon > 0$. Figure 7.2 provides a graphical representation of this instance. For this example, the

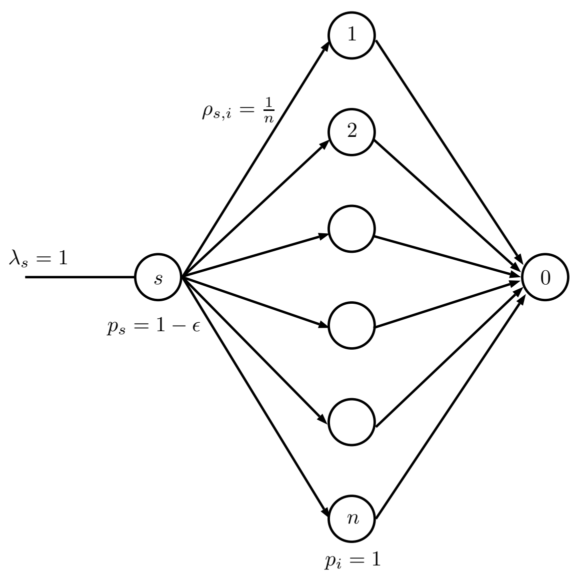

Figure 7.2: A bad example for Algorithm 7.2.

unconstrained optimal assortment is $U^* = \{1, \ldots, n\}$, and the greedy algorithm on U^* selects k items among U^*, meaning that a total revenue of k/n is obtained. However, the optimal solution of the constrained problem is to only offer item s, which gives a revenue of $1-\epsilon$. As ϵ tends to 0, the approximation ratio goes to $k/|U^*|$. $\square$

The poor performance of Algorithm 7.2 on the above example illustrates that an optimal assortment for the constrained problem may be very different from that of its unconstrained counterpart. Hence, searching within an unconstrained optimal solution for a good approximate solution to the constrained problem can be unfruitful in general. It is worth noting that the lower bound of $k/|U^*|$ for Algorithm 7.2 is tight, as stated in the following lemma, whose proof is given in Appendix D.3.

Lemma 7.4. *Algorithm 7.2 guarantees a $k/|U^*|$-approximation algorithm to* Cardinality-Assort.

The analysis of the two greedy variants for the cardinality constrained assortment optimization under the Markov chain model provides important insights that we use towards designing a good algorithm for the problem.

7.4 Local Ratio based Algorithm Design

In this section, we present the general framework of our approximation algorithm for the cardinality and capacity constrained assortment optimization under the Markov chain model.

7.4.1 High-Level Ideas

As the example in Figure 7.1 illustrates, Algorithm 7.1 could end up with a highly suboptimal solution due to picking items that cannibalize, i.e. block, the demand for higher price items. Picking the highest price item will eliminate such a concern. However, a high price item might only capture very little demand, and therefore, generate very small revenue as illustrated in the example in Figure 7.2. When there is a capacity constraint on the assortment, picking such items may not be an optimal use of the capacity. This motivates us to choose the highest price item in an appropriate consideration set. Intuitively, the consideration set will consist of items that generate sufficiently high incremental revenue.

We first give a high-level description of our algorithm that builds the solution iteratively. Let $\mathcal{M}_t$ denote the problem instance in any iteration t. The algorithm (ALG) considers the following two steps in each iteration t:

1. *Greedy Selection.* Define an appropriate consideration set C_t of items, and pick the "highest price" item from C_t.

2. *Instance Update.* Construct a new instance, $\mathcal{M}_{t+1}$, of the constrained assortment optimization problem with appropriately modified item prices and transition probabilities such that

$$\mathsf{ALG}(\mathcal{M}_t) = \Delta_t + \mathsf{ALG}(\mathcal{M}_{t+1}),$$

where $\mathsf{ALG}(\cdot)$ is the revenue of the solution obtained by the algorithm on a given instance, and Δ_t is the incremental revenue in the objective value from the item selected in iteration t.

The instance update step linearizes the revenue function even though the original revenue function is non-linear, which is crucial for our iterative solution approach. We can also view the update rule as a framework to capture the externality of our actions in each iteration of the algorithm. To completely specify the algorithm, we need to provide a precise definition for the consideration set in the greedy step and for the instance update step. For both cardinality and capacity constrained assortment optimization problems, the instance update step is similar, as explained in Section 7.4.2. The consideration set, however, depends on the particular optimization problem being considered and will be defined later on. The intuition is to include items whose incremental revenue is above an appropriately chosen threshold. Our algorithm can be viewed in a local-ratio framework (see, for instance, [18, 17, 19]). Therefore, we will interchangeably refer to the instance updates as local-ratio updates. However, we would like to note that the local-ratio framework does not provide a general recipe for designing an update rule or analyzing the performance bound. In most algorithms in this framework, the update rule follows from a primal-dual algorithm. However, for the capacity constrained assortment optimization problem under the Markov chain model, we do not even know of any good LP formulation and the instance update rule requires new ideas.

7.4.2 Instance Update in Local Ratio Algorithm

Notation. Given an instance $\mathcal{M}$ of the Markov chain model, we define an updated instance $\mathcal{M}(S)$ given that S is made absorbing by modifying the item prices as well as the probability transition matrix. Note that we index the updates by a set S. Therefore, the instance $\mathcal{M}_t$ introduced in the preceding discussion is going to be thought of as $\mathcal{M}(S_{t-1})$, where S_{t-1} denotes the set of items picked up to (and including) step $t-1$. For an instance $\mathcal{M}(S)$, we will denote by p_i^S the updated price of item i, and by ρ_{ij}^S the updated transition probabilities for every $i \in \mathcal{N}, j \in \mathcal{N}_+$. Note that we do not change the arrival rate to any state, i.e., $\lambda_i^S = \lambda_i$ for all $i \in \mathcal{N}$. We also denote by $R^S : 2^{\mathcal{N}} \to \mathbb{R}$ the revenue function associated with the instance $\mathcal{M}(S)$ and by $\mathbb{P}^S(\cdot)$ the probability of any event with respect to the instance $\mathcal{M}(S)$.

Price update. First, we introduce the price updates, such that when S is made absorbing, we account for the revenue generated by every state $j \in S$. To this end, consider a unit demand at state $i \notin S$. This unit demand generates a revenue of p_i when i is made absorbing. On the other hand, when i is not absorbing, this unit demand at i generates a revenue of

$$\sum_{j \in S} \mathbb{P}_i(j \prec S_+ \backslash \{j\}) \cdot p_j.$$

The above revenue (which was already accounted for by S) is lost when i is also made absorbing in addition to S. Hence, the net revenue per unit demand at i when we make it absorbing, provided that S is already absorbing, is

$$p_i - \sum_{j \in S} \mathbb{P}_i(j \prec S_+ \backslash \{j\}) p_j,$$

which we denote as the adjusted price p_i^S. Note that the adjusted prices can be negative, corresponding to the situation where adding an item decreases the overall revenue. The price update is explicitly described in Figure 7.3.

Transition probabilities update. Since the subset of states S is set to be absorbing, we will simply redirect the outgoing probabilities from all states in S to 0. This is described in Figure 7.3.

Price update:

$$p_i^S = \begin{cases} 0 & \text{if } i \in S \\ p_i - \sum_{j \in S} \mathbb{P}_i(j \prec S_+ \backslash \{j\}) p_j & \text{otherwise.} \end{cases}$$

Transition probabilities update:

$$\rho_{ij}^S = \begin{cases} 1 & \text{if } i \in S \text{ and } j = 0 \\ 0 & \text{if } i \in S \text{ and } j \neq 0 \\ \rho_{ij} & \text{otherwise.} \end{cases}$$

Figure 7.3: Instance update in local-ratio algorithm.

We would like to note that the probabilities $\mathbb{P}_i(j \prec S_+ \backslash \{j\})$, needed for our price updates, can be interpreted as the choice probability $\pi(j, S)$ for a modified instance with $\lambda_i = 1$ and $\lambda_\ell = 0$ for $\ell \neq i$. Therefore, these quantities can be efficiently computed via traditional Markov chain tools (see, for instance, [25]).

7.4.3 Structural Properties of the Updates

We first show that the local-ratio updates allow us to linearize the revenue function.

Lemma 7.5. *$R(S_1 \cup S_2) = R(S_1) + R^{S_1}(S_2)$ for every $S_1, S_2 \subseteq \mathcal{N}$.*

Proof. Assume without lost of generality that $S_1 \cap S_2 = \emptyset$, since the items in $S_1 \cap S_2$ all have 0 as their adjusted price and we can then apply the proof to $S_2 \backslash S_1$. Using the definition of the local ratio updates, we have

$$\begin{aligned}
R^{S_1}(S_2) &= \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) p_i^{S_1} \\
&= \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) \left(p_i - \sum_{j \in S_1} \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) p_j \right) \\
&= \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) p_i - \sum_{j \in S_1} \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) p_j.
\end{aligned}$$

With the definition of ρ^{S_1}, note that all items of S_1 are redirected to 0. This, together with the fact that $S_1 \cap S_2 = \emptyset$ implies that for all $i \in S_2$, we have $\mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) = \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\})$. Consequently,

$$\begin{aligned}
R(S_1) + R^{S_1}(S_2) = &\sum_{j \in S_1} \left(\mathbb{P}(j \prec S_{1+} \backslash \{j\}) - \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) \right) p_j \\
&+ \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) p_i \\
= &\sum_{j \in S_1} \left(\mathbb{P}(j \prec S_{1+} \backslash \{j\}) - \mathbb{P}(S_2 \prec j \prec S_{1+} \backslash \{j\}) \right) p_j \\
&+ \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) p_i \\
= &\sum_{j \in S_1} \mathbb{P}(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_j + \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) p_i \\
= &R(S_1 \cup S_2),
\end{aligned}$$

where the second equality holds since

$$\sum_{i\in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) = \mathbb{P}(S_2 \prec j \prec S_{1+} \backslash \{j\}),$$

as by the Markov property, both the left and right terms in the above equality denote the probability that we will visit some state in S_2 before any state in S_{1+}, followed by state $j \in S_1$ before any other state in S_{1+}. □

The next lemma shows that the composition of two local ratio updates over subsets S_1 and S_2 is equivalent to a single local ratio update over $S_1 \cup S_2$. This property is crucial for repeatedly applying local-ratio updates.

Lemma 7.6. *Let $S_1 \subseteq \mathcal{N}$ be some assortment, and let $\mathcal{M}_1 = \mathcal{M}(S_1)$. For any S_2 with $S_1 \cap S_2 = \emptyset$, the instance $\mathcal{M}_1(S_2)$ is identical to the instance $\mathcal{M}(S_1 \cup S_2)$ in terms of item prices and transition probabilities.*

It suffices to verify that $(p_i^{S_1})^{S_2} = p_i^{S_1 \cup S_2}$ for all S_1,S_2 and $i \notin S_1 \cup S_2$, as the above identity clearly holds for the transition matrix updates. The proof is similar to that of Lemma 7.5, and is presented in Appendix D.4. Putting the previous two lemmas together gives the following claim.

Lemma 7.7. $R^{S_1}(S_2 \cup S_3) = R^{S_1}(S_2) + R^{S_1 \cup S_2}(S_3)$ *for any pairwise-disjoint sets $S_1, S_2, S_3 \subseteq \mathcal{N}$.*

7.4.4 Warm-up: Exact algorithm for the Unconstrained Problem

As a warmup, we first present an alternative exact algorithm for the unconstrained assortment optimization problem under the Markov chain model by using the local-ratio framework. Our algorithm is based on the observation that it is always optimal to offer the highest price item for the unconstrained problem, as it does not cannibalize the demand of other items. The latter property is implied by a slightly more general claim, formalized as follows. For any $x \in \mathbb{R}$, let $[x]^+ = \max(x, 0)$.

Lemma 7.8. *Let $S \subseteq \mathcal{N}$. For any item $i \notin S$ with price $p_i \geq [\max_{j\in S} p_j]^+$, we have $R(S \cup \{i\}) \geq R(S)$.*

Proof. From Lemma 7.5, we have that

$$R(S \cup \{i\}) = R(S) + R^S(\{i\}) = R(S) + \mathbb{P}^S(i \prec 0) \cdot p_i^S.$$

Now, $p_i \geq [\max_{j\in S} p_j]^+$ and

$$p_i^S = p_i - \sum_{j\in S} \mathbb{P}_i(j \prec S_+ \setminus \{j\}) \cdot p_j \geq 0,$$

which implies $R(S \cup \{i\}) \geq R(S)$. □

The Algorithm. Based on the above lemma, we present an alternative exact algorithm for the unconstrained assortment optimization problem under the Markov chain model. In particular, we define the consideration set in each iteration to be the set of all items. Therefore, we select the highest adjusted price item in every iteration (breaking ties arbitrarily) and update the prices and transition probabilities according to the local ratio updates described in Figure 7.3. This selection and updating process is repeated until all adjusted prices are non-positive, as explained in Algorithm 7.3.

Algorithm 7.3 Local Ratio for Unconstrained Assortment

1: Let S be the set of states picked so far, starting with $S = \emptyset$.

2: While there exists $i \in \mathcal{N}\backslash S$ such that $p_i^S \geq 0$,

(a) Let i^* be the item for which p_i^S is maximized, breaking ties arbitrarily.

(b) Add i^* to S.

3: Return S.

Theorem 7.4. *Algorithm 7.3 computes an optimal solution for the unconstrained assortment optimization problem under the Markov chain model.*

Proof. The correctness of Algorithm 7.3 is based on the observation that it is always optimal to offer the highest adjusted price item, as long as this price is non-negative. Suppose item 1 is the highest price item. From Lemma 7.8, we get $R(S \cup \{1\}) \geq R(S)$ for any assortment S. Therefore, we can assume that item 1 belongs to the optimal assortment. From Lemma 7.5, we can write

$$\max_{S\subseteq\mathcal{N}} R(S) = R(\{1\}) + \max_{S'\subseteq\mathcal{N}\backslash\{1\}} R^{\{1\}}(S').$$

It remains to show that, when we get to an iteration where our current absorption set is X, and the adjusted price of every state in the modified instance $\mathcal{M}(X)$ is non-positive, then X is an optimal solution to $\mathcal{M}$. To see this, by repeated applications of Lemmas 7.5 and 7.6, we have

$$\max_{S\subseteq\mathcal{N}} R(S) = R(X) + \max_{S'\subseteq\mathcal{N}\backslash X} R^X(S').$$

However, since the adjusted price of every state in the instance $\mathcal{M}(X)$ is non-positive, we must have $R^X(S') \leq 0$ for all $S' \subseteq \mathcal{N}\backslash X$. Hence, it is optimal not to make any state in $\mathcal{M}(X)$ absorbing, which implies that X is an optimal solution to $\mathcal{M}$. $\square$

Implications. Our algorithm for the unconstrained assortment optimization over the Markov chain model provides interesting insights for some known results about the optimal stopping problem and the assortment optimization over the MNL model. Blanchet et al. [25] relate the unconstrained assortment problem to the optimal stopping time on a Markov chain (see [47]). In this problem, we need to decide at each state i whether to stop and get the reward p_i, or transition according to the transition probabilities of the Markov chain. Moreover, there is an absorbing state 0 with price $p_0 = 0$. Algorithm 7.3 for the unconstrained assortment optimization problem gives an alternative strongly polynomial time algorithm for the optimal stopping problem.

Blanchet et al. [25] prove that the MNL choice model is a special case of the Markov chain based choice model. Therefore, by analyzing Algorithm 7.3 to solve the assortment optimization over the MNL model, we can recover the structure of the optimal assortment being nested by prices, i.e., the optimal assortment consists of the ℓ top-priced items for some ℓ. We give an explicit expression for our local ratio updates when the underlying choice model is MNL in Appendix D.5. Talluri and Van Ryzin [122] prove that under the MNL choice model, the optimal assortment to the unconstrained assortment optimization problem is nested by prices. From our derived expression, it is not difficult to verify that these updates do not change the ranking of the adjusted item prices. Hence, combining the correctness of Algorithm 7.3 with the latter observation provides an alternative way of showing that the optimal assortment is nested by price under the MNL model.

7.5 Cardinality Constrained Assortment Optimization for General Case

In this section, we present a $(1/2 - \epsilon)$-approximation for the cardinality constrained assortment optimization under the Markov chain model, for any fixed $\epsilon > 0$. Following the local-ratio framework described in Section 7.4, our algorithm for the cardinality constrained case also selects a state with high adjusted price in each step from an appropriate consideration set. The consideration set is defined to avoid picking states that have a high adjusted price but capture very little demand. In particular, the consideration set includes only items whose incremental revenue is at least a certain threshold.

The Algorithm. Our algorithm is iterative and selects a single item in each step. Let S_t be the set of selected items by the end of step t, starting with $S_0 = \emptyset$. We use σ_t to denote the item picked in step t, meaning that $S_t = \{\sigma_1, \dots, \sigma_t\}$. At every step $t \geq 1$, we select the highest adjusted price item (with respect to $p^{S_{t-1}}$, breaking ties arbitrarily) among items in the following consideration set:

$$C_t = \left\{ i \in \mathcal{N} \backslash S_{t-1} : R^{S_{t-1}}(\{i\}) \geq \alpha \frac{R(S^*)}{k} \right\},$$

where S^* is the optimal solution, k is the cardinality bound, and $\alpha \in (0,1)$ is a parameter whose value will be optimized later. Note that C_t is defined at the beginning of step t, whereas S_t is defined at the end of step t, and includes the item selected in this step. Once the item σ_t is selected, we recompute the adjusted prices via the local ratio update described in Figure 7.3, and update the consideration set to get C_{t+1}. The algorithm terminates when either k items have already been picked (i.e., upon the completion of step k), or when the consideration set C_t becomes empty.

Guessing the value of $R(S^*)$. Since the optimal revenue $R(S^*)$ is not known a-priori, we need to describe how the value of $R(S^*)$ is approximately guessed to complete the algorithm's description. A natural upper bound for $R(S^*)$ is $R(U^*)$, when U^* is the optimal unconstrained solution. From Lemma 7.4, we know that $R(S^*) \geq \frac{k}{|U^*|} R(U^*)$. Now, given an accuracy parameter $0 < \epsilon < 1$, let

$$\begin{aligned} B_j &= \frac{k}{|U^*|} R(U^*)(1+\epsilon)^j, \quad j = 1, \dots, J \\ J &= \min \{ j \in \mathbb{N} \ : \ B_j \geq R(U^*) \}. \end{aligned} \tag{7.2}$$

Note that $J = O(\frac{1}{\epsilon} \log k)$. For each guess B_j for the true value of $R(S^*)$, we run the algorithm, and eventually return the best solution found over all runs. Algorithm 7.4 describes the resulting procedure for a particular choice of B_j and threshold α for the consideration set. Algorithm 7.5 describes the full procedure for any given $\epsilon > 0$.

