PRIOR ROBUST OPTIMIZATION

By

Balasubramanian Sivan

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2013

Date of final oral examination: July 2, 2013

This dissertation is approved by the following members of the Final Oral Committee:
Shuchi Chawla (Advisor), Assistant Professor, Computer Sciences
Eric Bach, Professor, Computer Sciences
Jin-Yi Cai, Professor, Computer Sciences
Marzena Rostek, Associate Professor, Economics
Jason D. Hartline, Associate Professor, Electrical Engineering and Computer Science,
Northwestern University

I dedicate this thesis to my parents,
Appadurai Sivan & Anandavalli Sivan.

Acknowledgments

I am indebted to my advisor Prof. Shuchi Chawla for shaping my views on research from the ground up, and for being understanding and patient in this process. Her ability to convincingly articulate the merit of a research direction was very effective in inspiring me to work on a problem, and also in guiding me on how to communicate my research in a talk/paper. Shuchi's crystal clear description of a research problem and the core argument to be constructed there in made research look much less complicated than I imagined. She was a great and convenient source of information and insight. Thanks for all the care and attention you gave me in these past five years Shuchi!

Many thanks to Prof. Jason Hartline for hosting me at Northwestern University several times and also for visiting Wisconsin often. My multiple visits to what we graduate students call as "Jason Hartline Summer Research Institute" will be among the fondest memories of my PhD life. Thanks for the long and fruitful collaboration Jason!

I am grateful to Dr. Nikhil Devanur for hosting me as a summer intern at Microsoft Research Redmond, and also for several subsequent visits. A large part of this thesis is based on joint work with Nikhil while I was at Redmond. I have benefited immensely from Nikhil's stress-free mentoring style, both academically and otherwise. Thanks for being such a joy to work with Nikhil!

Thanks also to Dr. Jennifer Chayes and Dr. Christian Borgs for hosting me as a summer inter at Microsoft Research New England, and for spending an hour or more in weekly meetings in the midst of their jam packed schedule! Both academically and socially, MSR New England was a real fun place to be, thanks to Dr. Madhu Sudan, Dr. Brendan Lucier, Vasilis Syrgkanis, Hu Fu, Nick Gravin and Mickey Brautbar.

Thanks to all my PhD committee members for their valuable time: Prof. Eric Bach, Prof. Jin-Yi Cai, Prof. Shuchi Chawla, Prof. Jason D. Hartline and Prof. Marzena Rostek, and my prelim committee member Prof. Dieter van Melkebeek.

Among the highlights of my stay at Wisconsin are my day long research discussions with David Malec. My stay at Wisconsin would have been much less interesting without you David. Thanks to all my fellow theory graduate students for both making the place more vibrant, and for all the fun stress-busting discussions.

Thanks to all my coauthors. Special thanks to Chris Wilkens and Molly Wilkens for hosting me for one full week at their residence in Berkeley when Chris and I were working on our Single-Call paper. It was really kind of them to have done this, and the stay was totally fun!

I am fortunate to have had Prof. C. Pandu Rangan as my undergraduate mentor. His inviting attitude towards research was almost entirely the reason for me deciding to do a PhD. The TCS lab at IIT Madras, which he manages, could not have been more vibrant!

My childhood friend and genuine well-wisher Suresh, who I am outrageously lucky to have found, was really helpful in maintaining my sanity in PhD life. Our weekly phone calls exchanging summaries of life in Chennai and Madison (and conference travels, and everything under the Sun) was quite refreshing.

For an international graduate student like me in the US, it is quite crucial to have a support system back in the home country with people like my aunt and uncle, Subha and Chandrasekharan. Their extraordinary support during my dad's surgery, which happened while I was in the US and couldn't go back to India, was very important in removing my restlessness and putting me at ease.

Thanks to the most recent entrant to my family, my wife Varsha, for all the words of encouragement and inspiration, and for filling my life with joy. Thanks also to my in-laws for their constant love and concern for my welfare.

Thanks to my grandmother Parvathi for her one of a kind love towards me. The warmth and affection I received from her as a child formed a crucial part of my emotionally secure childhood.

Most importantly, I thank my parents Appadurai Sivan and Anandavalli Sivan for supporting me heart and soul in all my endeavors. They have always put my welfare above everything else in their lives, and have made continued sacrifices in giving me only the best in everything. I affectionately dedicate this thesis to them.

Contents

Abstract

The focus of this thesis is optimization in the presence of uncertain inputs. In a broad class of algorithmic problems, uncertainty is modeled as input being drawn from one among a large known universe of distributions, however the specific distribution is unknown to the algorithm. The goal then is to develop a single algorithm that for every distribution in this universe, performs approximately as well as the optimal algorithm tailored for that specific distribution. Such algorithms are robust to assumptions on prior distributions.

Prior robust optimization retains the robustness of worst-case analysis while going beyond the pessimistic impossibility results of worst-case analysis. Apart from this theoretical appeal, the ability to use the same algorithm for every prior distribution makes prior robust algorithms well-suited for deployment in real systems. Indeed, most prior robust algorithms in literature are simple to implement and some of them have been observed to perform well in large-scale systems.

In this thesis, we design and analyze prior robust algorithms in two distinct areas of research: online algorithms and mechanism design. In online algorithms, we use a hybrid argument to develop near optimal online algorithms for a general framework of problems, called the resource allocation framework, with several well motivated applications to Internet ad serving. In mechanism design, we use sampling and supply limitation techniques to develop prior robust truthful approximately revenue optimal auctions, and the first prior robust truthful mechanisms for approximate makespan minimization in machine scheduling.

1 Introduction

1.1 Prior Robust Optimization

Two among the fundamental challenges that an algorithm designer for optimization problems contends with are:

1. limited computational resources: constrained to develop algorithms that use time and space at most a polynomial in input size;

2. limited access to input: for many problems, obtaining the entire input before the algorithm begins working on them is either a costly matter or simply impossible. The algorithm, therefore, has to to either bear the cost for figuring out the input accurately or make its decisions by knowing only parts of the input.

The first of these challenges has been addressed extensively in Computer Science literature via the theory of approximation algorithms ([Vaz01, WS11]). The focus of this thesis is on the second challenge: in particular, we focus on the challenges posed by:

- online inputs, where the input arrives piece-by-piece and the algorithm is required to act immediately after a piece arrives without knowing the future pieces;

- input being distributed across several selfish agents who reveal their piece of the input only when the algorithm incentivizes them. Incentivizing participants restricts what the algorithm can do on the input so obtained.

We begin by reviewing how one formally models input uncertainty which is at the core of the second challenge. Although there are several approaches to do this, there are two approaches that have gained currency in Computer Science literature, namely, competitive analysis and stochastic analysis.

Competitive Analysis. At one extreme is competitive analysis. Here, the algorithm designer faces input uncertainty while the benchmark is omniscient, i.e., it knows the entire input ahead of time. The performance is measured through what is called competitive ratio: the worst among all possible inputs of the ratio of the performance of the algorithm and that of the benchmark, namely,

$$\text{worst}_{\mathcal{I}} \frac{\text{ALG}(\mathcal{I})}{\text{OPT}_{\mathcal{I}}}.$$

Note that OPT has a subscript $\mathcal{I}$ denoting that the benchmark is *instance-wise optimal*. For problems with online inputs, this means that OPT knows the entire input ahead of time, for problems with selfish participants this again means that OPT has access to the entire input ahead of time and is therefore not required to incentivize participants to get these inputs. This benchmark is clearly a strong one since it is explicitly more powerful than ALG, and thus any positive result with this benchmark is remarkable. However because of its strength, it often leads to pessimistic bounds on what is achievable.

An immediate question is why should the benchmark not be the optimal algorithm that also faces input uncertainty, i.e., optimal online algorithm or the optimal incentive compatible mechanism. The trouble with such a benchmark is that the optimal online algorithm is not well defined: any online algorithm can be modified to perform slightly worse on one input and slightly better on some other input, and thus there is no such thing as an optimal online algorithm. Same goes with optimal incentive compatible algorithms: performance on one input can often be compromised to improve performance on some other input. To circumvent this problem, the benchmark is made omniscient and hence knows the entire input ahead of time.

Stochastic Analysis. At the other extreme is stochastic analysis. Here, it is assumed that the input is drawn from a *known* distribution, and both the algorithm and the benchmark know this distribution, but not the entire input. The performance is measured through the ratio of the expected performance of the algorithm and that of the benchmark, i.e.,

$$\frac{\mathbf{E}_{\mathcal{I}\sim F}\,\text{ALG}_F(\mathcal{I})}{\mathbf{E}_{\mathcal{I}\sim F}\,\text{OPT}_F(\mathcal{I})}.$$

Note that both ALG and OPT have just F in their subscript, i.e, they both know only the distribution, but not the entire instance. For problems with online inputs this corresponds to the benchmark of the expected optimal online solution, and for problems with selfish participants this corresponds to the benchmark of the expected optimal incentive compatible solution. The problems that occurred while defining these quantities in competitive analysis vanish when we use expectations in definition: since the relative importance of each input is embedded in the distribution, there is a well-defined optimal way to tradeoff between different inputs. Absence of such an optimal tradeoff was precisely the difficulty in defining this benchmark in competitive analysis.

Since OPT is not explicitly more powerful than ALG, one can shoot even for a performance as good as OPT, thus circumventing the pessimistic bounds in competitive analysis. However, this model is subject to two major criticisms: one, input distributions may be hard to obtain; two, we risk over-fitting by designing algorithms for specific distributions, and, estimation errors render such algorithms suboptimal.

Prior Robust Analysis. Given the above two extremes in modeling input uncertainty, a possible middle ground is as follows: the input is drawn from some distribution. But the algorithm designer is unaware of the specific distribution. He only knows the huge universe of distributions to which the actual distribution belongs. The benchmark, on the other hand, knows the exact distribution, and corresponds to the optimal algorithm tailored specifically for that distribution. This corresponds to the expected optimal online solution for problems with online inputs, and the expected optimal incentive compatible solution for problems with selfish participants. The performance is measured through the worst among all distributions in the universe, of the ratio of the expected performance of the algorithm and that of the benchmark, i.e.,

$$\mathrm{worst}_{F \in U} \frac{\mathbf{E}_{\mathcal{I} \sim F} \, \mathrm{ALG}(\mathcal{I})}{\mathbf{E}_{\mathcal{I} \sim F} \, \mathrm{OPT}_F(\mathcal{I})}. \tag{1.1}$$

Note that the benchmark has a subscript F denoting that the benchmark is optimally tailored for a specific distribution, where as the algorithm has to be the same for every distribution. So what we are asking for in prior robust optimization is *a single algorithm which, for every distribution in the universe, performs approximately as well as the optimal algorithm designed specifically for that distribution.* It is in this sense that these algorithms are prior robust, i.e., they are robust to the actual prior distribution on inputs. Such algorithms are also well suited for deployment in real world systems because they make very little assumptions on inputs, and hence are robust to input fluctuations.

An Even Stronger Benchmark. Sometimes, we can give our guarantees against an even stronger benchmark, namely, we measure our performance as:

$$\text{worst}_{F \in U} \frac{\mathbf{E}_{\mathcal{I} \sim F} \, \text{ALG}(\mathcal{I})}{\mathbf{E}_{\mathcal{I} \sim F} \, \text{OPT}_{\mathcal{I}}}. \tag{1.2}$$

Note that the benchmark is not just optimal for a distribution, but it is an instance-wise optimal benchmark just as in competitive analysis. In other words, for online inputs the benchmark is the expected optimal offline solution, and for problems with selfish participants, the benchmark is the expected optimal solution that is not constrained to be incentive compatible. This is the strongest benchmark possible in the presence of a distribution. Our results in Chapters 2 and 4 use this benchmark.

Pushing to the Two Extremes. Prior robust analysis can be pushed to the two extremes of competitive and stochastic analysis as follows: when the universe U contains all possible distributions, including point masses, prior robust analysis is the same as competitive analysis. When the universe contains exactly one distribution, prior robust analysis becomes stochastic analysis. The goal in general is to develop prior robust algorithms for as rich a universe as possible.

Approximation Ratio/Factor. We use the term approximation ratio/factor to denote the ratios in expressions (1.1) and (1.2). Which of these two is being used will be made explicit while we use it.

The focus of this thesis is the design and analysis of prior robust algorithms for two distinct areas of research: online algorithms and mechanism design. Along the way, we highlight the versatility of broadly applicable techniques in developing these algorithms.

1.2 Motivating Examples

Before stating our results formally, we provide two motivating examples for designing prior robust algorithms.

Revenue Maximization in Internet Advertising. Every popular website you are aware of is likely to display one or more banner advertisements alongside the content it offers. These advertisements are called "display ads" (different from sponsored search ads that you see in search engines), and are used by firms as a means to promote their brand. In terms of revenue, the display ads industry is a multi-billion dollar industry and is projected to get ahead of even the incredibly lucrative search ads industry in a few years. At a high level this is the mechanism to get an displayed in a website: the advertiser signs a contract with the website that reads something like "in the coming month 5 million impressions of my ad will be served to males in the age group 21 to 35 in the Chicago area for 10 cents per impression". The website signs such contracts with several advertisers, and, has to pay a penalty per impression when it fails to honor the contract. When an "impression" arrives, i.e., a user visits the website, the decision of which ad to display from among the many ads that have asked for being shown to this user has to be made. This decision is clearly an online problem because the decision has to be made without knowing which users will arrive in the future. The "supply" is relatively constant, i.e., the total number of user visits to a website doesn't vary much and hence is known relatively accurately. The website must therefore be judicious in using its limited supply of user visits, so as to honor all of its contract and still doing well in terms of revenue.

What constitutes a good algorithm in this setting? If we resort to competitive analysis to answer this question, we would be comparing with the offline optimal algorithm as our benchmark. Even in the special case where all the advertisers ask for the same number of impressions, all of them have the same price per impression and the same penalty per impression, nothing better than a $1-1/e$ ($\sim 63\%$) approximation is possible even with a randomized algorithm [MSVV05]. Can we get approximations close to 1 if we move away from competitive analysis? If we resort to stochastic analysis, the suggested algorithm will be very dependent on the distribution. Given this, it would be quite useful to have a single algorithm that performs very well and obtains a close-to-1 approximation regardless of the underlying distribution from which the impressions arrive. How do we go about designing algorithms with such strong guarantees? Answer in Chapter 2.

Revenue Maximization in Auctions. Think of your neighbor Bob who sells antiques in eBay via the standard second price auctions that eBay permits: the highest bidder wins the item and pays the seller the second highest bid. The seller is also allowed to put a minimum bid (called as a "reserve" bid) so that the winner pays the minimum of the reserve bid and the second highest bid. Naturally Bob likes to maximize his revenue and is unable to sleep well until he is convinced that he has done everything from his side to run an optimal auction. When can Bob be convinced that he is running a good auction?

Consider a simple instance with just two bidders: one has a value (the maximum price that a bidder is willing to pay) of \$100 and the other has a value of \$10. Bob obviously doesn't know the values of his bidders before the auction begins. What constitutes a good auction here? If Bob pessimistically uses competitive analysis to measure the goodness of his auction, he is out of luck: the optimal auction will place a reserve of \$100, and hence get a revenue of \$100. Bob on the other hand is completely in the dark about what reserve to place. Whatever reserve Bob places, there are bidder values for which the obtained revenue is a vanishingly small fraction of the optimal revenue. If Bob resorted to stochastic analysis, namely, assume that his bidders' values are drawn from distributions F_1 and F_2, then auction theory [Mye81] will tell him the optimal auction. But, the auction it suggests is quite complex and often outside what is permitted by eBay. Even in cases where the optimal auction turns out to be within the scope of eBay's permitted auctions, it requires Bob to know the distribution well. The best option for Bob therefore would be to make no assumptions about distributions, but still be able to provide a provable guarantee of his auction's performance. How exactly should Bob go about designing such an auction? Answer in Chapter 3.

1.3 Our Contributions

We now summarize our results in this thesis. For an in depth connection to related work and surrounding discussions, the reader is referred to the respective chapters. As mentioned earlier, in this thesis we design and analyze prior robust algorithms for two distinct areas of research: problems with online inputs and problems with selfish participants.

1.3.1 Prior Robust Optimization in Internet Advertising

There has been an increasing interest in online algorithms motivated by applications to online advertising. The most well known problem is the *Adwords* problem introduced by Mehta et al. [MSVV05], (defined formally in Chapter 2), that models the algorithmic problem involved in search engine revenue optimization . Here the algorithm needs to assign keywords arriving online to bidders to maximize profit, subject to budget constraints for the bidders. The problem has been analyzed in the traditional framework for online algorithms: worst-case competitive analysis. As with many online problems, worst case competitive analysis is not entirely satisfactory and there has been a drive in the last few years to go beyond the worst-case analysis. The predominant approach has been to assume that the input satisfies some stochastic property. For instance the *random permutation* model (introduced by Goel and Mehta [GM08]) assumes that the adversary picks the set of keywords, but the order in which the keywords arrive is chosen uniformly at random. We refer the reader to the discussion in Chapter 2 for the commonly assumed stochastic properties and how they compare against each other. For this work, we assume the closely related *i.i.d.* model: assume that the input (e.g., keywords in the Adwords problem) are i.i.d. samples from a fixed distribution, which is *unknown* to the algorithm. In this model, the universe of distributions U is the set of all possible i.i.d. distributions over the queries.

Informal Summary of Results in Chapter 2.

1. **Near-optimal online algorithm for resource allocation framework.** First, we define a general framework of online problems called the resource allocation framework, which is a significant generalization of several interesting special cases including display ads, Adwords, online network routing, online combinatorial auctions etc. Informally, m requests (think of keywords in Adwords) arrive online and have to be served using n available resources. Each resource has a capacity (think of advertiser budgets in Adwords) and cannot be used beyond capacity. There are different options to serve a request, and different options are associated with different profits (think of a query's bid differing across advertisers in Adwords). In the i.i.d. model, the requests are i.i.d samples from a fixed distributions that is unknown to the algorithm.

   We design a near-optimal prior robust algorithm, whose approximation ratio gets arbitrarily close to 1 when the revenue and capacity consumption of each request gets less significant compared to the total set of requests that arrive. Further, we show that the rate at which the approximation ratio tends to 1 is almost optimal: we show that no algorithm, even those that have complete knowledge of the distribution, can get a faster convergence to 1.

In contrast, in the competitive setting, no algorithm can get beyond a $1 - 1/e$ approximation even for the special case of Adwords even if all the bids are infinitesimally small compared to the advertiser budgets.

2. **Near-optimal online algorithm for adwords.** Due to its significance, we single out the special case of Adwords and design an online algorithm that goes beyond what's possible for the general resource allocation framework. We design an algorithm whose approximation ratio converges to 1 faster than our convergence rate for the general resource allocation framework, and, we show that this is the fastest convergence rate any algorithm can guarantee even if it knew the distribution completely.

   Unlike our algorithm for resource allocation, our Adwords algorithm needs the knowledge of a few parameters from the distribution in order to accomplish this faster convergence ratio. An intriguing open question is whether this improvement for Adwords is possible without any knowledge of distribution at all.

3. **Fast offline approximation algorithm for large mixed packing & covering IPs.** We consider a class of mixed covering and packing integer programs (IPs) inspired by the resource allocation framework, and design a fast *offline* approximation algorithm for these integer programs based on our online algorithm. Our algorithm is randomized, and solves the offline problem in an online manner via sampling. The basic idea is to conclude the feasibility of the original integer program using the feasibility of the sampled portion of the integer program (the latter is found by our online algorithm). Problems in the resource allocation framework where the instances are too large to use traditional algorithms occur fairly often, especially in the context of online advertising. Often approximate but quick solutions are preferable.

4. **Beyond i.i.d.: The Adversarial Stochastic Input (ASI) Model.** One

drawback of the i.i.d. model is that the distribution does not vary over time, i.e., at every time step, a request is drawn from the same distribution. In settings like Adwords or display ads, different periods of the day have different distributions, and inspired by this, we generalize the i.i.d. model to the adversarial stochastic input model. Informally, distributions are allowed to vary over time, and an adversary is allowed to pick, even adaptively, which distribution will get used at a particular time step. We provide three different generalizations of the i.i.d. model, and extend our results to all these three models with strong guarantees similar to the i.i.d. unknown model. Our guarantees are against the worst distribution the adversary picks. If given the help of a few distributional parameters, the same guarantee extends for the average of the distributions which the adversary picks.

1.3.2 Prior Robust Revenue Maximizing Auction Design

Consider the sale of a single item, with n interested buyers. Buyer i has a value v_i for the item. Assume that each v_i is drawn independently from a distribution F. For this setting auction theory [Mye81] suggests the revenue optimal auction. This auction however requires the knowledge of a distribution dependent parameter to be implemented. Can we somehow do without knowledge of the distribution? Bulow and Klemperer [BK96] showed that when the distribution F satisfies a technical regularity condition (see Chapter 3 for the definition) recruiting one more agent drawn from the same distribution F, and just running the second price auction on the $n+1$ agents will give at least as much revenue as the revenue optimal auction on n agents. Informally, advertising and spreading the word about the auction to boost the number of bidders (which is clearly prior robust) pays off better than performing market analysis to determine the bidder distributions.

While the technical regularity condition includes many natural distributions, it is also violated by simple ones like multimodal distributions (in fact, even bimodal ones). For instance, the convex combination of two regular distributions is likely to result in a bimodal distribution. This situation of a bidder's value being a convex combination of two regular distributions is among the most common reasons for regularity being violated in an auction situation: in Bob's eBay auction in the motivating example, it could well be that each bidder for Bob is a professional antique collector with probability p_p and an amateur with probability p_a, with respective distributions F_p

and F_a. While F_p and F_a could satisfy the technical regularity condition, their convex combination $p_pF_p + p_aF_a$ is often not regular. That is, a heterogeneous market is quite likely to result in bidders who violate the regularity condition. Is there a natural extension of Bulow and Klemperer's result to this widely prevalent setting?

Informal Summary of Results in Chapter 3.

1. **One extra bidder from every regular distribution.** When the distributions of each of the n agents can be expressed as convex combinations of k regular distributions (possibly different convex combinations, i.e., the n agents could be from different distributions), we prove that recruiting one agent from each of these k regular distributions, and running the second price auction on these $n + k$ agents without any reserve price will give at least half the revenue of the optimal auction on the original n agents. Informally, while advertising was the solution for regular distributions, a targeted advertising campaign is the solution for heterogeneous markets with irregular distributions. In Bob's auction with professionals and amateurs, if Bob manages to bring in one additional professional and amateur and run the second price auction, he gets at least half the optimal revenue in the original setting with n bidders.

2. **Just one extra bidder for hazard rate dominant distributions.** If the k underlying regular distributions are such that one of them hazard rate dominates the rest (see Chapter 3 for a definition; for example if all the k underlying distributions are exponentials with possibly different rates, or power law distributions with possibly different scales, or uniform distributions over possibly different intervals, one of them is guaranteed to hazard rate dominate the rest), recruiting just one additional agent from the hazard rate dominant distribution gives at least half as much revenue as the optimal auction run on the original n agents.

1.3.3 Prior Robust Mechanisms for Machine Scheduling

Scheduling jobs in machines to minimize the completion time of the last job (namely, makespan) is a central problem in Computer Science, particularly in the context of resource allocation. Even from an Economic viewpoint, minimizing makespan is relevant because it can be interpreted as maximizing fairness across machines. For the field of Algorithmic Mechanism Design, makespan minimization is the seminal problem and was introduced by Nisan and Ronen [NR99]. In the problem they introduce, there are m machines and n jobs. The machines are operated by selfish agents who hold the runtimes of jobs as private, and will reveal them only if the scheduling algorithm properly incentivizes them. Nisan and Ronen [NR99] design a truthful (incentive compatible) mechanism that obtains a m approximation in the competitive setting, i.e., the benchmark is the optimal makespan possible without the truthfulness constraint. Ashlagi, Dobzinski and Lavi [ADL09] later showed that this is almost the best one can hope for in the competitive setting: they showed that for a large class of natural mechanisms, nothing better than a m approximation is achievable.

In this work, just as for resource allocation problems, we ask, whether we can get better approximation guarantees if we go beyond competitive analysis. While the literature on prior robust mechanisms has primarily focused on the linear objective of revenue maximization [HR09, DRY10, RTCY12], in this work, with appropriate distributional assumptions, we give the first non-trivial guarantees for the non-linear objective of makespan minimization .

Informal Summary of Results in Chapter 4.

1. **An $O(n/m)$ approximation with i.i.d. machines and non-i.i.d. jobs.** When the runtimes of a job on different machines are i.i.d. random variables (but different jobs could have different distributions), we design a prior robust truthful mechanism that obtains a $O(n/m)$ approximation to $\mathrm{OPT}_{1/2}$, which is the optimal makespan without incentive compatibility constraints that can be obtained by using at most $m/2$ machines. This result says that if we augment our resources by doubling the number of machines and use our mechanism, we get a $O(n/m)$ approximation to OPT. When $n = O(m)$, this gives a constant factor approximation. In contrast, in the competitive setting, even when $n = m$ the Ashlagi et al. [ADL09] hardness of approximation of factor m remains intact.

   Further, for a large class of distributions, including those that satisfy the Monotone Hazard Rate (MHR) property (intuitively, those that have tails no heavier than the exponential distribution: this includes the uniform, normal, exponential distributions etc.), we show that $\mathrm{OPT}_{1/2}$ and OPT are within a factor of 4. That is, for this class of distributions, even without resource augmentation, we get a $O(n/m)$ approximation.

2. **Sublogarithmic approximations with i.i.d. machines and jobs.** The above result gives good approximations when n is relatively small. When $n \geq m \log m$ we design another mechanism, which, with the additional assumption that the jobs are also identically distributed, obtains a $O(\sqrt{\log m})$ approximation to $\mathrm{OPT}_{1/3}$. Further, if the distributions belong to the large class of distributions mentioned in the previous paragraph (including MHR distributions), we obtain $O(\log \log m)^2$ approximation to OPT.

1.4 Prerequisites and Dependencies

Prerequisites. All necessary technical prerequisites are provided in the respective chapters. Apart from those provided in the chapters, familiarity with basic probability, expectations, and big O notation is all that is assumed.

Dependencies. The chapters are self contained, and can be read in any order.

1.5 Bibliographic Notes

The work presented in this thesis is contained in the research papers [DJSW11, DSA12, SS13, CHMS13].

1. The results in Chapter 2 are based on two joint works [DJSW11, DSA12]. Our results for general resource allocation [DJSW11] are based on joint work with Nikhil Devanur, Kamal Jain and Chris Wilkens. The improved results for the special case of Adwords are based on joint work [DSA12] with Nikhil Devanur and Yossi Azar.

2. The results in Chapter 3 are based on joint work [SS13] with Vasilis Syrgkanis.

3. The results in Chapter 4 are based on joint work [CHMS13] with Shuchi Chawla, Jason D. Hartline and David Malec.

2 Prior Robust Optimization in Internet Advertising

Organization. In this chapter, we design and analyze prior robust algorithms for revenue maximization in Internet advertising. The chapter is organized as follows. In Section 2.1 we informally summarize our results, put them in context of related work with some additional discussion. In Section 2.2 we provide all the necessary preliminaries and the formal statements of all our results in this chapter. In Section 2.3 we design and analyze prior robust online algorithms for the resource allocation framework, which is a strict generalization of the Adwords problem, the "display ads" problem and other interesting special cases. In Section 2.4 we prove that the approximation factor we get in Section 2.3 is almost optimal. In Section 2.5, we consider the interesting special case of our resource allocation framework, namely, the Adwords problem, and design and analyze an online algorithm that obtains an improved approximation over what is possible for the general case in Section 2.3, and we prove its tightness in Section 2.6. In Section 2.7 we analyze the greedy algorithm and show that it obtains a $1 - 1/e$ approximation for the Adwords problem with a large bid-to-budget ratio, i.e., large γ, improving over the previous best known trivial factor of $1/2$. In Section 2.8 we use our online algorithms to give fast approximation algorithms for mixed packing and covering problems. In Section 2.9 we discuss some special cases of the resource allocation framework and conclude with some open questions in Section 2.10.

2.1 Introduction & Summary of Results

The results in this chapter fall into distinct categories of prior robust algorithms for online problems and fast approximation algorithms for offline problems. However they all share common techniques.

There has been an increasing interest in online algorithms motivated by applications to online advertising. The most well known is the *Adwords* problem introduced by Mehta et al. [MSVV05], where the algorithm needs to assign keywords arriving online to bidders to maximize profit, subject to budget constraints for the bidders. The problem has been analyzed in the traditional framework for online algorithms: worst-case competitive analysis. As with many online problems, the worst-case competitive analysis is not entirely satisfactory and there has been a drive in the last few years to go beyond the worst-case analysis. The predominant approach has been to assume that the input satisfies some stochastic property. For instance the *random permutation* model (introduced by Goel and Mehta [GM08]) assumes that the adversary picks the set of keywords, but the order in which the keywords arrive is chosen uniformly at random. A closely related model is the *i.i.d.* model: assume that the keywords are i.i.d. samples from a fixed distribution, which is *unknown* to the algorithm. Stronger assumptions such as i.i.d. samples from a *known* distribution have also been considered.

First Result: Near-Optimal Prior Robust Online Algorithms for Resource Allocation Problems. A key parameter on which many of the algorithms for Adwords depend is the bid to budget ratio, which measures how significant any single keyword/query is when compared to the total budget. For instance Mehta et al. [MSVV05] and Buchbinder, Jain and Naor [BJN07] design an algorithm that achieves a worst case competitive ratio that tends to $1 - 1/e$ as the bid to budget ratio (which we denote by γ) tends to 0. (Note that γ approaching zero is the easiest case. Even with γ approaching zero, $1 - 1/e$ is the best competitive ratio that any randomized algorithm can achieve in the worst case.) Devanur and Hayes [DH09] showed that in the random permutation model, the competitive ratio tends to 1 as γ tends to 0. This result showed that competitive ratio of algorithms in stochastic models could be much better than that of algorithms in the worst case. The important question since then has been to determine the optimal trade-off between γ and the competitive ratio. [DH09] showed how to get a 1- $O(\epsilon)$ competitive ratio when γ is at most $O(\frac{\epsilon^3}{n\log(mn/\epsilon)})$ where n is the number of advertisers and m is the number of keywords. Subsequently Agrawal, Wang and Ye [AWY09] improved the bound on γ to $O(\frac{\epsilon^2}{n\log(m/\epsilon)})$. The papers of Feldman et al. [FHK+10] and Agrawal, Wang and Ye [AWY09] have also shown that the technique of [DH09] can be extended to other online problems.

The first main result in this paper is the following 3-fold improvement of previous results: (Theorems 2.2 and 2.3)

1. We give an algorithm which guarantees a $1 - \epsilon$ approximation factor when $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$ This is almost *optimal*; we show that no algorithm, even if

it knew the distribution, can guarantee a $1-\epsilon$ approximation factor when $\gamma = \omega(\frac{\epsilon^2}{\log(n)})$.

2. The bound applies to a more general model of stochastic input, called the *adversarial stochastic input* model. This is a generalization of the i.i.d. model with unknown distribution, in that the distributions are allowed to change over time. We provide three different generalizations in Section 2.3.5.

3. Our results generalize to a more general class of online problems that we call the *resource allocation framework*. A formal definition of the framework is presented in Section 2.2.2 and a discussion of many interesting special cases is presented in Section 2.9.

Significance. Regarding the bound on γ, the removal of the factor of n is significant. Consider for instance the Adwords problem and suppose that the bids are all in [0,1]. The earlier bound implies that the budgets need to be of the order of n/ϵ^2 in order to get a $1-\epsilon$ competitive algorithm, where n is the number of advertisers. With realistic values for these parameters, it seems unlikely that this condition would be met. While with the improved bounds presented in this paper, we only need the budget to be of the order of $\log n/\epsilon^2$ and this condition is met for reasonable values of the parameters. Furthermore, in the more general resource allocation framework, the current highest upper bound on γ is from Agrawal, Wang and Ye [AWY09] and equals $O(\frac{\epsilon^2}{n\log(mK/\epsilon)})$. Here K is the number of available "options" (see Section 2.2.2) and in typical applications like network routing, k could be exponential in n, and thus, the factor saved by our algorithm becomes quadratic in n.

I.I.D. vs Random Permutations. We note here that so far, all the algorithms for the i.i.d. model (with unknown distribution) were actually designed for the random permutation model. It seems that any algorithm that works for one should also work for the other. However we can only show that our algorithm works in the i.i.d. model, so the natural question is if our algorithm works for the random permutation model. It would be very surprising if it didn't.

Adversarial Stochastic Input (ASI) Model. One drawback of the stochastic models considered so far is that they are time invariant, that is the input distribution remains the same for every request. The adversarial stochastic input model allows the input distribution to change over time. The formal definitions of these three generalizations are presented while we use them in Section 2.3.5

Second Result: Improved Online Algorithms for Adwords using a few Parameters from the Distribution. Due to its significance, we single out one special case of the resource allocation framework, namely, the Adwords problem, for further study. Is it possible to design algorithms that go beyond what is possible for the general resource allocation framework, i.e., is it to possible to get a $1-\epsilon$ approximation algorithm for the Adwords problem for $\gamma = \omega(\epsilon^2/\log(n))$. In our second result (Theorem 2.4), we design an online algorithm for adwords that obtains $1-\epsilon$ approximation for Adwords whenever $\gamma = O(\epsilon^2)$. We also show that this is the best one can hope for: no algorithm can guarantee, even if it knew the complete distribution, a $1-\epsilon$ approximation for $\gamma = \omega(\epsilon^2)$.

This algorithm however is not completely prior robust: the algorithm we design requires n parameters from the distribution. If we need a completely prior robust algorithm, then the slightly worse result for the general resource allocation framework (Theorem 2.2) is the best known currently. However we note here that requiring n parameters from the distribution is information theoretically strictly weaker than asking for the knowledge of the entire distribution which could have an exponential (even infinite) support.

Open Question. Design a prior robust algorithm for adwords that guarantees a $1-\epsilon$ approximation for adwords when $\gamma = O(\epsilon^2)$.

Third Result: Prior Robust $1-1/e$ approximation Greedy Algorithm for Adwords. A natural algorithm for the Adwords problem that is widely used for its simplicity is the greedy algorithm: always match an incoming query to the advertiser that has the maximum effective bid (the minimum of bid and remaining budget) for that query. Because of its wide use, previously the performance of the greedy algorithm has been analyzed by Goel and Mehta [GM08] who showed that in the random permutation and the i.i.d. models, it has a competitive ratio of $1-1/e$ with an assumption which is essentially that γ tends to 0.

It has been an important open problem to analyze the performance of greedy algorithm in a stochastic setting for unbounded γ, i.e., for all $0 \leq \gamma \leq 1$. The best factor known so far is $1/2$, and this works for the worst case also. Nothing better was known, even in the stochastic models. *The third result in this chapter is that for the Adwords problem in the i.i.d. unknown distributions model, with no assumption on γ (i.e., γ could be as big as 1), the greedy algorithm gets an approximation factor of $1 - 1/e$ against the optimal fractional solution to the expected instance (Theorem 2.8).*

We note here that there are other algorithms that achieve a $1 - 1/e$ approximation for the Adwords problem with unbounded γ, but the greedy algorithm is the only prior robust algorithm known, and it quite simple too. For example, our our second result (Theorem 2.4) will become a $1-1/e$ approximation at $\gamma = 1$ (see Section 2.5 for how we get $1 - 1/e$ at $\gamma = 1$), but it requires a few parameters from the distribution. Similarly Alaei et al. [AHL12] design a randomized algorithm that obtains a $1 - 1/e$ approximation, but requires the knowledge of the entire distribution.

Fourth Result: Fast Approximation Algorithms for Mixed Packing and Covering Integer Programs. Charles et al. [CCD+10] considered the following (offline) problem: given a lopsided bipartite graph $G = (L, R, E)$, that is a bipartite graph where $m = |L| \gg |R| = n$, does there exist an assignment $M : L \to R$ with $(j, M(j)) \in E$ for all $j \in L$, and such that for every vertex $i \in R$, $|M^{-1}(i)| \geq B_i$ for some given values B_i. Even though this is a classic problem in combinatorial optimization with well known polynomial time algorithms, the instances of interest are too large to use traditional approaches to solve this problem. (The value of m in particular is very large.) The approach used by [CCD+10] was to essentially design an online algorithm in the i.i.d. model: choose vertices from L uniformly at random and assign them to vertices in R in an online fashion. The online algorithm is guaranteed to be close to optimal, as long as sufficiently many samples are drawn. Therefore it can be used to solve the original problem (approximately): the online algorithm gets an almost satisfying assignment if and only if the original graph has a satisfying assignment (with high probability).

The fourth result in this chapter is a generalization of this result to get fast approximation algorithms for a wide class of mixed packing and covering integer programs (IPs) inspired by problems in the resource allocation framework (Theorem 2.9). Problems in the resource allocation framework where the instances are too large to use traditional algorithms occur fairly often, especially in the context of online advertising. Formal statements and a more detailed discussion are presented in Section 2.2.5.

High Level Description of Techniques. The underlying idea used for all these results can be summarized at a high level as thus: consider a hypothetical algorithm called *Hypothetical-Oblivious* that knows the distribution from which the input is drawn and uses an optimal solution w.r.t. this distribution. Now suppose that we can analyze the performance of Hypothetical-Oblivious by considering a potential function and showing that it decreases by a certain amount in each step. Now we can design an algorithm that does not know the distribution as follows: consider the same potential function, and in every step choose the option that minimizes the potential function. Since the algorithm minimizes the potential in each step, the decrease in the potential for this algorithm is better than that for Hypothetical-Oblivious and hence we obtain the same guarantee as that for Hypothetical-Oblivious. The choice of potential function varies across the results; also, whether we minimize or maximize the the potential function varies across the results.

For instance, our first result (Theorem 2.2), the performance of Hypothetical-Oblivious is analyzed using Chernoff bounds. The Chernoff bounds are proven by showing bounds on the expectation of the moment generating function of a random variable. Thus the potential function is the sum of the moment generating functions for all the random variables that we apply the Chernoff bounds to. The proof shows that in each step this potential function decreases by some multiplicative factor. The algorithm is then designed to achieve the same decrease in the potential function. A particularly pleasing aspect about this technique is that we obtain very simple proofs. For instance, the proof of Theorem 2.8 is extremely simple: the potential function in this case is simply the total amount of unused budgets and we show that this amount (in expectation) decreases by a factor of $1 - 1/m$ in each step where there are m steps in all.

Multiplicative-Weight Updates. Our techniques and the resulting algorithms for our first and fourth results (Theorem 2.2 and Theorem 2.9) bear a close resemblance to the algorithms of Young [You95, You01] for derandomizing randomized rounding and the fast approximation algorithms for solving covering/packing LPs of Plotkin, Shmoys and Tardos [PST91], Garg and Könemann [GK98],Fleischer [Fle00]. In fact Arora, Hazan and Kale [AHK05] showed that all these algorithms are related to the multiplicative weights update method for solving the *experts* problem and especially highlighted the similarity between the potential function used in the analysis of the multiplicative update method and the moment generating function used in the proof of Chernoff bounds and Young's algorithms. Hence it is no surprise that our algorithm which uses Chernoff bounds is also a multiplicative update algorithm. Our algorithm is closer in spirit to Young's algorithms than others. A basic difference of our algorithm from this previous set of results is that our algorithm uses the special structure of the polytope $\sum_k x_{j,k} \leq 1$ (as against the more general polytopes in these works) in giving a more efficient solution both for the offline and online versions of our problem. For instance, for our offline problem the number of oracle calls required will have a quadratic dependence on γm if we used the [PST91] algorithm, where as using the special structure of the polytope, we obtain a linear dependence on γm.

It is possible that our algorithm can also be interpreted as an algorithm for the experts problem. In fact Mehta et al. [MSVV05] asked if there is a $1 - o(1)$ competitive algorithm for Adwords in the i.i.d model with small bid to budget ratio, and in particular if the algorithms for experts could be used. They also conjectured that such an algorithm would iteratively adjust a budget discount factor based on the rate at which the budget is spent. Our algorithms for resource allocation problem when specialized for Adwords look exactly like that and with the connections to the experts framework, we answer the questions in [MSVV05] in the positive.

2.2 Preliminaries & Main Results

2.2.1 Resource Allocation Framework

We consider the following framework of optimization problems. There are n resources, with resource i having a capacity of c_i. There are m requests; each request j can be satisfied by a vector $\mathbf{x}_j \in \{0, 1\}^K$, with coordinates $x_{j,k}$, such that $\sum_k x_{j,k} \leq 1$. Think of vector $\mathbf{x}_j$ as picking an option to satisfy a request from a total of K options (We also overload K to denote the set of options). The vector $\mathbf{x}_j$ consumes $\mathbf{a}_{i,j} \cdot \mathbf{x}_j$ amount of resource i, and gives $\mathbf{w}_{i,j} \cdot \mathbf{x}_j$ amount of type i profit. The $\mathbf{a}_{i,j}$'s and $\mathbf{w}_{i,j}$'s are non-negative vectors of length K (and so are the $\mathbf{x}_j$'s). The co-ordinates of the vectors

$\mathbf{a}_{i,j}$ and $\mathbf{w}_{i,j}$ will be denoted by a_{ijk} and w_{ijk} respectively, i.e., the k^{th} option consumes a_{ijk} amount of resource i and gives a type i profit of w_{ijk}. The objective is to maximize the minimum among all types of profit subject to the capacity constraints on the resources. The following is the linear program relaxation of the resource allocation problem:

$$\begin{aligned}
&\text{Maximize } \min_i \sum_j \mathbf{w}_{i,j} \cdot \mathbf{x}_j \text{ s.t.} \\
&\forall\ i, \sum_j \mathbf{a}_{i,j} \cdot \mathbf{x}_j \leq c_i \\
&\forall\ j, \sum_k x_{j,k} \leq 1 \\
&\forall\ j, k, x_{j,k} \geq 0
\end{aligned}$$

Note that dropping a request by not picking any option at all is feasible too. For expositional convenience, we will denote not picking any option at all as having picked the $\perp$ option ($\perp$ may not be in the set K) for which $a_{ij\perp} = 0$ and $w_{ij\perp} = 0$ for all i, j.

We consider two versions of the above problem. The first is an online version with stochastic input: requests are drawn from an unknown distribution. The second is an offline problem when the number of requests is much larger than the number of resources, and our goal is to design a fast PTAS for the problem.

2.2.2 Near-Optimal Online Algorithm for Resource Allocation

We now consider an online version of the resource allocation framework. Here requests arrive online. We consider the i.i.d. model, where each request is drawn independently from a given distribution. The distribution is unknown to the algorithm. The algorithm knows m, the total number of requests. To define our benchmark, we now define the expected instance.

Expected Instance. Consider the following *expected instance* of the problem, where everything happens as per expectation. It is a single offline instance which is a function of the given distribution over requests and the total number of requests m. Every request in the support of the distribution is also a request in this instance. The capacities of the resources in this instance are the same as in the original instance.

Suppose request j has a probability p_j of arriving in the given distribution. The resource consumption of j in the expected instance is given by $mp_j\mathbf{a}_{i,j}$ for all i and the type i profit is $mp_j\mathbf{w}_{i,j}$. The intuition is that if the requests were drawn from this distribution then the expected number of times request j is seen is mp_j. To summarize, the *LP relaxations* of a random instance with set of requests R, and the expected instance are as follows (slightly rewritten for convenience).

Random Instance R	**Expected Instance**
Maximize λ s.t.	Maximize λ s.t.
$\forall\ i, \sum_{j\in R,k} w_{ijk}x_{j,k} \geq \lambda$	$\forall\ i, \sum_{j,k} mp_j w_{ijk}x_{j,k} \geq \lambda$
$\forall\ i, \sum_{j\in R,k} a_{ijk}x_{j,k} \leq c_i$	$\forall\ i, \sum_{j,k} mp_j a_{ijk}x_{j,k} \leq c_i$
$\forall\ j \in R, \sum_k x_{j,k} \leq 1$	$\forall\ j, \sum_k x_{j,k} \leq 1$
$\forall\ j \in R, k, x_{j,k} \geq 0.$	$\forall\ j, k, x_{j,k} \geq 0.$

LP 2.1: LP relaxations for random and expected instances

We now prove that the fractional optimal solution to the expected instance W_E is an upper bound on the expectation of W_R, where W_R is the offline fractional optimum of the actual sequence of requests in a random instance R.

Lemma 2.1 $W_E \geq \mathbf{E}[W_R]$

Proof: The average of optimal solutions for all possible sequences of requests is a feasible solution to the expected instance with a profit equal to $E[W_R]$. Thus the optimal profit for the expected instance could only be larger. ■

The approximation factor of an algorithm in the i.i.d. model is defined as the ratio of the expected profit of the algorithm to the fractional optimal profit W_E for the expected instance. Let $\gamma = \max\left(\left\{\frac{a_{ijk}}{c_i}\right\}_{i,j,k} \cup \left\{\frac{w_{ijk}}{W_E}\right\}_{i,j,k}\right)$ be the parameter capturing the significance of any one request when compared to the total set of requests that arrive online. The main result is that as γ tends to zero, the approximation factor ratio tends to 1. In fact, we give the almost optimal trade-off.

Theorem 2.2 *For any $\epsilon \geq 1/m$, Algorithm 2 achieves an objective value of $W_E(1 - O(\epsilon))$ for the online resource allocation problem with probability at least $1-\epsilon$, assuming $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$. Algorithm 2 does not require any knowledge of the distribution at all.*

Theorem 2.3 *There exist instances with $\gamma = \frac{\epsilon^2}{\log(n)}$ such that no algorithm, even with complete knowledge of the distribution, can get a $1 - o(\epsilon)$ approximation factor.*

Oracle Assumption. We assume that we have the following oracle available to us: given a request j and a vector $\mathbf{v}$, the oracle returns the vector $\mathbf{x}_j$ that maximizes $\mathbf{v}.\mathbf{x}_j$ among all $\mathbf{x}_j$s in $\{0,1\}^K$ satisfying $\sum_k x_{j,k} \leq 1$. Such an oracle can be implemented in polynomial time for most interesting special cases including Adwords, display ads, network routing. After seeing how this oracle is used in our algorithm, the reader will be able to quickly realize that for the Adwords and display ads problem (described below), this assumption boils down to being able to find the maximum among n numbers in polynomial time. For network routing (described in Section 2.9), this assumption corresponds to being able to find the shortest path in a graph in polynomial time. For combinatorial auctions (described in Section 2.9), this corresponds to the demand query assumption: given prices on various items, the buyer should be able to decide in polynomial time which bundle gives him the maximum utility. While this is not always achievable in polynomial time, there cannot be any hope of a posted pricing solution for combinatorial auction without this minimum assumption.

Extensions and Special Cases. The extensions of Theorem 2.2 to the various generalizations of the i.i.d. model to the adversarial stochastic input model are presented in Section 2.3.5. We refer the reader to Section 2.9 for a discussion on several problems that are special cases of the resource allocation framework and have been previously considered. Here, we discuss two special cases — the Adwords problem and display ads problem.

1. **Adwords.** In the adwords problem, there are n advertisers with a daily budget of B_i for advertiser i. There are m keywords/queries that arrive online, and advertiser i has a bid of b_{ij} for query j. Let x_{ij} denote the indicator variable for whether or not query j was allocated to agent i. After all allocation is over, agent i pays $\sum_j \min(b_{ij}x_{ij}, B_i)$, i.e., the minimum of the sum of the bids for queries allocated to i and his budget B_i. The objective is to maximize the sum

of the payments from all advertisers. Technically this is not a special case of the resource allocation framework because the budget constraint is not binding: the value of the allocated bids to an advertiser can exceed his budget, although the total payment from the advertiser will be at most the budget. But it is not difficult to see that the LP relaxation of the offline problem can be written as in LP 2.2, which is clearly a special case of resource allocation framework LP. Note that the benchmark is anyway an upper bound even on the expected optimal fractional solution. Therefore, any algorithm that gets an α approximation factor for resource allocation is also guaranteed to get the same approximation factor for Adwords. The only notable thing being that an algorithm for resource allocation when used for adwords will treat the budget constraints as binding, and obtain the guarantee promised in Theorem 2.2 (In our $1-1/e$ approximation algorithm for adwords in Section 2.7 that holds for all values of γ (≤ 1 of course), we use this facility to exceed budget).

2. **Display Ads.** In the display ads problem, there are n advertisers and m impressions arrive online. Advertiser i has wants c_i impressions in total and pays v_{ij} for impression j, and will get paid a penalty of p_i for every undelivered impression. If over-delivered, he will pay his bid for the first c_i impressions delivered. Letting $b_{ij} = v_{ij} + p_i$, we can write the LP relaxation of the offline display ads problem as in LP 2.2, which is clearly a special case of the resource allocation LP.

Adwords	**Display Ads**
Maximize $\sum_{i,j} b_{ij}x_{ij}$ s.t.	Maximize $\sum_{i,j} b_{ij}x_{ij}$ s.t.
$\forall\ i, \sum_j b_{ijk}x_{ij} \leq B_i$	$\forall\ i, \sum_j x_{ij} \leq c_i$
$\forall\ j, \sum_i x_{ij} \leq 1$	$\forall\ j, \sum_i x_{ij} \leq 1$
$\forall\ i, j, x_{ij} \geq 0.$	$\forall\ i, j, x_{ij} \geq 0.$

LP 2.2: LP relaxations for Adwords and Display Ads

2.2.3 Asymptotically Optimal Online Algorithm for Adwords

As mentioned in the introduction, we single out the Adwords problem, a special case of resource allocation framework, due to its significance, and ask if we can design algorithms that go beyond what's possible for the general resource allocation framework.

We show that for the adwords problem such an improvement is possible, namely we show that we can get a $1-\epsilon$ algorithm for all $\gamma = O(\epsilon^2)$, improving over our result in Theorem 2.2 which held only for all $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$. However our algorithm is not completely prior robust. It requires a few parameters from the distribution, namely, it requires the amounts of budget consumed for each advertiser by some optimal solution to the expected instance. As discussed in the introduction, this requirement is information theoretically strictly weaker than requiring to know the entire distribution.

For the adwords problem $\gamma = \max_{i,j} \frac{b_{i,j}}{B_i}$, and let $\gamma_i = \max_j \frac{b_{i,j}}{B_i}$. Clearly $\gamma = \max_i \gamma_i$. Let C_i be the amount of budget consumed from advertiser i (that is the amount of revenue from i) in *some* optimal fractional solution to the expected instance. Clearly the expected instance's optimal fractional solution's revenue is $\sum_i C_i$.

Theorem 2.4 *Given the budget consumption C_i from every advertiser i by some optimal solution to the distribution instance, Algorithm 5 achieves an expected revenue of $\sum_i C_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$, which is at least $\sum_i C_i(1 - O(\max_i \sqrt{\gamma_i}))$, thus implying an approximation of $1 - O(\sqrt{\gamma})$ to the optimal fractional revenue of $\sum_i C_i$ to the expected instance.*

Theorem 2.5 *There exist instances with $\gamma = \epsilon^2$ for which no algorithm, even with the complete knowledge of the distribution, can get a $1 - o(\epsilon)$ approximation factor.*

As a corollary we also get a result without this extra assumption in the case where the optimal solution saturates all the budgets.

Remark 2.6 *We note the following about Theorem 2.4*

1. *Often it could be the case that many advertisers have small γ_i's but a few advertisers have large γ_i's, thus driving the maximum γ up. But our algorithm's*

competitive ratio doesn't degrade solely based on the maximum γ. Rather, it depends on all γ_i's and is robust to a few outliers.

2. *Our algorithm performs slightly better than what is stated. Let $k_i = \frac{1}{\gamma_i}$. Then our algorithm attains a revenue of $\sum_i C_i(1 - \frac{k_i^{k_i}}{k_i! e^{k_i}}) \geq \sum_i C_i(1 - \frac{1}{\sqrt{2\pi k_i}}) = \sum_i C_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$. In the worst case where $\gamma_i = i$ for all i, we get a $1 - 1/e$ approximation.*

Improvements for Online Stochastic Matching. A useful application of Theorem 2.4 is restricting it to the special case of online stochastic matching problem introduced by Feldman et al. [FMMM09]. In this problem, all the budgets are equal to 1, and all the bids are either 0 or 1. [FMMM09] showed an approximation of 0.67, and this has been improved by Bahmani and Kapralov [BK10], Manshadi, Gharan and Saberi [MGS11] and finally by Haeupler, Mirrokni and Zadimoghaddam [HMZ11] to 0.7036. (This problem has also been studied in the random permutation setting by Karande, Mehta and Tripathi [KMT11], and Mahdian and Yan [MY11]. The best approximation factor known in 0.696 due to Mahdian and Yan [MY11]). Our observation is that if the budgets are increased to 2 from 1, the approximation given by Theorem 2.4 is already at 0.729, assuming that the algorithm is given the number of matches to every left-hand-side (LHS) vertex. It increases even further for the B-matching problem, to an approximation ratio of $1 - \frac{1}{\sqrt{2\pi B}}$. More generally for the $\mathbf{B}$-matching problem, where LHS vertex i can accept B_i matches, the matching size obtained is $\sum_i C_i(1 - \frac{1}{\sqrt{2\pi B_i}})$ thus giving a factor of at least $1 - \frac{1}{\sqrt{2\pi B}}$ where $B = \min_i B_i$.

Corollary 2.7 *For the online stochastic $\mathbf{B}$-matching problem, there is an online algorithm that achieves a revenue of $\sum_i C_i(1 - \frac{1}{\sqrt{2\pi B_i}})$, and thus an approximation factor of at least $1 - \frac{1}{\sqrt{2\pi B}}$, where $B = \min_i B_i$, provided the algorithm is given the number of matches C_i to every LHS vertex in some optimal solution to the expected instance.*

Note that this shows an interesting trade-off w.r.t. the results of Feldman et al. [FMMM09]. There is a big improvement possible by just letting the number of possible matches in the LHS to 2 instead of one, and this is evidently the case for most applications of matching motivated by online advertising. On the other hand, instead of having to know the distribution it is enough to know the optimal expected consumptions.

2.2.4 Greedy Algorithm for Adwords

As noted in the introduction, the greedy algorithm is widely implemented due to its simplicity, but its performance was known to be only a 1/2 approximation even in stochastic models. We show that the greedy algorithm obtains a $1-1/e$ approximation for all γ, i.e., $0 \leq \gamma \leq 1$.

Theorem 2.8 *The greedy algorithm achieves an approximation factor of* $1 - 1/e$ *for the Adwords problem in the i.i.d. unknown distributions model for all* γ*, i.e.,* $0 \leq \gamma \leq 1$.

We note here that the competitive ratio of $1 - 1/e$ is tight for the greedy algorithm [GM08]. It is however not known to be tight for an arbitrary algorithm.

2.2.5 Fast Approximation Algorithms for Large Mixed Packing and Covering Integer Programs

Charles et al. [CCD+10] consider the following problem: given a bipartite graph $G = (L, R, E)$ where $m = |L| \gg |R| = n$, does there exist an assignment $M : L \to R$ with $(j, M(j)) \in E$ for all $j \in L$, and such that for every vertex $i \in R$, $|M^{-1}(i)| \geq B_i$ for some given values B_i. Since m is very large classic matching algorithms are not useful. Charles et al. [CCD+10] gave an algorithm that runs in time linear[1] in the number of edges of an induced subgraph obtained by taking a random sample from R of size $O\left(\frac{m \log n}{\min_i\{B_i\}\epsilon^2}\right)$, for a gap-version of the problem with gap ϵ. Such an algorithm is very useful in a variety of applications involving ad assignment for online advertising, particularly when $\min_i\{B_i\}$ is large.

We consider a generalization of the above problem inspired by the resource allocation framework. In fact, we consider the following mixed covering-packing problem. Suppose that there are n packing constraints, one for each $i \in [n]$ of the form $\sum_{j=1}^{m} \mathbf{a}_{i,j} \cdot \mathbf{x}_j \leq c_i$ and n covering constraints, one for each $i \in [n]$ of the form $\sum_{j=1}^{m} \mathbf{w}_{i,j} \cdot \mathbf{x}_j \geq d_i$. Each $\mathbf{x}_j$ (with coordinates $x_{j,k}$) is constrained to be in $\{0, 1\}^K$ and satisfy $\sum_k x_{j,k} \leq 1$. The $\mathbf{a}_{i,j}$'s and $\mathbf{w}_{i,j}$'s (and hence $\mathbf{x}_j$'s) are non-negative vectors

[1]In fact, the algorithm makes a single pass through this graph.

of length K with coordinates a_{ijk} and w_{ijk}. Does there exist a feasible solution to this system of constraints? The gap-version of this problem is as follows. Distinguish between the two cases, with a high probability, say $1-\delta$:

YES: There is a feasible solution.

NO: There is no feasible solution even if all the c_i's are multiplied by $1+\epsilon$ and all the d_i's are multiplied by $1-\epsilon$.

We note that solving (offline) an optimization problem in the resource allocation framework can be reduced to the above problem through a binary search on the objective function value.

Oracle Assumption. We assume that we have the following oracle available to us: given a request j and a vector v, the oracle returns the vector $\mathbf{x}_j$ that maximizes $v \cdot \mathbf{x}_j$ among all $\mathbf{x}_j$'s in $\{0,1\}^K$ satisfying $\sum_k x_{j,k} \leq 1$.