7.5.1 Technical Lemmas

Prior to analyzing the performance guarantee of our algorithm, we present two technical lemmas. We start by arguing that the revenue function is sublinear for general item prices.

Algorithm 7.4 Algorithm with guess B_j and threshold α

1: Let S be the set of states picked so far, starting with $S = \emptyset$.

2: For all S, let $C(S) = \{i \in \mathcal{N} \backslash S : R^S(\{i\}) \geq \frac{\alpha \cdot B_j}{k}\}$.

3: While $|S| < k$ and $C(S) \neq \emptyset$,

(a) Let i^* be the item of $C(S)$ for which p_i^S is maximized, breaking ties arbitrarily.

(b) Add i^* to S.

4: Return S.

Algorithm 7.5 Local-ratio Algorithm for Cardinality-Assort with threshold α

1: Given any $\epsilon > 0$, let J and B_j, $j \in [J]$ be as defined in (7.2).

2: For all $j \in [J]$, let S_j be the solution returned by Algorithm 7.4 with guess B_j and threshold α

3: Return $\text{argmax}_{j \in [J]} R(S_j)$.

Lemma 7.9. *For all $S_1, S_2 \subseteq \mathcal{N}$ consisting only of non-negative priced items, $R(S_1 \cup S_2) \leq R(S_1) + R(S_2)$.*

Proof. We have that

$$
\begin{aligned}
R(S_1 \cup S_2) &= \sum_{j \in S_1} \mathbb{P}(j \prec (S_1 \cup S_2)_+ \setminus \{j\}) \cdot p_j + \sum_{j \in S_2 \setminus S_1} \mathbb{P}(j \prec (S_1 \cup S_2)_+ \setminus \{j\}) \cdot p_j \\
&\leq \sum_{j \in S_1} \mathbb{P}(j \prec (S_1)_+ \setminus \{j\}) \cdot p_j + \sum_{j \in S_2} \mathbb{P}(j \prec (S_2)_+ \setminus \{j\}) \cdot p_j \\
&= R(S_1) + R(S_2),
\end{aligned}
$$

where the first inequality follows as for any $j \in S_i$ $(i = 1, 2)$, $\mathbb{P}(j \prec (S_1 \cup S_2)_+ \setminus \{j\}) \leq \mathbb{P}(j \prec (S_i)_+ \setminus \{j\})$. □

Next, we establish a technical lemma that allows us to compare the revenue of the optimal solution $R(S^*)$ with the revenue of the set returned by our algorithm, $R(S_t)$. First, note that the consideration sets along different steps are nested (i.e., $C_1 \supseteq C_2 \supseteq \cdots$). Therefore, once an item disappears from the consideration set, it never reappears. This allows us to partition the items of S^* according to the moment they disappear from the consideration set (since either their adjusted revenue becomes too small or they get picked by the algorithm). More precisely, let $Z_0 = S^*$ and for all $t \geq 1$, we define the following sets:

- $Z_t = S^* \cap C_t$ denotes the items of S^* which are in the consideration set C_t.
- $Y_t = Z_{t-1} \backslash Z_t$ denotes the items of S^* which disappear from the consideration set during step $t-1$.
- $Y_t^+ = \{i \in Y_t : p_i^{S_{t-1}} \geq 0\}$ denotes the items of Y_t which have a non-negative adjusted price at step t.

Note that these sets are all defined at the beginning of step t. The following lemma relates the adjusted revenue of items in Z_{t-1} and Z_t in terms of the marginal change in revenue, $R(S_t) - R(S_{t-1})$.

Lemma 7.10. *For all $t \geq 1$, $R(S_t) - R(S_{t-1}) \geq R^{S_{t-1}}(Z_t) - (R^{S_t}(Z_{t+1}) + R^{S_t}(Y_{t+1}^+))$.*

Proof. Recall that, by definition, Z_t contains the items of S^* that are in the consideration set at the beginning of step t. Since our algorithm picks the highest adjusted price item, σ_t, in the consideration set C_t, we have $p_{\sigma_t}^{S_{t-1}} \geq p_i^{S_{t-1}} \geq 0$ for all items $i \in Z_t$. Therefore, by Lemma 7.8,

$$R^{S_{t-1}}(Z_t) \leq R^{S_{t-1}}(Z_t \cup \{\sigma_t\}). \tag{7.3}$$

We now consider two cases, depending on whether the item σ_t appears in the optimal solution S^* or not.

Case (a): $\sigma_t \notin S^*$. From Lemma 7.7, $R^{S_{t-1}}(Z_t \cup \{\sigma_t\}) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t)$. Consequently, from inequality (7.3), we have

$$\begin{aligned} R^{S_{t-1}}(Z_t) &\leq R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t) \\ &= R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_{t+1} \cup Y_{t+1}) \\ &\leq R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_{t+1} \cup Y_{t+1}^+) \\ &\leq R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_{t+1}) + R^{S_t}(Y_{t+1}^+), \end{aligned}$$

where the second inequality holds since removing all negative adjusted price items can only increase net revenue, and the last inequality follows from Lemma 7.9. Adding $R(S_{t-1})$ on both sides of the inequality yields the desired inequality by Lemma 7.5.

Case (b): $\sigma_t \in S^*$**.** From Lemma 7.7, $R^{S_{t-1}}(Z_t) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t \backslash \{\sigma_t\})$. Then, similar to the previous case, we have

$$R^{S_t}(Z_t \backslash \{\sigma_t\}) \le R^{S_t}((Z_{t+1} \cup Y_{t+1}^+) \backslash \{\sigma_t\}) \le R^{S_t}(Z_{t+1}) + R^{S_t}(Y_{t+1}^+ \backslash \{\sigma_t\}).$$

Note that $R^{S_t}(Y_{t+1}^+ \backslash \{\sigma_t\}) = R^{S_t}(Y_{t+1}^+)$ since $p_{\sigma_t}^{S_t} = 0$ and σ_t is an absorbing state in $\mathcal{M}(S_t)$. Adding $R(S_{t-1})$ on both sides of the inequality concludes the proof. $\square$

From the above result, we obtain the following claim.

Lemma 7.11. *For all $t \ge 0$, we have $R(S_t) \ge R(S^*) - (R^{S_t}(Z_{t+1}) + \sum_{j=1}^{t+1} R^{S_{j-1}}(Y_j^+))$.*

Proof. By summing the inequality stated in Lemma 7.10 over $j = 1, \ldots, t$, we obtain a telescopic sum which yields

$$R(S_t) \ge R(Z_1) - \left(R^{S_t}(Z_{t+1}) + \sum_{j=2}^{t+1} R^{S_{j-1}}(Y_j^+) \right).$$

Since every item in S^* must have non-negative price and $S^* = Z_1 \cup Y_1$ by definition, we have $R(S^*) \le R(Z_1) + R(Y_1)$ by sublinearity of the revenue function (see Lemma 7.9). Combining these two inequalities concludes the proof. $\square$

7.5.2 Analysis of the Local-Ratio Algorithm

We show that the local-ratio algorithm gives a $(1/2 - \epsilon)$-approximation for Cardinality-Assort for any fixed $\epsilon > 0$. In particular, we have the following theorem.

Theorem 7.5. *For any fixed $\epsilon > 0$, Algorithm 7.5 gives a $(1/2 - \epsilon/2)$-approximation for Cardinality-Assort. Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

Proof. For a fixed $\epsilon > 0$, let j^* be such that $\frac{R(S^*)}{1+\epsilon} \le B_{j^*} \le R(S^*)$. Let $B = B_{j^*}$ and consider the solution returned by Algorithm 7.4 with guess B and threshold α. We consider two cases based on the condition by which the algorithm terminates.

Case 1. If the algorithm stops after completing step k, then by linearity of the revenue when using the local ratio updates (Lemmas 7.5 and 7.6), the resulting solution S_k has a revenue of

$$R(S_k) = \sum_{t=1}^{k} R^{S_{t-1}}(\{\sigma_t\}) \ge \alpha B \ge \frac{\alpha}{1+\epsilon} \cdot R(S^*) \ge (1-\epsilon)\alpha R(S^*),$$

where the above inequality holds since the item σ_t belongs to the consideration set C_t, and therefore $R^{S_{t-1}}(\{\sigma_t\}) \geq \alpha B/k$.

Case 2. Now, suppose the algorithm stops at the end of step $k' < k$, after discovering that $C_{k'+1} = \emptyset$. From Lemma 7.11, we get

$$R(S_{k'}) + R^{S_{k'}}(Z_{k'+1}) \geq R(S^*) - \sum_{j=1}^{k'+1} R^{S_{j-1}}(Y_j^+).$$

Now, since $C_{k'+1} = \emptyset$, this implies that $Z_{k'+1} = \emptyset$. Moreover, from Lemma 7.9, we also have $R^{S_{j-1}}(Y_j^+) < |Y_j^+| \cdot \alpha \cdot B/k$ for all $j = 1, \ldots, k'+1$. Therefore,

$$\sum_{j=1}^{k'+1} R^{S_{j-1}}(Y_j^+) \leq \alpha \cdot \frac{B}{k} \cdot \sum_{j=1}^{k'+1} |Y_j^+| \leq \alpha B \leq \alpha R(S^*),$$

where the second inequality holds since $\sum_{j=1}^{k'+1} |Y_j^+| \leq k$ and the last inequality holds as $B \leq R(S^*)$. Therefore,

$$R(S_{k'}) \geq R(S^*) - \alpha R(S^*) = (1-\alpha) \cdot R(S^*).$$

This shows that the approximation ratio attained by our algorithm is

$$\min\{(1-\epsilon)\alpha,\ 1-\alpha\}.$$

Picking $\alpha = 1/2$ we obtain a $(1/2 - \epsilon/2)$-approximation for Cardinality-Assort.

Running time. Algorithm 7.5 considers $J = O(\frac{1}{\epsilon}\log n)$ guesses for $R(S^*)$. For any given guess B_j, the running time of Algorithm 7.4 is polynomial in the input size. Therefore, the overall running time of Algorithm 7.5 is polynomial in the input size and $1/\epsilon$. □

Tight example. We show that Algorithm 7.5 is tight in the following sense: consider Algorithm 7.4 with input guess as the true value of $R(S^*)$ and threshold $\alpha = 1/2$, then there are instances for which the approximation ratio is 1/2. In particular, we consider an instance with 3 items. The Markov chain has 4 states $\mathcal{N}_+ = \{s, 1, 2, 0\}$. The prices are: $p_s = 1, p_1 = p_2 = 2$. The arrival rate for state s is $\lambda_s = 1$ and all other states have an arrival rate of zero. The transition probabilities are given in Figure 7.4. Consider the cardinality constrained assortment problem with cardinality bound, $k = 1$. The optimal assortment is $S^* = \{s\}$ with $R(S^*) = 1$. With guess $R(S^*)$ and $\alpha = 1/2$, the consideration set in the first step is $\{s, 1, 2\}$, and therefore Algorithm 7.4 picks either 1 or 2, obtaining a revenue of $R(S^*)/2$.

We would like to note that our algorithm runs Algorithm 7.4 for different guesses $B_j, j = 1, \ldots, J$ and returns the best solution across all runs. Therefore, the performance bound of our algorithm is at least $(1/2 - O(\epsilon))$ and possibly better. In fact, in our computational study, we observe that the empirical performance of our algorithm is significantly better than the theoretical bound of $(1/2 - O(\epsilon))$. We describe the computational study in Section 7.7. It is an interesting open question to provide a tighter analysis of the approximation bound for Algorithm 7.5 that returns the best solution among several guesses of $R(S^*)$.

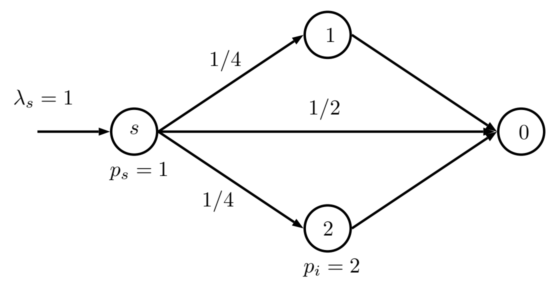

Figure 7.4: A tight example for Algorithm 7.5.

7.6 Capacity Constrained Assortment Optimization for General Case

In this section, we show how to approximate the capacity constrained problem under the Markov chain model within factor $1/3 - \epsilon$, for any fixed $\epsilon > 0$. Recall that, unlike the simpler cardinality case, now each item i has an arbitrary weight w_i, and we have an upper bound W on the available capacity. We assume without loss of generality that each item individually satisfies the capacity constraint, i.e., $w_i \leq W$ for all $i \in \mathcal{N}$.

The Algorithm. We describe a local-ratio based algorithm, similar in spirit to the one for the cardinality constrained problem, by suitably adapting the way consideration sets are defined. For this purpose, instead of considering items whose incremental absorption revenue exceeds a certain threshold, we only consider items whose incremental absorption revenue per unit of weight exceeds a certain certain threshold.

Again, our algorithm selects a single item in each step. Let S_t be the set of selected items by the end of step t, starting with $S_0 = \emptyset$. We use σ_t to denote the item picked in step t, meaning that $S_t = \{\sigma_1, \dots, \sigma_t\}$. At every step $t \geq 1$, we select the highest adjusted price item (with respect to $p^{S_{t-1}}$, breaking ties arbitrarily) among items in the following consideration set:

$$C_t = \left\{ i \in \mathcal{N} \backslash S_{t-1} : \frac{R^{S_{t-1}}(\{i\})}{w_i} \geq \alpha \frac{R(S^*)}{W} \right\},$$

where S^* is the optimal solution, W is the capacity bound, and $\alpha \in (0, 1)$ is a parameter whose value will be optimized later. Once the item σ_t is selected, we recompute the adjusted prices via the local ratio update described in Figure 7.3. This selection and update process is repeated in every step until either the consideration set becomes empty or adding the current item violates the capacity constraint. Let t' be such a step. In the former case, we stop and return $S_{t'-1}$. In the latter case, we take either $S_{t'-1}$ or $\{\sigma_{t'}\}$, depending on which of these sets has a larger total revenue.

Guessing $R(S^*)$**.** As in the case of cardinality constraints, since the value of $R(S^*)$ is unknown, we need to approximately guess the value $R(S^*)$. We will use a procedure similar to the one given in Section 7.5, with the exception of utilizing $\frac{1}{|U^*|} R(U^*)$ as a lower bound (see proof of Lemma 7.2 in Appendix D.2), where U^* is the optimal unconstrained solution. In particular, we consider the following guesses for $R(S^*)$.

$$\begin{aligned} B_j &= \frac{1}{|U^*|} R(U^*)(1 + \epsilon)^j, \ j = 1, \dots, J \\ J &= \min \{ j \in \mathbb{N} \ : \ B_j \geq R(U^*) \} . \end{aligned} \tag{7.4}$$

Note that $J = O(\frac{1}{\epsilon} \log n)$. Algorithm 7.6 provides a description of our approximation algorithm for Capacity-Assort, given a particular guess B_j for $R(S^*)$ and threshold α, while Algorithm 7.7 describes the complete procedure.

7.6.1 Analysis

To analyze the above algorithm, it is convenient to have a technical lemma similar to Lemma 7.11. By defining the same sets Y_t and Z_t with respect to the optimal assortment S^* to Capacity-Assort and the adapted consideration sets C_t, the exact same lemma holds. We therefore do not restate this claim and its proof, as these are identical to those of Lemma 7.11. The following theorem shows that the local-ratio algorithm gives a $(1/3 - \epsilon)$-approximation for Cardinality-Assort for any fixed $\epsilon > 0$.

Algorithm 7.6 Algorithm with guess B_j and threshold α

Let S be the set of states picked so far, starting with $S = \emptyset$.

For all S, let $C(S) = \{i \in \mathcal{N} : \frac{R^S(\{i\})}{w_i} \geq \alpha \cdot \frac{B_j}{W}\}$.

While $\sum_{i \in S} w_i < W$ and $C(S) \neq \emptyset$,

(a) Let i^* be the item of $C(S)$ for which p_i^S is maximized, breaking ties arbitrarily.

(b) If $\sum_{i \in S \cup \{i^*\}} w_i < W$, add i^* to S.

(c) Else return the highest revenue set among $\{i^*\}$ and S.

Return S.

Algorithm 7.7 Local-ratio Algorithm for Capacity-Assort with threshold α

Given any $\epsilon > 0$, let J and B_j, $j \in [J]$ be as defined in (7.4).

For all $j \in [J]$, let S_j be the solution returned by Algorithm 7.6 with guess B_j and threshold α

Return $\text{argmax}_{j \in [J]} R(S_j)$.

Theorem 7.6. *For any fixed $\epsilon > 0$, Algorithm 7.7 gives a $(1/3 - \epsilon/3)$-approximation for* Capacity-Assort*. Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

Proof. For a fixed $\epsilon > 0$, let j^* be such that $\frac{R(S^*)}{1+\epsilon} \leq B_{j^*} \leq R(S^*)$. Let $B = B_{j^*}$ and consider the solution returned by Algorithm 7.6 with guess B and threshold α. We consider two cases based on the condition by which the algorithm terminates. Let t' be the step at which the algorithm terminates.

Case 1. Suppose we stop the algorithm since adding the item $\sigma_{t'}$ violates the capacity constraint, that is, $\sum_{t=1}^{t'} w_{\sigma_t} > W$. In this case, we return either $S_{t'-1}$ or $\{\sigma_{t'}\}$, depending on which of these sets has a larger revenue. We argue that this choice guarantees a revenue of at least $\alpha R(S^*)/2$, since

$$\begin{aligned}
\max\left\{R(S_{t'-1}), R(\{\sigma_{t'}\})\right\} &\geq \max\left\{\sum_{t=1}^{t'-1} R^{S_t}(\{\sigma_t\}), R^{S_{t'-1}}(\{\sigma_{t'}\})\right\} \\
&\geq \max\left\{\alpha\frac{B}{W}\sum_{t=1}^{t'-1} w_{\sigma_t}, \alpha\frac{B}{W}w_{\sigma_{t'}}\right\} \\
&= \alpha\frac{B}{W}\cdot\max\left\{\sum_{t=1}^{t'-1} w_{\sigma_t}, w_{\sigma_{t'}}\right\} \\
&\geq \alpha\frac{B}{2} \\
&\geq \alpha\cdot\frac{R(S^*)}{2(1+\epsilon)} \\
&\geq (1-\epsilon)\alpha\cdot\frac{R(S^*)}{2},
\end{aligned}$$

where the third to last inequality holds since $\max\{\sum_{t=1}^{t'-1} w_{\sigma_t}, w_{\sigma_{t'}}\} \geq W/2$ and the second to last inequality follows as $B \geq R(S^*)/(1+\epsilon)$.