Let $\gamma = \max\left(\left\{\frac{a_{ijk}}{c_i}\right\}_{i,j,k} \cup \left\{\frac{w_{ijk}}{d_i}\right\}_{i,j,k}\right)$.

Theorem 2.9 *For any $\epsilon > 0$, Algorithm 6 solves the gap version of the mixed covering-packing problem with $\Theta(\frac{\gamma m \log(n/\delta)}{\epsilon^2})$ oracle calls.*

Chernoff Bounds. We present here the form of Chernoff bounds that we use throughout the rest of this paper. Let $X = \sum_i X_i$, where $X_i \in [0, B]$ are i.i.d random variables. Let $\mathbf{E}[X] = \mu$. Then, for all $\epsilon > 0$,

$$\mathbf{Pr}[X < \mu(1-\epsilon)] < \exp\left(\frac{-\epsilon^2\mu}{2B}\right).$$

Consequently, for all $\delta > 0$, with probability at least $1-\delta$,

$$X - \mu \geq -\sqrt{2\mu B \ln(1/\delta)}.$$

Similarly, for all $\epsilon \in [0, 2e-1]$,

$$\mathbf{Pr}[X > \mu(1+\epsilon)] < \exp\left(\frac{-\epsilon^2\mu}{4B}\right).$$

Consequently, for all $\delta > \exp(\frac{-(2e-1)^2\mu}{4B})$, with probability at least $1-\delta$,

$$X - \mu \leq \sqrt{4\mu B \ln(1/\delta)}.$$

For $\epsilon > 2e-1$,

$$\mathbf{Pr}[X > \mu(1+\epsilon)] < 2^{-(1+\epsilon)\mu/B}.$$

2.3 Near-Optimal Prior Robust Online Algorithms for Resource Allocation

For convenience, we begin by rewriting the LP relaxation of a random instance R of the online resource allocation problem. and the expected instance (already defined in Section 2.2.2 as LP 2.1)

Random Instance R	**Expected Instance**
Maximize λ s.t.	Maximize λ s.t.
$\forall\ i, \sum_{j\in R,k} w_{ijk}x_{j,k} \geq \lambda$	$\forall\ i, \sum_{j,k} mp_j w_{ijk}x_{j,k} \geq \lambda$
$\forall\ i, \sum_{j\in R,k} a_{ijk}x_{j,k} \leq c_i$	$\forall\ i, \sum_{j,k} mp_j a_{ijk}x_{j,k} \leq c_i$
$\forall\ j \in R, \sum_k x_{j,k} \leq 1$	$\forall\ j, \sum_k x_{j,k} \leq 1$
$\forall\ j \in R, k, x_{j,k} \geq 0.$	$\forall\ j, k, x_{j,k} \geq 0.$

LP 2.3: LPs for random and expected instances

We showed in Lemma 2.1 that $W_E \geq \mathbf{E}[W_R]$. All our approximation guarantees are w.r.t. the stronger benchmark of W_E which is the optimal fractional solution of the expected instance. We would like to remind the reader that while the benchmark is allowed to be fractional, the online algorithm of course is allowed to find only integral solutions.

We divide the rest of this section into four subsections. The subsections progressively weaken the assumptions on knowledge of the distribution of the input.

1. In section 2.3.1 we develop a hypothetical algorithm called Hypothetical-Oblivious-Conservative, denoted by $\widetilde{P}$, that achieves an objective value of $W_E(1-2\epsilon)$ w.p. at least $1-\epsilon$ assuming $\gamma = O\left(\frac{\epsilon^2}{\log(n/\epsilon)}\right)$. Theorem 2.11 is the main result of this section. The algorithm is hypothetical because it assumes knowledge of the entire distribution, where as the goal of this paper is to develop algorithms that work without distributional knowledge.

2. In section 2.3.2 we design an algorithm for the online resource allocation problem that achieves the same guarantee as the Hypothetical-Oblivious-Conservative algorithm $\widetilde{P}$, without any knowledge of the distribution except for a single parameter of the distribution — the value of W_E. Theorem 2.12 is the main result of this section.

3. In section 2.3.3 we design an algorithm for the online resource allocation problem that achieves an objective value of at least $W_E(1-O(\epsilon))$ w.p. at least $1-\epsilon$ assuming $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$ without any knowledge at all about the distribution. The algorithm in Section 2.3.2 serves as a good warm-up for the algorithm in this section. Theorem 2.2 is the main result of this section.

4. In section 2.3.5 we relax the assumption that the distribution from which the requests are drawn is i.i.d.; we give three different generalizations of the i.i.d. model with strong revenue guarantees as in the i.i.d. model.

2.3.1 Completely Known Distributions

When the distributions are completey known, we first compute the expected instance and solve its LP relaxation (LP 2.3) optimally. Let x^*_{jk} denote the optimal solution to the expected LP 2.3. The Hypothetical-Oblivious algorithm P works as follows: when request j arrives, it serves it using option k with probability x^*_{jk}. Let $X^*_{i,t}$ denote the amount of resource i consumed in step t for the algorithm P. Thus the total amount of resource i consumed over the entire m steps of algorithm P is $\sum_{t=1}^m X^*_{i,t}$. Note that $\mathbf{E}[X^*_{i,t}] = \sum_{j,k} p_j a_{ijk} x^*_{jk} \leq \frac{c_i}{m}$. Thus, we can bound the probability that $\mathbf{Pr}[\sum_{t=1}^m X^*_{i,t} \geq c_i(1+\epsilon)]$ using Chernoff bounds. We explicitly derive this bound here since we use this derivation in designing the algorithm in Section 2.3.2.

Since we cannot exceed c_i amount of resource consumption by any non-zero amount, we need to be more conservative than P. So we analyze the following algorithm $\widetilde{P}$, called Hypothetical-Oblivious-Conservative, instead of P: when request j arrives, it serves it using option k with probability $\frac{x^*_{jk}}{1+\epsilon}$, where ϵ is an error parameter of algorithm designer's choice. Let $\widetilde{X}_{i,t}$ denote the amount of resource i consumed in step t for the algorithm $\widetilde{P}$. Note that $\mathbf{E}[\widetilde{X}_{i,t}] \leq \frac{c_i}{(1+\epsilon)m}$. Thus, even with a $(1+\epsilon)$ deviation using Chernoff bounds, the resource consumption is at most c_i.

We begin by noting that $\widetilde{X}_{i,t} \leq \gamma c_i$ by the definition of γ. For all $\epsilon \in [0,1]$ we have,

$$
\begin{aligned}
\mathbf{Pr}\left[\sum_{t=1}^{m} \widetilde{X}_{i,t} \geq \frac{c_i}{1+\epsilon}(1+\epsilon)\right] &= \mathbf{Pr}\left[\frac{\sum_{t=1}^{m} \widetilde{X}_{i,t}}{\gamma c_i} \geq \frac{1}{\gamma}\right] \\
&= \mathbf{Pr}\left[(1+\epsilon)^{\frac{\sum_{t=1}^{m} \widetilde{X}_{i,t}}{\gamma c_i}} \geq (1+\epsilon)^{\frac{1}{\gamma}}\right] \\
&\leq \mathbf{E}\left[(1+\epsilon)^{\frac{\sum_{t=1}^{m} \widetilde{X}_{i,t}}{\gamma c_i}}\right] / (1+\epsilon)^{\frac{1}{\gamma}} \\
&= \mathbf{E}\left[\prod_{t=1}^{m} (1+\epsilon)^{\frac{\widetilde{X}_{i,t}}{\gamma c_i}}\right] / (1+\epsilon)^{\frac{1}{\gamma}} \\
&\leq \mathbf{E}\left[\prod_{t=1}^{m} \left(1+\epsilon\frac{\widetilde{X}_{i,t}}{\gamma c_i}\right)\right] / (1+\epsilon)^{\frac{1}{\gamma}} \\
&\leq \left[\prod_{t=1}^{m} \left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)\right] / (1+\epsilon)^{\frac{1}{\gamma}} \\
&\leq \left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\frac{1}{\gamma(1+\epsilon)}} \\
&\leq e^{\frac{-\epsilon^2}{4\gamma}\frac{1}{1+\epsilon}} \\
&\leq \frac{\epsilon}{2n}
\end{aligned}
$$

where the first inequality follows from Markov's inequality, the second from convexity of exponential function together with with the fact that $\widetilde{X}_{i,t} \leq \gamma c_i$, the third from $\mathbf{E}[\widetilde{X}_{i,t}] \leq \frac{c_i}{(1+\epsilon)m}$, and the fourth from $1+x \leq e^x$, the fifth is standard for all $\epsilon \in [0,1]$, and the sixth follows from $\gamma = O(\epsilon^2/\log(n/\epsilon))$ for an appropriate choice of constant inside the big-oh coupled with $n \geq 2$.

Remark 2.10 *At first sight this bound might seem anomalous — the bound $\frac{\epsilon}{2n}$ is increasing in ϵ, i.e., the probability of a smaller deviation is smaller than the probability of a larger deviation! The reason for this anomaly is that γ is related to ϵ as $\gamma =$*

$O(\frac{\epsilon^2}{\log(n/\epsilon)})$, and smaller the γ, the better revenue we can get (i.e., more granular requests leads to lesser wastage from errors, and hence more revenue). Thus a small deviation for small γ has a smaller probability than a larger deviation for a larger γ.

Similarly let $\widetilde{Y}_{i,t}$ denote the revenue obtained from type i profit in step t for the algorithm $\widetilde{P}$. Note that $\mathbf{E}[\widetilde{Y}_{i,t}] = \sum_{j,k} p_j w_{ijk} \frac{x^*_{jk}}{1+\epsilon} \geq \frac{W_E}{(1+\epsilon)m}$. By the definition of γ, we have $\widetilde{Y}_{i,t} \leq \gamma W_E$. For all $\epsilon \in [0,1]$ we have,

$$\begin{aligned}
\mathbf{Pr}\left[\sum_{t=1}^{m} \widetilde{Y}_{i,t} \leq \frac{W_E}{1+\epsilon}(1-\epsilon)\right] &= \mathbf{Pr}\left[\frac{\sum_{t=1}^{m} \widetilde{Y}_{i,t}}{\gamma W_E} \leq \frac{1-\epsilon}{\gamma(1+\epsilon)}\right] \\
&= \mathbf{Pr}\left[(1-\epsilon)^{\frac{\sum_{t=1}^{m} \widetilde{Y}_{i,t}}{\gamma W_E}} \geq (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}\right] \\
&\leq \mathbf{E}\left[(1-\epsilon)^{\frac{\sum_{t=1}^{m} \widetilde{Y}_{i,t}}{\gamma W_E}}\right] / (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}} \\
&= \mathbf{E}\left[\prod_{t=1}^{m} (1-\epsilon)^{\frac{\widetilde{Y}_{i,t}}{\gamma W_E}}\right] / (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}} \\
&\leq \mathbf{E}\left[\prod_{t=1}^{m} \left(1-\epsilon\frac{\widetilde{Y}_{i,t}}{\gamma W_E}\right)\right] / (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}} \\
&\leq \left[\prod_{t=1}^{m} \left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)\right] / (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}} \\
&\leq \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{\frac{1}{\gamma(1+\epsilon)}} \\
&\leq e^{\frac{-\epsilon^2}{2\gamma}\frac{1}{1+\epsilon}} \\
&\leq \frac{\epsilon}{2n}
\end{aligned}$$

Thus, we have all the capacity constraints satisfied, (i.e., $\sum_i \widetilde{X}_{i,t} \leq c_i$), and all resource profits are at least $\frac{W_E}{1+\epsilon}(1-\epsilon)$ (i.e., $\sum_i \widetilde{Y}_{i,t} \geq \frac{W_E}{1+\epsilon}(1-\epsilon) \geq W_E(1-2\epsilon)$), with probability at least $1 - 2n \cdot \epsilon/2n = 1-\epsilon$. This proves the following theorem:

Theorem 2.11 *For any $\epsilon > 0$, the Hypothetical-Oblivious-Conservative algorithm $\widetilde{P}$ achieves an objective value of $W_E(1-2\epsilon)$ for the online resource allocation problem with probability at least $1-\epsilon$, assuming $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$.*

2.3.2 Unknown Distribution, Known W_E

We now design an algorithm A without knowledge of the distribution, but just knowing a single parameter W_E. Let $A^s\widetilde{P}^{m-s}$ be a hybrid algorithm that runs A for the first s steps and $\widetilde{P}$ for the remaining $m-s$ steps. Let $\epsilon \in [0,1]$ be the error parameter, which is the algorithm desinger's choice. Call the algorithm a failure if at least one of the following fails:

1. For all i, $\sum_{t=1}^{m} X_{i,t}^{A} \leq c_i$.
2. For all i, $\sum_{t=1}^{m} Y_{i,t}^{A} \geq W_E(1-2\epsilon)$.

For any algorithm A, let the amount of resource i consumed in the t-th step be denoted by $X_{i,t}^{A}$ and the amount of resource i profit be denoted by $Y_{i,t}^{A}$. Let $S_s\left(X_i^A\right) = \sum_{t=1}^{s} X_{i,t}^{A}$ denote the amount of resource i consumed in the first s steps, and let $S_s\left(Y_i^A\right) = \sum_{t=1}^{s} Y_{i,t}^{A}$ denote the resource i profit in the first s steps. Similar to the derivation in Section 2.3.1 which bounded the failure probability of $\widetilde{P}$, we can bound the failure probability of any algorithm A, i.e.,

$$
\begin{aligned}
\mathbf{Pr}\left[\sum_{t=1}^{m} X_{i,t}^{A} \geq \frac{c_i}{1+\epsilon}(1+\epsilon)\right] &= \mathbf{Pr}\left[\frac{\sum_{t=1}^{m} X_{i,t}^{A}}{\gamma c_i} \geq \frac{1}{\gamma}\right] \\
&= \mathbf{Pr}\left[(1+\epsilon)^{\frac{\sum_{t=1}^{m} X_{i,t}^{A}}{\gamma c_i}} \geq (1+\epsilon)^{\frac{1}{\gamma}}\right] \\
&\leq \mathbf{E}\left[(1+\epsilon)^{\frac{\sum_{t=1}^{m} X_{i,t}^{A}}{\gamma c_i}}\right] / (1+\epsilon)^{\frac{1}{\gamma}} \\
&= \mathbf{E}\left[\prod_{t=1}^{m}(1+\epsilon)^{\frac{X_{i,t}^{A}}{\gamma c_i}}\right] / (1+\epsilon)^{\frac{1}{\gamma}} \qquad (2.1)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{Pr}\left[\sum_{t=1}^{m} Y_{i,t}^{A} \leq \frac{W_E}{1+\epsilon}(1-\epsilon)\right] &= \mathbf{Pr}\left[\frac{\sum_{t=1}^{m} Y_{i,t}^{A}}{\gamma W_E} \leq \frac{1-\epsilon}{\gamma(1+\epsilon)}\right] \\
&= \mathbf{Pr}\left[(1-\epsilon)^{\frac{\sum_{t=1}^{m} Y_{i,t}^{A}}{\gamma W_E}} \geq (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}\right] \\
&\leq \mathbf{E}\left[(1-\epsilon)^{\frac{\sum_{t=1}^{m} Y_{i,t}^{A}}{\gamma W_E}}\right] / (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}
\end{aligned}
$$

$$= \mathbf{E}\left[\prod_{t=1}^{m}(1-\epsilon)^{\frac{Y_{i,t}^A}{\gamma W_E}}\right] / (1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}} \tag{2.2}$$

In Section 2.3.1 our algorithm A was $\widetilde{P}$ $\left(\text{and therefore we can use } \mathbf{E}[\widetilde{X}_{i,t}] \leq \frac{c_i}{(1+\epsilon)m}\right.$ and $\left.\mathbf{E}[\widetilde{Y}_{i,t}] \geq \frac{W_E}{(1+\epsilon)m}\right)$, the total failure probability which is the sum of (2.1) and (2.2) for all the i's would have been $n \cdot \left[\frac{\epsilon}{2n} + \frac{\epsilon}{2n}\right] = \epsilon$. The goal is to design an algorithm A for stage r that, unlike P, does not know the distribution and knows just W_E, but obtains the same ϵ failure probability. That is we want to show that the sum of (2.1) and (2.2) over all i's is at most ϵ:

$$\sum_i \frac{\mathbf{E}\left[\prod_{t=1}^{m}(1+\epsilon)^{\frac{X_{i,t}^A}{\gamma c_i}}\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} + \sum_i \frac{\mathbf{E}\left[\prod_{t=1}^{m}(1-\epsilon)^{\frac{Y_{i,t}^A}{\gamma W_E}}\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}} \leq \epsilon$$

For the algorithm $A^s\widetilde{P}^{m-s}$, the above quantity can be rewritten as

$$\sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_s\left(X_i^A\right)}{\gamma c_i}} \prod_{t=s+1}^{m}(1+\epsilon)^{\frac{\widetilde{X}_{i,t}}{\gamma c_i}}\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} + \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_s\left(Y_i^A\right)}{\gamma W_E}} \prod_{t=s+1}^{m}(1-\epsilon)^{\frac{\widetilde{Y}_{i,t}}{\gamma W_E}}\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}},$$

which, by using $(1+\epsilon)^x \leq 1+\epsilon x$ for $0 \leq x \leq 1$, is in turn upper bounded by

$$\sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_s\left(X_i^A\right)}{\gamma c_i}} \prod_{t=s+1}^{m}\left(1+\epsilon\frac{\widetilde{X}_{i,t}}{\gamma c_i}\right)\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} + \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_s\left(Y_i^A\right)}{\gamma W_E}} \prod_{t=s+1}^{m}\left(1-\epsilon\frac{\widetilde{Y}_{i,t}}{\gamma W_E}\right)\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}.$$

Since for all t, the random variables $\widetilde{X}_{i,t}$, $X_{i,t}^A$, $\widetilde{Y}_{i,t}$ and $Y_{i,t}^A$ are all independent, and $\mathbf{E}[\widetilde{X}_{i,t}] \leq \frac{c_i}{(1+\epsilon)m}$ and $\mathbf{E}[\widetilde{Y}_{i,t}] \geq \frac{W_E}{(1+\epsilon)m}$, the above is in turn upper bounded by

$$\sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_s\left(X_i^A\right)}{\gamma c_i}} \left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-s}\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} + \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_s\left(Y_i^A\right)}{\gamma W_E}} \left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-s}\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}. \tag{2.3}$$

Let $\mathcal{F}[A^s\widetilde{P}^{m-s}]$ denote the quantity in (2.3), which is an upper bound on failure probability of the hybrid algorithm $A^s\widetilde{P}^{m-s}$. By Theorem 2.11, we know that $\mathcal{F}[\widetilde{P}^m] \leq$

ϵ. We now prove that for all $s \in \{0, 1, \ldots, m-1\}$, $\mathcal{F}[A^{s+1}\widetilde{P}^{m-s-1}] \le \mathcal{F}[A^s\widetilde{P}^{m-s}]$, thus proving that $\mathcal{F}[A^m] \le \epsilon$, i.e., running the algorithm A for all the m steps results in a failure with probability at most ϵ. To design such an A we closely follow the derivation of Chernoff bounds, which is what established that $\mathcal{F}[\widetilde{P}^m] \le \epsilon$ in Theorem 2.11. However the design process will reveal that unlike algorithm $\widetilde{P}$ which needs the entire distribution, just the knowledge of W_E will do for bounding the failure probability by ϵ.

Assuming that for all $s < p$, the algorithm A has been defined for the first $s+1$ steps in such a way that $\mathcal{F}[A^{s+1}\widetilde{P}^{m-s-1}] \le \mathcal{F}[A^s\widetilde{P}^{m-s}]$, we now define A for the $p+1$-th step in a way that will ensure that $\mathcal{F}[A^{p+1}\widetilde{P}^{m-p-1}] \le \mathcal{F}[A^p\widetilde{P}^{m-p}]$. We have

$$\begin{aligned}
\mathcal{F}[A^{p+1}\widetilde{P}^{m-p-1}] &= \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_{p+1}\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_{p+1}\left(Y_i^A\right)}{\gamma W_E}}\left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}} \\
&\le \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_p\left(X_i^A\right)}{\gamma c_i}}\left(1+\epsilon\frac{X_{i,p+1}^A}{\gamma c_i}\right)\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_p\left(Y_i^A\right)}{\gamma W_E}}\left(1-\epsilon\frac{Y_{i,p+1}^A}{\gamma W_E}\right)\left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}
\end{aligned} \tag{2.4}$$

Define

$$\phi_{i,s} = \frac{1}{c_i}\left[\frac{(1+\epsilon)^{\frac{S_s\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-s-1}}{(1+\epsilon)^{\frac{1}{\gamma}}}\right]$$

$$\psi_{i,s} = \frac{1}{W_E}\left[\frac{(1-\epsilon)^{\frac{S_s\left(Y_i^A\right)}{\gamma W_E}}\left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-s-1}}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}\right]$$

Define the step $p+1$ of algorithm A as picking the following option k^* for request j, where:

$$k^* = \arg\min_k \left\{ \sum_i a_{ijk} \cdot \phi_{i,p} - \sum_i w_{ijk} \cdot \psi_{i,p} \right\}. \tag{2.5}$$

For the sake of clarity, the entire algorithm is presented in Algorithm 1.

Algorithm 1 : Algorithm for stochastic online resource allocation with unknown distribution, known W_E

Input: Capacities c_i for $i \in [n]$, the total number of requests m, the values of γ and W_E, an error parameter $\epsilon > 0$.

Output: An online allocation of resources to requests

Initialize $\phi_{i,0} = \frac{1}{c_i}\left[\frac{\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-1}}{(1+\epsilon)^{\frac{1}{\gamma}}}\right]$, and, $\psi_{i,0} = \frac{1}{W_E}\left[\frac{\left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-1}}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}\right]$

for $s = 1$ to m **do**

    If the incoming request is j, use the following option k^*:

$$k^* = \arg\min_{k \in K \cup \{\perp\}} \left\{ \sum_i a_{ijk} \cdot \phi_{i,s-1} - \sum_i w_{ijk} \cdot \psi_{i,s-1} \right\}.$$

    $X^A_{i,s} = a_{ijk^*}$, $Y^A_{i,s} = w_{ijk^*}$

    Update $\phi_{i,s} = \phi_{i,s-1} \cdot \left[\frac{(1+\epsilon)^{\frac{X^A_{i,s}}{\gamma c_i}}}{1+\frac{\epsilon}{(1+\epsilon)\gamma m}}\right]$, and, $\psi_{i,s} = \psi_{i,s-1} \cdot \left[\frac{(1-\epsilon)^{\frac{Y^A_{i,s}}{\gamma W_E}}}{1-\frac{\epsilon}{(1+\epsilon)\gamma m}}\right]$

end for

By the definition of step $p+1$ of algorithm A given inequation (2.5), it follows that for any two algorithms with the first p steps being identical, and the last $m-p-1$ steps following the Hypothetical-Oblivious-Conservative algorithm $\widetilde{P}$, algorithm A's $p+1$-th step is the one that minimizes expression (2.4). In particular it follows that expression (2.4) is upper bounded by the same expression where the $p+1$-the step is according to $\widetilde{X}_{i,p+1}$ and $\widetilde{Y}_{i,p+1}$, i.e., we replace $X^A_{i,p+1}$ by $\widetilde{X}_{i,p+1}$ and $Y^A_{i,p+1}$ by $\widetilde{Y}_{i,p+1}$. Therefore we have

$$\mathcal{F}[A^{p+1}\widetilde{P}^{m-p-1}] \le \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_p\left(X^A_i\right)}{\gamma c_i}} \left(1+\epsilon\frac{\widetilde{X}_{i,p+1}}{\gamma c_i}\right)\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} \qquad +$$

$$\sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_p\left(Y_i^A\right)}{\gamma W_E}}\left(1-\epsilon\frac{\widetilde{Y}_{i,p+1}}{\gamma W_E}\right)\left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}$$

$$\leq \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_p\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1+\epsilon)^{\frac{1}{\gamma}}} +$$

$$\sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_p\left(Y_i^A\right)}{\gamma W_E}}\left(1-\frac{\epsilon}{(1+\epsilon)\gamma W_E}\right)\left(1-\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-p-1}\right]}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}$$

$$= \mathcal{F}[A^p \widetilde{P}^{m-p}]$$

This completes the proof of the following theorem.

Theorem 2.12 *For any $\epsilon > 0$, Algorithm 1 achieves an objective value of $W_E(1-2\epsilon)$ for the online resource allocation problem with probability at least $1-\epsilon$, assuming $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$. The algorithm A does not require any knowledge of the distribution except for the single parameter W_E.*

2.3.3 Completely Unknown Distribution

We first give a high-level overview of this section before going into the details. In this section, we design an algorithm A without any knowledge of the distribution at all. The algorithm is similar in spirit to the one in Section 2.3.2 except that since we do not have knowledge of W_E, we divide the algorithm into many stages. In each stage, we run an algorithm similar to the one in Section 2.3.2 except that instead of W_E, we use an estimate of W_E that gets increasingly accurate with each successive stage.

More formally, the algorithm runs in l stages $\{0, 1, \ldots, l-1\}$, where l is such that $\epsilon 2^l = 1$, and $\epsilon \in [1/m, 1/2]$ (we need $\epsilon \leq 1/2$ so that l is at least 1) is the error parameter of algorithm designer's choice. Further we need $m \geq \frac{1}{\epsilon}$ so that $\epsilon m \geq 1$. We assume that ϵm is an integer for clarity of exposition. Stage r handles $t_r = \epsilon m 2^r$ requests for $r \in \{0, \ldots l-1\}$. The first ϵm requests are used just for future estimation, and none of them are served. For convenience we sometimes call this pre-zero stage as stage -1, and let $t_{-1} = \epsilon m$. Stage $r \geq 0$ serves $t \in [t_r + 1, t_{r+1}]$. Note that in the optimal solution to the expected instance of stage r, no resource i gets consumed by more than $\frac{t_r c_i}{m}$, and every resource i gets a profit of $\frac{t_r W_E}{m}$, i.e., consumption and profit have been scaled down by a factor of $\frac{t_r}{m}$. As in the previous sections, with a high probability, we can only reach close to $\frac{t_r W_E}{m}$. Further, since stage r consists of only t_r requests, which is much smaller than m for small r, it follows that for small r, our error in how close to we get to $\frac{t_r W_E}{m}$ will be higher. Indeed, instead of having the same error parameter of ϵ in every stage, we set stage-specific error parameters which get progressively smaller and become close to ϵ in the final stages. These parameters are chosen such that the overall error is still $O(\epsilon)$ because the later stages having more requests matter more than the former. There are two sources of error/failure which we detail below.

1. The first source of failure stems from not knowing W_E. Instead we estimate a quantity Z_r which is an approximation we use for W_E in stage r, and the approximation gets better as r increases. We use Z_r to set a profit target of $\frac{t_r Z_r}{m}$ for stage r. Since Z_r could be much smaller than W_E our algorithm could become very suboptimal. We prove that with a probability of at least $1 - 2\delta$ we have $W_E(1 - 3\epsilon_{x,r-1}) \le Z_r \le W_E$ (see next for what $\epsilon_{x,r}$ is), where $\delta = \frac{\epsilon}{3l}$. Thus for all the l stages, these bounds are violated with probability at most $2l\delta = 2\epsilon/3$.

2. The second source of failure stems from achieving this profit target in every stage. We set error parameters $\epsilon_{x,r}$ and $\epsilon_{y,r}$ such that for every i stage r consumes at most $\frac{t_r c_i}{m}(1 + \epsilon_{x,r})$ amount of resource i, and for every i we get a profit of at least $\frac{t_r Z_r}{m}(1 - \epsilon_{y,r})$, with probability at least $1 - \delta$. Thus the overall failure probability, as regards falling short of the target $\frac{t_r Z_r}{m}$ by more than $\epsilon_{y,r}$ and exceeding $\frac{t_r c_i}{m}$ by more than $\epsilon_{x,r}$, for all the l stages together is at most $\delta \cdot l = \epsilon/3$.

Thus summing over the failure probabilities we get $\epsilon/3 + 2\epsilon/3 = \epsilon$. We have that with probability at least $1-\epsilon$, for every i, the total consumption of resource i is at most $\sum_{r=0}^{l-1} \frac{t_r c_i}{m}(1+\epsilon_{x,r})$, and total profit from resource i is at least $\sum_{r=0}^{l-1} \frac{t_r W_E}{m}(1-3\epsilon_{x,r-1})(1-\epsilon_{y,r})$. We set $\epsilon_{x,r} = \sqrt{\frac{4\gamma m \ln(2n/\delta)}{t_r}}$ for $r \in \{-1, 0, 1, \ldots, l-1\}$, and $\epsilon_{y,r} = \sqrt{\frac{2w_{\max} m \ln(2n/\delta)}{t_r Z_r}}$ for $r \in \{0, 1, \ldots, l-1\}$ (we define $\epsilon_{x,r}$ starting from $r = -1$, with $t_{-1} = \epsilon m$, just for technical convenience). From this it follows that $\sum_{r=0}^{l-1} \frac{t_r c_i}{m}(1 + \epsilon_{x,r}) \le c_i$ and $\sum_{r=0}^{l-1} \frac{t_r W_E}{m}(1 - 3\epsilon_{x,r-1})(1 - \epsilon_{y,r}) \ge W_E(1 - O(\epsilon))$, assuming $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$. The algorithm is described in Algorithm 2. This completes the high-level overview of the proof. All that is left to prove is the points 1 and 2 above, upon which we would have proved our main theorem, namely, Theorem 2.2, which we recall below.