Case 2. On the other hand, suppose the algorithm terminates since $C_{t'+1} = \emptyset$. Using Lemma 7.11 adapted to the capacitated case, we have

$$R(S_{t'}) + R^{S_{t'}}(Z_{t'+1}) \geq R(S^*) - \sum_{j=1}^{t'+1} R^{S_{j-1}}(Y_j^+).$$

Since $C_{t'+1} = \emptyset$, this implies that $Z_{t'+1} = \emptyset$. Moreover, from Lemma 7.9, for all $j = 1, \ldots, t'+1$, we have

$$R^{S_{j-1}}(Y_j^+) < \alpha B \cdot \frac{\sum_{i\in Y_j^+} w_i}{W}.$$

Since our algorithm stopped prior to reaching the capacity constraint, we have $\sum_{j=1}^{t'+1}\sum_{i\in Y_j^+} w_i \leq W$. Consequently, $\sum_{j=1}^{t'+1} R^{S_{j-1}}(Y_j^+) < \alpha B \leq \alpha R(S^*)$, and therefore,

$$R(S_{t'}) \geq R(S^*) - \alpha R(S^*) = (1-\alpha)R(S^*).$$

As a result, the approximation ratio attained by our algorithm is

$$\min\left\{(1-\epsilon)\frac{\alpha}{2},\ 1-\alpha\right\}.$$

By setting $\alpha = 2/3$, we obtain an approximation factor of $(1/3 - \epsilon/3)$.

Running Time . Algorithm 7.7 considers $J = O(\frac{1}{\epsilon}\log n)$ guesses of $R(S^*)$. Each run of Algorithm 7.6 for a given guess is polynomial time. Therefore, the overall running time of Algorithm 7.7 is polynomial in the input size and $1/\epsilon$. □

Tight example. Our analysis is tight in the following sense. When Algorithm 7.7 is run with the true value of $R(S^*)$, there are instances for which the approximation ratio is 1/3. For example, consider the instance given in Figure 7.5. For a capacity bound of $W = 1$, the optimal assortment is $S^* = \{b, c\}$. Initially, all the items are in the consideration set and the algorithm picks item a, the highest price item. In the next step, no item can be added to the assortment. The algorithm therefore returns $S = \{a\}$ since $R(\{a\}) > R(\{d\})$ and yields a revenue of $R(S^*)/3 + O(\epsilon)$. When ϵ goes to 0, the approximation ratio goes to 1/3.

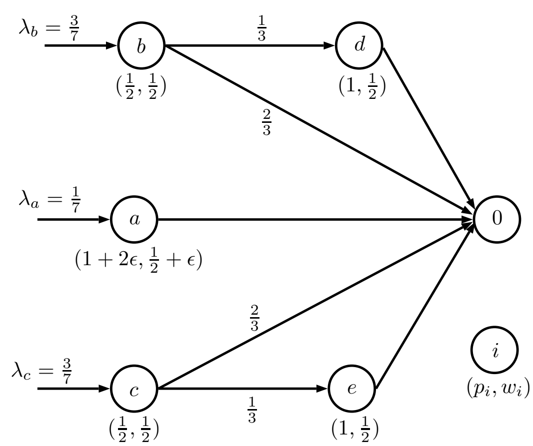

Figure 7.5: A tight example for Algorithm 7.7.

7.7 Computational Experiments

In this section, we present our results from a computational study to test the performance of Algorithm 7.5 for the cardinality constrained assortment optimization for the Markov chain choice model. In particular, we focus on testing: i) the performance of our algorithm with respect to an optimal algorithm, and ii) the running time of this algorithm. We first present a mixed-integer programming (MIP) formulation of Cardinality-Assort.

7.7.1 A Mixed-Integer Programming Formulation

We show that the following mixed-integer program (MIP) is an exact reformulation of Cardinality-Assort.

$$
\begin{aligned}
\max \ & \sum_{i=1}^{n} \alpha_i p_i \\
\text{s.t. } & \alpha_i + \beta_i - \sum_{j=1}^{n} \rho_{ji}\beta_j = \lambda_i, && \forall i = 1, \dots, n \\
& y_i \geq \alpha_i, && \forall i = 1, \dots, n \\
& \sum_{i=1}^{n} y_i \leq k \\
& \alpha_i \geq 0, \beta_i \geq 0, y_i \in \{0,1\}, && \forall i = 1, \dots, n.
\end{aligned}
\tag{7.5}
$$

Lemma 7.12. *The mixed-integer program* (7.5) *is an exact reformulation of Cardinality-Assort.*

Proof. Consider the following LP:

$$
\begin{aligned}
\max \ & \sum_{i=1}^{n} \alpha_i p_i \\
\text{s.t. } & \alpha_i + \beta_i - \sum_{j=1}^{n} \rho_{ji}\beta_j = \lambda_i, && \forall i = 1, \dots, n \\
& \alpha_i \geq 0, \beta_i \geq 0, && \forall i = 1, \dots, n.
\end{aligned}
\tag{7.6}
$$

Let (α, β) be an extreme point solution to the above LP, and let $S = \{i : \alpha_i > 0\}$. Feldman and Topaloglu [58] show that α_i is the choice probability $\pi(i, S)$ when the assortment S is offered under the Markov chain choice model. Hence, the objective value $\sum_{i=1}^{n} \alpha_i r_i$ equals to $R(S)$. By adding the indicator variables y_i, we are restricting ourselves to the subset of feasible solutions where at most k of the α_i-s are allowed to be strictly positive. Note that the extreme points of this polytope, corresponding to the projection of the feasible space of the MIP down to the (α, β) coordinates, are exactly the set of assortments S with cardinality at most k. Hence, (7.5) is a mixed-integer formulation of the cardinality constrained assortment optimization problem. □

7.7.2 Settings Tested

We proceed by describing the families of random instances being tested in our computational experiments. Here, each item's price p_i is uniformly distributed over the interval $[0, 1]$. Note that since we present statistics regarding approximation factors, any constant here will give identical results, so the choice of 1 is arbitrary. In each instance, we compute the optimal unconstrained assortment U^* using the LP given by [25]. We then choose the cardinality constraint k uniformly between 1 and $|U^*|/2$. For the transition probabilities ρ_{ij} and the arrival rates λ_i, we test our algorithm on three different settings:

1. We generate n^2 independent random variables X_{ij}, each picked uniformly over the interval $[0, 1]$. We then set $\rho_{ij} = X_{ij}/\sum_{j=0}^{n} X_{ij}$ for all i, j such that $i \neq j$. Since we do not allow self-loops (i.e. $\rho_{ii} = 0$), the number of random variables needed is n^2. For the arrival rates, we then generate n independent random variables Y_i, each picked uniformly over the interval $[0, 1]$, and set $\lambda_i = Y_i/\sum_{j=1}^{n} Y_j$ for all $i \neq 0$.

2. In this setting, we sparsify the transition matrix of setting 1. More precisely, we additionally generate n^2 independent random variable Z_{ij}, each following a Bernoulli distribution with parameter 0.2. For all i, j such that $i \neq j$, we set $\rho_{ij} = Z_{ij}X_{ij}/\sum_{j=0}^{n} Z_{ij}X_{ij}$, where X_{ij} are generated as in setting 1. This is equivalent to eliminating each transition (i, j) with probability 0.8 and then renormalizing. The arrival rates are generated similarly to setting 1.

3. The transition matrix in this last setting is one of a random walk. More precisely, we generate n^2 independent random variable X_{ij}, each following a Bernoulli distribution with parameter 0.5. We then set $\rho_{ij} = X_{ij}/\sum_{j=0}^{n} X_{ij}$ for all i, j such that $i \neq j$. We also generate n random variables Y_i, each following a Bernoulli distribution with parameter 0.5, and set $\lambda_i = Y_i/\sum_{j=1}^{n} Y_j$ for all $i \neq 0$.

7.7.3 Results

We examine how our algorithm performs in term of both approximation and running time. Table 7.1 shows the approximation ratio of Algorithm 7.5 (with $\epsilon = 0.1$) for the different settings and the different values of n. We use the MIP formulation given in (7.5) to compute the optimal assortment. As can be observed, the actual performance of our algorithm is significantly better than its worst case theoretical guarantee. Indeed, in all settings tested, the average approximation ratio is always above 0.97. Moreover, the worst approximation ratio over all instances is above 0.77.

Setting	n	Approximation Ratio		# instances within x% of OPT				# instances
		Average	Minimum	2%	5%	10%	20%	
1	30	0.9783	0.7771	664	812	972	998	1,000
2	30	0.9784	0.7734	662	858	956	995	1,000
3	30	0.9830	0.7693	708	884	976	998	1,000
1	60	0.9803	0.8671	622	838	997	1,000	1,000
2	60	0.9796	0.8094	621	888	982	1,000	1,000
3	60	0.9854	0.8885	693	941	998	1,000	1,000
1	100	0.9763	0.9132	52	79	100	100	100
2	100	0.9782	0.8882	59	91	99	100	100
3	100	0.9848	0.9142	70	97	100	100	100

Table 7.1: Performance of Algorithm 7.5 for Cardinality-Assort.

The running time of our algorithm also scales nicely. Table 7.2 shows the performance of Algorithm 7.5 in terms of running time for setting 2. The running times are very similar for the other settings. On the other hand, while the MIP running time can be competitive in some cases, it blows up when the number of products n gets large (see Table 7.2). Note that for $n = 100$, 12 out of the 100

n	Average Running Time		Maximum Running Time		# instances
	Algorithm 7.5	MIP	Algorithm 7.5	MIP	
30	0.18	0.17	0.67	0.25	1,000
60	0.74	0.67	1.25	29.34	1,000
100	3.18	278.20	9.16	10,226.98	100
200	31.98	**	47.38	**	20

Table 7.2: Running time of Algorithm 7.5 and the MIP for setting 2. ** Denotes the cases when we set a time limit of 2 hours.

instances had a running time of at least 30 minutes. For $n = 200$, we set a time limit of 2 hours for the MIP. Out of the 20 random instances generated, 16 reached the time limit without terminating. These numerical experiments suggest that Algorithm 7.5 is computationally efficient and that its numerical performance is significantly better than the theoretical worst-case guarantee.

We also compare the performance of Algorithm 7.5 with the best solution found by the MIP solver within a time limit that is equal to the running time for Algorithm 7.5 for the corresponding instance. Table 7.3 shows the ratio between the performance of Algorithm 7.5 and that of the best feasible solution found by the MIP within the allowed time limit as well as the duality gaps for the best feasible solution. We observe that although the solver might not even terminate, it finds good solutions within the time limit allowed. On average, the best MIP solution computed within the time limit is slightly better than the solution computed by Algorithm 7.5. Although, for several instances, Algorithm 7.5 outperforms the best MIP solution within the time limit (about 20% instances for $n = 30$ and 10% for $n = 60$). Therefore, the MIP solver spends a significant fraction of the time in reducing the duality gap and proving optimality for large instances. It is interesting open question to find a stronger LP formulation for the problem.

n	Algorithm 5/(Best MIP Solution)		# instances with ratio > 1	Duality gap		# instances
	Average	Maximum		Average	Maximum	
30	0.9800	1.0851	200	2.64 %	34.85 %	1,000
60	0.9805	1.0108	101	10.42 %	69.68 %	1,000
100	0.9787	1.0100	6	15.21 %	52.18 %	100
200	0.9821	1.0029	4	19.06 %	99.60 %	100

Table 7.3: Comparison of Algorithm 7.5 with the best MIP solution when we allow the solver the same time limit.

7.8 Discussion

As mentioned in Section 7.4, the unconstrained assortment problem to the optimal stopping time on a Markov chain. In this problem, we need to decide at each state i whether to stop and get the reward p_i, or transition according to the transition probabilities of the Markov chain. Moreover, there is an absorbing state 0 with price $p_0 = 0$. The optimal stopping problem can be viewed as a special case of Markov Decision Process (MDP). A standard methodology for solving a MDP is by solving a set of optimality equations, whose solution dictates the optimal action to take when in each state (in our case, to stop or to continue transitioning). Solving the optimality equations is a well studied problem in the MDP literature. The value iteration algorithm as well as the linear program used in [25] are related to some of the standard algorithms used for solving the optimality equations (see e.g. Chapters 6 and 7 of [124]). It is worth mentioning that Algorithm 7.3 gives an alternative procedure to solve the optimal stopping problem. Our algorithm is fundamentally different than the existing algorithm as it is sequential in nature: it first decides on a state that it will stop at according to prices, and subsequently modifies the problem in response to its decision and iteratively solve the subproblem. This sequential exact algorithm resembles in spirit the Elimination algorithm presented in [115]. The main difference is that the Elimination algorithm selects states that it will not optimal to stop at. Whether there exists a sequential algorithm (building on our work and that of [115]) for solving a general class of MDP is an interesting open question.

Building on our established connection further, the cardinality constraint assortment optimization problem under the Markov Chain model is analogous to an optimal stopping problem where we impose a constraint on the total number of states we can choose to stop at. To our knowledge, such a restriction on the policy space has not been studied in the optimal stopping literature, and the tractability of such problems is unknown until our work. For instance, the linear programming formulation given by [25] turns into an integer program whose integrality gap is not bounded by any constant. On the other hand, the approximation algorithms we present in Sections 7.5 and 7.6 can be adapted to solve optimal stopping problems involving certain coupling constraints on the actions that a policy can take. Hence, these algorithms are by no means limited to assortment planning problems, and may be of interest to a much broader audience. Perhaps some of the ideas of our algorithms can be extended to solve other MDPs with certain restriction on the policy space.

Another open research direction is to improve the approximation constant of Algorithms 7.5 and 7.7. For instance, our current algorithms use the same threshold α in constructing our consideration set. One potential improvement would be to consider a threshold function that varies from iteration to iteration. Unfortunately, we show in Appendix D.6 that no iteration varying implementation of Algorithm 7.5 can improve the approximation constant using our current lines of analysis. Broadly speaking, the criteria our sequential algorithm to select a new item in each iteration is a combination of the incremental revenue that it brings (which indirectly incorporates the popularity of an item) versus its (adjusted[51]) profitability per unit of sale. Right now our algorithms account for this trade-off by using the incremental revenue as a screening process to construct our consideration set and the adjusted per unit profitability as the selection criteria. Note that we can potentially improve the approximation ratio by considering other functions forms to address the trade-off between the two aforementioned criteria.

[51] taking into account the externality it imposes on other items if we were to offer it

Part I

Bibliography

Bibliography

[1] Atila Abdulkadiroglu and Tayfun Sonmez. Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems. *Econometrica*, 66(3):689–702, 1998.

[2] Atila Abdulkadiroglu and Tayfun Sonmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

[3] Paola Alimonti and Viggo Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1):123–134, 2000.

[4] Noga Alon, Michal Feldman, Ariel D. Procaccia, and Moshe Tennenholtz. Strategyproof approximation mechanisms for location on networks. *CoRR*, abs/0907.2049:3432–3435, 2009.

[5] Noga Alon, Michal Feldman, Ariel D. Procaccia, and Moshe Tennenholtz. Strategyproof approximation of the minimax on networks. *Math. Oper. Res.*, 35(3):513–526, 2010.

[6] Noga Alon, Michal Feldman, Ariel D. Procaccia, and Moshe Tennenholtz. Walking in circles. *Discrete Mathematics*, 310(23):3432–3435, 2010.

[7] Ali Aouad, Vivek Farias, Retsef Levi, and Danny Segev. The approximability of assortment optimization under ranking preferences, 2015. Working paper, available online as SSRN report 2612947.

[8] Stergios Athanassoglou and Jay Sethuraman. House allocation with fractional endowments. *International Journal of Game Theory*, 40(3):481–513, 2011.

[9] H. Aziz, S. Gasper, S. Mackenzie, and T. Walsh. Fair assignment of indivisible objects under ordinal preferences. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 1305–1312, 2014.

[10] Haris Aziz. Random assignment with multi-unit demands. *CoRR*, abs/1401.7700, 2014.

[11] Haris Aziz, Pang Luo, and Christine Rizkallah. Incompatibility of efficiency and strategyproofness in the random assignment setting with indifferences. *CoRR*, abs/1604.07540, 2016.

[12] Haris Aziz and Paul Stursberg. A generalization of probabilistic serial to randomized social choice. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 559–565. AAAI Press, 2014.

[13] Haris Aziz and Chun Ye. Cake cutting algorithms for piecewise constant and piecewise uniform valuations. In Tie-Yan Liu, Qi Qi, and Yinyu Ye, editors, *Web and Internet Economics: 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings*, pages 1–14. Springer International Publishing, 2014.

[14] Egon Balas. Disjunctive programs: Cutting planes from logical conditions, in o.l. mangasarian et al., eds. *Nonlinear Programming*, 2:279–312, 1974.

[15] Hans-Jurgen Bandelt. First euro summer institute networks with condorcet solutions. *European Journal of Operational Research*, 20(3):314 – 326, 1985.

[16] H.J. Bandelt and Labbé M. How bad can a voting location be. *Social Choice and Welfare*, 3(2):125–145, 1986.

[17] Reuven Bar-Yehuda, Keren Bendel, Ari Freund, and Dror Rawitz. The local ratio technique and its application to scheduling and resource allocation problems. In *Graph Theory, Combinatorics and Algorithms*, pages 107–143. Springer, 2005.

[18] Reuven Bar-Yehuda and Shimon Even. A local-ratio theorem for approximating the weighted vertex cover problem. *North-Holland Mathematics Studies*, 109:27–45, 1985.

[19] Reuven Bar-Yehuda and Dror Rawitz. A tale of two methods. In *Theoretical Computer Science: Essays in Memory of Shimon Even*, pages 196–217. Springer, 2006.

[20] Salvador Barberà, Jordi Massó, and Shigehiro Serizawa. Strategy-proof voting on compact ranges. *Games and Economic Behavior*, 25(2):272–291, 1998.

[21] Daniel Bienstock. Approximate formulations for 0-1 knapsack sets. *Oper. Res. Lett.*, 36(3):317–320, 2008.