Theorem 2.2 *For any $\epsilon \geq 1/m$, Algorithm 2 achieves an objective value of $W_E(1 - O(\epsilon))$ for the online resource allocation problem with probability at least $1-\epsilon$, assuming $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$. Algorithm 2 does not require any knowledge of the distribution at all.*

Algorithm 2 : Algorithm for stochastic online resource allocation with unknown distribution

---

Input: Capacities c_i for $i \in [n]$, the total number of requests m, the value of γ, an error parameter $\epsilon > 1/m$.
Output: An online allocation of resources to requests

Set $l = \log(1/\epsilon)$
Initialize $t_{-1} : t_{-1} \leftarrow \epsilon m$
for $r = 0$ to $l - 1$ **do**
  Compute e_{r-1}: the optimal solution to the t_{r-1} requests of stage $r - 1$ with capacities capped at $\frac{t_{r-1}c_i}{m}$.
  Set $Z_r = \frac{e_{r-1}}{1+\epsilon_{x,r-1}} \cdot \frac{m}{t_{r-1}}$
  Set $\epsilon_{x,r} = \sqrt{\frac{4\gamma m \ln(2n/\delta)}{t_r}}$, $\epsilon_{y,r} = \sqrt{\frac{2w_{\max} m \ln(2n/\delta)}{t_r Z_r}}$
  Set $\phi^r_{i,0} = \frac{\epsilon_{x,r}}{\gamma c_i}\left[\frac{\left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)^{t_r-1}}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}}\right]$, and, $\psi^r_{i,0} = \frac{\epsilon_{y,r}}{w_{\max}}\left[\frac{\left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)^{t_r-1}}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}}\right]$
  for $s = 1$ to t_r **do**
    If the incoming request is j, use the following option k^*:

$$k^* = \arg\min_{k \in K \cup \{\perp\}} \left\{ \sum_i a_{ijk} \cdot \phi^r_{i,s-1} - \sum_i w_{ijk} \cdot \psi^r_{i,s-1} \right\}.$$

    $X^A_{i,t_r+s} = a_{ijk^*}$, $Y^A_{i,t_r+s} = w_{ijk^*}$
    Update $\phi^r_{i,s} = \phi^r_{i,s-1} \cdot \left[\frac{(1+\epsilon_{x,r})^{\frac{X^A_{i,t_r+s}}{\gamma c_i}}}{1+\frac{\epsilon_{x,r}}{m\gamma}}\right]$, and, $\psi^r_{i,s} = \psi^r_{i,s-1} \cdot \left[\frac{(1-\epsilon_{y,r})^{\frac{Y^A_{i,t_r+s}}{w_{\max}}}}{1-\frac{\epsilon_{y,r}}{m\gamma}}\right]$
  end for
end for

Detailed Description and Proof. We begin with the first point in our high-level overview above, namely by describing how Z_r is estimated and proving its concentration around W_E. After stage r (including stage -1), the algorithm computes the optimal fractional objective value e_r to the following instance $\mathcal{I}_r$: the instance has the t_r requests of stage r, and the capacity of resource i is capped at $\frac{t_r c_i}{m}$. Using the optimal fractional objective value e_r of this instance, we set $Z_{r+1} = \frac{e_r}{1+\epsilon_{x,r}} \cdot \frac{m}{t_r}$. The first task now is to prove that Z_{r+1} as estimated above is concentrated enough around W_E. This basically requires proving concentration of e_r.

Lemma 2.13 *With a probability at least* $1 - 2\delta$*, we have*

$$\frac{t_r W_E}{m}(1 - 2\epsilon_{x,r}) \leq e_r \leq \frac{t_r W_E}{m}(1 + \epsilon_{x,r}).$$

Proof: We prove that the lower and upper bound hold with probability $1 - \delta$ each, thus proving the lemma.

We begin with the lower bound on e_r. Note that the expected instance of the instance $\mathcal{I}_r$ has the same optimal solution x^*_{jk} as the optimal solution to the full expected instance (i.e., the one without scaling down by $\frac{t_r}{m}$). Now consider the algorithm $\widetilde{P}(r)$, which is the same as the $\widetilde{P}$ defined in Section 2.3.1 except that ϵ is replaced by $\epsilon_{x,r}$, i.e., it serves request j with option k with probability $\frac{x^*_{jk}}{1+\epsilon_{x,r}}$. When $\widetilde{P}(r)$ is run on instance $\mathcal{I}_r$, with a probability at least $1 - \frac{\delta}{2n}$, at most $\frac{t_r c_i}{m}$ amount of resource i is consumed, and with probability at least $1 - \frac{\delta}{2n}$, at least $\frac{t_r W_E}{m} \frac{1-\epsilon_{x,r}}{1+\epsilon_{x,r}}$ resource i profit is obtained. Thus with a probability at least $1 - 2n \cdot \frac{\delta}{2n} = 1 - \delta$, $\widetilde{P}(r)$ achieves an objective value of at least $\frac{t_r W_E}{m}(1 - 2\epsilon_{x,r})$. Therefore the optimal objective value e_r will also be at least $\frac{t_r W_E}{m}(1 - 2\epsilon_{x,r})$.

To prove the upper bound, we consider the primal and dual LPs that define e_r in LP 2.4 and the primal and dual LPs defining the expected instance in LP 2.5. In the latter, for convenience, we use $mp_j\beta_j$ as the dual multiplier instead of just β_j.

Note that the set of constraints in the dual of LP 2.5 is a superset of the set of constraints in the dual of LP 2.4, making any feasible solution to dual of LP 2.5 also feasible to dual of LP 2.4. In particular, the optimal solution to dual of LP 2.5 given by β^*_j's, α^*_i's and ρ^*_i's is feasible for dual of LP 2.4. Hence the value of e_r is upper bounded the objective of dual of LP 2.4 at β^*_j's, α^*_i's and ρ^*_i's. That is we have

$$e_r \leq \sum_{j \in \mathcal{I}_r} \beta^*_j + \frac{t_r}{m} \sum_i \alpha^*_i c_i.$$

Primal defining e_r	**Dual defining** e_r
Maximize λ s.t.	Minimize $\sum_{j\in\mathcal{I}_r}\beta_j + \frac{t_r}{m}\sum_i \alpha_i c_i$ s.t.
$\forall\ i,\ \sum_{j\in\mathcal{I}_r,k} w_{ijk}x_{j,k} \geq \lambda$	$\forall\ j\in\mathcal{I}_r,\ \beta_j + \sum_i(\alpha_i a_{ijk} - \rho_i w_{ijk}) \geq 0$
$\forall\ i,\ \sum_{j\in\mathcal{I}_r,k} a_{ijk}x_{j,k} \leq \frac{t_r c_i}{m}$	$\sum_i \rho_i \geq 1$
$\forall\ j\in\mathcal{I}_r,\ \sum_k x_{j,k} \leq 1$	$\forall\ i,\ \rho_i \geq 0, \alpha_i \geq 0$
$\forall\ j\in\mathcal{I}_r, k,\ x_{j,k} \geq 0.$	$\forall\ j\in\mathcal{I}_r,\ \beta_j \geq 0$

LP 2.4: Primal and dual LPs defining e_r

Primal for the expected instance	**Dual for the expected instance**
Maximize λ s.t.	Minimize $\sum_j mp_j\beta_j + \sum_i \alpha_i c_i$ s.t.
$\forall\ i,\ \sum_{j,k} mp_j w_{ijk}x_{j,k} \geq \lambda$	$\forall\ j,\ mp_j\left(\beta_j + \sum_i(\alpha_i a_{ijk} - \rho_i w_{ijk})\right) \geq 0$
$\forall\ i,\ \sum_{j,k} mp_j a_{ijk}x_{j,k} \leq c_i$	$\sum_i \rho_i \geq 1$
$\forall\ j,\ \sum_k x_{j,k} \leq 1$	$\forall\ i,\ \rho_i \geq 0, \alpha_i \geq 0$
$\forall\ j, k,\ x_{j,k} \geq 0.$	$\forall\ j,\ \beta_j \geq 0$

LP 2.5: Primal and dual LPs defining the expected instance

We now upper bound the RHS by applying Chernoff bounds on $\sum_{j\in\mathcal{I}_r}\beta^*_j$. Since the dual LP in LP 2.5 is a minimization LP, the constraints there imply that $\beta^*_j \leq w_{\max}$. Applying Chernoff bounds we have,

$$\begin{aligned} e_r \quad &\leq t_r\sum_j p_j\beta^*_j + \sqrt{4t_r(\sum_j p_j\beta^*_j)w_{\max}\ln(1/\delta)} + \frac{t_r}{m}\sum_i \alpha^*_i c_i \\ &\leq \frac{t_r W_E}{m} + \frac{t_r W_E}{m}\epsilon_{x,r} \end{aligned}$$

where the first inequality holds with probability at least $1-\delta$ and the second inequality uses the fact the optimal value of the expected instance (dual of LP 2.5) is W_E. This proves the required upper bound on e_r that $e_r \leq \frac{t_r W_E}{m}(1+\epsilon_{x,r})$ with probability at least $1-\delta$.

Going back to our application of Chernoff bounds above, in order to apply it in the form above, we require that the multiplicative deviation from mean $\sqrt{\frac{4w_{\max}\ln(1/\delta)}{t_r\sum_j p_j\beta_j^*}} \in [0, 2e-1]$. If $\sum_j p_j\beta_j^* \geq \frac{\epsilon W_E}{m}$, then this requirement would follow. Suppose on the other hand that $\sum_j p_j\beta_j^* < \frac{\epsilon W_E}{m}$. Since we are happy if the excess over mean is at most $\frac{t_r W_E}{m}\epsilon_{x,r}$, let us look for a multiplicative error of $\frac{\frac{t_r W_E \epsilon_{x,r}}{m}}{t_r\sum_j p_j\beta_j^*}$. Based on the fact that $\sum_j p_j\beta_j^* < \frac{\epsilon W_E}{m}$ and that $\epsilon_{x,r} > \epsilon$ for all r, the multiplicative error can be seen to be at least a constant, and can be made larger than $2e-1$ depending on the constant inside the big O of γ. We now use the version of Chernoff bounds for multiplicative error larger than $2e-1$, which gives us that a deviation of $\frac{t_r W_E}{m}\epsilon_{x,r}$ occurs with a probability at most $2^{-\left(1+\frac{\frac{t_r W_E \epsilon_{x,r}}{m}}{t_r\sum_j p_j\beta_j^*}\right)\frac{t_r\sum_j p_j\beta_j^*}{w_{\max}}}$, where the division by $w_{\max}$ is because of the fact that $\beta_j^* \leq w_{\max}$. Noting that $w_{\max} \leq \gamma W_E$, we get that this probability is at most δ/n which is at most δ.

■

Lemma 2.13 implies that $W_E(1-3\epsilon_{x,r-1}) \leq Z_r \leq W_E,\ \forall r \in \{0, 1, \ldots, l-1\}$. The rest of the proof is similar to that of Section 2.3.2, and is focused on the second point in the high-level overview we gave in the beginning of this section 2.3.3. In Section 2.3.2 we knew W_E and obtained a $W_E(1-2\epsilon)$ approximation with no resource i consumed beyond c_i with probability $1-\epsilon$. Here, instead of W_E we have an approximation for W_E in the form of Z_r which gets increasingly accurate as r increases. We set a target of $\frac{t_r Z_r}{m}$ for stage r, and show that with a probability of at least $1-\delta$ we get a profit of $\frac{t_r Z_r}{m}(1-\epsilon_{y,r})$ from every resource i and no resource i consumed beyond $\frac{t_r c_i}{m}(1+\epsilon_{x,r})$ capacity.[2]

As in Section 2.3.2, call stage r of algorithm A a failure if at least one of the following fails:

1. For all i, $\sum_{t=t_r+1}^{t_{r+1}} X_{i,t}^A \leq \frac{t_r c_i}{m}(1+\epsilon_{x,r})$.

[2]Note that we are allowed to consume a bit beyond $\frac{t_r c_i}{m}$ because our goal is just that over all we don't consume beyond c_i, and not that for every stage we respect the $\frac{t_r c_i}{m}$ constraint. In spite of this $(1+\epsilon_{x,r})$ excess consumption in all stages, since stage -1 consumes nothing at all, we will see that no excess consumption occurs at the end.

2. For all i, $\sum_{t=t_r+1}^{t_{r+1}} Y_{i,t}^A \geq \frac{t_r Z_r}{m}(1-\epsilon_{y,r})$.

Let $S_s^r(X_i) = \sum_{t=t_r+1}^{t_r+s} X_{i,t}$ denote the amount of resource i consumed in the first s steps of stage r, and let $S_s^r(Y_i) = \sum_{t=t_r+1}^{t_r+s} Y_{i,t}$ denote the resource i profit in the first s steps of stage r.

$$\begin{aligned}
\mathbf{Pr}\left[\sum_{t=t_r+1}^{t_{r+1}} X_{i,t}^A \geq \frac{t_r c_i}{m}(1+\epsilon_{x,r})\right] &= \mathbf{Pr}\left[\frac{\sum_{t=t_r+1}^{t_{r+1}} X_{i,t}^A}{\gamma c_i} \geq \frac{t_r}{m\gamma}(1+\epsilon_{x,r})\right] \\
&= \mathbf{Pr}\left[(1+\epsilon_{x,r})^{\frac{\sum_{t=t_r+1}^{t_{r+1}} X_{i,t}^A}{\gamma c_i}} \geq (1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}\right] \\
&\leq \mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{\sum_{t=t_r+1}^{t_{r+1}} X_{i,t}^A}{\gamma c_i}}\right] / (1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}} \\
&= \frac{\mathbf{E}\left[\prod_{t=t_r+1}^{t_{r+1}} (1+\epsilon_{x,r})^{\frac{X_{i,t}^A}{\gamma c_i}}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} \qquad (2.6)
\end{aligned}$$

$$\begin{aligned}
\mathbf{Pr}\left[\sum_{t=t_r+1}^{t_{r+1}} Y_{i,t}^A \leq \frac{t_r Z_r}{m}(1-\epsilon_{y,r})\right] &= \mathbf{Pr}\left[\frac{\sum_{t=t_r+1}^{t_{r+1}} Y_{i,t}^A}{w_{\max}} \leq \frac{t_r Z_r}{m w_{\max}}(1-\epsilon_{y,r})\right] \\
&= \mathbf{Pr}\left[(1-\epsilon_{y,r})^{\frac{\sum_{t=t_r+1}^{t_{r+1}} Y_{i,t}^A}{w_{\max}}} \geq (1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}\right] \\
&\leq \mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{\sum_{t=t_r+1}^{t_{r+1}} Y_{i,t}^A}{w_{\max}}}\right] / (1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}} \\
&= \frac{\mathbf{E}\left[\prod_{t=t_r+1}^{t_{r+1}} (1-\epsilon_{y,r})^{\frac{Y_{i,t}^A}{w_{\max}}}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}} \qquad (2.7)
\end{aligned}$$

If our algorithm A was P $\left(\text{and therefore we can use } \mathbf{E}[X_{i,t}^*] \leq \frac{c_i}{m} \text{ and } \mathbf{E}[Y_{i,t}^*] \geq \frac{W_E}{m} \geq \frac{Z_r}{m}\right)$, the total failure probability for each stage r which is the sum of (2.6) and (2.7) for all the i's would have been $n \cdot \left[e^{\frac{-\epsilon_{x,r}^2}{4\gamma}\frac{t_r}{m}} + e^{\frac{-\epsilon_{y,r}^2}{2}\frac{t_r Z_r}{m w_{\max}}}\right] = n \cdot \left[\frac{\delta}{2n} + \frac{\delta}{2n}\right] = \delta$. The goal is to design an algorithm A for stage r that, unlike P, does not know the distribution but also obtains the same δ failure probability, just as we did in Section 2.3.2. That is we want to show that the sum of (2.6) and (2.7) over all i's is at most δ:

$$\sum_i \frac{\mathbf{E}\left[\prod_{t=t_r+1}^{t_{r+1}}(1+\epsilon_{x,r})^{\frac{X_{i,t}^A}{\gamma c_i}}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} + \sum_i \frac{\mathbf{E}\left[\prod_{t=t_r+1}^{t_{r+1}}(1-\epsilon_{y,r})^{\frac{Y_{i,t}^A}{w_{\max}}}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}} \le \delta.$$

For the algorithm $A^s P^{t_r-s}$, the above quantity can be rewritten as

$$\sum_i \frac{\mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{S_s^r\left(X_i^A\right)}{\gamma c_i}} \prod_{t=t_r+s+1}^{t_{r+1}}(1+\epsilon_{x,r})^{\frac{X_{i,t}^*}{\gamma c_i}}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} +$$
$$\sum_i \frac{\mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{S_s^r\left(Y_i^A\right)}{w_{\max}}} \prod_{t=t_r+s+1}^{t_{r+1}}(1-\epsilon_{y,r})^{\frac{Y_{i,t}^*}{w_{\max}}}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}}$$

which, by using $(1+\epsilon)^x \le 1+\epsilon x$ for $0 \le x \le 1$, is in turn upper bounded by

$$\sum_i \frac{\mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{S_s^r\left(X_i^A\right)}{\gamma c_i}} \prod_{t=t_r+s+1}^{t_{r+1}}\left(1+\epsilon_{x,r}\frac{X_{i,t}^*}{\gamma c_i}\right)\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} +$$
$$\sum_i \frac{\mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{S_s^r\left(Y_i^A\right)}{w_{\max}}} \prod_{t=t_r+s+1}^{t_{r+1}}\left(1-\epsilon_{y,r}\frac{Y_{i,t}^*}{w_{\max}}\right)\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}}.$$

Since for all t, the random variables $X_{i,t}^*$, $X_{i,t}^A$, $Y_{i,t}^*$ and $Y_{i,t}^A$ are all independent, and $\mathbf{E}[X_{i,t}^*] \le \frac{c_i}{m}$, $\mathbf{E}[Y_{i,t}^*] \ge \frac{W_E}{m}$, and $\frac{W_E}{w_{\max}} \ge \frac{1}{\gamma}$, the above is in turn upper bounded by

$$\sum_i \frac{\mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{S_s^r\left(X_i^A\right)}{\gamma c_i}} \left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)^{t_r-s}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} + \sum_i \frac{\mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{S_s^r\left(Y_i^A\right)}{w_{\max}}} \left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)^{t_r-s}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}}. \tag{2.8}$$

Let $\mathcal{F}_r[A^s P^{t_r-s}]$ denote the quantity in (2.8), which is an upper bound on failure probability of the hybrid algorithm $A^s P^{t_r-s}$ for stage r. We just showed that $\mathcal{F}_r[P^{t_r}] \le \delta$. We now prove that for all $s \in \{0, 1, \ldots, t_r - 1\}$, $\mathcal{F}_r[A^{s+1} P^{t_r-s-1}] \le \mathcal{F}_r[A^s P^{t_r-s}]$, thus proving that $\mathcal{F}_r[A^{t_r}] \le \delta$, i.e., running the algorithm A for all the t_r steps of stage r results in a failure with probability at most δ.

Assuming that for all $s < p$, the algorithm A has been defined for the first $s+1$ steps in such a way that $\mathcal{F}_r[A^{s+1}P^{t_r-s-1}] \leq \mathcal{F}_r[A^s P^{t_r-s}]$, we now define A for the $p+1$-th step of stage r in a way that will ensure that $\mathcal{F}_r[A^{p+1}P^{t_r-p-1}] \leq \mathcal{F}_r[A^p P^{t_r-p}]$. We have

$$\begin{aligned}
\mathcal{F}_r[A^{p+1}P^{m-p-1}] &= \sum_i \frac{\mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{S^r_{p+1}\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{S^r_{p+1}\left(Y_i^A\right)}{w_{\max}}}\left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}} \\
&\leq \sum_i \frac{\mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{S^r_{p}\left(X_i^A\right)}{\gamma c_i}}\left(1+\epsilon_{x,r}\frac{X^A_{i,t_r+p+1}}{\gamma c_i}\right)\left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{S^r_{p}\left(Y_i^A\right)}{w_{\max}}}\left(1-\epsilon_{y,r}\frac{Y^A_{i,t_r+p+1}}{w_{\max}}\right)\left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}}
\end{aligned} \tag{2.9}$$

Define

$$\begin{aligned}
\phi^r_{i,s} &= \frac{\epsilon_{x,r}}{\gamma c_i}\left[\frac{(1+\epsilon_{x,r})^{\frac{S^r_s\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)^{t_r-s-1}}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}}\right] \\
\psi^r_{i,s} &= \frac{\epsilon_{y,r}}{w_{\max}}\left[\frac{(1-\epsilon_{y,r})^{\frac{S^r_s\left(Y_i^A\right)}{w_{\max}}}\left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)^{t_r-s-1}}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}}\right]
\end{aligned}$$

Define the step $p+1$ of algorithm A as picking the following option k^* for request j:

$$k^* = \arg\min_k \left\{ \sum_i a_{ijk} \cdot \phi^r_{i,p} - \sum_i w_{ijk} \cdot \psi^r_{i,p} \right\}.$$

By the above definition of step $p+1$ of algorithm A (for stage r), it follows that for any two algorithms with the first p steps being identical, and the last $t_r - p - 1$ steps following the Hypothetical-Oblivious algorithm P, algorithm A's $p+1$-th step is the one that minimizes expression (2.9). In particular it follows that expression (2.9) is upper bounded by the same expression where the $p+1$-the step is according to X^*_{i,t_r+p+1} and Y^*_{i,t_r+p+1}, i.e., we replace X^A_{i,t_r+p+1} by X^*_{i,t_r+p+1} and Y^A_{i,t_r+p+1} by Y^*_{i,t_r+p+1}. Therefore we have

$$\begin{aligned}
\mathcal{F}_r[A^{p+1}P^{m-p-1}] \leq \sum_i & \frac{\mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{S^r_p\left(X^A_i\right)}{\gamma c_i}}\left(1+\epsilon_{x,r}\frac{X^*_{i,t_r+p+1}}{\gamma c_i}\right)\left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} + \\
& \sum_i \frac{\mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{S^r_p\left(Y^A_i\right)}{w_{\max}}}\left(1-\epsilon_{y,r}\frac{Y^*_{i,t_r+p+1}}{w_{\max}}\right)\left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}} \\
\leq \sum_i & \frac{\mathbf{E}\left[(1+\epsilon_{x,r})^{\frac{S^r_p\left(X^A_i\right)}{\gamma c_i}}\left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)\left(1+\frac{\epsilon_{x,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1+\epsilon_{x,r})^{(1+\epsilon_{x,r})\frac{t_r}{m\gamma}}} + \\
& \sum_i \frac{\mathbf{E}\left[(1-\epsilon_{y,r})^{\frac{S^r_p\left(Y^A_i\right)}{w_{\max}}}\left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)\left(1-\frac{\epsilon_{y,r}}{m\gamma}\right)^{t_r-p-1}\right]}{(1-\epsilon_{y,r})^{(1-\epsilon_{y,r})\frac{t_r Z_r}{m w_{\max}}}} \\
= \mathcal{F}_r[A^pP^{t_r-p}] &
\end{aligned}$$

This completes the proof of Theorem 2.2.

2.3.4 Approximate Estimations

Our Algorithm 2 in Section 2.3.3 required periodically computing the optimal solution to an offline instance. Similarly, our Algorithm 1 in Section 2.3.2 requires the value of W_E to be given. Suppose we could only approximately estimate these quantities, do our results carry through approximately? That is, suppose the solution to the offline instance is guaranteed to be at least $\frac{1}{\alpha}$ of the optimal, and the stand-in that we are given for W_E is guaranteed to be at least $\frac{1}{\alpha}$ of W_E. Both our Theorem 2.2 and Theorem 2.12 go through with just the W_E replaced by W_E/α. Every step of the proof of the exact version goes through in this approximate version, and so we skip the formal proof for this statement.

2.3.5 Adversarial Stochastic Input

In this section, we relax the assumption that requests are drawn i.i.d. every time step. We give three different models of Adversarial Stochastic Input (ASI) with equally good revenue guarantees.

2.3.5.1 ASI Model 1

In this model, the distribution for each time step need not be identical, but an adversary gets to decide which distribution to sample a request from. The adversary could even use how the algorithm has performed in the first $t-1$ steps in picking the distribution for a given time step t. The guarantee we give is against the worst distribution over all time steps picked by the adversary. More formally, let $W_E(t)$ denote the optimal profit for the expected instance of distribution of time step t. Note that $W_E(t)$ is a random variable because the distribution adversary picks at step t could be dependent on what transpired in the previous $t-1$ stages. Suppose $W_E(t) \geq W_E$ for all t, then, given the value of W_E, our Algorithm 1 in Section 2.3.2 will guarantee a revenue of $W_E(1-2\epsilon)$ with a probability of at least $1-\epsilon$ assuming $\gamma = O(\frac{\epsilon^2}{\log(n/\epsilon)})$, just like the guarantee in Theorem 2.12.

Algorithm 1 works for this ASI model because, the proof did not use the similarity of the distributions beyond the fact that $\mathbf{E}[X^*_{i,t}|X^*_{i,t'}\forall t' < t] \leq \frac{c_i}{m}$ for all values of $X^*_{i,t'}$, and $\mathbf{E}[Y^*_{i,t}|Y^*_{i,t'}\forall t' < t] \geq \frac{W_E}{m}$ for all values of $Y^*_{i,t'}$ (Here $X^*_{i,t}$ and $Y^*_{i,t}$ denote the random variables following from allocation according the optimal solution to the expected instance of the distribution used in stage t). In other words, distributions being identical and independent is not crucial, but the fact that the expected instances of these distributions have a minimum profit guarantee inspite of all the dependencies between the distributions is sufficient. Both of these inequalities remain true in this model of ASI also, and thus it easy to verify that Algorithm 1 works for this model.

2.3.5.2 ASI Model 2

In this model, as in Model 1, distributions are picked by the adversary every time step. However we tradeoff one property of Model 1 for another. In Model 1, we asked for knowledge of $W_E = \min_t W_E(t)$. Here instead, we don't ask for W_E, but require that $W_E(t)$ remains the same for all t, call it W_E. We now use Algorithm 2 from Section 2.3.3, and it is easy to see that we get the same guarantee for this model as the guarantee for the i.i.d. model in Section 2.3.3 (Theorem 2.2). The reason again is that we have $\mathbf{E}[X^*_{i,t}|X^*_{i,t'}\forall t' < t] \leq \frac{c_i}{m}$ and $\mathbf{E}[Y^*_{i,t}|Y^*_{i,t'}\forall t' < t] = \frac{W_E}{m}$ in this model and that is sufficient for Algorithm 2 to give this guarantee.

2.3.5.3 ASI Model 3

In this model, the distributions are picked by the adversary in the above 2 models, and the expected instances of these models can have arbitrary guarantees on profit. In other words the profit for expected instance of distribution for time step t could be an arbitrary $W_E(t)$. Our benchmark is $W_E = \frac{\sum_{t=1}^m W_E(t)}{m}$, unlike $\min_t W_E(t)$ of model 1: this is clearly a much stronger benchmark. As a tradeoff, what we ask for is that we know $W_E(t)$ for every t. While the adversary could adaptively pick the distributions, we ask for the $W_E(t)$ for all t at the beginning of the algorithm.

The relevance of this model for the real world is that for settings like display ads, the distribution fluctuates over the day. In general a day is divided into many chunks and within a chunk, the distribution is assumed to remain i.i.d. and that the optimizer has fair idea of the revenue to be extracted for any given chunk. This is exactly captured by this model.

A slight modification of our Algorithm 1 in Section 2.3.2 will give a revenue of $\frac{\sum_{t=1}^m W_E(t)}{m}(1-2\epsilon)$ with probability at least $1-\epsilon$, i.e., $W_E(1-2\epsilon)$ w.p. at least $(1-\epsilon)$. Among the two potential functions $\phi_{i,s}$ and $\psi_{i,s}$, we modify $\psi_{i,s}$ in the most natural way to account for the fact that distributions change every step.