[22] Daniel Bienstock and Benjamin McClosky. Tightening simple mixed-integer sets with guaranteed bounds. *Math. Program.*, 133(1-2):337–363, 2012.

[23] Daniel Bienstock, Baruch M. Schieber, Jay Sethuraman, and Chun Ye. Approximation algorithms for the incremental knapsack problem, 2016. Working paper.

[24] Garret Birkhoff. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucuman Revista , Serie A*, 5:147–151, 1946.

[25] Jose Blanchet, Guillermo Gallego, and Vineet Goyal. A markov chain approximation to choice modeling. *forthcoming at Operations Research*, 2016.

[26] A. Bogomolnaia and H. Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004.

[27] Anna Bogomolnaia. Random assignment: Redefining the serial rule. *Journal of Economic Theory*, 158, Part A:308 – 318, 2015.

[28] Anna Bogomolnaia and Eun Jeong Heo. Probabilistic assignment of objects: Characterizing the serial rule. *Journal of Economic Theory*, 147(5):2072 – 2082, 2012.

[29] Anna Bogomolnaia and Hervé Moulin. A New Solution to the Random Assignment Problem. *Journal of Economic Theory*, 100(2):295–328, October 2001.

[30] Anna Bogomolnaia and Hervé Moulin. A simple random assignment problem with a unique solution. *Economic Theory*, 19(3):623–636, 2002.

[31] Steven J. Brams. *Mathematics and Democracy: Designing Better Voting and Fair-Division Procedures*. Princeton University Press, 2008.

[32] Steven J. Brams, Michal Feldman, Jamie Morgenstern, John K. Lai, and Ariel D. Procaccia. On maxsum fair cake divisions. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 26:1285–1291, 2012.

[33] Steven J. Brams and Alan D. Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.

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

[35] Margaret L Brandeau and Samuel S Chiu. Parametric facility location on a tree network with an l p-norm cost function. *Transportation Science*, 22(1):59–69, 1988.

[36] Simina Brânzei and P. B. Miltersen. Equilibrium analysis in cake cutting. In *In Proceedings of the 12th International Conference on Autonomous Agent and Multi-Agent Systems (AAMAS)*, pages 327–334, 2013.

[37] Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1433–1452, 2014.

[38] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061 – 1103, 2011.

[39] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Designing random allocation mechanisms: Theory and applications. *American Economic Review*, 103(2):585–623, 2013.

[40] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[41] Christopher P. Chambers. Consistency in the probabilistic assignment model. *Journal of Mathematical Economics*, 40(8):953 – 962, 2004.

[42] Chandra Chekuri and Sanjeev Khanna. Approximation algorithms for minimizing average weighted completion time. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 2004.

[43] Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, 35(3):713–728, 2005.

[44] Yiling Chen, John K. Lai, David C. Parkes, and Ariel D. Procaccia. Truth, justice, and cake cutting. *Games and Economic Behavior*, 77(1):284–297, 2013.

[45] Yukun Cheng and Sanming Zhou. *A Survey on Approximation Mechanism Design Without Money for Facility Games*, pages 117–128. Springer International Publishing, 2015.

[46] Maurice Cheung and David B. Shmoys. A primal-dual approximation algorithm for min-sum single-machine scheduling problems. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings*, volume 6845 of *Lecture Notes in Computer Science*, pages 135–146. Springer, 2011.

[47] Yuan Shih Chow, Herbert Robbins, and David Siegmund. *Great expectations: The theory of optimal stopping.* Houghton Mifflin Boston, 1971.

[48] Yuga J. Cohler, John K. Lai, David C. Parkes, and Ariel D. Procaccia. Optimal envy-free cake cutting. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence.* pages 626—631, 2011.

[49] Hervé Crés and Hervé Moulin. Scheduling with opting out: Improving upon random priority. *Operations Research*, 49(4):pp. 565–577, 2001.

[50] Vladimir I Danilov. The structure of non-manipulable social choice rules on a tree. *Mathematical Social Sciences*, 27(2):123–131, 1994.

[51] James Davis, Guillermo Gallego, and Huseyin Topaloglu. Assortment planning under the multinomial logit model with totally unimodular constraint structures, 2013. Technical Report.

[52] Antoine Désir and Vineet Goyal. Near-optimal algorithms for capacity constrained assortment optimization, 2014. Available at SSRN 2543309.

[53] Antoine Désir, Vineet Goyal, Danny Segev, and Chun Ye. Capacity constrained assortment optimization under the markov chain choice model, 2016. under review at Operations Research.

[54] Nikhil Devanur, Christos H. Papadimitriou, Amin Saberi, and Vijay Vazirani. Market equilibrium via a primal–dual algorithm for a convex program. *Journal of the ACM*, 55, 2008.

[55] Robert J. Dolan. Incentive mechanisms for priority queuing problems. *The Bell Journal of Economics*, 9(2):pp. 421–436, 1978.

[56] U. Feige and J. Vondrák. Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 667–676, Oct 2006.

[57] Itai Feigenbaum, Jay Sethuraman, and Chun Ye. Approximately optimal mechanisms for strategyproof facility location: Minimizing lp norm of costs. *to appear in Mathematics of Operations Research*, 2016.

[58] Jacob B Feldman and Huseyin Topaloglu. Revenue management under the markov chain choice model, 2014.

[59] Jacob B Feldman and Huseyin Topaloglu. Capacity constraints across nests in assortment optimization under the nested logit model. *forthcoming in Operations Research*, 2016.

[60] Michal Feldman and Yoav Wilf. Strategyproof facility location and the least squares objective. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, EC '13, pages 873–890, New York, NY, USA, 2013. ACM.

[61] Lisa Fleischer, Michel X. Goemans, Vahab S. Mirrokni, and Maxim Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *SODA*, pages 611–620. ACM Press, 2006.

[62] Guillermo Gallego, Richard Ratliff, and Sergey Shebalov. A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research*, 63(1):212–232, 2015.

[63] Guillermo Gallego and Huseyin Topaloglu. Constrained assortment optimization for the nested logit model. *Management Science*, 60(10):2583–2601, 2014.

[64] Giorgio Gallo, Michael Grigoriadis, and Robert Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18:30–55, 1989.

[65] Iftah Gamzu and Danny Segev. A polynomial-time approximation scheme for the airplane refueling problem. *CoRR*, abs/1512.06353, 2015.

[66] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[67] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *In Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, pages 323–336, 2011.

[68] Allan Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[69] Pierre Hansen and Jacques-Franois Thisse. Outcomes of voting and planning. *Journal of Public Economics*, 16(1):1 – 15, 1981.

[70] Jeff Hartline and Alexa Sharp. An incremental model for combinatorial maximization problems. In Carme Àlvarez and Maria J. Serna, editors, *WEA*, volume 4007 of *Lecture Notes in Computer Science*, pages 36–48. Springer, 2006.

[71] Jeffrey Hartline. *Incremental Optimization*. PhD thesis, Cornell University, 2008.

[72] Tadashi Hashimoto, Daisuke Hirata, Onur Kesten, Morimitsu Kurino, and M. Utku Unver. Two axiomatic approaches to the probabilistic serial mechanism. *Theoretical Economics*, 9(1), January 2014.

[73] Refael Hassin and Moshe Haviv. *To queue or not to queue : equilibrium behavior in queuing systems*. International series in operations research and management science. Kluwer Academic Publ., Boston, Dordrecht, 2002.

[74] Wiebke Hóhn. *Complex Single Machine Scheduling: Theoretical and Practical Aspects of Sequencing*. PhD thesis, TU Berlin, 2014.

[75] Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):pp. 293–314, 1979.

[76] Akshay-Kumar Katta and Jay Sethuraman. A solution to the random assignment problem on the full preference domain. *Journal of Economic Theory*, 131(1):231–250, November 2006.

[77] Fuhito Kojima. Random assignment of multiple indivisible objects. *Mathematical Social Sciences*, 57(1):134 – 142, 2009.

[78] Fuhito Kojima and Mihai Manea. Incentives in the probabilistic serial mechanism. *Journal of Economic Theory*, 145(1):106 – 123, 2010.

[79] A. Gürhan Kök and Marshall L. Fisher. Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research*, 55(6):1001–1021, 2007.

[80] Ariel Kulik, Hadas Shachnai, and Tami Tamir. Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Mathematics of Operations Research*, 38(4):729–739, 2013.

[81] David Kurokawa, Ariel D. Procaccia, and Nisarg Shah. Leximin allocations in the real world. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, pages 345–362, New York, NY, USA, 2015. ACM.

[82] Martine Labbe. First euro summer institute outcomes of voting and planning in single facility location problems. *European Journal of Operational Research*, 20(3):299 – 313, 1985.

[83] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4):795–806, 2010.

[84] Avishay Maya and Noam Nisan. Incentive compatible two player cake cutting. In *Proceedings of the 8th International Workshop on Internet and Network Economics (WINE)*, pages 170–183, 2012.

[85] Daniel McFadden, Kenneth Train, et al. Mixed mnl models for discrete response. *Journal of applied Econometrics*, 15(5):447–470, 2000.

[86] Nicole Megow and José Verschae. *Automata, Languages, and Programming: 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, chapter Dual Techniques for Scheduling on a Machine with Varying Speed, pages 745–756. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[87] Haim Mendelson. Pricing computer services: Queueing effects. *Commun. ACM*, 28(3):312–321, March 1985.

[88] Haim Mendelson and Seungjin Whang. Optimal incentive-compatible priority pricing for the m/m/1 queue. *Operations Research*, 38(5):pp. 870–883, 1990.

[89] Elchanan Mossel and Omer Tamuz. Truthful fair division. In S. Kontogiannis, E. Koutsoupias, and P. Spirakis, editors, *Proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT) volume 6386 of* Lecture Notes in Computer Science (LNCS)., pages 288–199. Springer-Verlag, 2010.

[90] Hervé Moulin. On strategy-proofness and single-peakedness. *Public Choice*, 35:437–455, 1980.

[91] Hervé Moulin. *Fair Division and Collective Welfare*. The MIT Press, 2003.

[92] Hervé Moulin. One dimensional mechanism design. *forthcoming in Theoretical Economics*, 2016.

[93] P. Naor. The regulation of queue size by levying tolls. *Econometrica*, 37(1):pp. 15–24, 1969.

[94] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

[95] Antonio Nicoló and Yan Yu. Strategic divide and choose. *Games and Economic Behavior*, 646(1):268–289, 2008.

[96] Ariel D. Procaccia. Cake cutting: Not just child's play. *Communications of the ACM*, 56(7):78–87, 2013.

[97] Ariel D. Procaccia. Cake cutting algorithms. *Chapter 13 of the Handbook of Computational Social Choice*, 2016.

[98] Ariel D Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. *ACM Transactions on Economics and Computation*, 1(4):18, 2013.

[99] Marek Pycia. Assignment with multiple-unit demands and responsive preferences, 2011. Manuscript.

[100] J. H. Reijnierse and J. A. M. Potters. On finding an envy-free pareto-optimal division. *Mathematical Programming*, 83:291–311, 1998.

[101] Jack M. Robertson and William A. Webb. *Cake Cutting Algorithms: Be Fair If You Can*. A. K. Peters, 1998.

[102] Alvin E. Roth. Incentive compatibility in a market with indivisible goods. *Economics Letters*, 9(2):127 – 132, 1982.

[103] Alvin E. Roth, Tayfun Sonmez, and M. Utku Unver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188, December 2005.

[104] Paat Rusmevichientong, Zuo-Jun Max Shen, and David B Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680, 2010.

[105] Daniel Saban and Jay Sethuraman. The complexity of computing the random priority allocation matrix. *Mathematics of Operations Research*, 40(4):1005–1014, 2015.

[106] 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.

[107] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, 1986.

[108] James Schummer. Strategy-proofness versus efficiency on restricted domains of exchange economies. *Social Choice and Welfare*, 14:47–56, 1997.

[109] James Schummer and Rakesh V Vohra. Strategy-proof location on a network. *Journal of Economic Theory*, 104(2):405–428, 2002.

[110] Petra Schuurman and Gerhard J. Woeginger. Approximation schemes – a tutorial, 2007.

[111] Jay Sethuraman and Chun Ye. A note on the assignment problem with uniform preferences. *Operations Research Letters*, 43(3):283 – 287, 2015.

[112] Jay Sethuraman and Chun Ye. A generalization of the probabilistic serial mechanism and its relationship to the leximin allocation, 2016. Working paper.

[113] Alexa Sharp. *Incremental Algorithms: Solving Problems in a Changing World.* PhD thesis, Cornell University, 2007.

[114] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993.

[115] Isaac Sonin. The state reduction and related algorithms and their applications to the study of markov chains, graph theory, and the optimal stopping problem. *Advances in Mathematics*, 145(2):159–188, 1999.

[116] Tayfun Sönmez and Utku Unver. Matching, allocation, and exchange of discrete resources. Boston College Working Papers in Economics 717, Boston College Department of Economics, 2009.

[117] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[118] Walter Stromquist. How to cut a cake fairly. *American Mathematical Monthly*, 87(8):640–644, 1980.

[119] Francis E. Su. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly*, 10:930–942, 1999.

[120] Jeroen Suijs. On incentive compatibility and budget balancedness in public decision making. *Economic design*, 2(1):193–209, 1996.

[121] Lars-Gunnar Svensson. Queue allocation of indivisible goods. *Social Choice and Welfare*, 11(4):323–330, 1994.

[122] K. Talluri and G. Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.

[123] Yuan Tian. Strategy-proof and efficient offline interval scheduling and cake cutting. In Y. Chen and N. Immorlica, editors, *Proc. of 9th WINE*, pages 436–437. editors, LNCS, 2013.

[124] Henk C Tijms. *A first course in stochastic models.* John Wiley and Sons, 2003.

[125] Vijay V. Vazirani. *Approximation Algorithms.* Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[126] Vijay V. Vazirani. chapter 10: Combinatorial algorithms for market equilibria. In N. Nisan and T. Roughgarden, editors, *Algorithmic Game Theory*, pages 103–134. Cambridge University Press, 2007.

[127] John Von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *In Contributions to the Theory of Games*, 2:512, 1953.

[128] Jens Vygen. *Approximation algorithms for facility location problems.* Lecture Notes, University of Bonn., 2005.

[129] H.C.W.L. Williams. On the formation of travel demand models and economic evaluation measures of user benefit. *Environment and Planning A*, 3(9):285–344, 1977.

[130] Dan Zhang and William L Cooper. Revenue management for parallel flights with customer-choice behavior. *Operations Research*, 53(3):415–431, 2005.

[131] Lin Zhou. On a conjecture by gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123 – 135, 1990.

[132] Roie Zivan, Miroslav Dudík, Steven Okamoto, and Katia Sycara. Reducing untruthful manipulation in envy-free pareto optimal resource allocation. In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pages 391–398, 2010.

Part II

Appendices

Appendix A

Cake Cutting Algorithms for Piecewise Constant and Piecewise Uniform Valuations

Proof of Proposition 3.2

We begin with some notations. Let $len(X)$ denote the length of $X \subseteq [0, 1]$. Since the value density function is piecewise uniform, it suffices to consider the length of pieces of the cake that are desired by each agent.

Let $S \subseteq N$ be a coalition of agents who misreport their value density function.

Let I denote the instance where every agent reports truthfully and I' denote the instance where agents in S misreport.

Let $D_1, \ldots, D_n \subseteq [0, 1]$ denote the pieces of cake that are truly desired by each agent.

Let $D'_1, \ldots, D'_n \subseteq [0, 1]$ denote the desired pieces of cake that are reported by each agent. In other words, $D'_i = D_i$ if and only if $i \notin S$.

Let $X_1, \ldots, X_n \subseteq [0, 1]$ denote the allocation received by each agent under truthful reports.

Let $X'_1, \ldots, X'_n \subseteq [0, 1]$ denote the allocation received by each agent when the agents in S misreport.

Let $B_1, \ldots, B_k$ be the bottleneck sets of agents arranged in the order that they are being allocated by the algorithm in instance I.

Let $B'_1, \ldots, B'_p$ be the bottleneck sets of agents arranged in the order that they are being allocated by the algorithm in instance I'.

Moreover, since every agent belonging to the same bottleneck set receives an allocation of the same length under CCEA for this special case. We let $len(B_i)$ denote the length of the allocation each agent receives in the bottleneck set B_i. Let

$$B_l^+ = \{i \in B_l \mid len(X_i' \cap D_i) \geq len(X_i \cap D_i) = len(X_i)\}^{52},$$

and

$$B_l^- = \{i \in B_l \mid len(X_i' \cap D_i) \leq len(X_i)\}.$$

In other words B_l^+ is the subset of agents of B_l who weakly gain in utility when the agents in S misreport, and B_l^- is the subset of agents of B_l who weakly lose in utility when the agents in S misreport. We will show that for all $l = 1, \ldots, k$, $B_l = B_l^-$. This would then directly imply that no one in coalition S can strictly benefit by misreporting.

We will prove this result via induction on l. In order to carry on with the induction, we will show that no agent in B_1 appears in the coalition S. We begin with a lemma.

Lemma A.1. *For every agent i, $len(X_i' \cap D_i) \geq len(B_1)$.*

Proof. Suppose not, let i be an agent such that $len(X_i' \cap D_i) < len(B_1)$. It must be the case that i reported his valuation truthfully. Consequently, the following set of inequalities hold for agent i:

$$len(X_i') = len(X_i' \cap D_i') = len(X_i' \cap D_i) < len(B_1),$$

where the first equality follows the free disposal assumption. The second equality follows from $D_i = D_i'$. Let B_l' be the bottleneck set that i belongs to in the instance I'. Then we have $len(B_1') \leq len(B_l') = len(X_i') < len(B_1)$. This is because, the length of allocation of agents is non-decreasing with respect to the index of bottleneck sets. We refer the readers to Lemma 3.4 of [44] for a proof of this fact. It is clear that B_1' cannot contain an agent who misreports in I', since a misreporting agent in B_1' only receives a piece of cake with length $len(B_1') < len(B_1)$, which is strictly less than what he would've gotten had he reported truthfully. Hence, every agent in B_1' must report his true preference in I'.

[52] The fact that $len(X_i \cap D_i) = len(X_i)$ makes use of the free disposal property, i.e. the allocation that the mechanism gives agent i is a subset of the pieces desired by agent i under truthful reports.