Define

$$\phi_{i,s} = \frac{1}{c_i}\left[\frac{(1+\epsilon)^{\frac{S_s\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-s-1}}{(1+\epsilon)^{\frac{1}{\gamma}}}\right]$$

$$\psi_{i,s} = \frac{1}{W_E(s+1)}\left[\frac{(1-\epsilon)^{\frac{S_s\left(Y_i^A\right)}{\gamma W_E}}\prod_{t=s+2}^m\left(1-\frac{\epsilon W_E(t)}{(1+\epsilon)W_E\gamma m}\right)}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}\right]$$

Note that when $W_E(t) = W_E$ for all t, then we get precisely the $\psi_{i,s}$ defined in Section 2.3.2 for Algorithm 1. We present our algorithm below in Algorithm 3.

We skip the proof for the profit guarantee of $W_E(1-2\epsilon)$ since it is almost identical to the proof in Section 2.3.2 for Algorithm 1.

Algorithm 3 : Algorithm for stochastic online resource allocation in ASI model 3

Input: Capacities c_i for $i \in [n]$, the total number of requests m, the values of γ and $W_E(t)$ for $t \in [m]$, an error parameter $\epsilon > 0$.
Output: An online allocation of resources to requests

1: Initialize $\phi_{i,0} = \frac{1}{c_i}\left[\frac{\left(1+\frac{\epsilon}{(1+\epsilon)\gamma m}\right)^{m-1}}{(1+\epsilon)^{\frac{1}{\gamma}}}\right]$, and, $\psi_{i,0} = \frac{1}{W_E(1)}\left[\frac{\prod_{t=2}^{m}\left(1-\frac{\epsilon W_E(t)}{(1+\epsilon)W_E\gamma m}\right)}{(1-\epsilon)^{\frac{1-\epsilon}{\gamma(1+\epsilon)}}}\right]$

2: **for** $s = 1$ to m **do**

3: If the incoming request is j, use the following option k^*:

$$k^* = \arg\min_{k \in K \cup \{\perp\}} \left\{ \sum_i a_{ijk} \cdot \phi_{i,s-1} - \sum_i w_{ijk} \cdot \psi_{i,s-1} \right\}.$$

4: $X_{i,s}^A = a_{ijk^*}$, $Y_{i,s}^A = w_{ijk^*}$

5: Update $\phi_{i,s} = \phi_{i,s-1} \cdot \left[\frac{(1+\epsilon)^{\frac{X_{i,s}^A}{\gamma c_i}}}{1+\frac{\epsilon}{(1+\epsilon)\gamma m}}\right]$, and, $\psi_{i,s} = \psi_{i,s-1} \cdot \left[\frac{W_E(s)}{W_E(s+1)}\right] \cdot \left[\frac{(1-\epsilon)^{\frac{Y_{i,s}^A}{\gamma W_E}}}{1-\frac{\epsilon}{(1+\epsilon)\gamma m}}\right]$

6: **end for**

2.4 Proof of Near-Optimality of Online Algorithm for Resource Allocation

In this section, we construct a family of instances of the resource allocation problem in the i.i.d. setting for which $\gamma = \omega(\epsilon^2/\log n)$ will rule out a competitive ratio of $1 - O(\epsilon)$. The construction closely follows the the construction by Agrawal et al. [AWY09] for proving a similar result in the random-permutation model.

The instance has $n = 2^z$ resources with B units of each resource, and $Bz(2 + 1/\alpha) + \sqrt{Bz}$ requests where $\alpha < 1$ is some scalar. Each request has only one "option", i.e., each request can either be dropped, or if served, consumes the same number of units of a specific subset of resources (which we construct below). This means that a request is simply a scalar times a binary string of length 2^z, with the ones (or the scalars) representing the coordinates of resources that are consumed by this request, if served.

The requests are classified into z categories. Each category in expectation consists of $m/z = B(2 + 1/\alpha) + \sqrt{B/z}$ requests. A category, indexed by i, is composed of two different binary vectors v_i and w_i (each of length 2^z). The easiest way to visualize these vectors is to construct two $2^z \times z$ $0-1$ matrices, with each matrix consisting of all possible binary strings of length z, written one string in a row. The first matrix lists the strings in ascending order and the second matrix in descending order. The i-th column of the first matrix multiplied by the scalar α is the vector v_i and the i-th column of the second matrix is the vector w_i. There are two properties of these vectors that are useful for us:

1. The vectors v_i/α and w_i are complements of one another
2. Any matrix of z columns, with column i being either v_i/α or w_i has exactly one row with all ones in it.

We are ready to construct the i.i.d. instance now. Each request is drawn from the following distribution. A given request could be, for each i, of type:

1. v_i and profit 4α with probability $\frac{B}{\alpha zm}$
2. w_i and profit 3 with probability $\frac{B}{zm}$
3. w_i and profit 2 with probability $\sqrt{\frac{B}{zm}}$
4. w_i and profit 1 with probability $\frac{B}{zm}$
5. Zero vector with zero profit with probability $1 - \frac{2B}{zm} - \sqrt{\frac{B}{zm}} - \frac{B}{\alpha zm}$

We use the following notation for request types: a $(2, w_i)$ request stands for a w_i type request of profit 2. Observe that the expected instance has an optimal profit of $\mathrm{OPT} = 7B$. This is obtained by picking for each i, the $\frac{B}{\alpha z}$ vectors of type v_i and profit 4α, along with $\frac{B}{z}$ vectors of type w_i with profit 3. Note that this exhausts every unit of every item, and thus, combined with the fact that the most profitable requests have been served, the value of $7B$ is indeed the optimal value. This means, any algorithm that obtains a $1 - \epsilon$ competitive ratio must have an expected profit of at least $7B - 7\epsilon B$.

Let $r_i(w)$ and $r_i(v)$ be the random variables denoting the number of vectors of type w_i and v_i picked by some $1 - \epsilon$ competitive algorithm ALG. Let $a_i(v)$ denote the total number of vectors of type v_i that arrived in this instance.

Lemma 2.14 *For some constant k, the $r_i(w)$'s satisfy*

$$\sum_i \mathbf{E}\left[|r_i(w) - B/z|\right] \leq 7\epsilon B + 4\sqrt{\alpha k B z}.$$

Proof: Let Y denote the set of indices i for which $r_i(w) > B/z$. One way to upper bound the total number of vectors of type v picked by ALG is the following. Split the set of indices into Y and $X = [z] \setminus Y$. The number of v's from Y is, by chosen notation, $\sum_{i \in Y} r_i(v)$. The number of v's from X, we show, is at most $\frac{B - \sum_{i \in Y} r_i(w)}{\alpha}$. Note that since there are only B copies of every item, it follows that $\alpha[\sum_i r_i(v)] \leq B$, and $\sum_i r_i(w) \leq B$. Further, by property 2 of v_i's and w_i's, we have that $\alpha[\sum_{i \in X} r_i(v)] + \sum_{i \in Y} r_i(w) \leq B$. This means that the number of v's from X is at most $\frac{B - \sum_{i \in Y} r_i(w)}{\alpha}$.

Let $P = \sum_{i \in Y}(r_i(w) - B/z)$, and $M = \sum_{i \in X}(B/z - r_i(w))$. Showing $\mathbf{E}[P + M] \leq 7\epsilon B + 4\sqrt{\alpha k B z}$ proves the lemma. By an abuse of notation, let ALG also be the profit obtained by the algorithm ALG and let $\text{best}w_i(t)$ denote the most profitable t requests of type w_i in a given instance. Note that $4B + \sum_{i=1}^{z} \text{best}w_i(B/z) \leq 7B = \text{OPT}$. We upper-bound $\mathbf{E}[ALG]$ as:

$$
\begin{aligned}
\mathbf{E}[ALG] \leq \mathbf{E}\left[\sum_{i=1}^{z} \text{best}w_i(r_i(w))\right] &+ 4\alpha\left[\frac{B - \sum_{i \in Y}\mathbf{E}[r_i(w)]}{\alpha} + \sum_{i \in Y}\mathbf{E}[r_i(v)]\right] \\
&\leq \mathbf{E}\left[\sum_{i=1}^{z} \text{best}_i(B/z) + 3P - M\right] + 4\left(B - \mathbf{E}\left[\sum_{i \in Y}(r_i(w) - B/z) + |Y|B/z\right]\right) \\
&\qquad + 4\alpha\,\mathbf{E}\left[\sum_{i \in Y} r_i(v)\right] \\
&\leq \text{OPT} - \mathbf{E}[P + M] + 4\alpha\,\mathbf{E}\left[\sum_{i \in Y}\left(r_i(v) - \frac{B}{\alpha z}\right)\right] \\
&\left(\text{Since } P = \sum_{i \in Y}(r_i(w) - B/z)\ \right) \\
&\leq \text{OPT} - \mathbf{E}[P + M] + 4\alpha\,\mathbf{E}\left[\sum_{i \in Y}\left(a_i(v) - \frac{B}{\alpha z}\right)\right] (\text{Since } r_i(v) \leq a_i(v)) \\
&\leq \text{OPT} - \mathbf{E}[P + M] + 4\alpha\,\mathbf{E}\left[\sum_{i : a_i(v) \geq \frac{B}{\alpha z}}\left(a_i(v) - \frac{B}{\alpha z}\right)\right] \\
&\leq \text{OPT} - \mathbf{E}[P + M] + 4\alpha \cdot z \cdot k' \cdot \sqrt{\frac{B}{\alpha z}} \\
&\text{(where } k' \text{ is some constant from Central Limit Theorem)} \\
&\leq \text{OPT} - \mathbf{E}[P + M] + 4\sqrt{\alpha k B z} \text{ (where } k \text{ is } k'^2)
\end{aligned}
$$

The inequality that follows from CLT uses the fact that for a random variable $X \sim (m, c/m)$ (X is binomially distributed with success probability of c/m), whenever $c = \omega(1)$, and $c \leq m$, we have that $E[X|X \geq c] = c + k'\sqrt{c}$, for some constant k'. In this case, we have $\frac{B}{\alpha z}$ in place of c. For example, if $n = \log(m)$ (and thus $z = \log n = \log\log m$), as long as $B = \omega(\log\log m)$ and $B \leq m$, the CLT inequality will hold. Note that α could have been any constant and this argument still holds. ■

We are now ready to prove Theorem 2.3, which we restate here for convenience.

Theorem 2.3 *There exist instances with $\gamma = \frac{\epsilon^2}{\log(n)}$ such that no algorithm, even with complete knowledge of the distribution, can get a $1 - o(\epsilon)$ approximation factor.*

Proof: We first give the overview of the proof before providing a detailed argument.

Overview. Lemma 2.14 says that $r_i(w)$ has to be almost always close to B/z for all i. In particular, the probability that $\sum_i |r_i(w) - B/z| \leq 4\left(7\epsilon B + 4\sqrt{\alpha k B z}\right)$ is atleast 3/4. In this proof we show, in an argument similar to the one in [AWY09], that if this has to be true, one has to lose a revenue of $\Omega(\sqrt{Bz}) - 4(7\epsilon B + 4\sqrt{\alpha k B z})$. Since α can be set to any arbitrary constant, this means that we lose a revenue of $\Omega(\sqrt{Bz}) - 28\epsilon B$. Since OPT is $7B$, to get a $1 - \epsilon$ approximation, we require that $\Omega(\sqrt{Bz}) - 28\epsilon B \leq 7\epsilon B$. Thus, we need $B \geq \Omega(\frac{\log m}{\epsilon^2})$. In other words, we require $\gamma = \frac{1}{B} \leq O(\frac{\epsilon^2}{\log m})$.

In Detail. We now proceed to prove the claim that a revenue loss of $\Omega(\sqrt{Bz}) - 4(7\epsilon B + \sqrt{\alpha k B z})$ is inevitable. We just showed that with a probability of at least 3/4, $\sum_i |r_i(w) - B/z| \leq 4\left(7\epsilon B + 4\sqrt{\alpha k B z}\right)$. For now we assume that $r_i(w)$ should be exactly B/z and later account for the probability 1/4 leeway and also the $4\left(7\epsilon B + 4\sqrt{\alpha k B z}\right)$ error that is allowed by Lemma 2.14. With this assumption, we show that for each i there is a loss of $\Omega(\sqrt{B/z})$.

For each i let o_i denote the number of $(1, w_i)$ requests that the algorithm served in total. With a constant probability the number of 3's and 2's (of type w_i) exceed B/z. If $o_i = \Omega(\sqrt{B/z})$ there is a loss of at least $\Omega(\sqrt{B/z})$ because of picking ones instead of 2's or 3's. This establishes the $\Omega(\sqrt{B/z})$ loss that we wanted to prove, for this case.

Suppose $o_i < \Omega(\sqrt{Bz})$. For each i, let R_i be the set of requests of type w_i with profit either 1 or 3. For every i, with a constant probability $2B/z - 2\sqrt{B/z} \leq |R_i| \leq 2B/z + 2\sqrt{B/z}$. Conditional on the set R_i we make the following two observations:

- the types of requests in R_i are independent random variables that take value 1 or 3 with equal probability.

- the order of requests in R_i is a uniformly random permutation of R_i

Now consider any $(2, w_i)$ request, say t-th request, of profit 2. With a costant probability this request can be served without violating any capacity constraints, and thus, the algorithm has to decide whether or not to serve this request. In at least 1/2 of the random permutations of R_i, the number of bids from set R_i before the bid t is less than B/z. Conditional on this event, the profits of requests in R_i before t, with a constant probability could:

1. take values such that there are enough $(3, w_i)$ requests after t to make the total number of w_i requests picked by the algorithm to be at least B/z;

2. take values such that even if all the $(3, w_i)$ requests after t were picked, the total number of w_i requests picked is at most $B/z - \sqrt{B/z}$ with a constant probability.

In the first kind of instances (where number of $(3, w_i)$ requests are more than B/z) retaining $(2, w_i)$ causes a loss of 1 as we could have picked a 3 instead. In the second kind, skipping $(2, w_i)$ causes a loss of 1 since we could have picked that 2 instead of a 1. Thus there is an inevitable constant probability loss of 1 per $(2, w_i)$ request. Thus in expectation, there is a $\Omega(\sqrt{B/z})$ loss.

Thus whether $o_i = \sqrt{B/z}$ or $o_i < \sqrt{B/z}$, we have established a loss of $\Omega(\sqrt{B/z})$ per i and thus a total expected loss of $\Omega(\sqrt{Bz})$. This is under the assumption that $r_i(w)$ is exactly B/z. There is a leeway of $4\left(7\epsilon B + 4\sqrt{\alpha k B z}\right)$ granted by Lemma 2.14. Even after that leeway, since α can be made an arbitrarily small constant and Lemma 2.14 still holds, we have the loss at $\Omega(\sqrt{Bz}) - 28\epsilon B$. Now after the leeway, the statement $\sum_i |r_i(w) - B/z| \leq 4\left(7\epsilon B + 4\sqrt{\alpha k B z}\right)$ has to hold only with probability 3/4. But even this puts the loss at $\Omega(\sqrt{Bz}) - 21\epsilon B$

Therefore, $E[ALG] \leq \text{OPT} - \Omega(\sqrt{Bz}) - 21\epsilon B$. Since $\text{OPT} = 7B$, we have $E[ALG] \leq \text{OPT}(1 - \Omega(\sqrt{z/B}) - 21\epsilon)$, and in order to get $1 - O(\epsilon)$ approximation we need $\Omega(\sqrt{z/B} - 21\epsilon) \leq O(\epsilon)$, implying that $B \geq \Omega(z/\epsilon^2) = \Omega(\log m/\epsilon^2)$. ■

2.5 Asymptotically Optimal Prior Robust Online Algorithms for Adwords

In this section, we study the special case of the Adwords problem defined in Section 2.2.2. We begin by defining saturated instances for the adwords problem: any instance that is drawn from a distribution for which there is *some optimal* solution to the expected instance which completely consumes the budget B_i for every advertiser i is called a saturated instance. In Section 2.5.2, we design a completely distribution independent independent algorithm for saturated instances. In Section 2.5.4, for general instances where the optimal consumption in the expected instance for advertiser i is $C_i \leq B_i$, we design algorithms with the knowledge of the C_i's: these are total n parameters one for each advertiser. In saturated instances since C_i's are equal to B_i's (and B_i's are known to the algorithm designer), there is no requirement for the distribution dependent parameter C_i's, where as in general instances we need them.

2.5.1 Saturated Instances: Completely Known Distribution

Like in Section 2.3 we begin with the case where we completely know the distribution, and design a hypothetical algorithm called Hypothetical-Oblivious that achieves a revenue of $\sum_i B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$ (recall that the total revenue achievable is at most $\sum_i B_i$). The Hypothetical-Oblivious algorithm uses an optimal solution to the expected instance to perform the online allocation of queries. Note that since the distribution is unknown to the algorithm designer, the expected instance cannot be computed, and thus Hypothetical-Oblivious is a hypothetical algorithm. Let $\{x^*_{ij}\}$ denote some optimal solution to the expected instance. When query j arrives, the algorithm Hypothetical-Oblivious assigns it to advertiser i with probability x^*_{ij}. Thus Hypothetical-Oblivious is a non-adaptive algorithm that follows the same assignment probabilities irrespective of the previously arrived queries and their assignments. Even if the budget of an advertiser has been fully consumed, Hypothetical-Oblivious does not make any alterations to the said assignment rule, i.e., it will get zero revenue from such allocations.

2.5.1.1 Hypothetical-Oblivious Algorithm on Single-Bid Instances

For now, we restrict attention to instances where advertiser i's bids are either b_i or zero, and use the analysis for this case to derive the analysis for the more general case in Section 2.5.1.2. The Hypothetical-Oblivious algorithm, at any given time-step, assigns a query to advertiser i with probability $\sum_j p_j b_{ij} x^*_{ij} = B_i/m$, where $b_{ij} \in \{0, b_i\}$, and the equality follows from the fact that the solution $\{x^*_{ij}\}$ consumes the entire budget in the expected instance. When bids are 0 or b_i, observe that the Hypothetical-Oblivious algorithm is simply a balls and bins process which, in each time-step throws a ball into bin i with probability $\frac{B_i}{b_i m}$. Note that full consumption implies that $m \geq \sum_i \frac{B_i}{b_i}$. We now show that the expected number of balls(queries) in bin(advertiser) i at the end of this process is at least $\frac{B_i}{b_i}\left(1 - \sqrt{\frac{b_i}{2\pi B_i}}\right) = \frac{B_i}{b_i}(1 - \sqrt{\frac{\gamma_i}{2\pi}})$. Since each ball is worth b_i, this proves that the revenue in bin i at the end is $B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$.

Lemma 2.15 *In a balls and bins process where a given bin with capacity k receives a ball with probability k/m at each step, the expected number of balls in the given bin after m steps is at least $k(1 - \frac{1}{\sqrt{2\pi k}})$.*

Proof: Let X_m be the binomial random variable denoting the number of balls thrown at the bin, when m is the total number of balls, and k/m is the probability with which a ball is thrown at the bin . Clearly $\mathbf{E}[X_m] = k$. The quantity we are interested in, that is the expected number of balls in the bin after m steps, is $\mathbf{E}[\min(X_m, k)]$. This quantity, as proved in [Yan11], monotonically decreases in m. In other words, more balls get wasted (due to overflow) if we throw $m+1$ balls with probability $\frac{k}{m+1}$ each, instead of m balls with probability $\frac{k}{m}$ each. Therefore the quantity $\mathbf{E}[\min(X_m, k)]$ attains its minimum as $m \to \infty$, and equals $k(1 - \frac{k^k}{k!e^k}) \geq k(1 - \frac{1}{\sqrt{2\pi k}})$ [Yan11]. ■

Thus the competitive ratio achieved by Hypothetical-Oblivious is at least $1 - \frac{k^k}{k!e^k}$ where $k = \lfloor 1/\gamma \rfloor$.

Corollary 2.16 *For any single-bid instance of the adwords problem that is saturated, the Hypothetical-Oblivious algorithm achieves a revenue of $\sum_i B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$.*

2.5.1.2 Extending Hypothetical-Oblivious to Arbitrary Bids

We now show that the Hypothetical-Oblivious algorithm can actually achieve the same revenue of $\sum_i B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$ for arbitrary bids.

In the arbitrary bids case, advertiser i's bids are in $[0, b_i]$ (instead of $\{0, b_i\}$ of the previous section). That is, b_i is just the maximum bid and several other smaller bids are also possible. The Hypothetical-Oblivious algorithm for such instances is like a balls and bins process albeit the balls can be fractional balls. That is in each step, a ball of size $s \in [0, b_i]$ is thrown into the bin i, where $s = b_{ij}$ with probability $p_j x_{ij}$, and thus, the expected "amount" of ball aimed at bin i in a single step is $\sum_j p_j b_{ij} x_{ij} = B_i/m$. Our argument is that for *every* bin, any arbitrary bid instance (which we also refer to as fractional bid instance) never performs worse in expectation compared to the corresponding single-bid instance (which we also refer to as integral bid instance). Thus, from now on we fix some particular bin, say i, and the random variables we define in the rest of this section 2.5.1.2 are with respect to this particular bin i, though we drop this index.

Let the random variables X_F^j and X_I^j denote respectively, the amount of ball aimed at a given bin in step j, when the bids are in $[0, b]$ and $\{0, b\}$. Since budgets are fully consumed, we have that for a given bin of capacity k, the expectations of X_F^j and X_I^j are equal, i.e., $\mathbf{E}[X_F^j] = \mathbf{E}[X_I^j] = k/m$.

Let the random variables X_F and X_I denote respectively, the total amount of balls aimed at the given bin over all the steps, in the fractional bid and the integral bid case. That is, $X_F = \sum_j X_F^j$, and $X_I = \sum_j X_I^j$. The amount of ball that has landed in the given bin (in the fractional bid case) is given by $\min(k, X_F)$, and thus $\mathbf{E}[\min(k, X_F)]$ is the quantity we are interested in. Similarly $\mathbf{E}[\min(k, X_I)]$ is the quantity of interest for integral bid case. By Lemma 2.15, we know that in the integral bid case the expected number of balls landed in the bin is $\mathbf{E}[\min(k, X_I)] \geq k(1 - \frac{k^k}{k!e^k})$. If we prove this inequality for X_F also, that completes the proof. For a given expected amount of ball in the balls and bins process (in this case we have k/m amount of ball thrown in expectation in each step), the maximum wastage of balls (due to overflow) occurs when the distribution of ball size is extreme, i.e., either b or zero. Thus, the wastage for the $[0, b]$ case is at most the wastage for $\{0, b\}$ case, and hence $\mathbf{E}[\min(k, X_F)] \geq \mathbf{E}[\min(k, X_I)]$. This fact follows immediately, for example, from Corollary 4 of [LP03]. Thus we have the following:

Corollary 2.17 *For any saturated instance of the adwords problem, the Hypothetical-Oblivious algorithm achieves a revenue of* $\sum_i B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$.

2.5.2 Saturated Instances: Completely Unknown Distribution

We now proceed to construct a distribution independent algorithm for saturated instances of the adwords problem that achieves at least as much revenue as achieved by the hypothetical algorithm Hypothetical-Oblivious. While our algorithm, given by Algorithm 4, remains the same for integral(single-bid) and fractional (arbitrary) bids case, the argument is easier for integral bid case. Therefore, we begin with the integral case first.

Algorithm 4 Distribution independent algorithm for saturated instances

Input: Budgets B_i for $i \in [n]$, maximum possible bids $b_i = \max_j b_{ij}$ for $i \in [n]$, and the total number of queries m

Output: An online assignment of queries to advertisers

1: Initialize $R_i^0 = B_i$ for all i
2: **for** $t = 1$ to m **do**
3: Let j be the query that arrives at time t
4: For each advertiser i, compute using Equation (2.10)

$$\Delta_i^t = \min(b_{ij}, R_i^{t-1}) + \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1} - \min(b_{ij}, R_i^{t-1}), m-t\right) - \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1}, m-t\right)$$

5: Assign the query to the advertiser $i^* = \operatorname{argmax}_{i \in [n]} \Delta_i^t$
6: Set $R_i^t = R_i^{t-1}$ for $i \neq i^*$ and set $R_{i^*}^t = R_{i^*}^{t-1} - \min(b_{i^*j}, R_{i^*}^{t-1})$
7: **end for**

2.5.2.1 Our Algorithm for Single-bid instances (or Integral Instances)

The idea of our algorithm is quite simple. When a query arrives, do the following: assuming that the Hypothetical-Oblivious algorithm will be implemented for the rest of steps, find the advertiser i, who when assigned the query will maximize the sum of the revenue obtained in this step together with the residual expected revenue that can be obtained in the remaining steps (where the residual expected revenue is calculated taking the remaining steps to be Hypothetical-Oblivious). Since the Hypothetical-Oblivious algorithm is just a balls and bins process that throws balls of value b_i into bin i with probability $\frac{B_i}{b_i m}$, the algorithm is the following: assuming that the remaining steps follow the simple and non-adaptive balls and bins process that throws a ball of value b_i into bin i with probability $\frac{B_i}{b_i m}$, compute the bin that when assigned the ball will maximize the sum of this step's revenue and expected residual revenue.

More formally, let j be the t-th query. We compute the difference in the expected revenue contributed by i, when it is assigned query j and when it is not assigned query j. That advertiser who maximizes this difference is assigned the query. The expected residual revenue $\mathcal{R}(p, b, k, l)$ is a function of the following four quantities:

- the probability p with which a ball is thrown into this bin in the balls and bins process;
- the value b of each ball;
- the remaining space k in the bin;
- the remaining number of balls l;

Let R_i^{t-1} denote the remaining budget of advertiser i, when the t-th query/ball arrives. Then for each advertiser i, we compute the difference

$$\Delta_i^t = \min(b_{ij}, R_i^{t-1}) + \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1} - \min(b_{ij}, R_i^{t-1}), m - t\right) - \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1}, m - t\right),$$

and assign the query to the advertiser $i^* \in \operatorname{argmax}_i \Delta_i^t$. This is precisely what Algorithm 4 describes.

The quantity $\mathcal{R}(p, b, k, l)$ is computed as follows. The residual expected revenue can be seen as the difference of two quantities

- the expected amount of balls to be aimed at the bin in the remaining steps = blp;
- the expected amount of wasted balls, where waste occurs when $\lceil \frac{k}{b} \rceil$ or more balls are thrown, and is given by $\sum_{r=\lceil \frac{k}{b} \rceil}^{l} (rb - k)\binom{l}{r} p^r (1-p)^{l-r}$.

Thus we have

$$\mathcal{R}(p, b, k, l) = blp - \sum_{r=\lceil \frac{k}{b} \rceil}^{l} (rb - k)\binom{l}{r} p^r (1-p)^{l-r} \tag{2.10}$$

Lemma 2.18 *Algorithm 4 obtains an expected revenue at least as much as the Hypothetical-Oblivious algorithm.*

Proof: We prove the lemma by a hybrid argument. Let $A^r P^{m-r}$ represent a hybrid algorithm that runs our Algorithm 4 for the first r steps, and the Hypothetical-Oblivious algorithm for the remaining $m-r$ steps. If for all $1 \leq r \leq m$ we prove that $\mathbf{E}[A^r P^{m-r}] \geq \mathbf{E}[A^{r-1} P^{m-r+1}]$, we would have proved that $\mathbf{E}[A^m] \geq \mathbf{E}[P^m]$, and thus the lemma. But $\mathbf{E}[A^r P^{m-r}] \geq \mathbf{E}[A^{r-1} P^{m-r+1}]$ follows immediately from the definition of A which makes the expected revenue maximizing choice at any given step assuming Hypothetical-Oblivious on integer bids for the remaining steps. ■

2.5.2.2 Our Algorithm for Arbitrary Bid Instances (or Fractional Instances)

We now move on to fractional instances. Our algorithm is the same as for the integral instances, namely Algorithm 4. However, the hybrid argument here is a bit more subtle. Let P_F^m denote running the Hypothetical-Oblivious algorithm on fractional bids for m steps, and let P_I^m denote the same for integral bids. That is, P_I^m is a balls and bins process where bin i receives a ball of value b_i with probability $\frac{B_i}{b_i m}$ and zero otherwise, whereas P_F^m is a fractional balls and bins process where the expected amount of ball thrown into bin i is $\frac{B_i}{m}$ as in the integral Hypothetical-Oblivious, but the value of the balls can be anywhere in $[0, b_i]$. Our proof is going to be that $\mathbf{E}[A^m] \geq \mathbf{E}[P_I^m]$. Note that we do not compare the quantities that one would want to compare on first thought: namely $\mathbf{E}[A^m]$ and $\mathbf{E}[P_F^m]$, and the inequality could possibly go either way in this comparison. Instead we just compare our algorithm with Hypothetical-Oblivious working on integral instances. Since, we know from Corollary 2.16 we know that P_I^m has a good competitive ratio, it is enough to prove that $\mathbf{E}[A^m] \geq \mathbf{E}[P_I^m]$.

Lemma 2.19 $\mathbf{E}[A^m] \geq \mathbf{E}[P_I^m]$

Proof: We prove this lemma, like Lemma 2.18, by a hybrid argument, albeit with two levels. Let $A^r P_F P_I^{m-r-1}$ denote a hybrid which runs our Algorithm 4 for the first r steps (on the actual instance, which might be fractional), followed by a single step of Hypothetical-Oblivious on the actual instance (again, this could be fractional), followed by $m-r-1$ steps of Hypothetical-Oblivious on the integral instance (an integral instance which throws the same expected amount of ball into a bin in a given step as the fractional instance does). For all $1 \leq r \leq m$, we prove that

$$\mathbf{E}[A^r P_I^{m-r}] \geq \mathbf{E}[A^{r-1} P_F P_I^{m-r}] \geq \mathbf{E}[A^{r-1} P_I^{m-r+1}] \tag{2.11}$$

Chaining inequality (2.11) for all r, we get the Lemma.

The first inequality in (2.11) follows from the definition of A because it chooses that bin which maximizes the expected revenue assuming the remaining steps are integral Hypothetical-Oblivious.

The second inequality has a bin-by-bin proof. Fix a bin, and the proof follows from a discussion similar to that in Section 2.5.1.2, namely the maximum wastage occurs when the distribution of ball size is extreme, i.e., b_i or zero. Formally let X_F^r be the random variable representing the amount thrown by fractional Hypothetical-Oblivious in the r-th step, and let the corresponding integral variable X_I^r. If λ_{r-1} is the remaining budget after $r-1$ steps of A in the given bin, we are interested in how $\mathbf{E}[\min(\lambda_{r-1}, X_F^r + \sum_{t=r+1}^m X_I^t)]$ compares with $\mathbf{E}[\min(\lambda_{r-1}, \sum_{t=r}^m X_I^t)]$. Among all distributions with a fixed expectation, X_I^r is the distribution with extreme ball size and hence results in maximum wastage due to overflow (and hence minimum expected revenue), and thus we have the second inequality. This follows from Corollary 4 of [LP03].

■

Lemma 2.19 together with Corollary 2.16 proves the restriction of Theorem 2.4 for saturated instances.

General Instances. We now move onto general instances, i.e., instances that are not necessarily saturated. In this case, our algorithm requires to be given as input the budget consumption from every advertiser, by some optimal algorithm for the distribution instance. Let C_i denote the budget consumption from advertiser i by the optimal solution. The C_i's are part of the input to the algorithm. Note that $C_i \leq B_i$.

2.5.3 General Instances: Completely Known Distribution

Like in Section 2.5.1, before presenting our algorithm, we first describe the hypothetical Hypothetical-Oblivious algorithm and show that it also achieves a revenue of $\sum_i C_i(1-$

$\sqrt{\frac{\gamma_i}{2\pi}}) = \sum_i C_i \left(1 - \sqrt{\frac{b_i}{2\pi B_i}}\right)$.

2.5.3.1 Hypothetical-Oblivious Algorithm on Single-Bid Instances

Like in Section 2.5.1, we begin with the integral instances, i.e., advertiser i has bids of b_i or 0. Let $k_i = \frac{B_i}{b_i}$. To prove that Hypothetical-Oblivious achieves a revenue of $\sum_i C_i(1 - \sqrt{\frac{\gamma_i}{2\pi}}) = \sum_i C_i \left(1 - \sqrt{\frac{b_i}{2\pi B_i}}\right)$, it is enough to prove that the expected revenue from advertiser i is at least $C_i(1 - \frac{k_i^{k_i}}{k_i! e^{k_i}}) \geq C_i(1 - \frac{1}{\sqrt{2\pi k_i}})$.

Note that when interpreted as a balls and bins process, Hypothetical-Oblivious corresponds to throwing a ball of value b_i with probability $\frac{C_i}{b_i m}$ into bin i at every step (and not with probability $\frac{B_i}{b_i m}$). To prove that this process achieves a revenue of $C_i(1 - \frac{k_i^{k_i}}{k_i! e^{k_i}})$ is equivalent to proving that the value of wasted balls is at most $C_i \frac{k_i^{k_i}}{k_i! e^{k_i}}$. Dropping subscripts, and setting $b_i = 1$ since it doesn't affect analysis below, this means we have to prove that in a bin of capacity B, when a ball is thrown at every step with probability C/m, the expected number of wasted balls is at most $\frac{B^B}{B! e^B} \times C$, i.e., $\frac{B^{B+1}}{B! e^B} \times \frac{C}{B}$. From Section 2.5.1 Lemma 2.15 we know that $\frac{B^{B+1}}{B! e^B}$ is the expected number of wasted balls when a ball is thrown with probability B/m at every step. All we have to prove now is that when a ball is thrown with probability C/m, the expected number of wasted balls is at most C/B fraction of the same when the probability was B/m.

Lemma 2.20 *In a balls and bins process where a given bin with capacity B receives a ball with probability C/m at each step, the expected number of balls in the given bin after m steps is at least $(1 - \frac{B^B}{B! e^B}) \times C$.*

Proof: Let the random variable Y denote the number of balls wasted after m steps. Then our goal is to prove that $\mathbf{E}[Y] \leq \frac{B^B}{B! e^B} \times C = \frac{B^{B+1}}{B! e^B} \times \frac{C}{B}$. We have

$$\begin{aligned}
\mathbf{E}[Y] &= \sum_{r=B+1}^{m} (r - B) \binom{m}{r} \left(\frac{C}{m}\right)^r \left(1 - \frac{C}{m}\right)^{m-r} \\
&= \sum_{r=B+1}^{m} (r - B) \binom{m}{r} \left(\frac{B}{m}\right)^r \left(1 - \frac{B}{m}\right)^{m-r} \times \left(\frac{C/m}{B/m}\right)^r \left(\frac{1 - C/m}{1 - B/m}\right)^{m-r}
\end{aligned} \tag{2.12}$$

By Lemma 2.15 we know that

$$\sum_{r=B+1}^{m}(r-B)\binom{m}{r}\left(\frac{B}{m}\right)^{r}\left(1-\frac{B}{m}\right)^{m-r}\leq\frac{B^{B+1}}{B!e^{B}}.$$

Thus, it's enough if we prove that for all $r \geq B+1$, $f(r) = \left(\frac{C/m}{B/m}\right)^{r}\left(\frac{1-C/m}{1-B/m}\right)^{m-r} \leq C/B$. Since $C \leq B$ and hence$(\frac{C}{B})(\frac{1-B/m}{1-C/m}) \leq 1$, the function $f(r)$ decreases with r, and thus it's enough to prove that $f(B+1) \leq C/B$.

$$\frac{f(B+1)}{C/B}=\left(\frac{C}{B}\right)^{B}\left(1+\frac{B-C}{m-B}\right)^{m-(B+1)}\leq\left(\frac{C}{B}\right)^{B}\left(1+\frac{B-C}{m-B}\right)^{m-B}\leq\left(\frac{C}{B}\right)^{B}e^{B-C}$$

We now have to prove that $\left(\frac{C}{B}\right)^{B}e^{B-C} \leq 1$. Let $B = tC$, and thus $t \geq 1$. Thus, what we need to prove is that $e^{(t-1)C} \leq t^{tC}$ for $t \geq 1$. It is a straight forward exercise in calculus to prove that $e^{t-1} \leq t^{t}$ for all $t \geq 1$.

Thus $f(B+1) \leq C/B$, and this proves the lemma. ■

2.5.3.2 Extending Hypothetical-Oblivious to general bids

Extending Hypothetical-Oblivious to general bids is identical to the discussion in Section 2.5.1.2, and we omit it here.

2.5.4 General Instances: Partially Known Distribution

We now proceed to give an algorithm that has reduced dependence on distributions, namely, it uses just the C_i's as inputs as against the entire distributional knowledge. The algorithm is very similar to the one presented in Section 2.5.2 and is presented below as Algorithm 5.

The only difference between Algorithm 4 and Algorithm 5 is that the calculation of Δ_i^t is done using a probability of $\frac{C_i}{b_i m}$ instead of $\frac{B_i}{b_i m}$. Apart from this difference, the two algorithms are identical, and Lemmas analogous to Lemmas 2.18 and 2.19 can be proven, and combining them we get Theorem 2.4.

Algorithm 5 Partially distribution dependent algorithm for general instances

Input: Budgets B_i for $i \in [n]$, Consumption C_i for $i \in [n]$, maximum possible bids $b_i = \max_j b_{ij}$ for $i \in [n]$ and the total number of queries m

Output: An online assignment of queries to advertisers

1: Initialize $R_i^0 = B_i$ for all i
2: **for** $t = 1$ to m **do**
3: Let j be the query that arrives at time t
4: For each advertiser i, compute using Equation (2.10)

$$\Delta_i^t = \min(b_{ij}, R_i^{t-1}) + \mathcal{R}\left(\frac{C_i}{b_i m}, b_i, R_i^{t-1} - \min(b_{ij}, R_i^{t-1}), m - t\right) - \mathcal{R}\left(\frac{C_i}{b_i m}, b_i, R_i^{t-1}, m - t\right)$$

5: Assign the query to the advertiser $i^* = \operatorname{argmax}_{i \in [n]} \Delta_i^t$
6: Set $R_i^t = R_i^{t-1}$ for $i \neq i^*$ and set $R_{i^*}^t = R_{i^*}^{t-1} - \min(b_{i^*j}, R_{i^*}^{t-1})$
7: **end for**

2.5.5 Approximate Estimations

Our Algorithm 5 required the budget consumptions C_i's to be given. What if these quantities could be estimated only approximately? That is, suppose we have an algorithm that can estimate C_i' such that $\sum_i C_i' \geq \frac{1}{\alpha} C_i$, then our Algorithm 5 when run with these estimates C_i', will get a revenue of $\sum_i C_i'(1 - \sqrt{\frac{\gamma_i}{2\pi}})$.

2.6 Proof of Asymptotic Optimality of Online Algorithm for Adwords

We now show a simple example that shows that even when the distributions are known, no algorithm can give a $1 - o(\sqrt{\gamma})$ approximation, and hence proving Theorem 2.5 which we restate here for convenience.

Theorem 2.5 *There exist instances with $\gamma = \epsilon^2$ for which no algorithm, even with the complete knowledge of the distribution, can get a $1 - o(\epsilon)$ approximation factor.*

Consider two advertisers 1 and 2. Advertiser 1 has a budget of $2B$ and 2 has a budget of B. There are four types of queries: 0-query, 1-query, 2-query and $1 - 2$

query.

1. The 0-query is worth nothing to both advertisers
2. The 1-query is worth 1 to advertiser 1 and zero to advertiser 2
3. The 2-query is worth 2 to advertiser 2 and zero to advertiser 1
4. The 12-query is worth 1 to advertiser 1 and 2 to advertiser 2

There are totally m queries that arrive online. The 2-query occurs with probability $\frac{B}{2m}$, the 1-query with probability $\frac{B-\sqrt{B}}{m}$, the 12-query with probability $\frac{\sqrt{B}}{m}$ and the 0-query with remaining probability. Notice that the γ for this instance is $\frac{1}{B}$. Thus it is enough to show that a loss of $\Theta(\sqrt{B})$ cannot be avoided.

First the distribution instance has $\frac{B}{2}$ 2-queries, $B-\sqrt{B}$ 1-queries, $\sqrt{B}$ 12-queries and remaining zero queries. This means that the distribution instance has a revenue of $2B$, which is our benchmark.

Now consider the 12 queries. By Chernoff bounds, with a constant probability at least a $\Theta(\sqrt{B})$ of such queries occur in an instance. In such instances, at least a constant fraction of these 12-queries, i.e., $\Theta(\sqrt{B})$ 12-queries, occur at such a point in the algorithm where,

- with a constant probability the remaining 2-queries could completely exahust advertiser 2's budget,
- and with a constant probability the remaining 2-queries could fall short of the budget of advertiser 2 by $\Theta(\sqrt{B})$

Note that by Chernoff bounds these events occur with a constant probability. This is the situation that confuses the algorithm. Whom to assign such a 12-query to? Giving it to advertiser 2 will fetch one unit of revenue more, but then with a constant probability situation 1 occurs in which case it is correct in hindsight to have assigned this query to advertiser 1, thus creating a loss of 1. On the other hand if the algorithm assigns this 12-query to advertiser 1, with a constant probability situation 2 occurs, thus making it necessary for each of these $\Theta(\sqrt{B})$ queries to have been given to advertiser 2. Thus for $\Theta(\sqrt{B})$ queries, there is a constant probability that the algorithm will lose one unit of revenue irrespective of what it decides. This costs the algorithm a revenue loss of $\Theta(\sqrt{B})$, thus proving asymptotic tightness, and hence Theorem 2.5

2.7 Greedy Algorithm for Adwords

In this section, we give a simple proof of Theorem 2.8, which we restate below for conveniene.

Theorem 2.8 *The greedy algorithm achieves an approximation factor of* $1 - 1/e$ *for the Adwords problem in the i.i.d. unknown distributions model for all* γ*, i.e.,* $0 \leq \gamma \leq 1$.

As noted in Section 2.2.2 where the Adwords problem was introduced, the budget constraints are not hard, i.e., when a query j arrives, with a bid amount $b_{ij} >$ remaining budget of i, we are still allowed to allot that query to advertiser i, but we only earn a revenue of the remaining budget of i, and not the total value b_{ij}.

Goel and Mehta [GM08] prove that the greedy algorithm gives a $(1 - 1/e)$ approximation to the adwords problem when the queries arrive in a random permutation or in i.i.d., but under an assumption which almost gets down to γ tending to zero, i.e., bids being much smaller than budgets. We give a much simpler proof for a $(1 - 1/e)$ approximation by greedy algorithm for the i.i.d. unknown distributions case, and our proof works for all γ.

Let p_j be the probability of query j appearing in any given impression. Let $y_j = mp_j$. Let x_{ij} denote the offline fractional optimal solution for the expected instance. Let $w_i(t)$ denote the amount of money spent by advertiser i at time step t, i.e., for the t-th query in the greedy algorithm (to be described below). Let $f_i(0) = \sum_j b_{ij}x_{ij}y_j$. Let $f_i(t) = f_i(0) - \sum_{r=1}^{t} w_i(r)$. Let $f(t) = \sum_{i=1}^{n} f_i(t)$. Note that $f_i(0)$ is the amount spent by i in the offline fractional optimal solution to the expected instance.

Consider the greedy algorithm which allocates the query j arriving at time t to the advertiser who has the maximum effective bid for that query, i.e., $\operatorname*{argmax}_i \min\{b_{ij}, B_i - \sum_{r=1}^{t-1} w_i(r)\}$. We prove that this algorithm obtains a revenue of $(1 - 1/e)\sum_{i,j} b_{ij}x_{ij}y_j$ and thus gives the desired $1 - 1/e$ competitive ratio against the fractional optimal solution to the expected instance. Consider a hypothetical algorithm that allocates queries to advertisers according to the x_{ij}'s. We prove that this hypothetical algorithm obtains an expected revenue of $(1 - 1/e)\sum_{i,j} b_{ij}x_{ij}y_j$, and argue that the greedy algorithm only performs better. Let $w_i^h(t)$ and $f_i^h(t)$ denote the quantities analogous to $w_i(t)$ and $f_i(t)$ for the hypothetical algorithm, with the initial value $f_i^h(0) = f_i(0) =$

$\sum_j b_{ij}x_{ij}y_j$. Let $f^h(t) = \sum_{i=1}^n f_i^h(t)$. Let $\mathrm{EXCEED}_i(t)$ denote the set of all j such that b_{ij} is strictly greater than the remaining budget at the beginning of time step t, namely $b_{ij} > B_i - \sum_{r=1}^{t-1} w_i^h(r)$.

Lemma 2.21 $\mathbf{E}[w_i^h(t)|f_i^h(t-1)] \geq \frac{f_i^h(t-1)}{m}$

Proof: The expected amount amount of money spent at time step t, is given by

$$\mathbf{E}[w_i^h(t)|f_i^h(t-1)] = \sum_{j\in \mathrm{EXCEED}_i(t)} \left(B_i - \sum_{r=1}^{t-1} w_i^h(r)\right)\frac{x_{ij}y_j}{m} + \sum_{j\notin \mathrm{EXCEED}_i(t)} b_{ij}\frac{x_{ij}y_j}{m} \tag{2.13}$$

If $\sum_{j\in \mathrm{EXCEED}_i(t)} x_{ij}y_j \geq 1$, then by (2.13),

$$\mathbf{E}[w_i^h(t)|f_i^h(t-1)] \geq \frac{B_i - \sum_{r=1}^{t-1} w_i^h(r)}{m} \geq \frac{f_i^h(0) - \sum_{r=1}^{t-1} w_i^h(r)}{m} = \frac{f_i^h(t-1)}{m}.$$

Suppose on the other hand $\sum_{j\in \mathrm{EXCEED}_i(t)} x_{ij}y_j < 1$. We can write $\mathbf{E}[w_i^h(t)|f_i^h(t-1)]$ as

$$\mathbf{E}[w_i^h(t)|f_i^h(t-1)] = \frac{f_i^h(0)}{m} - \sum_{j\in \mathrm{EXCEED}_i(t)} \left(b_{ij} - (B_i - \sum_{r=1}^{t-1} w_i^h(r))\right)\frac{x_{ij}y_j}{m}. \tag{2.14}$$

Since $b_{ij} \leq B_i$, and $\sum_{j\in \mathrm{EXCEED}_i(t)} x_{ij}y_j < 1$, (2.14) can be simplified to

$$\begin{aligned}\mathbf{E}[w_i^h(t)|f_i^h(t-1)] &> \frac{f_i^h(0)}{m} - \frac{\sum_{r=1}^{t-1} w_i^h(r)}{m} \\ &= \frac{f_i^h(t-1)}{m}.\end{aligned}$$

■

Lemma 2.22 *The hypothetical algorithm satisfies the following:* $\mathbf{E}[f^h(t)|f^h(t-1)] \leq f^h(t-1)(1-1/m)$

Proof: From the definition of $f_i^h(t)$, we have

$$\begin{aligned}f_i^h(t) &= f_i^h(t-1) - w_i^h(t) \\ \mathbf{E}[f_i^h(t)|f_i^h(t-1)] &= f_i^h(t-1) - \mathbf{E}[w_i^h(t)|f_i^h(t-1)] \leq f_i^h(t-1)(1-\frac{1}{m}),\end{aligned}$$

where the inequality is due to Lemma 2.21. Summing over all i gives the Lemma. ∎

Lemma 2.23 $\mathbf{E}[\text{GREEDY}] \geq (1-1/e)\sum_{i,j} b_{ij}x_{ij}y_j$

Proof: Lemma 2.22 proves that for the hypothetical algorithm, the value of the difference $f^h(t-1) - \mathbf{E}[f^h(t)|f^h(t-1)]$, which is the expected amount spent at time t by all the advertisers together, conditioned on $f^h(t-1)$, is at least $\frac{f^h(t-1)}{m}$. But by definition, conditioned on the amount of money spent in first $t-1$ steps, the greedy algorithm earns the maximum revenue at time step t . Thus, for the greedy algorithm too, the statement of the lemma 2.22 must hold, namely, $\mathbf{E}[f(t)|f(t-1)] \leq f(t-1)(1-1/m)$. This means that $\mathbf{E}[f(m)] \leq f(0)(1-1/m)^m \leq f(0)(1/e)$. Thus the expected revenue earned is

$$\begin{aligned}
\mathbf{E}[\sum_{r=1}^{m} w(r)] &= f(0) - \mathbf{E}[f(m)] \\
&\geq f(0)\,(1-1/e) \\
&= (1-1/e)\sum_{i,j} b_{ij}x_{ij}y_j
\end{aligned}$$

and this proves the lemma. ∎

Lemma 2.23 proves Theorem 2.8.

2.8 Fast Approximation Algorithm for Large Mixed Packing & Covering Integer Programs

In this section, we consider the mixed packing-covering problem stated in Section 2.2.5. and prove Theorem 2.9. We restate the LP for the mixed covering-packing problem here.

$$
\begin{aligned}
&\forall\ i, \sum_{j,k} a_{ijk} x_{j,k} \leq c_i \\
&\forall\ i, \sum_{j,k} w_{ijk} x_{j,k} \geq d_i \\
&\forall\ j, \sum_{k} x_{j,k} \leq 1 \\
&\forall\ j, k, x_{j,k} \geq 0.
\end{aligned} \tag{2.15}
$$

The goal is to check if there is a feasible solution to this LP. We solve a gap version of this problem. Distinguish between the two cases with a high probability, say $1 - \delta$:

YES: There is a feasible solution.

NO: There is no feasible solution even if all of the c_i's are multiplied by $1 + 2\epsilon$ and all of the d_i's are multiplied by $1 - 2\epsilon$.

We use $1 + 2\epsilon$ and $1 - 2\epsilon$ instead of just $1 + \epsilon$ and $1 - \epsilon$ purely to reduce notational clutter in what follows.

Like in the online problem, we refer to the quantities indexed by j as requests, a_{ijk} as resource i consumption, and w_{ijk} as resource i profit, and the quantities indexed by k as options. There a total of m requests, n resources, and K options, and the "zero" option denoted by $\perp$. Recall that the parameter γ for this problem is defined by $\gamma = \max\left(\left\{\frac{a_{ijk}}{c_i}\right\}_{i,j,k} \cup \left\{\frac{w_{ijk}}{d_i}\right\}_{i,j,k}\right)$. Our algorithm needs the value of m, n and γ (an upper bound on the value of γ also suffices).

High-level overview. We solve this offline problem in an online manner via random sampling. We sample $T = \Theta(\frac{\gamma m \log(n/\delta)}{\epsilon^2})$ requests j from the set of possible requests uniformly at random with replacement, and then design an algorithm that allocates resources online for these requests. At the end of serving T requests we check if the solution given proportionally satisfies the constraints of LP 2.15. If yes, we declare YES as the answer and declare NO otherwise. At the core of the soluton is the online sampling algorithm we use, which is identical to the techniques used to develop the online algorithm in Sections 2.3.2 and 2.3.3. We describe our algorithm in Algorithm 6

Algorithm 6 : Online sampling algorithm for offline mixed covering-packing problems

Input: The mixed packing and covering LP 2.15, faliure probability $\delta > 0$, and an error parameter $\epsilon > 0$.

Output: Distinguish between the cases 'YES' where there is a feasible solution to LP 2.15, and 'NO' where there is no feasible solution to LP 2.15 even if all the c_i's are multiplied by $1 + 2\epsilon$ and all of the d_i's are multiplied by $1 - 2\epsilon$.

Set $T = \Theta(\frac{\gamma m \log(n/\delta)}{\epsilon^2})$

Initialize $\phi_{i,0} = \frac{1}{c_i}\left[\frac{\left(1+\frac{\epsilon}{m\gamma}\right)^{T-1}}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}}\right]$, and, $\psi_{i,0} = \frac{1}{d_i}\left[\frac{\left(1-\frac{\epsilon}{m\gamma}\right)^{T-1}}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}}\right]$

for $s = 1$ to T **do**

  Sample a request j uniformly at random from the total pool of m requests

  If the incoming request is j, use the following option k^*:

$$k^* = \arg\min_{k \in K \cup \{\perp\}} \left\{ \sum_i a_{ijk} \cdot \phi_{i,s-1} - \sum_i w_{ijk} \cdot \psi_{i,s-1} \right\}.$$

  $X^A_{i,s} = a_{ijk^*}$, $Y^A_{i,s} = w_{ijk^*}$

  Update $\phi_{i,s} = \phi_{i,s-1} \cdot \left[\frac{(1+\epsilon)^{\frac{X^A_{i,s}}{\gamma c_i}}}{1+\frac{\epsilon}{m\gamma}}\right]$, and, $\psi_{i,s} = \psi_{i,s-1} \cdot \left[\frac{(1-\epsilon)^{\frac{Y^A_{i,s}}{\gamma d_i}}}{1-\frac{\epsilon}{m\gamma}}\right]$

end for

if $\forall i \sum_{t=1}^{T} X^A_{i,t} < \frac{Tc_i}{m}(1+\epsilon)$, and, $\sum_{t=1}^{T} Y^A_{i,t} > \frac{Td_i}{m}(1-\epsilon)$ **then**

  Declare YES

else

  Declare NO

end if

The main theorem of this section is Theorem 2.9, which we restate here:

Theorem 2.9 *For any $\epsilon > 0$, Algorithm 6 solves the gap version of the mixed covering-packing problem with $\Theta(\frac{\gamma m \log(n/\delta)}{\epsilon^2})$ oracle calls.*

Detailed Description and Proof. The proof is in two parts. The first part proves that our algorithm indeed answers YES when the actual answer is YES with a probability at least $1 - \delta$. The second part is the identical statement for the NO case.

The YES case. We begin with the case where the true answer is YES. Let x^*_{jk} denote some feasibel solution to LP 2.15. In a similar spirit to Sections 2.3.1, 2.3.2 and 2.3.3, we define the algorithm P as follows. It samples a total of $T = \Theta(\frac{\gamma m \log(n/\delta)}{\epsilon^2})$

requests uniformly at random, with replacement, from the total pool of m requests. When request j is sampled, P serves j using option k with probability x^*_{jk}. Thus, if we denote by $X^*_{i,t}$ the consumption of resource i in step t of P, then we have $\mathbf{E}[X^*_{i,t}] = \sum_{j=1}^{m} \frac{1}{m} \sum_k a_{ijk} x_{j,k}* \leq \frac{c_i}{m}$. This inequality follows from x^*_{jk} being a feasible solution to LP 2.15. Similarly let $Y^*_{i,t}$ denote the resource i profit in step t of P. We have $\mathbf{E}[Y^*_{i,t}] \geq \frac{d_i}{m}$. We now write the probability that our condition for YES is violated for some algorithm A.

$$
\begin{aligned}
\mathbf{Pr}\left[\sum_{t=1}^{m} X^A_{i,t} \geq \frac{Tc_i}{m}(1+\epsilon)\right] &= \mathbf{Pr}\left[\frac{\sum_{t=1}^{T} X^A_{i,t}}{\gamma c_i} \geq \frac{T}{m\gamma}(1+\epsilon)\right] \\
&= \mathbf{Pr}\left[(1+\epsilon)^{\frac{\sum_{t=1}^{T} X^A_{i,t}}{\gamma c_i}} \geq (1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}\right] \\
&\leq \mathbf{E}\left[(1+\epsilon)^{\frac{\sum_{t=1}^{T} X^A_{i,t}}{\gamma c_i}}\right] / (1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}} \\
&= \frac{\mathbf{E}\left[\prod_{t=1}^{T}(1+\epsilon)^{\frac{X^A_{i,t}}{\gamma c_i}}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}}
\end{aligned} \tag{2.16}
$$

$$
\begin{aligned}
\mathbf{Pr}\left[\sum_{t=1}^{T} Y^A_{i,t} \leq \frac{Td_i}{m}(1-\epsilon)\right] &= \mathbf{Pr}\left[\frac{\sum_{t=1}^{T} Y^A_{i,t}}{\gamma d_i} \geq \frac{T}{m\gamma}(1-\epsilon)\right] \\
&= \mathbf{Pr}\left[(1-\epsilon)^{\frac{\sum_{t=1}^{T} Y^A_{i,t}}{\gamma d_i}} \geq (1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}\right] \\
&\leq \mathbf{E}\left[(1-\epsilon)^{\frac{\sum_{t=1}^{T} Y^A_{i,t}}{\gamma d_i}}\right] / (1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}} \\
&= \frac{\mathbf{E}\left[\prod_{t=1}^{T}(1-\epsilon)^{\frac{Y^A_{i,t}}{\gamma d_i}}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}}
\end{aligned} \tag{2.17}
$$

If our algorithm A was P $\left(\text{and therefore we can use } \mathbf{E}[X^*_{i,t}] \leq \frac{c_i}{m} \text{ and } \mathbf{E}[Y^*_{i,t}] \geq \frac{d_i}{m}\right)$, the total failure probability in the YES case, which is the sum of (2.16) and (2.17) for all the i's would have been at most δ, if $T = \Theta(\frac{\gamma m \log(n/\delta)}{\epsilon^2})$ for an appropriate constant inside Θ. The goal is to design an algorithm A that, unlike P, does not first solve LP 2.15 and then use x^*_{jk}'s to allocate resources, but allocates online and also obtains the same δ failure probability, just as we did in Sections 2.3.2 and 2.3.3. That is we want to show that the sum of (2.16) and (2.17) over all i's is at most δ:

$$\frac{\mathbf{E}\left[\prod_{t=1}^{T}(1+\epsilon)^{\frac{X^A_{i,t}}{\gamma c_i}}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}}+\frac{\mathbf{E}\left[\prod_{t=1}^{T}(1-\epsilon)^{\frac{Y^A_{i,t}}{\gamma d_i}}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}}\leq\delta.$$

For the algorithm A^sP^{T-s}, the above quantity can be rewritten as

$$\sum_i\frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_s\left(X^A_i\right)}{\gamma c_i}}\prod_{t=s+1}^{T}(1+\epsilon)^{\frac{X^*_{i,t}}{\gamma c_i}}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}}+\sum_i\frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_s\left(Y^A_i\right)}{\gamma d_i}}\prod_{t=s+1}^{T}(1-\epsilon)^{\frac{Y^*_{i,t}}{\gamma d_i}}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}},$$

which, by using $(1+\epsilon)^x\leq 1+\epsilon x$ for $0\leq x\leq 1$, is in turn upper bounded by

$$\sum_i\frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_s\left(X^A_i\right)}{\gamma c_i}}\prod_{t=s+1}^{T}\left(1+\epsilon\frac{X^*_{i,t}}{\gamma c_i}\right)\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}}+\sum_i\frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_s\left(Y^A_i\right)}{\gamma d_i}}\prod_{t=s+1}^{T}\left(1-\epsilon\frac{Y^*_{i,t}}{\gamma d_i}\right)\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}}.$$

Since for all t, the random variables $X^*_{i,t}$, $X^A_{i,t}$, $Y^*_{i,t}$ and $Y^A_{i,t}$ are all independent, and $\mathbf{E}[X^*_{i,t}]\leq\frac{c_i}{m}$, $\mathbf{E}[Y^*_{i,t}]\geq\frac{d_i}{m}$, the above is in turn upper bounded by

$$\sum_i\frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_s\left(X^A_i\right)}{\gamma c_i}}\left(1+\frac{\epsilon}{m\gamma}\right)^{T-s}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}}+\sum_i\frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_s\left(Y^A_i\right)}{\gamma d_i}}\left(1-\frac{\epsilon}{m\gamma}\right)^{T-s}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}}.\quad(2.18)$$

Let $\mathcal{F}[A^sP^{T-s}]$ denote the quantity in (2.18), which is an upper bound on failure probability of the hybrid algorithm A^sP^{T-s}. We just said that $\mathcal{F}[P^T]\leq\delta$. We now prove that for all $s\in\{0,1,\ldots,T-1\}$, $\mathcal{F}[A^{s+1}P^{T-s-1}]\leq\mathcal{F}[A^sP^{T-s}]$, thus proving that $\mathcal{F}[A^T]\leq\delta$, i.e., running the algorithm A for all the T steps of stage r results in a failure with probability at most δ.