On the other hand, since B_1' is the first bottleneck set in I', we have that

$$\begin{aligned} len(B_1') &= \frac{len((\cup_{j \in B_1'} D_j') \cap [0,1])}{|B_1'|} = \frac{len((\cup_{j \in B_1'} D_j) \cap [0,1])}{|B_1'|} \\ &\geq \frac{len((\cup_{j \in B_1} D_j) \cap [0,1])}{|B_1|} = len(B_1), \end{aligned}$$

which leads to a contradiction. □

Lemma A.2. *It is the case that $B_1 = B_1^-$. In other words, no agent in B_1 is strictly better off when some subset of agents misreport their preference.*

Proof. Suppose not, then there exists some $j \in X_1$ such that $len(X_j' \cap D_j) > len(X_j)$. We also established in the previous lemma that $len(X_i' \cap D_i) \geq len(B_1) = len(X_i)$ for all $i \in B_1$. Summing over $i \in B_1$, we get that

$$len(\cup_{i \in B_1}(X_i' \cap D_i)) = \sum_{i \in B_1} len(X_i' \cap D_i) > \sum_{i \in B_1} len(X_i) = len(\cup_{i \in B_1} X_i) = len(\cup_{i \in B_1} D_i),$$

where the first two equalities follow from the fact that the X_i's and $X_i' \cap D_i$'s are disjoint subsets and the third equality follows from the way the algorithm allocates to the agents in the smallest bottleneck set. But this set of inequalities contradict the fact that $\cup_{i \in B_1}(X_i' \cap D_i) \subseteq \cup_{i \in B_1} D_i$, which implies that $len(\cup_{i \in B_1}(X_i' \cap D_i)) \leq len(\cup_{i \in B_1} D_i)$. Hence, it must be the case that for every $i \in B_1$, we have that $len(X_i' \cap D_i) = len(X_i)$, which implies that $i \in B_1^-$. □

Lemma A.3. *No agent in B_1 appears in the coalition S and B_1 is also the first bottleneck set for I'.*

Proof. By the previous lemma, no agent in B_1 is strictly better off by misreporting his preference. Thus, if any agent in B_1 is in S, then he makes himself no worse off and simultaneously make some other agent in S strictly better off. Let's examine the collective allocation of agents in B_1 in the instance I'. We get that

$$len(\cup_{i \in B_1} X_i') \geq len(\cup_{i \in B_1}(X_i' \cap D_i)) \geq len(\cup_{i \in B_1} X_i),$$

where the last inequality follows from Lemma 1. This implies that the agents in B_1 collectively obtain an allocation that is no smaller than the allocation they would get had they reported truthfully. Thus, having a subset of agents in X_1 misreport will not benefit the other agents in the coalition S. Hence, without lost of generality, we may assume that no agent in B_1 appears in the coalition S. Provided that every agent in B_1 also reports truthfully in I', there is no incentive for an agent that belongs to a subsequent bottleneck set in I to misreport and prevent B_1 from being the first bottleneck set in I' since that would make the misreporting agent strictly worse off.[53]

□

Since no agent in B_1 appears in the coalition S and B_1 is also the first bottleneck set for I', we can remove B_1 from N and $\cup_{i \in B_1} X_i$ from $[0,1]$ and induct on the set of remaining agents $N \backslash B_1$ and the remaining piece of cake $[0,1] \backslash \cup_{i \in B_1} X_i$ to be allocated. The proof is complete by invoking the inductive hypothesis with B_2 being the first bottleneck set in the new instance.

Proof of Proposition 3.4 The first impossibility result assumes that the algorithm disposes the intervals desired by no agent. Consider the following two agent profiles.

Profile 1:

$$v_1(x) = 1 \text{ if } x \in [0, 0.2],\ v_1(x) = 0 \text{ if } x \in (0.2, 1]$$
$$v_2(x) = 0 \text{ if } x \in [0, 0.6],\ v_2(x) = 1 \text{ if } x \in (0.6, 1]$$

The interval $(0,2,0.6]$ is discarded because no agent desires it. The uniform allocation rule gives us the allocation:

$$X_1 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}(0.6, 1]$$
$$X_2 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}(0.6, 1]$$

where $p[a, b]$ for some $0 \leq p \leq 1$ denotes a subinterval of $[a, b]$ with length p times that of $[a, b]$. Let $A \subset (0.6, 1]$ be the allocation that agent 2 receives in this case.

Now consider profile 2:

$$v_1(x) = 1 \text{ if } x \in [0, 0.2],\ v_1(x) = 0 \text{ if } x \in (0.2, 1]$$

[53] Note that the lexicographical tie breaking rule for bottleneck sets is needed in this case.

$$v_2(x) = 0 \text{ if } x \in [0,1]\backslash A,\ v_2(x) = 1 \text{ if } x \in A$$

In this case, all intervals other than $[0, 0.2]$ and A are discarded because no agent desires them.The uniform allocation rule gives us the allocation:

$$X_1 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}A$$
$$X_2 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}A$$

Hence, agent 2 in profile 2 would misreport so that the reported profile is profile 1.

The second impossibility result assumes that the algorithm does not dispose the intervals desired by no agent. Consider the following two agent profiles.

Profile 1:

$$v_1(x) = 1 \text{ if } x \in [0, 0.2],\ v_1(x) = 0 \text{ if } x \in (0.2, 1]$$

$$v_2(x) = 1 \text{ if } x \in [0, 0.2],\ v_2(x) = 0 \text{ if } x \in (0.2, 1]$$

The uniform allocation rule gives us the allocation:

$$X_1 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}(0.2, 1]$$
$$X_2 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}(0.2, 1]$$

Let $A \subset (0.2, 1]$ be the allocation that agent 1 receives in this case.

Now consider profile 2:

$$v_1(x) = 1 \text{ if } x \in [0, 0.2] \cup A,\ v_1(x) = 0 \text{ otherwise}$$

$$v_2(x) = 1 \text{ if } x \in [0, 0.2],\ v_2(x) = 0 \text{ if } x \in (0.2, 1]$$

The uniform allocation rule gives us the allocation:

$$X_1 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}A \cup \frac{1}{2}(0.2, 1]\backslash A$$
$$X_2 = \frac{1}{2}[0, 0.2] \cup \frac{1}{2}A \cup \frac{1}{2}(0.2, 1]\backslash A$$

Hence, agent 1 in profile 2 would misreport so that the reported profile is profile 1.

Proof of Proposition 3.5

We first prove that MCSD is well-defined and results in a feasible allocation in which each agent gets $1/n$ size of the cake. Let $\mathcal{J}' = \{J_1, \ldots, J_\ell\}$ be a partitioning of the interval $[0, 1]$ induced by the discontinuity points in agent valuations and the cake cuts in the $n!$ cake allocations. We make a couple of claims about $\mathcal{J}'$ that following from the way $\mathcal{J}'$ is constructed.

Claim A.1. *An agent is completely indifferent over each subinterval in $\mathcal{J}'$.*

Claim A.2. *Let X_i^π denote a maximum preference cake piece of size $1/n$ chosen by agent i in the serial order π. For each $J \in \mathcal{J}'$ either X_i^π contains J completely or it does not contain any part of J.*

Now consider a matrix of dimension $n! \times \ell$: $B = (b_{ij})$ such that $b_{ij} = 1$ if $J_j \subset X_i^\pi$ and $b_{ij} = 0$ if $J_j \not\subset X_i^\pi$. Since for each $\pi \in \Pi^N$, each agent $i \in N$ gets $1/n$ of the cake in X_i^π, then it follows that $\sum_{i=1}^{n!} \sum_{j=1}^{\ell} b_{ij} len(J_j) = n!/n$. Hence,

$$\sum_{j=1}^{\ell} \sum_{i=1}^{n!} \frac{b_{ij} len(J_j)}{n!} = \frac{1}{n}.$$

Also consider a matrix of dimension $n \times \ell$: $P = (p_{ij})$ such that p_{ij} denotes the fraction of J_j that agent i gets in Y_i. From the algorithm MCSD, we know that $p_{ij} = \frac{count(i,J_j)}{n!}$ where $count(i, J_j)$ is the number of permutations in which i gets J_j. It is immediately seen that each column sums up to 1. Hence each J_j is complete allocated to the agents. We now prove that each agent gets a total cake piece of size $1/n$. We do so by showing that $\sum_{j=1}^{\ell} p_{ij} len(J_j) = 1/n$.

$$\begin{aligned} 1/n &= \sum_{j=1}^{\ell} \sum_{i=1}^{n!} b_{ij} len(J_j)/n! = \sum_{j=1}^{\ell} (\sum_{i=1}^{n!} b_{ij}) len(J_j)/n! = \sum_{j=1}^{\ell} (count(i, J_j)) len(J_j)/n! \\ &= \sum_{j=1}^{\ell} (\frac{count(i, J_j)}{n!}) len(J_j) = \sum_{j=1}^{\ell} p_{ij} len(J_j). \end{aligned}$$

Hence $X = (X_1, \ldots, X_n)$ the allocation returned by MCSD is a proper allocation of the cake in which each agent gets a total cake piece of size $1/n$.

Proof of Lemma 3.1

Define $\bar{v}_i = v_i$ for $a \leq x \leq b$ and $\bar{v}_i(x) = \bar{v}_i(x + (b-a))$ recursively for x outside of $[a, b]$ (i.e. replicating the function v_i on $[a, b]$). Since $\bar{v}_i$ is periodic, then it suffices to show that

$$E_U[V_i(A)] = \int_a^b \int_x^{x+\alpha(b-a)} \bar{v}_i(y)dy \frac{1}{b-a} dx = \alpha \int_a^b \bar{v}_i(x)dx = \alpha V_i([a,b]).$$

Since $\bar{v}_i$ is periodic, we have that

$$\frac{1}{b-a} \int_a^b \int_x^{x+\alpha(b-a)} \bar{v}_i(y)dydx = \frac{1}{b-a} \int_a^b \bar{v}_i(y) \int_{y-\alpha(b-a)}^{y} dxdy = \alpha \int_a^b \bar{v}_i(x)dx,$$

which proves the lemma.

Proof of Proposition 3.8

$$a = [0, 0.25], b = (0.25, 0.5], c = (0.5, 0.75], d = (0.75, 1]$$

Consider the following two profiles of valuations.

Profile 1:

$$v_1(x) = 4 \text{ if } x \in a,\ v_1(x) = 3 \text{ if } x \in b,\ v_1(x) = 2 \text{ if } x \in c,\ v_1(x) = 1 \text{ if } x \in d$$

$$v_2(x) = 3 \text{ if } x \in a,\ v_2(x) = 4 \text{ if } x \in b,\ v_2(x) = 1 \text{ if } x \in c,\ v_2(x) = 2 \text{ if } x \in d$$

Running MCSD gives us:

$$X_1 = \frac{1}{2}a \cup \frac{1}{2}b \cup \frac{1}{2}c \cup \frac{1}{2}d$$
$$X_2 = \frac{1}{2}a \cup \frac{1}{2}b \cup \frac{1}{2}c \cup \frac{1}{2}d$$

Profile 2:

$$v_1(x) = 4 \text{ if } x \in a,\ v_1(x) = 2 \text{ if } x \in b,\ v_1(x) = 3 \text{ if } x \in c,\ v_1(x) = 1 \text{ if } x \in d$$

$$v_2(x) = 2 \text{ if } x \in a,\ v_2(x) = 4 \text{ if } x \in b,\ v_2(x) = 1 \text{ if } x \in c,\ v_2(x) = 3 \text{ if } x \in d$$

Running MCSD gives us the allocation:

$$X_1 = a \cup c$$

$$X_2 = b \cup d$$

Note that agents with true valuation in profile 1 would misreport together to profile 2, which gives them a higher utility of 1.5 each as opposed to 1.25 had they reported truthfully. This means that MCSD is not group strategyproof even for 2 agents. MCSD is not strategyproof because its allocation in profile 1 is Pareto dominated by its allocation in profile 2.

Proof of Proposition 3.9

Consider a profile of three agents, each with piecewise uniform valuation function.

$$v_1(x) = 1.5 \text{ if } x \in [0, 2/3],\ 0 \text{ otherwise}$$

$$v_2(x) = 1.5 \text{ if } x \in [0, 1/3] \cup (2/3, 1],\ 0 \text{ otherwise}$$

$$v_3(x) = 1.5 \text{ if } x \in (1/3, 1],\ 0 \text{ otherwise}$$

Let $a = [0, 1/3]$, $b = (1/3, 2/3]$, $c = (2/3, 1]$.

We adopt the following implementation of MCSD: when it is agent i's turn to pick, out of the pieces of the remaining cake that he likes, he takes the *left-most* such piece with length 1/n, where n is the number of agents.

If the priority ordering were $1, 2, 3$, then a feasible assignment that respects the preferences is 1 $\leftarrow$ a, 2 $\leftarrow$ c, 3 $\leftarrow$ b.

If the priority ordering were 1,3,2, then a feasible assignment that respects the preferences is $1 \leftarrow a$, $3 \leftarrow b$, $2 \leftarrow c$.

If the priority ordering were $2, 1, 3$, then a feasible assignment that respects the preferences is $2 \leftarrow a$, $1 \leftarrow b$, $3 \leftarrow c$.

If the priority ordering were $2, 3, 1$, then a feasible assignment that respects the preferences is $2 \leftarrow a$, $3 \leftarrow b$, $1 \leftarrow c$.

If the priority ordering were $3, 1, 2$, then a feasible assignment that respects the preferences is $3 \leftarrow b$, $1 \leftarrow a$, $2 \leftarrow c$.

If the priority ordering were $3, 2, 1$, then a feasible assignment that respects the preferences is $3 \leftarrow b$, $2 \leftarrow a$, $1 \leftarrow c$.

Then, the MCSD allocation is as follows.

$$\begin{aligned}
X_1 = &\quad \frac{1}{2}[0, 1/3] \cup \frac{1}{6}(1/3, 2/3] \cup \frac{1}{3}(2/3, 1] \\
X_2 = &\quad \frac{1}{2}[0, 1/3] \cup \frac{1}{2}(2/3, 1] \\
X_3 = &\quad \frac{5}{6}(1/3, 2/3] \cup \frac{1}{6}(2/3, 1]
\end{aligned}$$

Clearly, agent 1 envies agent 3 in this case.

Appendix B

Approximately Optimal Mechanisms for Strategyproof Facility Location: Minimizing L_p Norm of Costs

B.1 An Alternative Definition of Individual Cost

Let g be a strictly increasing and convex $\mathcal{C}_1$ function on $[0, \infty)$ with $g(0) = g'(0) = 0$. Note that $g(x) = x^p$ satisfies this description for all $p > 1$. We consider a scenario where the cost of agent i is $C(x_i, y) = g(|x_i - y|)$ when the mechanism is deterministic and locates the facility at y. Similarly $C(x_i, \pi) = \mathbb{E}_{y \sim \pi}[g(|x_i - y|)]$ when the mechanism is randomized and locates the facility according to distribution π. The social cost function $h(|x_1 - y|, |x_2 - y|)$ is only assumed to be (1) anonymous ($h(d, d') = h(d', d)$) and (2) satisfy that for all $a \in (\min\{x_1, x_2\}, \max\{x_1, x_2\})$ where $x_1 \neq x_2$, $h(|x_1 - a|) + h(|x_2 - a|) < h(|x_2 - x_1|)$. Note that for $p > 1$, the L_p norm of the distances and the L_p norm of the costs (for the general g above) both satisfy these conditions. We show that in this case, no randomized strategyproof mechanism satisfying shift invariance, scale invariance and ex-post Pareto efficiency for $n = 2$ can help us improve the approximation ratio relatively to the median mechanism.

Theorem B.1. *Let f be a randomized mechanism satisfying shift invariance and scale invariance, and ex-post Pareto efficiency for $n = 2$. Assume f is strategyproof with respect to the individual cost function $C(x_i, y) = g(|x_i - y|)$, where g is a strictly increasing and convex $\mathcal{C}_1$ function on $[0, \infty)$ with*

$g(0) = g'(0) = 0$. *If the social cost function satisfies (1) and (2), then the approximation ratio of f is at least as large as the median's.*

Proof. Using a proof similar to that of Lemma 4, we may assume without loss of generality that f is symmetric. Consider a profile where $n = 2$ and $x_1 = 0$, $x_2 = 1$. Let $Y = f(0,1)$. We would like that $\mathbb{P}(Y \in (0,1)) = 0$. Suppose for the sake of contradiction that there there exists $x \in (0, \frac{1}{2})$ such that $\mathbb{P}(Y \in (x, 1-x)) = q > 0$. Now suppose agent 2 now misreports his location to $1+\epsilon$ for some small $\epsilon > 0$ such that $\frac{1}{1+\epsilon} > 1 - x$. By shift and scale invariance, $f(0, 1+\epsilon) = (1+\epsilon)Y$. Then the difference in cost for agent 2 between the two profile of reports is

$$\begin{aligned}
\mathbb{E}[g\big(|1-(1+\epsilon)Y|\big)] - \mathbb{E}[g\big(|1-Y|\big)] &= -\int_0^{\frac{1}{1+\epsilon}} (g(1-y) - g(1-(1+\epsilon)y))dF(y) \\
&+ \int_{\frac{1}{1+\epsilon}}^1 (g((1+\epsilon)y - 1) - g(1-y))dF(y) \\
&\leq \mathbb{P}(Y \in [\frac{1}{1+\epsilon}, 1])g(\epsilon) - q\Big(g(1-x^*) - g(1-(1+\epsilon)x^*)\Big)
\end{aligned}$$

where $x^* \in \arg\min_{y \in [x, 1-x]} g(1-y) - g(1-(1+\epsilon)y)$. The inequality follows from the fact that $g((1+\epsilon)y-1) - g(1-y) \leq g(\epsilon)$ for all $y \in [\frac{1}{1+\epsilon}, 1]$ and that $g(1-y) - g(1-(1+\epsilon)y) \geq g(1-x^*) - g(1-(1+\epsilon)x^*)$ for all $y \in [x, 1-x]$. Note that

$$\begin{aligned}
\lim_{\epsilon \to 0^+} \frac{\mathbb{E}[g\big(|1-(1+\epsilon)Y|\big)] - \mathbb{E}[g\big(|1-Y|\big)]}{\epsilon} &\leq \lim_{\epsilon \to 0^+} \mathbb{P}(Y \in [\frac{1}{1+\epsilon}, 1])\frac{g(\epsilon)}{\epsilon} - q\frac{g(1-x^*) - g(1-(1+\epsilon)x^*)}{\epsilon} \\
&\leq \mathbb{P}(Y = 1)g'(0) - qg'(1-x^*)x^* < 0
\end{aligned}$$

The third inequality follows from $g'(0) = 0$ and $g'(1-x^*) > 0$ (since g is strictly convex). This implies that $\mathbb{E}[g\big(|1-(1+\epsilon)Y|\big)] - \mathbb{E}[g\big(|1-Y|\big)] < 0$ for ϵ sufficiently small, implying that there is a profitable deviation for agent 2. □

B.2 Omitted Proofs from Section 4.2

Proof of Lemma 6. First, let us prove that the two conditions imply strategyproofness. By shift invariance and anonymity, it suffices to check strategyproofness for profiles where $x_1 = 0$ and $x_2 \geq 0$. Moreover, any scale invariant mechanism is trivially strategyproof with respect to the profile $(0,0)$ since scale invariance implies $f(0,0) = 0$, which means that no agent has incentive to misreport his location.[54] Thus, we can assume that $x_2 > 0$. It suffices to show that agent 2 cannot benefit by

[54] $f(0,0) = 0$ follows from, say, $f(0,0) = f(0 \cdot 1, 0 \cdot 1) = 0 \cdot f(1,1) = 0$, where the second equality is by scale invariance.

deviating from his true location if the two aforementioned conditions hold. Since $x_2 > 0$, we can denote agent 2's deviation x_2' as cx_2 for some $c \in \mathbb{R}$. Moreover, we can assume that $c \geq 0$. This can be justified as follows. Assume $c < 0$. Note that by symmetry, in any fixed profile $\mathbf{z}$, the closer a point is to $m_{\mathbf{z}}$, the smaller the expected distance of the facility is from that point. In particular, this implies that $C(x_2, f(0, -cx_2)) \leq C(-x_2, f(0, -cx_2))$. But also note that by scale invariance, $C(-x_2, f(0, -cx_2)) = C(x_2, f(0, cx_2))$. Thus, $C(x_2, f(0, -cx_2)) \leq C(x_2, f(0, cx_2))$. Consequently, if reporting cx_2 is a profitable deviation for agent 2 for some $c < 0$, then reporting $-cx_2$ is also a profitable deviation for the agent.