Assuming that for all $s<p$, the algorithm A has been defined for the first $s+1$ steps in such a way that $\mathcal{F}[A^{s+1}P^{T-s-1}]\leq\mathcal{F}[A^sP^{T-s}]$, we now define A for the $p+1$-th

step of stage r in a way that will ensure that $\mathcal{F}[A^{p+1}P^{T-p-1}] \leq \mathcal{F}[A^p P^{T-p}]$. We have

$$\begin{aligned}
\mathcal{F}[A^{p+1}P^{m-p-1}] &= \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_{p+1}\left(X_i^A\right)}{\gamma c_i}} \left(1+\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}} \qquad + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_{p+1}\left(Y_i^A\right)}{\gamma d_i}} \left(1-\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}} \\
&\leq \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_p\left(X_i^A\right)}{\gamma c_i}} \left(1+\epsilon\frac{X_{i,p+1}^A}{\gamma c_i}\right)\left(1+\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}} \qquad + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_p\left(Y_i^A\right)}{\gamma d_i}} \left(1-\epsilon\frac{Y_{i,p+1}^A}{\gamma d_i}\right)\left(1-\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}}
\end{aligned} \tag{2.19}$$

Define

$$\phi_{i,s} = \frac{1}{c_i}\left[\frac{(1+\epsilon)^{\frac{S_s\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon}{m\gamma}\right)^{T-s-1}}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}}\right]; \qquad \psi_{i,s} = \frac{1}{d_i}\left[\frac{(1-\epsilon)^{\frac{S_s\left(Y_i^A\right)}{\gamma d_i}}\left(1-\frac{\epsilon}{m\gamma}\right)^{T-s-1}}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}}\right]$$

Define the step $p+1$ of algorithm A as picking the following option k for request j:

$$k^* = \arg\min_{k \in K \cup \{\perp\}} \left\{\sum_i a_{ijk} \cdot \phi_{i,p} - \sum_i w_{ijk} \cdot \psi_{i,p}\right\}.$$

By the above definition of step $p+1$ of algorithm A, it follows that for any two algorithms with the first p steps being identical, and the last $T-p-1$ steps following the Hypothetical-Oblivious algorithm P, algorithm A's $p+1$-th step is the one that minimizes expression (2.19). In particular it follows that expression (2.19) is upper bounded by the same expression where the $p+1$-the step is according to $X^*_{i,p+1}$ and

$Y^*_{i,p+1}$, i.e., we replace $X^A_{i,p+1}$ by $X^*_{i,p+1}$ and $Y^A_{i,p+1}$ by $Y^*_{i,p+1}$. Therefore we have

$$
\begin{aligned}
\mathcal{F}[A^{p+1}P^{T-p-1}] &\leq \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_p\left(X_i^A\right)}{\gamma c_i}}\left(1+\epsilon\frac{X^*_{i,p+1}}{\gamma c_i}\right)\left(1+\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}} \quad + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_p\left(Y_i^A\right)}{\gamma d_i}}\left(1-\epsilon\frac{Y^*_{i,T+p+1}}{\gamma d_i}\right)\left(1-\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}} \\
&\leq \sum_i \frac{\mathbf{E}\left[(1+\epsilon)^{\frac{S_p\left(X_i^A\right)}{\gamma c_i}}\left(1+\frac{\epsilon}{m\gamma}\right)\left(1+\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1+\epsilon)^{(1+\epsilon)\frac{T}{m\gamma}}} \quad + \\
&\qquad \sum_i \frac{\mathbf{E}\left[(1-\epsilon)^{\frac{S_p\left(Y_i^A\right)}{\gamma d_i}}\left(1-\frac{\epsilon}{m\gamma}\right)\left(1-\frac{\epsilon}{m\gamma}\right)^{T-p-1}\right]}{(1-\epsilon)^{(1-\epsilon)\frac{T}{m\gamma}}} \\
&= \mathcal{F}[A^pP^{T-p}]
\end{aligned}
$$

The NO case. We now proceed to prove that when the real answer is NO, our algorithm says NO with a probability at least $1-\delta$, i.e.,

Lemma 2.24 *For a* NO *instance, if* $T \geq \Theta(\frac{\gamma m \log(n/\delta)}{\epsilon^2})$, *then*

$$\mathbf{Pr}\left[\max_i \frac{S_T\left(X_i^A\right)}{c_i} < \frac{T}{m}(1+\epsilon)\ \&\ \min_i \frac{S_T\left(Y_i^A\right)}{d_i} > \frac{T}{m}(1-\epsilon)\right] \leq \delta.$$

Proof: Let R denote the set of requests sampled. Consider the following LP.

If the primal in LP 2.6 has an optimal objective value at least $\frac{T\epsilon}{m}$, then our algorithm would have declared NO. We now show that by picking $T = \Theta(\frac{\gamma m \ln(n/\delta)}{\epsilon^2})$, the above LP will have its optimal objective value at least $\frac{T\epsilon}{m}$, with a probability at least $1-\delta$. This makes our algorithm answer NO with a probability at least $1-\delta$.

Now, the primal of LP (2.6) has an optimal value equal to that of the dual which in turn is lower bounded by the value of dual at any feasible solution. One such feasible solution is $\alpha^*, \beta^*, \rho^*$, which is the optimal solution to the full version of the dual in LP (2.6), namely the one where $R = [m]$, $T = m$. This is because the set of

Sampled primal LP	**Sampled dual LP**
Minimize λ s.t.	Maximize $\frac{T}{m}\sum_i(\rho_i - \alpha_i) - \sum_{j\in R}\beta_j$ s.t.
$\forall\ i, \lambda - \sum_{j\in R,k}\frac{a_{ijk}x_{j,k}}{c_i} \geq -\frac{T}{m}$	$\forall\ j \in R, k,\ \beta_j \geq \sum_i\left(\rho_i\frac{w_{ijk}}{d_i} - \alpha_i\frac{a_{ijk}}{c_i}\right)$
$\forall\ i, \lambda + \sum_{j\in R,k}\frac{w_{ijk}x_{j,k}}{d_i} \geq \frac{T}{m}$	$\sum_i(\alpha_i + \rho_i) \leq 1$
$\forall\ j \in R, \sum_k x_{j,k} \leq 1$	$\forall\ i, \alpha_i, \rho_i \geq 0$
$\forall\ j, k, x_{j,k} \geq 0$	$\forall\ j \in R, \beta_j \geq 0$
$\lambda \geq 0$	

LP 2.6: Sampled primal and dual LPs

constraints in the full version of the dual is clearly a superset of the constraints in the dual of LP (2.6). Thus, the optimal value of the primal of LP (2.6) is lower bounded by value of dual at $\alpha^*, \beta^*, \rho^*$, which is

$$= \frac{T}{m}(\sum_i \rho_i^* - \alpha_i^*) - \sum_{j\in R}\beta_j^* \tag{2.20}$$

For proceeding further in lower bounding (2.20), we apply Chernoff bounds to $\sum_{j\in R}\beta_j^*$. The fact that the full version of LP 2.6 is a maximization LP, coupled with the constraints there in imply that $\beta_j^* \leq \gamma$. Further, let τ^* denote the optimal value of the full version of LP (2.6), i.e., $\sum_i(\rho_i^* - \alpha_i^*) - \sum_j \beta_j^* = \tau^*$. Now, the constraint $\sum_i(\alpha_i^* + \rho_i^*) \leq 1$ coupled with the fact that $\tau^* \geq 0$ implies $\sum_j \beta_j^* \leq 1$. We are now ready to lower bound the quantity in (2.20). We have the optimal solution to primal of LP (2.6)

$$
\begin{aligned}
&\geq \frac{T}{m}(\sum_i \rho_i^* - \alpha_i^*) - \sum_{j \in R} \beta_j^* \\
&\geq \frac{T}{m}\sum_i (\rho_i^* - \alpha_i^*) - \left(\frac{T \sum_j \beta_j^*}{m} + \sqrt{\frac{4T(\sum_j \beta_j^*)\gamma \ln(1/\delta)}{m}} \right) \quad \left(\text{Since } \beta_j^* \in [0,\gamma]\right) \\
&\geq \frac{T\tau^*}{m} - \sqrt{\frac{4T\gamma\ln(1/\delta)}{m}} \\
&= \frac{T\tau^*}{m}\left[1 - \sqrt{\frac{\gamma m \ln(1/\delta)}{T} \cdot \frac{4}{\tau^{*2}}}\right] \qquad (2.21)
\end{aligned}
$$

where the second inequality is a "with probability at least $1-\delta$" inequality, i.e., we apply Chernoff bounds for $\sum_{j \in S} \beta_j^*$, along with the observation that each $\beta_j^* \in [0,\gamma]$. The third inequality follows from $\sum_j \beta_j^* \leq 1$ and $\sum_i (\rho_i^* - \alpha_i^*) - \sum_j \beta_j^* = \tau^*$. Setting $T = \Theta(\frac{\gamma m \ln(n/\delta)}{\epsilon^2})$ with a appropriate constant inside the Θ, coupled with the fact that $\tau^* \geq 2\epsilon$ in the NO case, it is easy to verify that the quantity in (2.21) is at least $\frac{T\epsilon}{m}$.

Going back to our application of Chernoff bounds above, in order to apply it in the form above, we require that the multiplicative deviation from mean $\sqrt{\frac{4\gamma m \ln(1/\delta)}{T \sum_j \beta_j^*}} \in [0, 2e-1]$. If $\sum_j \beta_j^* \geq \epsilon^2$, then this requirement would follow. Suppose on the other hand that $\sum_j \beta_j^* < \epsilon^2$. Since we are happy if the excess over mean is at most $\frac{T\epsilon}{m}$, let us look for a multiplicative error of $\frac{\frac{T\epsilon}{m}}{\frac{T\sum_j \beta_j^*}{m}}$. Based on the fact that $\sum_j \beta_j^* < \epsilon^2$ the multiplicative error can be seen to be at least $1/\epsilon$ which is larger than $2e-1$ when $\epsilon < \frac{1}{2e-1}$. We now use the version of Chernoff bounds for multiplicative error larger than $2e-1$, which gives us that a deviation of $\frac{T\epsilon}{m}$ occurs with a probability at most $2^{-\left(1+\frac{\frac{T\epsilon}{m}}{\frac{T\sum_j \beta_j^*}{m}}\right)\frac{T\sum_j \beta_j^*}{m\gamma}}$, where the division by γ is because of the fact that $\beta_j^* \leq \gamma$. This probability is at most $(\frac{\delta}{n})^{1/\epsilon}$, which is at most δ.

■

The proofs for the YES and NO cases together prove Theorem 2.9

2.9 Special Cases of the Resource Allocation Framework

We now list the problems that are special cases of the resource allocation framework and have been previously considered.

2.9.1 Network Routing and Load Balancing

Consider a graph (either undirected or directed) with edge capacities. Requests arrive online; a request j consists of a source-sink pair, (s_j, t_j) and a bandwidth ρ_j. In order to satisfy a request, a capacity of ρ_j must be allocated to it on every edge along some path from s_j to t_j in the graph. In the *throughput maximization* version, the objective is to maximize the number of satisfied requests while not allocating more bandwidth than the available capacity for each edge (Different requests could have different values on them, and one could also consider maximizing the total value of the satisfied requests). Our algorithm 2 for resource allocation framework directly applies here and the approximation guarantee there directly carries over. Kamath, Palmon and Plotkin [KPP96] considered a variant of this problem with the requests arriving according to a stationary Poisson process, and show a competitive ratio that is very similar to ours.

2.9.2 Combinatorial Auctions

Suppose we have n items for sale, with c_i copies of item i. Bidders arrive online, and bidder j has a utility function $U_j : 2^{[n]} \to \mathbf{R}$. If we posted prices p_i for each item i, then bidder j buys a bundle S that maximizes $U_j(S) - \sum_{i \in S} p_i$. We assume that bidders can compute such a bundle. The goal is to maximize social welfare, the total utility of all the bidders, subject to the supply constraint that there are only c_i copies of item i. Firstly, incentive constraints aside, this problem can be written as an LP in the resource allocation framework. The items are the resources and agents arriving online are the requests. All the different subsets of items form the set of options. The utility $U_j(S)$ represents the profit $w_{j,S}$ of serving agent j through option S, i.e. subset S. If an item $i \in S$, then $a_{i,j,S} = 1$ for all j and zero otherwise. Incentive constraints aside, our algorithm for resource allocation at step s, will choose the option k^* (or equivalently the bundle S) as specified in point 8 of Algorithm 2, i.e., minimize the potential function. That is, if step s falls in stage r,

$$k^* = \arg\min_k \left\{ \sum_i a_{ijk} \cdot \phi^r_{i,s-1} - w_{j,k} \cdot \psi^r_{s-1} \right\}$$

(note that unlike Algorithm 2 there is no subscripting for $w_{j,k}$). This can be equivalently written as

$$k^* = \arg\max_k \left\{ w_{j,k} \cdot \psi^r_{s-1} - \sum_i a_{ijk} \cdot \phi^r_{i,s-1} \right\}$$

Now, maximizing the above expression at step s is the same as picking the k to maximize $w_{j,k} - \sum_i p_i(s) a_{ijk}$, where $p_i(s) = \frac{\phi^r_{i,s-1}}{\psi^r_{s-1}}$. Thus, if we post a price of $p_i(s)$ on item i for bidder number s, he will do exactly what the algorithm would have done otherwise. Suppose that the bidders are i.i.d samples from some distribution (or they arrive as in the adversarial stochastic input model). We can use Theorem 2.2 to get an incentive compatible posted price auction[3] with a competitive ratio of $1 - O(\epsilon)$ whenever $\gamma = \min_i \{c_i\} \geq O\left(\frac{\log(n/\epsilon)}{\epsilon^2}\right)$. Further if an analog of Theorem 2.2 also holds in the random permutation model then we get a similar result for combinatorial auctions in the offline case: we simply consider the bidders one by one in a random order.

2.9.3 Adwords and Display Ads Problems

The Adwords and Display ads special cases were already discussed in Section 2.2.2.

2.10 Conclusion

Our work raises the following open questions.

- **I.I.D. vs random permutations.** As mentioned in the introduction, we can show that our algorithm works in the i.i.d model, so the natural question is if our algorithm works for the random permutation model. There is no separation known between these two models so far, although any result for the random permutation model also works for the i.i.d. model.

[3]Here we assume that each agent reveals his true utility function *after* he makes his purchase. This information is necessary to compute the prices to be charged for future agents.

- **Do without periodic estimation?** Currently in our algorithm for online resource allocation (Algorithm 2), we need to estimate the optimum objective function value periodically. For this we need to solve (at least approximately) an offline instance of the problem repeatedly. Is there an algorithm that avoids this? We show in Section 2.3.2 that we can avoid this if we are given a single parameter from the distribution, namely W_E. But can we do without that parameter too?

- **Do without the distribution dependent parameters C_i's for adwords?** Our improved algorithm for Adwords (Algorithm 5) is not completely prior robust in that it requires a few parameters from the distribution, namely, the consumption information in some optimal solution to the expected instance. Can we design asymptotically optimal algorithm for Adwords that are also prior robust?

- **Better algorithms for Adwords for large γ.** We prove that the greedy algorithm for Adwords gets a $1 - 1/e$ for Adwords even with $\gamma = 1$. While this is tight for the greedy algorithm, are there other algorithms that can perform better. There doesn't appear to be any reason to get stuck at $1 - 1/e$.

3 Prior Robust Revenue Maximizing Auction Design

Organization. In this chapter, we design and analyze prior robust mechanisms for maximizing revenue. The chapter is organized as follows. In Section 3.1 we informally summarize our results, put them in context of related work with some additional discussion. In Section 3.2 we provide all the necessary preliminaries. In Section 3.3 we show that for non-i.i.d. irregular distributions, recruiting one extra bidder (targeted advertising) each from every underlying population group and running a second-price auction compares well with the optimal revenue. In Section 3.4 we obtain guarantees for a non-targered advertising campaign. In Section 3.5 we obtain guarantees for the Vickrey auction with a single (anonymous) reserve price. We conclude with an open question in Section 3.7.

3.1 Introduction & Summary of Results

Simplicity and detail-freeness are two much sought-after themes in auction design. The celebrated classic result of Bulow and Klemperer [BK96] says that in a standard single-item auction with n bidders, when the valuations of bidders are drawn i.i.d from a distribution that satisfies a regularity condition, running a Vickrey auction (second-price auction) with one extra bidder drawn from the same distribution yields at least as much revenue as the optimal auction for the original n bidders. The Vickrey auction is both simple and detail-free since it doesn't require any knowledge of bidder distributions. Given this success story for i.i.d. regular distributions, we ask in this paper, what is the analogous result when we go beyond i.i.d regular settings? Our main result is a version of Bulow and Klemperer's result to non-i.i.d irregular settings. Our work gives the first positive results in designing simple mechanisms for irregular distributions, by parameterizing irregular distributions, i.e., quantifying the amount of irregularity in a distribution. Our parameterization is motivated by real world market structures and suggests that most realistic markets will not be highly irregular with respect to this metric. Our results enable the first positive approximation bounds on the revenue of the second-price auction with an anonymous reserve in both i.i.d. and non-i.i.d. irregular settings.

Before explaining our results, we briefly describe our setting. We consider a single-item auction setting with bidders having quasi-linear utilities. That is the utility of a bidder is his value for the item if he wins, less the price he is charged by the auction. We study auctions in the Bayesian setting, i.e. the valuations of bidders are drawn from known distributions[1]. We make the standard assumption that bidder valuations are drawn from independent distributions.

Irregular Distributions are Common. The technical regularity condition in Bulow and Klemperer's result is quite restrictive, and indeed irregular distributions are quite common in markets. For instance, any distribution with more than a single mode violates the regularity condition. The most prevalent reason for a bidder's valuation distribution failing to satisfy the regularity condition is that a bidder in an auction is randomly drawn from a heterogeneous population. The population typically is composed of several groups, and each group has its characteristic preferences. For instance the population might consist of students and seniors, with each group typically having very different preferences from the other. While the distribution of preferences within any one group might be relatively well-aligned and the value distribution might have a single mode and satisfy the regularity condition, the distribution of a bidder drawn from the general population, which is a mixture of such groups, is some convex combination of these individual distributions. Such a convex combination violates regularity even in the simplest cases.

For a variety of reasons, including legal reasons and absence of good data, a seller might be unable to discriminate between the buyers from different population groups and thus has to deal with the market as if each buyer was arriving from an irregular distribution. However, to the least, most sellers do know that their market consists of distinct segments with their characteristic preferences.

Measure of Irregularity. The above description suggests that a concrete measure of irregularity of a distribution is the number of regular distributions required to describe it. We believe that such a measure could be of interest in both designing mechanisms and developing good provable revenue guarantees for irregular distributions in many settings. It is a rigorous of irregularity for any distribution since any

[1]One of the goals of this work is to design detail-free mechanisms, i.e., minimize the dependence on knowledge of distributions. Thus most of our results make little or no use of knowledge of distributions. We state our dependence precisely while stating our results.

distribution can be well-approximated almost everywhere by a sufficient number of regular ones and if we allow the number of regular distributions to grow to infinity then any distribution can be exactly described. Irregular distributions that typically arise in practice are combinations of a small number of regular distributions and this number can be considered almost a constant with respect to the market size. In fact there exist evidence in recent [JM03, GS11] and classical [Rob33] microeconomic literature that irregularity of the value distribution predominantly arises due to market segmentation in a small number of parts (e.g. loyal customers vs. bargain-hunters [JM03], luxury vs. low income buyers [GS11] etc). Only highly pathological distributions require a large number of regular distributions to be described — such a setting in a market implies that the population is heavily segmented and each segment has significantly different preferences from the rest.

Motivated by this, we consider the following setting: the market/population consists of k underlying population groups, and the valuation distribution of each group satisfies the regularity condition. Each bidder is drawn according to some probability distribution over these groups. That is bidder i arrives from group t with probability $p_{i,t}$. Thus if F_t is the cumulative distribution function (cdf) of group t, the cdf of bidder i is $F_i = \sum_t p_{i,t} F_t$. For example, consider a market for a product that consists of two different population groups, say students and seniors. Now suppose that two bidders come from two cities with different student to senior ratios. This would lead to the probability p_i^t's to be different for different i's. This places us in a non-i.i.d. irregular setting. All our results also extend to the case where these probabilities $p_{i,t}$ are arbitrarily correlated.

Benchmark. We use as our benchmark the revenue optimal incentive compatible auction (given by Myerson [Mye81]) that is tailored for the input distribution.

Example 3.1 (An illustrative example) *Consider an eBay auction for iPads. One could think of the market as segmented mainly into two groups of buyers: young and elder audience. These two market segments have quite value distributions. Suppose for instance, that the value distribution of young people is distributed as a normal distribution $N(\mu_y, \sigma)$ while the elders is distributed as a normal distribution $N(\mu_e, \sigma)$ with $\mu_y > \mu_e$. In addition, suppose that the eBay buyer population is composed of a fraction p_y young people and $p_e < p_y$ of elders. Thus the eBay seller is facing an* *irregular valuation distribution that is a mixture of two Gaussian distribution with mixture probabilities p_y and p_e*

The eBay seller has two ways of increasing the revenue that he receives: a) increasing the market competition by bringing extra bidders through advertising (possibly even targeted advertising), and b) setting appropriately his reserve price in the second price auction that he runs. Observe that he has no means of price discriminating, or running discriminatory auctions such as Myerson's optimal auction (which is tailored for a given input distribution) for this irregular distribution. This raises the main questions that we address in this paper: how should he run his advertising campaign? How many people more (either through targeted or non-targeted advertising) should he bring to the auction to get a good approximation to the optimal revenue? What approximation of the optimal revenue is he guaranteed by running a Vickrey auction with a single anonymous reserve?

None of the existing literature in approximately optimal auctions provides any positive results for irregular distributions, and hence, would be unable to give any provable guarantees for these questions, even in this simple example. On the contrary, giving a sneak preview of our main results, our paper gives positive results to all the above questions: 1) bringing just one extra young bidder in the auction (targeted advertising) and running a Vickrey auction with no reserve would yield revenue at least $1/2$ *of the optimal revenue (Theorem 3.5), 2) bringing* 2 *extra bidders drawn from the distribution of the combined population (non-targeted advertising) would yield at least* $\frac{1}{2}\left(1-\frac{1}{e}\right)$ *of the optimal revenue (Theorem 3.8), 3) running a Vickrey auction among the original* n *bidders with a reserve price which is drawn according to the distribution of maximum of two draws from* $(N(\mu_y, \sigma)))$ *will yield* $\frac{1}{2}\left(1-\frac{1}{e}\right)\frac{n}{n+2}$ *of the optimal revenue (Corollary 3.10).*

Our Results

First result (Section 3.3.1): Targeted Advertising for Non-i.i.d. Irregular Settings. We show that by recruiting an extra bidder from each underlying group and running the Vickrey auction, we get a revenue that is at least half of the optimal auction's revenue in the original setting. While the optimal auction is manifestly impractical in a non-i.i.d. irregular setting due to its complicated rules, delicate dependence on knowledge of distribution and its discriminatory nature[2], the Vickrey auction with extra bidders is simple and detail-free: it makes no use of the distributions of bidders. The auctioneer must just be able to identify that his market is composed of different groups and must conduct a targeted advertising campaign to recruit one extra bidder from each group. This result can be interpreted as follows: while advertising was the solution proposed by Bulow and Klemperer [BK96] for i.i.d. regular distributions, *targeted advertising* is the right approach for non-i.i.d. irregular distributions.

Tightness. While we do not know if the the factor 2 approximation we get is tight, Hartline and Roughgarden [HR09] show that even in a non-i.i.d. regular setting with just two bidders it is impossible to get better than a 4/3-approximation by duplicating the bidders, i.e., recruiting n more bidders distributed identically to the original n bidders. This lower bound clearly carries over to our setting also: there are instances where recruiting only one bidder from each different population group cannot give anything better than a 4/3-approximation. Thus there is no hope of getting a Bulow-Klemperer type result where the second price auction with the extra bidders gets at least as much revenue as the optimal auction. An interesting open question, however, would be if we can prove a Bulow-Klemperer type "outperforming" result for i.i.d. irregular distributions, for which the Hartline and Roughgarden [HR09] lower bound does not apply.

Open Question. In an i.i.d. irregular setting where the irregular distribution is composed of a mixture of k regular distributions, does a second price auction with k extra bidders, one from each group, get at least as much revenue as the optimal auction's revenue?

Second Result (Section 3.3.2): Just One Extra Bidder for Hazard Rate Dominant Distributions. If the k underlying distributions are such that one of them stochastically dominates, hazard-rate wise, the rest, then we show that recruiting just one extra bidder from the hazard rate dominant distribution and running the Vickrey auction gets at least half of the optimal revenue for the original setting.

[2] The optimal auction in a non-i.i.d. setting will award the item to the bidder with the highest virtual value and this is not necessarily the bidder with the highest value. In addition, typically a different reserve price will be set to different bidders. This kind of discrimination is often illegal or impractical. Also, the exact form of the irregular distribution will determine which region's of bidder valuations will need to be "ironed", i.e. treated equally.

Further, this hazard rate dominance requirement is not artificial: for instance, if all the k underlying distributions were from the same family of distributions like the uniform, exponential, Gaussian or even power law, then one of them is guaranteed to hazard rate dominate the rest. Though several common distributions satisfy this hazard rate dominance property, it has never been previously exploited in the context of approximately optimal auctions.

Third Result (Section 3.4): Non-targeted Advertising for i.i.d. Irregular Distributions. When the bidders are identically distributed, i.e., the probability $p_{i,t}$ of distribution t getting picked for bidder i is the same for all i (say p_t), we show that if each $p_t \geq \delta$, then bringing $\Theta\left(\frac{log(k)}{\delta}\right)$ extra bidders drawn from the original distribution (and not from one of the k underlying distributions) yields a constant approximation to the optimal revenue. Further in the special case where one of the underlying regular distributions hazard rate dominates the rest and its mixture probability is δ then $\Theta\left(\frac{1}{\delta}\right)$ bidders drawn from the original distribution are enough to yield a constant approximation. This shows that when each of the underlying population groups is sufficiently thick, then recruiting a few extra bidders from the original distribution is all that is necessary.