When agent 2 reports his location to be cx_2, where $c > 1$, the change in cost incurred by agent 2 is (where C_{orig} is the expected cost of agent 2 under truthful reporting and C_{dev} is the expected cost of agent 2 under misreporting):

$$
\begin{aligned}
C_{dev} - C_{orig} &= -(c-1) \int_{(-\infty, \frac{x_2}{c})} y dF(y) + \int_{[\frac{x_2}{c}, x_2)} ((c+1)y - 2x_2) dF(y) + (c-1) \int_{(x_2, \infty)} y dF(y) + \\
&+ (c-1)x_2 \mathbb{P}(Y = x_2) \\
&= -(c-1) \int_{(-\infty, x_2)} y dF(y) + \int_{[\frac{x_2}{c}, x_2)} (2cy - 2x_2) dF(y) + (c-1) \int_{(x_2, \infty)} y dF(y) \\
&+ (c-1)x_2 \mathbb{P}(Y = x_2) \\
&\geq -(c-1) \int_{(-\infty, x_2)} y dF(y) + (c-1) \int_{(x_2, \infty)} y dF(y) + (c-1)x_2 \mathbb{P}(Y = x_2).
\end{aligned}
$$

Hence, when condition 1 holds, we have that $-(c-1)\int_{(-\infty,x_2)} y dF(y) + (c-1)\int_{(x_2,\infty)} yF(y) + (c-1)x_2\mathbb{P}(Y = x_2) \geq 0$, which means that $C_{dev} - C_{orig} \geq 0$.

Similarly, when $0 \leq c < 1$, the change in cost incurred by agent 2 is:

$$
\begin{aligned}
C_{dev} - C_{orig} &= (1-c) \int_{(-\infty, x_2)} y dF(y) + \int_{(x_2, \frac{x_2}{c}]} (2x_2 - (c+1)y) dF(y) - (1-c) \int_{(\frac{x_2}{c}, \infty)} y dF(y) + \\
&+ (1-c)x_2 \mathbb{P}(Y = x_2) \\
&= (1-c) \int_{(-\infty, x_2)} y dF(y) + \int_{(x_2, \frac{x_2}{c}]} (2x_2 - 2cy) dF(y) - (1-c) \int_{(x_2, \infty)} y dF(y) \\
&+ (1-c)x_2 \mathbb{P}(Y = x_2) \\
&\geq (1-c) \int_{(-\infty, x_2)} y dF(y) - (1-c) \int_{(x_2, \infty)} y dF(y) + (1-c)x_2 \mathbb{P}(Y = x_2).
\end{aligned}
$$

Hence, when condition 2 holds, we have that $(1-c)\int_{(-\infty,x_2)} ydF(y) - (1-c)\int_{(x_2,\infty)} ydF(y) + (1-c)x_2\mathbb{P}(Y=x_2) \geq 0$, which means that $C_{dev} - C_{orig} \geq 0$. Hence, the mechanism is strategyproof for any profile $\mathbf{x}$ with $x_1 = 0 < x_2$.

To prove the other direction, suppose condition 1 does not hold for some profile $\mathbf{x}$ with $x_1 = 0 < x_2$. Then there exists $\epsilon > 0$ small enough such that $-\int_{(-\infty,x_2)} ydF(y) + \int_{(x_2,\infty)} ydF(y) + x_2\mathbb{P}(Y=x_2) \leq -\epsilon$ for some $x_2 > 0$. We choose $c > 1$ s.t. $\mathbb{P}(Y \in [\frac{x_2}{c}, x_2)) < \frac{\epsilon}{4x_2}$, then we have that

$$\begin{aligned} C_{dev} - C_{orig} &= -(c-1)\int_{(-\infty,x_2)} ydF(y) + \int_{[\frac{x_2}{c},x_2)} (2cy - 2x_2)dF(y) + (c-1)\int_{(x_2,\infty)} ydF(y) \\ &\quad + (c-1)x_2\mathbb{P}(Y=x_2) \\ &\leq (c-1)(-\int_{(-\infty,x_2)} ydF(y) + \int_{[\frac{x_2}{c},x_2)} (2x_2)dF(y) + \int_{(x_2,\infty)} ydF(y) + x_2\mathbb{P}(Y=x_2)) \\ &< -(c-1)\frac{\epsilon}{2} < 0, \end{aligned}$$

which contradicts strategyproofness of the mechanism.

Similarly, suppose condition 2 does not hold for some profile $\mathbf{x}$ with $x_1 = 0 < x_2$. Then there exists $\epsilon > 0$ small enough such that $\int_{(-\infty,x_2)} ydF(y) - \int_{(x_2,\infty)} ydF(y) + x_2\mathbb{P}(Y=x_2) \leq -\epsilon$ for some $x_2 > 0$. We choose $0 < c < 1$ s.t. $\mathbb{P}(Y \in (x_2, \frac{x_2}{c}])) < \frac{\epsilon}{4x_2}$, then we have that

$$\begin{aligned} C_{dev} - C_{orig} &= (1-c)\int_{(-\infty,x_2)} ydF(y) + \int_{(x_2,\frac{x_2}{c}]} (2x_2 - 2cy)dF(y) - (1-c)\int_{(x_2,\infty)} ydF(y) \\ &\quad + (1-c)x_2\mathbb{P}(Y=x_2) \\ &\leq (1-c)(\int_{(-\infty,x_2)} ydF(y) + \int_{[\frac{x_2}{c},x_2)} (2x_2)dF(y) - \int_{(x_2,\infty)} ydF(y) + x_2\mathbb{P}(Y=x_2)) \\ &< -(1-c)\frac{\epsilon}{2} < 0, \end{aligned}$$

which contradicts strategyproofness of the mechanism. $\square$

Proof of Lemma 7. Let f be as given above. Assume f violates both (1) and (2) on some profile $\mathbf{x}$ (otherwise, there is nothing to prove: we can take $g = f$). By shift invariance we may assume without loss of generality that $x_1 = 0$. We may assume by anonymity and shift invariance that

$x_1 = 0 < x_2$. Let $Y \sim f(\mathbf{x})$. Let $p_1 = \mathbb{P}(Y \in (m_{\mathbf{x}}, x_2)) + \frac{\mathbb{P}(Y=m_{\mathbf{x}})}{2} = \mathbb{P}(Y \in (x_1, m_{\mathbf{x}})) + \frac{\mathbb{P}(Y=m_{\mathbf{x}})}{2}$, $p_2 = \mathbb{P}(x_2, \infty) = \mathbb{P}(-\infty, x_1)$, $z_1 = \frac{\mathbb{E}[Y\mathbb{1}(Y\in(x_1,m_{\mathbf{x}}))]+m_{\mathbf{x}}\mathbb{P}(Y=m_{\mathbf{x}})/2}{p_1}$, and $z_1' = \mathbb{E}[Y|Y \in (-\infty, x_1)]$, $z_2 = \frac{\mathbb{E}[Y\mathbb{1}(Y\in(m_{\mathbf{x}},x_2))]+m_{\mathbf{x}}\mathbb{P}(Y=m_{\mathbf{x}})/2}{p_1}$, $z_2' = \mathbb{E}[Y|Y \in (x_2, \infty)]$.[55]

Consider a random variable Y'' obtained from Y as follows: $\mathbb{P}(Y'' \in \{z_1', x_1, z_1, z_2, x_2, z_2'\}) = 1$, $\mathbb{P}(Y'' = z_1') = \mathbb{P}(Y'' = z_2') = p_2$, $\mathbb{P}(Y'' = z_1) = \mathbb{P}(Y'' = z_2) = p_1$, and $\mathbb{P}(Y'' = x_1) = \mathbb{P}(Y'' = x_2) = \mathbb{P}(Y = x_1) = \mathbb{P}(Y = x_2)$. Clearly, Y'' is symmetric about the midpoint $m_{\mathbf{x}}$. Since the social cost function is convex, it follows that $\mathbb{E}[sc(\mathbf{x}, Y'')] \leq \mathbb{E}[sc(\mathbf{x}, Y)]$.

Now, consider a random variable Y' obtained from Y'' as follows. We construct Y' from Y'' by shifting parts of the probability mass at z_1 and z_1' to x_1 as well as by shifting parts of the probability mass at z_2 and z_2' to x_2 while ensuring that $\mathbb{E}[Y'] = \mathbb{E}[Y'']$. Specifically, since $z_1 < x_1 < z_1'$, we can write $x_1 = \lambda z_1 + (1-\lambda)z_1'$ for some $0 < \lambda < 1$. One way to shift the probability mass is to subtract probability λp and $(1-\lambda)p$ from z_1 and z_1' respectively and add probability p to x_1 for p sufficiently small (do the same transformation for points z_2, z_2', and x_2). This transformation ensures $\mathbb{E}[Y'] = \mathbb{E}[Y'']$ because

$$(p_1 - \lambda p)z_1 + (p_2 - (1-\lambda)p)z_2 + (\mathbb{P}(Y'' = x_1) + p)x_1 = p_1 z_1 + p_2 z_2 + \mathbb{P}(Y'' = x_1)x_1.$$

In order to maximize the shift in probability mass, we choose the largest p possible or $p = \min(\frac{p_1}{\lambda}, \frac{p_2}{1-\lambda})$. If $p = \frac{p_1}{\lambda}$, then $\mathbb{P}(Y' \in \{z_1', x_1, x_2, z_2'\}) = 1$, as $\mathbb{P}(Y' = z_1') = \mathbb{P}(Y' = z_2') = p_2 - (1-\lambda)p$, and $\mathbb{P}(Y' = x_1) = \mathbb{P}(Y' = x_2) = \mathbb{P}(Y'' = x_1) + p$. Else if $p = \frac{p_2}{1-\lambda}$, then $\mathbb{P}(Y' \in \{x_1, z_1, z_2, x_2\}) = 1$, $\mathbb{P}(Y' = z_1) = \mathbb{P}(Y' = z_2) = p_1 - \lambda p$, and $\mathbb{P}(Y' = x_1) = \mathbb{P}(Y' = x_2) = \mathbb{P}(Y'' = x_1) + p$. It is clear from construction that Y' is symmetric about $m_{\mathbf{x}}$. Convexity implies $\mathbb{E}[sc(\mathbf{x}, Y')] \leq \mathbb{E}[sc(\mathbf{x}, Y'')]$, and so $\mathbb{E}[sc(\mathbf{x}, Y')] \leq \mathbb{E}[sc(\mathbf{x}, Y)]$.

Now, let g be a mechanism that locates the facility according to Y' given profile $\mathbf{x}$. Note that there is a unique way to extend the definition of g to all other two-agent profiles such that g is shift and scale invariant as well as symmetric; let us extend the definition of g that way. Furthermore, this extension is easily seen to imply the following:

---

[55] Note that if $\mathbb{P}(Y = m_{\mathbf{x}}) = 0$, then z_1 is the conditional expectation of Y given that $Y \in (x_1, m_{\mathbf{x}})$. When $\mathbb{P}(Y = m_{\mathbf{x}}) > 0$, imagine that whenever $Y = m_{\mathbf{x}}$, we flip a fair coin; then z_1 is the conditional expectation of Y given that $Y \in (x_1, m_{\mathbf{x}})$ or $Y = m_{\mathbf{x}}$ and the coin lands on heads. z_2 can be defined in a similar manner (replace $(x_1, m_{\mathbf{x}})$ with $(m_{\mathbf{x}}, x_2)$ and heads with tails). From this description it is clear that $z_1 \in (x_1, m_{\mathbf{x}}]$, $z_2 \in [m_{\mathbf{x}}, x_2)$, and that they are symmetric about $m_{\mathbf{x}}$.

1. Since $\mathbb{E}[sc(\mathbf{x}, g(\mathbf{x}))] \leq \mathbb{E}[sc(\mathbf{x}, f(\mathbf{x}))]$ for the profile $\mathbf{x}$, the social cost obtained by mechanism g via the extension is no more than the one obtained by mechanism f for all two-agent profiles.

2. If $\mathbb{P}(g(\mathbf{x}) \in (x_1, x_2)) = 0$, then $\mathbb{P}(g(\mathbf{q}) \in (q_1, q_2)) = 0$ for all two-agent profiles $\mathbf{q}$. Similarly, if $\mathbb{P}(g(\mathbf{x}) \in (-\infty, x_1) \cup (x_2, \infty)) = 0$, then $\mathbb{P}(g(\mathbf{q}) \in (-\infty, q_1) \cup (q_2, \infty)) = 0$ for all two-agent profiles $\mathbf{q}$.

Thus, all that is left for us to do is to show strategyproofness of g. We can do so by verifying the conditions in Lemma 6 (the fact that it holds for all the required profiles is then again immediate by shift and scale invariance). When $p = \frac{p_1}{\lambda}$, we claim that it suffices to show that:

$$-z_1'(p_2 - (1-\lambda)p) + z_2'(p_2 - (1-\lambda)p) + x_2(\mathbb{P}(Y' = x_2) + p) \geq -z_1'p_2 - z_1p_1 - z_2p_1 + z_2'p_2 + x_2\mathbb{P}(Y = x_2),$$

and that

$$z_1'(p_2 - (1-\lambda)p) - z_2'(p_2 - (1-\lambda)p) + x_2(\mathbb{P}(Y' = x_2) + p) \geq z_1'p_2 + z_1p_1 + z_2p_1 - z_2'p_2 + x_2\mathbb{P}(Y = x_2).$$

To justify this claim, we need to show that the right hand sides are always greater than or equal to 0. But note that z_1, z_2, p_1, z_1', z_2', and p_2 were defined so that the right hand sides amount exactly to the conditions of Lemma 6 for f on the profile $\mathbf{x}$, and thus must be greater than or equal to zero. After some algebra, the two inequalities above reduce to:

$$z_1'(1-\lambda)p - z_2'(1-\lambda)p + x_2p \geq -z_1p_1 - z_2p_1, \tag{B.1}$$

and

$$-z_1'(1-\lambda)p + z_2'(1-\lambda)p + x_2p \geq z_1p_1 + z_2p_1. \tag{B.2}$$

To show (B.1), we know that

$$z_1'(1-\lambda)p + z_1p_1 = (z_1'(1-\lambda) + z_1\lambda)p = x_1p = 0, \quad \text{that is} \quad z_1'(1-\lambda)p = -z_1p_1,$$

and that

$$x_2p = (z_2'(1-\lambda) + z_2\lambda)p \geq z_2'(1-\lambda)p - z_2p_1, \quad \text{that is} \quad x_2p - z_2'(1-\lambda)p \geq -z_2p_1.$$

Combining the two above expressions gives us the desired result. Similarly, (B.2) follows from the fact that $z_1'(1-\lambda)p + z_1p_1 = 0$ and that $x_2p = (z_2'(1-\lambda) + z_2\lambda)p \geq -z_2'(1-\lambda)p + z_2p_1$. The proof for the case where $p = \frac{p_2}{1-\lambda}$ is similar and so will be omitted. □

B.3 Alternative Assumptions in Section 4.2

Theorem 4 holds if we replace the assumption of shift invariance with symmetry. It is clear from the structure of the proof that it is enough to replace Lemma 4 with the following lemma:

Lemma B.1. *Given any strategyproof, symmetric and scale invariant mechanism, there exists a strategyproof, symmetric, scale and shift invariant mechanism with a weakly smaller worst-case approximation ratio.*

Proof. Given a mechanism f, define $g(\mathbf{x}) = f(0, x_2 - x_1) + x_1$. Assume f is strategyproof, symmetric and scale invariant. We claim that g is strategyproof, symmetric, scale and shift invariant with a weakly smaller worst-case approximation ratio. The fact that g is shift invariant and has a weakly smaller worst-case approximation ratio than f is immediate. Let $Y_{x_1,x_2} \sim f(\mathbf{x})$ and $Y'_{x_1,x_2} \sim g(\mathbf{x})$; the relevant equalities below are in distribution.

1. g is symmetric: let $\mathbf{x} \in \mathbb{R}^2$, and let $b \in \mathbb{R}$. Then $\mathbb{P}(Y'_{x_1,x_2} \geq m_{\mathbf{x}}+b) = \mathbb{P}(Y_{0,x_2-x_1} \geq m_{\mathbf{x}}+b-x_1) = \mathbb{P}(Y_{0,x_2-x_1} \geq m_{(0,x_2-x_1)} + b) = \mathbb{P}(Y_{0,x_2-x_1} \leq m_{(0,x_2-x_1)} - b) = \mathbb{P}(Y_{0,x_2-x_1} \leq m_{\mathbf{x}} - b - x_1) = \mathbb{P}(Y_{0,x_2-x_1} + x_1 \leq m_{\mathbf{x}} - b) = \mathbb{P}(Y'_{x_1,x_2} \leq m_{\mathbf{x}} - b)$.

2. g is scale invariant: let $\mathbf{x} \in \mathbb{R}^2$ and let $c \in \mathbb{R}$. Then $Y_{cx_1,cx_2} = Y_{0,c(x_2-x_1)}+cx_1 = cY_{0,x_2-x_1}+cx_1 = c(Y_{0,x_2-x_1} + x_1) = cY'_{x_1,x_2}$. The second equality follows from scale invariance of f.

3. g is strategyproof: let $\mathbf{x} \in \mathbb{R}^2$, $b, x'_2 \in \mathbb{R}$. There are two cases:

   (a) Assume $\mathbb{E}[|x_2-Y'_{x_1,x_2}|] > \mathbb{E}[|x_2-Y'_{x_1,x'_2}|]$. Note that $\mathbb{E}[|x_2-Y'_{x_1,x_2}|] = \mathbb{E}[|(x_2-x_1)-Y_{0,x_2-x_1}|]$ and $\mathbb{E}[|x_2 - Y'_{x_1,x'_2}|] = \mathbb{E}[|(x_2 - x_1) - Y_{0,x'_2-x_1}|]$. Thus, it follows that when agent 1's location is 0 and agent 2's location is $x_2 - x_1$, agent 2 can benefit under f when reporting $x'_2 - x_1$ instead, violating strategyproofness of f. Contradiction.

   (b) Assume $\mathbb{E}[|x_1 - Y'_{x_1,x_2}|] > \mathbb{E}[|x_1 - Y'_{x_1+b,x_2}|]$. Note that $\mathbb{E}[|x_1 - Y'_{x_1,x_2}|] = \mathbb{E}[| - Y_{0,x_2-x_1}|] = \mathbb{E}[|(x_2 - x_1) - Y_{0,x_2-x_1}|]$, where the last equality follows from symmetry of f. Also note that $\mathbb{E}[|x_1 - Y'_{x_1+b,x_2}|] = \mathbb{E}[| - b - Y_{0,x_2-x_1-b}|] = \mathbb{E}[|(x_2 - x_1) - Y_{0,x_2-x_1-b}|]$, where again the last equality follows from symmetry of f. Thus, when agent 1's true location is 0 and agent 2's true location is $x_2 - x_1$, then agent 2 benefits under f by reporting $x_2 - x_1 - b$, violating strategyproofness of f. Contradiction.

□

Appendix C

Approximation Algorithms for the Incremental Knapsack Problem

C.1 Proof of Proposition 6.2

We will define a greedy algorithm, like that of value-to-weight ratio greedy algorithm for the standard knapsack. The algorithm ensures that the solution it constructs till time t satisfies the following two properties for all t:

1. the total value of items packed in the knapsack at time t' has value at least half of the optimal solution to the LP relaxation of the standard knapsack problem with capacity $B_{t'}$ for every $t' \leq t$.

2. the solution respects the precedence constraints at each time.

Index the items in non-increasing ratio order (break ties arbitrarily). Let $W_i = \sum_{i'=1}^{i} w_i$. Note that whenever the knapsack capacity is B, where $W_{i-1} \leq B < W_i$ (with $W_0 = 0$), the greedy solution either choose the set of items $\{1, \ldots, i-1\}$ or $\{i\}$, depending on which set gives a higher objective value. We will construct a sequence of nested solutions $S_1 \subseteq S_2 \subseteq \ldots S_T$ as follows. Suppose we want to construct a 1/2-approximation solution for time t given a sequence of nested solutions $S_1 \subseteq S_2 \subseteq \ldots S_{t-1}$ inductively. If $W_{i-1} \leq B_t < W_i$ for some i. If $W_{i-1} \leq B_{t-1} < W_i$, then we can set $S_t = S_{t-1}$ and still maintain a 1/2-approximate solution. So suppose $B_{t-1} < W_{i-1}$, then it can be shown through our construction that $S_{t-1} \subseteq \{1, \ldots, i-1\}$. There are two cases to consider.

1. Ratio greedy picks $\{1, \dots i-1\}$ in time period t. Since $S_{t-1} \subseteq \{1, \dots, i-1\}$, setting $S_t = \{1, \dots i-1\}$ maintains the precedence constraint for time t.

2. Ratio greedy picks $\{i\}$ in time period t. Since item i fits into the knapsack initially, we can reset $S_1 = S_2 = \dots S_{t-1} = S_t = \{i\}$ and still maintain a 1/2-approximate solution.

This completions the proof. Note that the approximation is valid for all possible set of discounting factors.

C.2 Proof of Lemma C.2

For every nonempty polyhedron $Q^{\sigma,h}$, we begin by showing (6.13) holds for every $S^{k,h}$ with the help of the following auxiliary LP. The auxiliary LP has constraints similar to those governing the feasible region of $Q^{\sigma,h}$, except that we only focus on the variables from the value class $S^{k,h}$. Instead of the knapsack capacity constraint, we require that the total weight of items from value class $S^{k,h}$ packed into the knapsack in each time period is no more than that of $\bar{x}$.

$$\begin{aligned}
\max \quad & \sum_{t=1}^{T'} \Delta_t \sum_{i=1}^{|S^{k,h}|} v'_{k_i} x_{k_i,t} \qquad \text{(C.1)} \\
\text{s.t.} \quad & \sum_{i=1}^{|S^{k,h}|} w_{k_i} x_{k_i,t} \le \sum_{i=1}^{|S^{k,h}|} w_{k_i} \bar{x}_{k_i,t} \quad \forall t \\
x_{k_1,t} = x_{k_2,t} = \dots = x_{k_{|S^{k,h}|},t} = 0 & \quad \forall (k,t) \text{ s.t. } \sigma_t^k = 0 \\
x_{k_1,t} = x_{k_2,t} = \dots = x_{k_{\sigma_t^k},t} = 1, & \\
x_{k_{\sigma_t^k}+1,t} = \dots = x_{k_{|S^{k,h}|},t} = 0 & \quad \forall (k,t) \text{ s.t. } 1 \le \sigma_t^k < J \text{ and } \sigma_t^k < |S^{k,h}| \\
x_{k_1,t} = x_{k_2,t} = \dots = x_{k_{\sigma_t^k},t} = 1 & \quad \forall (k,t) \text{ s.t. } \sigma_t^k = J \text{ and } \sigma_t^k < |S^{k,h}| \\
x_{k_1,t} = x_{k_2,t} = \dots = x_{k_{|S^{k,h}|},t} = 1 & \quad \forall (k,t) \text{ s.t. } \sigma_t^k \ge |S^{k,h}| \\
x_{k_i,t-1} \le x_{k_i,t} & \quad \forall k_i, \text{and } t = 2, 3, \dots, T' \\
x_{k_i,t-1} \in [0,1] & \quad \forall k_i, t.
\end{aligned}$$

Lemma C.1. *For every $S^{k,h}$, there exists an optimal solution to the auxiliary LP that contains at most one fractional variable $x_{k_i,t}$ in each time period t.*

Proof. We only need to focus on the case when there exists t such that $\sigma_t^k = J$ and $J < |S^{k,h}|$, otherwise the LP feasible region has at most one feasible point, which is an integer.

Let $t^\star$ be the first (smallest) period in which we are in the case $\sigma_t^k = J$ and $\sigma_t^k < |S^{k,h}|$. Ignoring the precedence constraints for a moment, then the auxiliary LP can be broken up into $T - t^\star + 1$ single period LPs of the following form, one for each $t \geq t^\star$.

$$LP_t = \max \quad \sum_{i=1}^{|S^{k,h}|} v'_{k_i} x_{k_i,t}$$

$$\text{s.t.} \quad \sum_{i=1}^{|S^{k,h}|} w_{k_i} x_{k_i,t} \leq \sum_{i=1}^{|S^{k,h}|} w_{k_i} \bar{x}_{k_i,t} \tag{C.2}$$

$$x_{k_1,t} = x_{k_2,t} = \ldots = x_{k_{\sigma_t^k},t} = 1 \tag{C.3}$$

$$x_{k_i,t-1} \in [0,1] \quad \forall k_i.$$

Notice that in the modified instance of the problem, all items have the same value within a value class $S^{k,h}$. Hence, an optimal solution to LP_t is simply to pack the items in increasing order of their weight, starting from the smallest weight item first. Moreover, notice that this set of optimal solutions satisfy the precedence constraints. □

Lemma C.2. *Let $\bar{x}$ be an optimal solution to the optimization problem* $\max\{\sum_{t=1}^{T'} \Delta_t \sum_{i \in S^h} v'_i x_{i,t} : x \in Q^{\sigma,h}\}$ *over a non-empty $Q^{\sigma,h}$ for some $\sigma \in \{0,\ldots,J\}^{T'K}$ and suppose $h \in S$. There exists an integer feasible solution $x^{\sigma,h}$ to the auxiliary LP such that*

$$\sum_{t=1}^{T'} \sum_{i=1}^{|S^{k,h}|} v'_{k_i} x_{k_i,t}^{\sigma,h} \geq (1-\epsilon) \sum_{t=1}^{T'} \sum_{i=1}^{|S^{k,h}|} v'_{k_i} \bar{x}_{k_i,t}, \tag{C.4}$$

for every $S^{k,h}$ and

$$\sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v'_i x_{i,t}^{\sigma,h} \geq \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v'_i \bar{x}_{i,t} - \Delta_T \epsilon v_h. \tag{C.5}$$

Proof. We first show the validity of inequality C.4. Without lost of generality, let $\hat{x}$ be the optimal solution to the auxiliary LP (C.1). For a time period t where $\sigma_t^k < J$ and $\sigma_t^k < |S^{k,h}|$ or when $\sigma_t^k \geq |S^{k,h}|$, we don't need to round the variables $\hat{x}_{k_i,t}$ since they are already integral. Hence, we set $x_{k_i,t}^{\sigma,h} = \hat{x}_{k_i,t}$ for all the variables these periods, which implies that

$$\sum_{i=1}^{|S^{k,h}|} v'_{k_i} x_{k_i,t}^{\sigma,h} = \sum_{i=1}^{|S^{k,h}|} v'_{k_i} \hat{x}_{k_i,t} \geq \sum_{i=1}^{|S^{k,h}|} v'_{k_i} \bar{x}_{k_i,t}$$

for such a period t.

For a time period t where $\sigma_t^k = J$ and $\sigma_t^k \geq |S^{k,h}|$, by Lemma C.1, there is at most one fractional $\hat{x}_{k_i,t}$ in such a time period. Consequently, we round down this fractional variable while keeping others the same. Since all the variables have the same value within a value class $S^{k,h}$, we have that

$$\frac{\sum_{i=1}^{|S^{k,h}|} v'_{k_i} \hat{x}_{k_i,t} - \sum_{i=1}^{|S^{k,h}|} v'_{k_i} x_{k_i,t}^{\sigma,h}}{\sum_{i=1}^{|S^{k,h}|} v'_{k_i} \hat{x}_{k_i,t}} \quad = \quad \frac{\sum_{i=1}^{|S^{k,h}|} \hat{x}_{k_i,t} - \sum_{i=1}^{|S^{k,h}|} x_{k_i,t}^{\sigma,h}}{\sum_{i=1}^{|S^{k,h}|} \hat{x}_{k_i,t}} \leq \frac{1}{J} < \epsilon,$$

which implies that

$$\sum_{i=1}^{|S^{k,h}|} v'_{k_i} x_{k_i,t}^{\sigma,h} \geq (1-\epsilon) \sum_{i=1}^{|S^{k,h}|} v'_{k_i} \hat{x}_{k_i,t} \geq (1-\epsilon) \sum_{i=1}^{|S^{k,h}|} v'_{k_i} \bar{x}_{k_i,t}.$$

Summing up the above inequalities over all time periods gives us the desired result.

Now we show the validity of inequality C.5. Consider the following auxiliary LP:

$$\begin{aligned}
\max \quad & \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v'_{k_i} x_{i,t} \\
\text{s.t.} \quad & \sum_{i \in T^h} w_i x_{i,t} \leq \sum_{i \in T^h} w_i \bar{x}_{i,t} \quad \forall t \\
& x_{i,t-1} \leq x_{i,t} \quad \forall i \in T^h, \text{and } t = 2, 3, \ldots, T' \\
& x_{i,t-1} \in [0,1] \quad \forall i, t.
\end{aligned}$$

An optimal solution of the LP above would be to greedily pack items in the order of non-increasing value-to-weight ratio. Let $\hat{x}$ be such an optimal solution. It is clear that $\hat{x}$ has at most one fractional variable in each time period. We round down such a fractional variable in each time period to 0 to obtain an integer solution $x^{\sigma,h}$. Consequently, we have that

$$\sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' \hat{x}_{i,t} - \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' x_{i,t}^{\sigma} \leq \frac{\epsilon v_h}{T'} \sum_{t=1}^{T'} \Delta_t \leq \Delta_{T'} \epsilon v_h,$$

where the first inequality follows from the fact that every item in T^h has value no more than $\frac{\epsilon v_h}{T'}$, and the second inequality follows from the fact that Δ_t is non-decreasing in t. Rearranging the terms, we get

$$\sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' x_{i,t}^{\sigma} \geq \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' \hat{x}_{i,t} - \Delta_{T'} \epsilon v_h \geq \sum_{t=1}^{T'} \Delta_t \sum_{i \in T^h} v_i' \bar{x}_{i,t} - \Delta_{T'} \epsilon v_h.$$

□

C.3 Proof of Proposition 6.5

We prove each of the three cases via an interchange argument. Suppose there exists an optimal schedule OPT that does not follow the index rule. Then there will be a pair of adjacent items i and j (not belong to the same indifference class) that disobeys the index rule, i.e. $\sigma^{-1}(i) < \sigma^{-1}(j)$ but i is scheduled after j in OPT. Consider a schedule MOD that switches the order between i and j while keeping the ordering of the rest of the jobs the same. We measure $V(MOD) - V(OPT)$ and show that the difference is positive, contradicting the optimality of OPT. Let w be the total capacity of items packed into the knapsack right before both items i and j.

- When $\Delta(s) = 1$,

$$\begin{aligned} V(MOD) - V(OPT) &= v_i(T - \frac{1}{c}(w + w_i)) + v_j(T - \frac{1}{c}(w + w_i + w_j)) \\ &\quad - v_j(T - \frac{1}{c}(w + w_j)) - v_i(T - \frac{1}{c}(w + w_i + w_j)) \\ &= v_i w_j - v_j w_i > 0 \end{aligned}$$

- When $\Delta(s) = e^{-rs}$,

$$
\begin{aligned}
V(MOD) - V(OPT) &= v_i \int_{\frac{1}{c}(w+w_i)}^{T} e^{-rs} ds + v_j \int_{\frac{1}{c}(w+w_i+w_j)}^{T} e^{-rs} ds \\
&\quad - v_j \int_{\frac{1}{c}(w+w_j)}^{T} e^{-rs} ds - v_i \int_{\frac{1}{c}(w+w_i+w_j)}^{T} e^{-rs} ds \\
&= v_i \int_{\frac{1}{c}(w+w_i)}^{\frac{1}{c}(w+w_i+w_j)} e^{-rs} ds - v_j \int_{\frac{1}{c}(w+w_j)}^{\frac{1}{c}(w+w_i+w_j)} e^{-rs} ds \\
&= \frac{e^{-rw}}{r} \big(v_i (e^{-rw_i} - e^{-r(w_i+w_j)}) - v_j (e^{-rw_j} - e^{-r(w_i+w_j)}) \big) \\
&= \frac{e^{-rw}}{r} \big(v_i e^{-rw_i} (1 - e^{-rw_j}) - v_j e^{-rw_j} (1 - e^{-rw_i}) \big) > 0
\end{aligned}
$$

- When $\Delta(s) = (1+r)^{-s}$, since we can rewrite $(1+r)^{-s}$ as $e^{-\ln(1+r)s}$, the same derivation follows.

C.4 Proof of Proposition 6.4

Proof. Assume w.l.o.g. that $b_1 \geq b_2 \geq \ldots \geq b_q$. We proceed via induction on $p+q$.

Base case: the result holds true for $p = 1$ and any q trivially as $a_1 = \sum_{i=1}^{q} b_i$.

Inductive step: since $a_1 \geq \max_{i=1,\ldots,q} b_i$, there exists $m \geq 1$ such that $a_1 \geq \sum_{i=1}^{m} b_i$ and $a_1 < \sum_{i=1}^{m+1} b_i$. Write $b_{m+1} = b_{m+1,1} + b_{m+1,2}$ such that $a_1 = \sum_{i=1}^{m} b_i + b_{m+1,1}$. Note that

$$
a_1^2 = (\sum_{i=1}^{m} b_i)^2 + b_{m+1,1}^2 + 2(\sum_{i=1}^{m} b_i) b_{m+1,1} \geq (\sum_{i=1}^{m} b_i)^2 + b_{m+1,1}^2 + 2 b_{m+1,2} b_{m+1,1}, \tag{C.6}
$$

as $\sum_{i=1}^{m} b_i \geq b_1 \geq b_{m+1,2}$. Now, we apply the inductive hypothesis on the sets $\{a_2, \ldots, a_p\}$ and $\{b_{m+1,2}, b_{m+2} \ldots, b_q\}$ (note that these two sets satisfy the assumptions of the lemma) and get

$$
\sum_{i=2}^{p} a_i^2 \geq b_{m+1,2}^2 + \sum_{i=m+2}^{p} b_i^2. \tag{C.7}
$$

Adding both sides of equations (C.6) and (C.7) gives us the desired result. $\square$

Appendix D

Capacity Constrained Assortment Optimization under the Markov Chain based Choice Model

D.1 Proofs of Theorems 7.1 and 7.2

Proof. Proof of Theorem 7.1. Consider an instance $\mathcal{I}$ of VCC, consisting of a cubic graph $G = (V, E)$ on n vertices $V = \{v_1, \ldots, v_n\}$. We can assume that $k > |E|/3$, or otherwise, the distinction between the two cases above is easy. We construct an instance $\mathcal{M}(\mathcal{I})$ of Cardinality-Assort as follows. Each vertex $v_i \in V$ corresponds to an item i of $\mathcal{N}$. In addition, we also have the no-purchase item 0. For each vertex $v \in V$, let $N(v)$ denote the neighborhood of v in G, i.e., $N(v) = \{u : (u, v) \in E\}$, consisting of exactly 3 vertices. Now, for all $(i, j) \in \mathcal{N} \times \mathcal{N}_+$ the transition probabilities are defined as

$$\rho_{ij} = \begin{cases} 1/4 & \text{if } v_j \in N(v_i) \text{ or } j = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Finally, for all items $i \in \mathcal{N}$, we have an arrival rate of $\lambda_i = 1/n$ and a price of $p_i = 1$. Out of these items, at most k can be selected.

The goal in VCC is to choose a minimum-cardinality set of vertices such that every edge is incident to at least one of the chosen vertices. Let $U^* \subseteq V$ be a minimum vertex cover in G. We show that the instance $\mathcal{M}(\mathcal{I})$ satisfies the following properties:

(a) $|U^*| \leq k \quad \Rightarrow \quad R(S^*) \geq \frac{3}{4} + \frac{k}{4n}$,

(b) $|U^*| \geq (1+\alpha)k \quad \Rightarrow \quad R(S^*) \leq \frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16}$,

where S^* is the optimal assortment for $\mathcal{M}(\mathcal{I})$. This implies that Cardinality-Assort cannot be approximated within factor larger than $1 - \frac{\alpha}{16}$, unless P = NP. To see this, note that the ratio between $\frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16}$ and $\frac{3}{4} + \frac{k}{4n}$ is monotone-increasing in k, meaning that the maximum value attained is $1 - \frac{\alpha}{16}$.

Case (a): $|U^*| \leq k$**.** In this case, we can augment U^* with $k - |U^*|$ additional vertices chosen arbitrarily from $V \setminus U^*$, and obtain a (not-necessarily minimum) vertex cover U with $|U| = k$. Now, consider the assortment $S = \{i : v_i \in U\}$, which is indeed a feasible solution. Since all prices are equal to 1, we can write the expected revenue of this set as

$$R(S) = \mathbb{P}(S \prec 0) = \sum_{i \in S} \lambda_i + \sum_{i \notin S} \lambda_i \mathbb{P}_i(S \prec 0) = \frac{k}{n} + \frac{1}{n} \sum_{i \notin S} \mathbb{P}_i(S \prec 0). \tag{D.1}$$

When starting at any state $i \notin S$, the Markov chain moves to 0 with probability 1/4 and gets absorbed. With probability 3/4, the Markov chain moves from i to one of the vertices in $N(i)$. Since U is a vertex cover, it follows that $N(i) \subseteq S$. Therefore, $\mathbb{P}_i(S \prec 0) = 3/4$ for all $i \notin S$. Based on these observations for the optimal assortment S^*, we have

$$R(S^*) \geq R(S) = \frac{k}{n} + \frac{3(n-k)}{4n} = \frac{3}{4} + \frac{k}{4n}.$$

Case (b): $|U^*| \geq (1+\alpha)k$**.** Let S be some assortment consisting of k items. In this case, equation (D.1) is still a valid decomposition of $R(S)$, and we need to consider two cases for items $i \notin S$. If $N(i) \subseteq S$, then $\mathbb{P}_i(S \prec 0) = 3/4$ as in case (a). However, when $N(i) \nsubseteq S$, there exists $j \in N(i)$ such that $j \notin S$. Therefore, there is a probability of 1/16 that starting from i the Markov chain moves to j and from there to 0. Consequently, for such items, $\mathbb{P}_i(S \prec 0) \leq \frac{3}{4} - \frac{|N(i) \backslash S|}{16}$. Therefore,

$$\begin{aligned} R(S) &= \frac{k}{n} + \frac{1}{n} \sum_{i \notin S, N(i) \subseteq S} \frac{3}{4} + \frac{1}{n} \sum_{i \notin S, N(i) \nsubseteq S} \mathbb{P}_i(S \prec 0) \\ &\leq \frac{3}{4} + \frac{k}{4n} - \frac{1}{16n} \sum_{i \notin S, N(i) \nsubseteq S} |N(i) \backslash S|. \end{aligned} \tag{D.2}$$

To upper bound the latter term, let $V(S)$ be the set of vertices of V corresponding to S, i.e., $V(S) = \{v_i : i \in S\}$. Let $\bar{E}(S)$ be the set of edges that are not covered by $V(S)$. We have $(2 \cdot |\bar{E}(S)|) =$

$\sum_{i \notin S, N(i) \not\subseteq S} |N(i) \backslash S|$. The important observation is that $|\bar{E}(S)| \geq \alpha k$. Otherwise, $V(S)$ can be augmented to a vertex cover via the addition of fewer than αk vertices, contradicting $|U^*| \geq (1+\alpha)k$. Now,

$$|\bar{E}(S)| \geq \alpha k \geq \frac{\alpha}{3} \cdot |E| = \frac{\alpha n}{2},$$

where the second inequality follows from $k > |E|/3$, and the last equality holds since $|E| = 3n/2$, as G is cubic. By inequality (D.2), we have

$$R(S) \leq \frac{3}{4} + \frac{k}{4n} - \frac{|\bar{E}(S)|}{8n} \leq \frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16} .$$

Since the above upper bound on $R(S)$ holds for any assortment S of k items, this must also be true for the maximum-revenue one, S^*. □ □

Proof. Proof of Theorem 7.2. Aouad et al. [7] show that unconstrained assortment optimization over the distribution over permutations model is hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$ even for the case where the number of preference lists is equal to the number of items, i.e., $K = n$.

We consider an instance $\mathcal{I}$ of the assortment optimization problem over distribution over permutations model with n preference lists: $L_1, \dots, L_n$. We construct a corresponding instance $\mathcal{M}(\mathcal{I})$ of the assortment optimization under the Markov chain model as follows. Each of the original items in $\mathcal{N}$ has a separate copy as a state in $\mathcal{M}(\mathcal{I})$ for every list that contains it. More precisely, for every list L_i and for every $1 \leq j \leq |L_i|$, we have a state (j,i) corresponding to the j-th most preferred item in L_i. In addition, there is a state 0 corresponding to the no-purchase option. Therefore, the set of states is:

$$\mathcal{S} = \{0\} \cup \{(j,i) \ : \ i = 1, \dots, n, \ j = 1, \dots, |L_i|\}.$$

The transition probabilities between these states are given by:

$$\rho_{((j,i),s)} = \begin{cases} 1 & \text{if } j < |L_i| \text{ and } s = (j+1,i) \\ 1 & \text{else if } j = |L_i| \text{ and } s = 0 \\ 0 & \text{otherwise.} \end{cases}$$

In other words, for each list there is a directed path (with transition probabilities 1) over its corresponding states in decreasing order of preference, ending at the no-purchase option. This is illustrated in Figure D.1. Finally, the arrival rates are defined by

$$\lambda_{(j,i)} = \begin{cases} \psi_i & \text{if } j = 1 \\ 0 & \text{otherwise,} \end{cases}$$

where ψ_i is the probability of list L_i. With this construction, each row corresponds to a list, and each column correspond to an item.

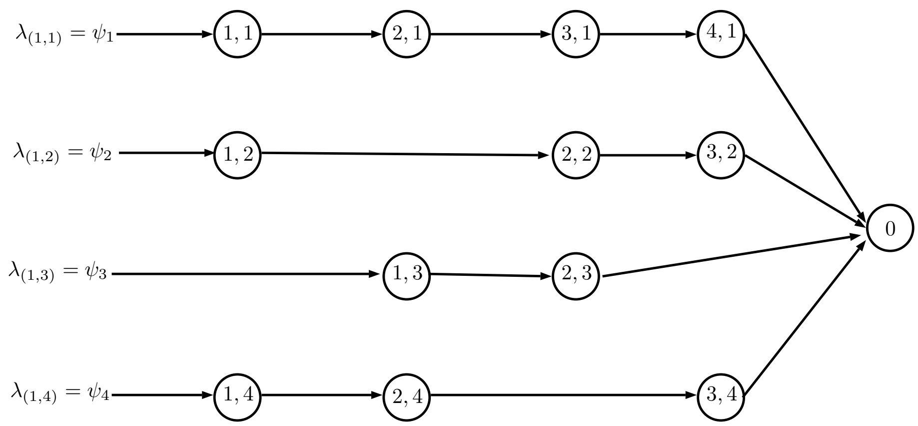

Figure D.1: Sketch of our construction for an instance on 4 items, where $L_1 = (1 \succ 2 \succ 3 \succ 4)$, $L_2 = (1 \succ 3 \succ 4)$, $L_3 = (2 \succ 3)$, and $L_4 = (1 \succ 2 \succ 4)$. Note, for example, that the state $(2, 2)$ corresponds to the second item of L_2, but actually corresponds to item 3.

In order to obtain a one-to-one correspondence between the solutions to $\mathcal{I}$ and $\mathcal{M}(\mathcal{I})$, it remains to ensure that, when item i is offered in $\mathcal{I}$, all of its corresponding copies (appearing in the same column) are offered in $\mathcal{M}(\mathcal{I})$, and vice versa. This restriction can be captured by the constraints $x_{(j,i)} = x_{(k,\ell)}$, for every $i, \ell \in \{1, \dots, n\}$ such that $j \leq |L_i|, k \leq |L_\ell|$ and such that the j^{th} item in L_i is the k^{th} item in L_ℓ. This way, we guarantee that each column is either completely picked or completely unpicked in the instance $\mathcal{M}(\mathcal{I})$. The resulting set of inequalities specifies a constraint matrix with a single appearance of $+1$ and -1 in each row, where all other entries are 0. Such matrices are well-known to be totally-unimodular (see, for example, [107]).

To complete the proof, note that the original instance $\mathcal{I}$ consists of n items and n preference lists and therefore, the Markov chain instance $\mathcal{M}(\mathcal{I})$ has $O(n^2)$ states. Since the former problem is NP-hard to approximate within factor $O(n^{1-\epsilon})$, for any fixed $\epsilon > 0$, it follows that TU-Assort cannot be efficiently approximated within $O(n^{1/2-\epsilon})$, unless $P = NP$. $\square$ $\square$

D.2 Proof of Lemma 7.2

This result is an immediate corollary of the following (more general) claim: Let S^g be the solution returned by Algorithm 7.1, and let S be any subset of states. Then,

$$R(S^g) \geq \frac{R(S)}{|S|}.$$

To prove this claim, let g be the first item selected by Algorithm 7.1, which necessarily exists as long as there is an item i with $p_i > 0$. Then, by definition of the greedy algorithm, we have $R(\{g\}) \geq R(\{i\})$ for every item $i \in S$. Therefore,

$$R(S^g) \geq R(\{g\}) \geq \frac{1}{|S|} \cdot \sum_{i \in S} R(\{i\}) \geq \frac{R(S)}{|S|},$$

where the last inequality follows from the sublinearity of the revenue function (Lemma 7.9).

D.3 Proof of Lemma 7.4

Let S^{gu} be the set of states selected by Algorithm 7.2. Note that for every $i \in S^{gu}$, we have that $\mathbb{P}(i \prec S^{gu}_{+} \backslash \{i\}) \geq \mathbb{P}(i \prec U^{*}_{+} \backslash \{i\})$ since S^{gu} is a subset of U^*. Thus,

$$\begin{aligned}
R(S^{gu}) &= \sum_{i \in S^{gu}} \mathbb{P}(i \prec S^{gu}_{+} \backslash \{i\}) p_i \\
&\geq \sum_{i \in S^{gu}} \mathbb{P}(i \prec U^{*}_{+} \backslash \{i\}) p_i \\
&\geq \frac{k}{|U^*|} \sum_{i \in U^*} \mathbb{P}(i \prec U^{*}_{+} \backslash \{i\}) p_i \\
&= \frac{k}{|U^*|} R(U^*) \\
&\geq \frac{k}{|U^*|} R(S^*),
\end{aligned}$$

where S^* is the optimal solution to $\textsf{Cardinality-Assort}$. Here, the second inequality holds due to picking the top k states in terms of $\mathbb{P}(i \prec U^{*}_{+} \backslash \{i\})$ values. The last inequality holds since the optimal unconstrained revenue provides an upper bound on the optimal revenue in the constrained case.

D.4 Proof of Lemma 7.6

It suffices to verify that $(p_i^{S_1})^{S_2} = p_i^{S_1 \cup S_2}$ for all S_1,S_2 and $i \notin S_1 \cup S_2$, as the above identity clearly hold for the transition matrix updates. We have

$$\begin{aligned}(p_i^{S_1})^{S_2} =& p_i^{S_1} - \sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) p_j^{S_1} \\ =& p_i - \underbrace{\sum_{l \in S_1} \mathbb{P}_i(l \prec S_{1+} \backslash \{l\}) p_l}_{A} - \underbrace{\sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) p_j^{S_1}}_{B} .\end{aligned}$$

Using the definition of the updated prices,

$$\begin{aligned} B &= \sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) p_j - \sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) \sum_{l \in S_1} \mathbb{P}_j(l \prec S_{1+} \backslash \{l\}) p_l \\ &= \sum_{j \in S_2} \mathbb{P}_i(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_j - \underbrace{\sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) \sum_{l \in S_1} \mathbb{P}_j(l \prec S_{1+} \backslash \{l\}) p_l}_{C} .\end{aligned}$$

We can now combine A and C,

$$\begin{aligned} A - C &= \sum_{l \in S_1} \left(\mathbb{P}_i(l \prec S_{1+} \backslash \{l\}) - \sum_{j \in S_2} \mathbb{P}_i(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) \mathbb{P}_j(l \prec S_{1+} \backslash \{l\}) \right) p_l \\ &= \sum_{l \in S_1} \left(\mathbb{P}_i(l \prec S_{1+} \backslash \{l\}) - \mathbb{P}_i(S_2 \prec l \prec S_{1+} \backslash \{l\}) \right) p_l \\ &= \sum_{l \in S_1} \mathbb{P}_i(l \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_l. \end{aligned}$$

Putting everything together, we get

$$(p_i^{S_1})^{S_2} = p_i - \sum_{j \in (S_2 \cup S_1)} \mathbb{P}_i(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_j = p_i^{S_1 \cup S_2}.$$

□

D.5 Application of Algorithm 7.3 to MNL

In the MNL model, we are given a collection of items, $1, \ldots, n$, along with the no-purchase option, which is denoted by item 0. Each item i has a utility parameter u_i and a price p_i. Without loss of generality, we can assume that $\sum_{i=0}^{n} u_i = 1$. For any given assortment S, each item $i \in S$ is picked with probability

$$\pi(i, S) = \frac{u_i}{u_0 + \sum_{i \in S} u_i},$$

making the expected revenue

$$R(S) = \sum_{i \in S} \frac{u_i}{u_0 + \sum_{\ell \in S} u_\ell} p_i.$$

Blanchet et al. [25] prove that the MNL choice model is a special case of the Markov chain model. More precisely, when $\rho_{ij} = u_j$ for all j and $\lambda_i = u_i$ for all i, the choice probabilities of the two models are identical. In this special case, our local ratio updates can be written as

$$p_i^S = \begin{cases} 0 & \text{if } i \in S \\ p_i - \sum_{j \in S} \frac{u_j}{u_0 + \sum_{\ell \in S} u_\ell} p_j & \text{otherwise.} \end{cases}$$

Note that in the above update, the subtracted term is independent of i. Therefore, the ordering of the prices does not change after each update. Since we are picking the highest adjusted price item at each step, it follows that the optimal assortment is nested by price, i.e., consists of the top ℓ priced items, for some ℓ. This is a well known structural property that we recover here as a direct consequence of our algorithm. Moreover, the updated prices provide a criteria for when to stop adding items to the assortment.

D.6 Algorithm 7.5 with Varying Threshold

Recall the consideration set used in Algorithm 7.5:

$$C_t = \left\{ i \in \mathcal{N} \backslash S_{t-1} : R^{S_{t-1}}(\{i\}) \geq \alpha \frac{R(S^*)}{k} \right\}.$$

Now instead of using the same threshold constant α, we allow it to vary with the iteration t. In other words, let threshold of the consideration set C_t in iteration t be $\alpha_{t,k} \frac{R(S*)}{k}$ for some $\alpha_{t,k} \geq 0$ for all $t = 1, \ldots, k$.

Proposition D.1. *No choice of $\alpha_{t,k}$ can give an approximation ratio better than $1/2 + O(1/k)$ using our current line of analysis, i.e. relying on direct implications of Lemmas 7.9 and 7.11.*

Proof. Suppose the algorithm terminates on an instance after some $k' \leq k$ iterations because the consideration set empties. By Lemma 7.9, we get an approximation bound of $\frac{R(S^*)}{k} \sum_{t=1}^{k'} \alpha_{t,k}$. On the other hand, one can strengthen Lemma 7.11 to say that the lost in incremental revenue of any individual item in $S^*\backslash S_{k'}$ is upper bounded by $\alpha_{\max_{k'},k}\frac{R(S^*)}{k}$, where $\alpha_{\max_{k'},k} = \max_{t=1,\ldots,k'} \alpha_{t,k}$. Together with Lemma 7.9, we get

$$R(S_{k'}) \geq R(S^*) - |S^*\backslash S_{k'}| \cdot \alpha_{\max,k} \cdot R(S^*)/k \geq (1-\alpha_{\max,k})R(S^*).$$

Putting the two bounds together, for a fixed $k' \leq k$, we get an approximation bound of

$$\max\big(\frac{R(S^*)}{k}\sum_{t=1}^{k'}\alpha_{t,k},\ (1-\alpha_{\max_{k'},k})R(S^*)\big)$$

Thus, the worst case bound is

$$\min_{k'\leq k}\max\big(\frac{R(S^*)}{k}\sum_{t=1}^{k'}\alpha_{t,k},\ (1-\alpha_{\max_{k'},k})R(S^*)\big).$$

We would like to show that the above expression is at most $1/2 + O(1/k)$.

Given a sequence of $\alpha_{1,k},...,\alpha_{k,k}$. Take k^* denote the smallest $t \leq k$ such that $\alpha_{t,k} \geq 1/2$. (If k^* does not exist, then in the event that the consideration set is not empty after iteration t, our current analysis leads to an approximation factor worse than 1/2.) Now, we have

$$\frac{R(S^*)}{k}\sum_{t=1}^{k*}\alpha_{t,k} \leq R(S^*)(1/2+1/k),$$

and

$$\big(1-\alpha_{\max_{k'},k}\big)R(S^*) = \big(1-\alpha_{k^*,k}\big)R(S^*) \leq R(S^*)/2$$

Thus, we get

$$\min_{k'\leq k}\max\big(\frac{R(S^*)}{k}\sum_{t=1}^{k'}\alpha_{t,k},\ (1-\alpha_{\max_{k'},k})R(S^*)\big) \leq R(S^*)(1/2+1/k),$$

as desired. □