Remark 3.2 *For the latter result it is not necessary that the decomposition of the irregular distribution into many regular distributions should resemble the actual underlying population groups. Even if the market is highly fragmented with several population groups, as long as there is mathematically some way to decompose the irregular distribution into the convex combination of a few regular distributions our third result still holds.*

Fourth Result (Section 3.5): Vickrey with a Single (Anonymous) Reserve. Suppose we are unable to recruit extra bidders. What is the next simplest non-discriminatory auction we could hope for? The Vickrey auction with a single reserve price. We show that when the non-i.i.d irregular distributions all arise as different convex combinations of the k underlying regular distributions, there exists a reserve such that the Vickrey auction with this single reserve obtains a $4k$ approximation to the optimal revenue. Though the factor of approximation is not small, it is the first non-trivial approximation known for non-i.i.d irregular distributions via Vickrey with anonymous reserve. In addition, as we already explained, in typical market applications we expect the number of different population groups k to be some small constant.

What is the best one can hope for a non-i.i.d irregular setting? Chawla, Hartline and Kleinberg [CHK07] show that for general non-i.i.d irregular distributions it is impossible to get a $o(\log n)$ approximation using Vickrey auction with a single reserve price, and it is unknown if this lower bound is tight, i.e., we do not yet know of a $\Theta(\log n)$ approximation. However the $o(\log n)$ impossibility exists only for arbitrary non-i.i.d irregular settings and doesn't apply when you assume some natural structure on the irregularity of the distributions, which is what we do.

To put our results in context: Single reserve price Vickrey auctions were also analyzed by Hartline and Roughgarden [HR09] for non-i.i.d *regular* settings, that showed that there exists a single reserve price that obtains a 4-approximation. Chawla et al. [CHMS10] show that when bidders are drawn from non-i.i.d irregular distributions, a Vickrey auction with a distribution-specific reserve price obtains a 2-approximation. Thus if there are k different distributions, k different reserve prices are used in this result. This means that if we insist on placing a single (anonymous) reserve price, this result guarantees a $O(1/k)$ approximation. In particular, when all distributions are different, i.e. $k = n$, this boils down to a $O(1/n)$ approximation.

In contrast, our result shows that even when all the distributions are different, as long as every irregular distribution can be described as some convex combination of k regular distributions, Vickrey with a single reserve price gives a factor $4k$ approximation. Further the factor does not grow with the number of players n.

Remark 3.3 *We also show that if the bidders are distributed with identical mixtures and the mixture probability is at least δ then Vickrey auction with a single reserve achieves a $\Theta\left(1 + \frac{\log(k)}{n\delta}\right)$ approximation. If one of the regular distribution hazard rate dominates the rest and has mixture probability δ, then Vickrey with a single reserve achieves a $\Theta\left(1 + \frac{1}{n\delta}\right)$ approximation.*

Observe that if all k regular distributions in the mixture have equal probability of arriving, then our results shows that a Vickrey auction with a single reserve achieves at least a $\Theta\left(1 + \frac{k\log(k)}{n}\right)$ of the optimal revenue. Observe that the approximation ratio becomes better as the number of bidders increases, as long as the number of underlying regular distributions remains fixed. If the number of underlying distributions increases linearly with the number of bidders, then the latter result implies a $\Theta(\log(n))$ approximation, matching the lower bound of [CHMS10].

Related Work. Studying the trade-off between simple and optimal auctions has been a topic of interest for long in auction design. The most famous result is the already discussed result of Bulow and Klemperer [BK96] for single-item auctions in i.i.d regular settings. Hartline and Roughgarden [HR09] generalize [BK96]'s result for settings beyond single-item auctions: they consider auctions where the set of buyers who can be simultaneously served form the independent set of a matroid; further they also relax the i.i.d constraint and deal with non-i.i.d settings. Dhangwatnotai, Roughgarden and Yan [DRY10] study revenue approximations via VCG mechanisms with multiple reserve prices, where the reserve prices are obtained by using the valuations of bidders as a sample from the distributions. Their results apply for matroidal settings when the distributions are regular, and for general downward closed settings when the distributions satisfy the more restrictive monotone hazard rate condition. As previously discussed, Chawla et al. [CHMS10] show that for i.i.d irregular distributions, Vickrey auction with a single reserve price gives a 2-approximation to the optimal revenue and for non-i.i.d distributions Vickrey auction with a distribution-specific reserve price guarantees a 2-approximation; Chawla et al. [CHK07] show that it is impossible to achieve a $o(\log n)$ approximation via Vickrey auction with a single reserve price for non-i.i.d irregular distributions. Single-item Vickrey auctions with bidder specific monopoly reserve prices were also studied in Neeman [Nee03] and Ronen [Ron01]. Approximate revenue maximization via VCG mechanisms with supply limitations were studied in Devanur et al. [DHKN11] and Roughgarden et al. [RTCY12].

3.2 Preliminaries

Basic Model. We study single item auctions among n bidders. Bidder i has a value v_i for a good, and the valuation profile for all the n players together is denoted by $\mathbf{v} = (v_1, v_2, \ldots, v_n)$. In a sealed bid auction each player submits a bid, and the bid profile is denoted by $\mathbf{b} = (b_1, b_2, \ldots, b_n)$. An auction is a pair of functions $(\mathbf{x}, \mathbf{p})$, where x maps a bid vector to outcomes $\{0, 1\}^n$, and p maps a bid vector to $\mathbf{R}_+^n$, i.e., a non-negative payment for each player. The players have quasi-linear utility functions, i.e., their utilities have a separable and linear dependence on money, given by $u_i(v_i, \mathbf{v}_{-i}) = v_i x_i(\mathbf{v}) - p_i(\mathbf{v})$. An auction is said to be dominant strategy truthful if submitting a bid equal to your value yields no smaller utility than any other bid in every situation, i.e., for all $\mathbf{v}_{-i}$, $v_i x_i(\mathbf{v}) - p_i(\mathbf{v}) \geq v_i x_i(b_i, \mathbf{v}_{-i}) - p_i(b_i, \mathbf{v}_{-i})$. Since we focus on truthful auctions in this paper $\mathbf{b} = \mathbf{v}$ from now on.

Distributions. We study auctions in a Bayesian setting, i.e., the valuations of bidders are drawn from a distribution. In particular, we assume that valuation of bidder i is drawn from distribution F_i, which is independent from but not necessarily identical to F_j for $j \neq i$. For ease of presentation, we assume that these distributions are continuous, i.e., they have density function f_i. We assume that the support of these distributions are intervals on the non-negative real line, with non-zero density everywhere in the interval.

Regularity and Irregularity. The hazard rate function of a distribution is defined as $h(x) = \frac{f(x)}{1-F(x)}$. A distribution is said to have a Monotone Hazard Rate(MHR) if $h(x)$ is monotonically non-decreasing. A weaker requirement on distributions is called regularity: the function $\phi(x) = x - \frac{1}{h(x)}$ is monotonically non-decreasing. We do not assume either of these technical conditions for our distributions. Instead we assume that the market of bidders consists of k groups and each group has a regular distribution G_i over valuations. Each bidder is drawn according to some (potentially different) convex combination of these k regular distributions, i.e., $F_i(x) = \sum_{t=1}^{k} p_{i,t} G_t(x)$. Such a distribution $F_i(\cdot)$ in most cases significantly violates the regularity condition.

Almost all irregular distributions used as examples in the literature (eg. Sydney Opera house distribution) can be written as a convex combination of a small number of regular distributions. In fact, mathematically, any irregular distribution can be approximated by a convex combination of sufficiently many regular distributions and as we take the number of regular distributions to infinity then it can be described exactly. Thus the number of regular distributions needed to describe an irregular distribution is a valid measure of irregularity that is well-defined for any distribution.

Revenue Objective. The objective in this paper to design auctions to maximize expected revenue, i.e., the expectation of the sum of the payments of all agents. Formally, the objective is to maximize $\mathbf{E}_{\mathbf{v}}[\sum_i p_i(\mathbf{v})]$. Myerson [Mye81] characterized the expected revenue from any auction as its expected virtual surplus, i.e. the expected sum of virtual values of the agents who receive the item, where the virtual value of an agent is $\phi(v) = v - \frac{1}{h(v)}$. Formally, for all bidders i, $\mathbf{E}_{\mathbf{v}}[p_i(\mathbf{v})] = \mathbf{E}_{\mathbf{v}}[\phi_i(v_i)x_i(\mathbf{v})]$. The equality holds even if we condition on a fixed v_{-i}, i.e., $\mathbf{E}_{v_i}[p_i(v_i, \mathbf{v}_{-i})] = \mathbf{E}_{v_i}[\phi(v_i)x_i(v_i, \mathbf{v}_{-i})]$.

3.3 Targeted Advertising and the Non-i.i.d. Irregular Setting

In this section we give our version of Bulow and Klemperer's result [BK96] for non-i.i.d irregular distributions.

3.3.1 One Extra Bidder from Every Population Group

Theorem 3.4 *Consider an auction among n non-i.i.d irregular bidders where each bidder's distribution F_i is some mixture of k regular distributions $\{G_1, \dots, G_k\}$ (the set of regular distributions is the same for all bidders but the mixture probabilities could be different). The revenue of the optimal auction in this setting is at most twice the revenue of a Vickrey auction with k extra bidders, where each bidder is drawn from a distinct distribution from $\{G_1, \dots, G_k\}$.*

Proof: Bidder i's distribution $F_i(x) = \sum_{t=1}^{k} p_{i,t} G_t(x)$ can be thought of as being drawn based on the following process: first a biased k-valued coin is flipped that decides from which distribution G_t player i's value will come from (according to the probabilities $p_{i,t}$), and then a sample from G_t is drawn. Likewise, the entire valuation profile can be thought of as being drawn in a similar way: first n independent, and possibly non-identically biased, k-valued coin tosses, decide the regular distribution from each bidder's value is going to be drawn from. Subsequently a sample is drawn from each distribution.

Let the random variable q_i be the index of the regular distribution that bidder i's value is going to be drawn, i.e., q_i is the result of the coin toss for bidder i. Let q denote the index profile of all players. Let $p(q) = \prod_{i=1}^{n} p_{i,q_i}$ be the probability that the index profile q results after the n coin tosses. Let $G(q) = \times_i G_{q_i}$ be the joint product distribution of players' values conditioned on the profile being q.

Let M_q be the optimal mechanism when bidders' distribution profile is q. Let $\mathcal{R}^q_M$ be the expected revenue of mechanism M_q. Let $R^q_M(\mathbf{v})$ denote the revenue of the mechanism when bidders have value $\mathbf{v}$. The revenue of the optimal mechanism M which cannot learn and exploit the actual distribution profile q is upper bounded by the revenue of the optimal mechanism that can first learn q. Therefore we have,

$$\mathcal{R}_M \leq \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R^q_M(\mathbf{v})] \tag{3.1}$$

Now, $\mathbf{E}_{\mathbf{v}\sim G(q)}[R^q_M(\mathbf{v})]$ corresponds to the optimal expected revenue when bidder i's distribution is the regular distribution G_{q_i}. Let $k(q)$ denote the number of distinct regular distributions contained in the profile q. Note that $k(q) \leq k$ for all q. Thus the above expectation corresponds to the revenue of a single-item auction where players can be categorized in $k(q)$ groups and bidders within each group t are distributed i.i.d. according to a regular distribution G_t. Theorem 6.3 of [RTCY12] applies to such a setting and shows that the optimal revenue for each of these non-i.i.d regular settings will be at most twice the revenue of Vickrey auction with one extra bidder for each distinct distribution in the profile q.

Hence,

$$\mathcal{R}_M \leq \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R^q_M(\mathbf{v})] \leq 2\sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_{SP_{n+k(q)}}(\mathbf{v})] \tag{3.2}$$

$$\leq 2\sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_{SP_{n+k}}(\mathbf{v})] \tag{3.3}$$

Since, the Vickrey auction with k extra bidders doesn't depend on the index profile q the RHS of (3.3) corresponds to the expected revenue of SP_{n+k} when bidders come from the initial i.i.d irregular distributions. ∎

The above proof actually proves an even stronger claim: the revenue from running the Vickrey auction with k extra bidders is at least half approximate even if the auctioneer could distinguish bidders by learning the bidder distribution profile q and run the corresponding optimal auction R^q_M.

Lower bound. A corner case of our theorem is when each bidder comes from a different regular distribution. From Hartline and Roughgarden [HR09] we know that a lower bound of 4/3 exists for such a case. In other words there exists two regular distributions such that if the initial bidders came each from a different distribution among these, then adding two extra bidders from those distributions will not give the optimal revenue but rather a 4/3 approximation to it. In other words, we cannot hope to get a Bulow Klemperer type of a result where the revenue of the second price auction with extra bidders outperforms the optimal auction. However, as mentioned in the introduction, an interesting open question is if we can get a Bulow-Klemperer type result for i.i.d. irregular distributions, i.e., can a second-price auction with extra bidders outperform optimal auction's revenue. Note that the lower bound result of Hartline and Roughgarden [HR09] doesn't apply to i.i.d. irregular settings.

3.3.2 Just One Extra Bidder in Total for Hazard Rate Dominant Distributions

In this section we examine the setting where among the k underlying regular distributions there exists one distribution that stochastically dominates the rest in the sense of hazard rate dominance. Hazard rate dominance is a standard dominance concept used while establishing revenue guarantees for auctions (see for example [Kir11]) and states the following: A distribution F hazard rate dominates a distribution G iff for every x in the intersection of the support of the two distributions: $h_F(x) \leq h_G(x)$. We denote such a domination by $F \succeq_{hr} G$.

In such a setting it is natural to ask whether adding just a single player from the dominant distribution is enough to produce good revenue guarantees. We actually show that adding only one extra person coming from the dominant distribution achieves exactly the same worst-case guarantee as adding k extra bidders one from each underlying distribution.

Theorem 3.5 *Consider an auction among n non-i.i.d irregular bidders where each bidder's distribution F_i is some mixture of k regular distributions $\{G_1, \ldots, G_k\}$ such that $G_1 \succeq_{hr} G_t$ for all t. The revenue of the optimal auction in this setting is at most twice the revenue of a Vickrey auction with one extra bidder drawn from G_1.*

The proof is based on a new lemma for the regular distribution setting: bidders are drawn from a family of k regular distributions such that one of them hazard-rate dominates the rest. This lemma can be extended to prove Theorem 3.5 in a manner identical to how Theorem 6.3 of Roughgarden et al. [RTCY12] was extended to prove

Theorem 3.4 in our paper. We don't repeat that extension here, and instead just prove the lemma. The lemma uses the notion of commensurate auctions defined by Hartline and Roughgarden [HR09], which we define (and reproduce a related lemma from [HR09]) in Appendix 3.6 for completeness.

Lemma 3.6 *Consider a non-i.i.d. regular setting where each player's value comes from some set of distributions $\{F_1, \ldots, F_k\}$ such that $F_1 \succeq_{hr} F_t$ for all t. The optimal revenue of this setting is at most twice the revenue of Vickrey auction with one extra bidder drawn from F_1.*

Proof: Let $\mathbf{v}$ denote the valuation profile of the initial n bidders and let v^* the valuation of the extra bidder from the dominant distribution. Let $R(\mathbf{v}, v^*)$ and $S(\mathbf{v}, v^*)$ denote the winners of the optimal auction (M) and of the second price auction with the extra bidder (SP_{n+1}) respectively. We will show, that the two auctions are commensurate, which boils down to showing that:

$$\mathbf{E}_{\mathbf{v},v^*}[\phi_{S(\mathbf{v},v^*)}(v_{S(\mathbf{v},v^*)})|S(\mathbf{v}, v^*) \neq R(\mathbf{v}, v^*)] \geq \ 0 \tag{3.4}$$

$$\mathbf{E}_{\mathbf{v},v^*}[\phi_{R(\mathbf{v},v^*)}(v_{R(\mathbf{v},v^*)})|S(\mathbf{v}, v^*) \neq R(\mathbf{v}, v^*)] \leq \ \mathbf{E}_{\mathbf{v},v^*}[p_{S(\mathbf{v},v^*)}|S(\mathbf{v}, v^*) \neq R(\mathbf{v}, v^*)] \tag{3.5}$$

where p_S is the price paid by the winner of the second price auction. The proof of equation (3.5) is easy and very closely follows the proof in [HR09] above.

We now prove equation (3.4). Since $F_1 \succeq_{hr} F_t$ we have that for all x in the intersection of the support of F_1 and F_t: $h_1(x) \leq h_t(x)$, which in turn implies that $\phi_1(x) \leq \phi_t(x)$, since $\phi_t(x) = x - \frac{1}{h_t(x)}$. By the definition of winner in Vickrey auction we have $\forall i : v_{S(\mathbf{v},v^*)} \geq v_i$. In particular, $v_{S(\mathbf{v},v^*)} \geq v^*$. If v^* is in the support of $F_{S(\mathbf{v},v^*)}$, then the latter, by regularity of distributions, implies that $\phi_{S(\mathbf{v},v^*)}(v_{S(\mathbf{v},v^*)}) \geq \phi_{S(\mathbf{v},v^*)}(v^*)$. Now $F_1 \succeq_{hr} F_t$ implies that $\phi_{S(\mathbf{v},v^*)}(v^*) \geq \phi_1(v^*)$ (since by definition v^* must also be in the support of F_1). If v^* is not in the support of $F_{S(\mathbf{v},v^*)}$, then since $v^* < v_{S(\mathbf{v},v^*)}$ and all the supports are intervals, it must be that v^* is below the lower bound L of the support of $F_{S(\mathbf{v},v^*)}$. Wlog we can assume that the support of F_1 intersects the support of every other distribution. Hence, since v^* is below L and the support of F_1 is an interval, L will also be in the support of F_1. Thus L is in the intersection of the two supports. By regularity of $F_{S(\mathbf{v},v^*)}, F_1$ and by the hazard rate dominance assumption, we have $\phi_{S(\mathbf{v},v^*)}(v_{S(\mathbf{v},v^*)}) \geq \phi_{S(\mathbf{v},v^*)}(L) \geq \phi_1(L) \geq \phi_1(v^*)$.

Thus in any case $\phi_{S(\mathbf{v},v^*)}(v_{S(\mathbf{v},v^*)}) \geq \phi_1(v^*)$. Hence, we immediately get that:

$$\mathbf{E}_{\mathbf{v},v^*}[\phi_{S(\mathbf{v},v^*)}(v_{S(\mathbf{v},v^*)})|S(\mathbf{v},v^*) \neq R(\mathbf{v},v^*)] \geq \mathbf{E}_{\mathbf{v},v^*}[\phi_1(v^*)|S(\mathbf{v},v^*) \neq R(\mathbf{v},v^*)]$$

Conditioned on $\mathbf{v}$ the latter expectation becomes:

$$\mathbf{E}_{v^*}[\phi_1(v^*)|S(\mathbf{v},v^*) \neq R(\mathbf{v},v^*), \mathbf{v}]$$

But conditioned on $\mathbf{v}$, $R(\mathbf{v},v^*)$ is some fixed bidder i. Hence, the latter expectation is equivalent to: $\mathbf{E}_{v^*}[\phi_1(v^*)|S(\mathbf{v},v^*) \neq i]$ for some i. We claim that for all i the latter expectation must be positive. Conditioned on $\mathbf{v}$, the event $S(\mathbf{v},v^*) \neq i$ happens only if v^* is sufficiently high, i.e., there is a threshold $\theta(\mathbf{v})$ such that $S(\mathbf{v},v^*) \neq i$ happens only if $v^* \geq \theta(\mathbf{v})$ (if i was the maximum valued bidder in the profile $\mathbf{v}$ then $\theta(\mathbf{v}) = v_i$, else $\theta(\mathbf{v}) = 0$.) By regularity of distributions, $v^* \geq \theta(\mathbf{v})$ translates to $\phi_1(v^*) \geq \phi_1(\theta)$. So we now have to show that: $\mathbf{E}_{v^*}[\phi_1(v^*)|\phi_1(v^*) \geq \phi_1(\theta)] \geq 0$. Since the unconditional expectation of virtual value is already non-negative, the expectation conditioned on a lower bound on virtual values is clearly non-negative. ■ ■

3.3.2.0.1 Examples and Applications. There are many situations where a hazard-rate dominant distribution aactually exists in the market. We provide some examples below.

Uniform, Exponential, Power-law distributions. Suppose the k underlying distributions were all uniform distributions of the form $U[a_i, b_i]$. The hazard rate $h_i(x) = \frac{1}{b_i - x}$. Clearly, the distribution with a larger b_i hazard-rate dominates the distribution with a larger b_i. If the k underlying distributions were all exponential distributions, i.e., $G_i(x) = 1 - e^{-\lambda_i x}$, then the hazard rate $h_i(x) = \lambda_i$. Thus the distribution with the smallest λ_i hazard rate dominates the rest. If the k underlying distributions were all power-law distributions, namely, $G_i(x) = 1 - \frac{1}{x^{\alpha_i}}$, then the hazard rate $h_i(x) = \frac{\alpha_i}{x}$. Thus the distribution with the smallest α_i hazard-rate dominates the rest.

A general condition. If all the k underlying regular distributions were such that for any pair i, j they satisfy $1 - G_i(x) = (1 - G_j(x))^{\theta_{ij}}$, then it is easy to verify that there always exists one distribution that hazard-rate dominates the rest of the distributions. For instance, the family of exponential distributions, and the family of power-law distributions are special cases of this general condition.

3.4 Non-Targeted Advertising and the i.i.d. Irregular Setting

In this section we consider the setting where all the bidders are drawn from the same distribution F. We assume that F can be written as a convex combination of k regular distributions $F_1, \ldots, F_k$, i.e. $F = \sum_{t=1}^{k} p_t F_t$ and such that the mixture probability p_t for every distribution is at least some constant δ: $\forall t \in [1, \ldots, k] : p_t \geq \delta$. A natural question to ask in an i.i.d. setting is how many extra bidders should be recruited from the original distribution to achieve a constant fraction of the optimal revenue (i.e., by running a non-targeted advertising campaign)?

In this section answer the above question as a function of the number of underlying distributions k and the minimum mixture probability δ. We remark that our results in this section don't require the decomposition of F into the F_t's resemble the distribution of the underlying population groups. Even if the number of underlying population groups is very large, as long as there is some mathematical way of decomposing F into k regular distributions with a minimum mixture probability of δ, our results go through. Hence, one can optimize our result for each F by finding the decomposition that minimizes our approximation ratio.

Theorem 3.7 *Consider an auction among n i.i.d. irregular bidders where the bidders' distribution F can be decomposed into a mixture of k regular distributions $\{G_1, \ldots, G_k\}$ with minimum mixture probability δ. The revenue of the optimal auction in this setting is at most $2\frac{k+1}{k}$ the revenue of a Vickrey auction with $\Theta\left(\frac{log(k)}{\delta}\right)$ extra bidders drawn from distribution F.*

Proof: Suppose that we bring n^* extra bidders in the auction. Even if the decomposition of the distribution F doesn't correspond to an actual market decomposition, we can always think of the value of each of the bidders drawn as follows: first we draw a number t from 1 to k according to the mixture probabilities p_t and then we draw a value from distribution G_t.

Let $\mathcal{E}$ be the event that all numbers 1 to k are represented by the n^* random numbers drawn to produce the value of the n^* extra bidders. The problem is a generalization of the coupon collector problem where there are k coupons and each coupon arrives with probability $p_t \geq \delta$. The relevant question is, what is the probability that all the coupons are collected after n^* coupon draws? The probability that a coupon t is not collected after n^* draws is: $(1-p_t)^{n^*} \leq (1-\delta)^{n^*} \leq e^{-n^*\delta}$. Hence, by the union bound, the probability that some coupon is not collected after n^* draws is at most $ke^{-n^*\delta}$. Thus the probability of event $\mathcal{E}$ is at least $1-ke^{-n^*\delta}$. Thus if $n^* = \frac{\log(k)+\log(k+1)}{\delta}$ then the probability of $\mathcal{E}$ is at least $1-\frac{1}{k+1}$.

Conditional on event $\mathcal{E}$ happening we know that the revenue of the auction is the revenue of the initial auction with at least one player extra drawn from each of the underlying k regular distributions. Thus we can apply our main theorem 3.4 to get that the expected revenue conditional on $\mathcal{E}$ is at least $\frac{1}{2}$ of the optimal revenue with only the initial n bidders. Thus:

$$\mathcal{R}_{SP_{n+n^*}} \geq \left(1-\frac{1}{k+1}\right) \mathbf{E}_{v,\tilde{v}\sim F^{n+n^*}}[R_{SP_{n+n^*}}(v,\tilde{v})|\mathcal{E}] \geq \left(1-\frac{1}{k+1}\right)\frac{1}{2}\mathcal{R}_M \tag{3.6}$$

■

Theorem 3.8 *Consider an auction among n i.i.d. irregular bidders where the bidders' distribution F can be decomposed into a mixture of k regular distributions $\{G_1,\ldots,G_k\}$ such that G_1 hazard rate dominates G_t for all $t>1$. The revenue of the optimal auction in this setting is at most $2\frac{e}{e-1}$ the revenue of a Vickrey auction with $\frac{1}{p_1}$ extra bidders drawn from distribution F.*

Proof: Similar to theorem 3.7 conditional on the even that an extra player is drawn from the hazard rate distribution, we can apply Lemma 3.6 to get that this conditional expected revenue is at least half the optimal revenue with the initial set of players.

If we bring n^* extra players then the probability of the above event happening is $1-(1-p_1)^{n^*} \geq 1-e^{-n^*p_1}$. Setting $n^* = \frac{1}{p_1}$ we get the theorem. ■

3.5 Vickrey with Single Reserve for Irregular Settings

In this section we prove revenue guarantees for Vickrey auction with a single reserve in irregular settings. We start by presenting results for the i.i.d. irregular setting presented in the previous paragraph — we give an approximation guarantee that is logarithmic in the number of mixture distributions that the irregular distribution can be decomposed into. We then prove Theorem 3.11 for the general non-i.i.d. case that doesn't require any minimum mixture probability.

Our revenue guarantees for the i.i.d. irregular setting are an immediate corollary of Theorems 3.7 and 3.8, by simply noticing that the all bidders in the augmented bidder setting are distributed identically. Hence, by symmetry the revenue contribution of each one of them is the same. Moreover, we can simulate the effect of the extra bidders needed, by simply running a Vickrey auction with a random reserve price drawn from the maximum value of the extra bidders.

Corollary 3.9 *Consider an auction among n i.i.d. irregular bidders where the bidders' distribution F can be decomposed into a mixture of k regular distributions $\{G_1, \ldots, G_k\}$ with minimum mixture probability δ. The revenue of the optimal auction in this setting is at most $2\frac{k+1}{k}\left(1+\frac{1}{n}\lceil\frac{2log(k)}{\delta}\rceil\right) = \Theta\left(1+\frac{log(k)}{n\delta}\right)$ the revenue of Vickrey auction with a single random reserve price drawn from the distribution of the maximum of $\lceil\frac{2log(k)}{\delta}\rceil$ independent random samples of F.*

Corollary 3.10 *Consider an auction among n i.i.d. irregular bidders where the bidders' distribution F can be decomposed into a mixture of k regular distributions $\{G_1, \ldots, G_k\}$ such that G_1 hazard rate dominates G_t for all $t > 1$. The revenue of the optimal auction is at most $2\frac{e}{e-1}\left(1+\frac{1}{n}\lceil\frac{1}{p_1}\rceil\right)$ the revenue of Vickrey auction with a single random reserve drawn from the distribution of the maximum of $\lceil\frac{1}{p_1}\rceil$ independent random samples of F.*

These results give good approximations with respect to the optimal revenue, but require more structure from the bidder distributions (i.e. i.i.d. and minimum mixture probability). In our final result we give an approximation guarantee of the Vickrey auction with a single anonymous reserve for the general non-i.i.d. irregular setting that we consider (i.e. player-specific mixtures are allowed and there is no minimum mixture probability).

Theorem 3.11 *Consider an auction among n non-i.i.d irregular bidders where each bidder's distribution F_i is some mixture of k regular distribution $\{G_1, \ldots, G_k\}$ (the set of regular distributions is the same for all bidders but the mixture probabilities could be different). The revenue of the optimal auction in the above setting is at most $4k$ times the revenue of a second price auction with a single reserve price which corresponds to the monopoly reserve price of one of the k distributions G_i.*

Proof: We use the same notation as in Section 3.3. In particular, we let q denote the index profile of distributions for all players and $p(q) = \prod_{i=1}^{n} p_{i,q_i}$ be the probability that an index profile arises. Let $G(q) = \times_i G_{q_i}$ be the product distribution that corresponds to how players values are distributed conditional on the coin tosses having value q.

Let M_q be the optimal mechanism when bidders' distribution profile is q. Let $\mathcal{R}_M^q$ be the expected revenue of mechanism M_q. By equation (3.2) in Section 3.3 we have,

$$\mathcal{R}_M \leq \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_M^q(\mathbf{v})] \leq 2 \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_{SP_{n+k(q)}}(\mathbf{v})] \tag{3.7}$$

Consider the auction $SP_{n+k(q)}$. If instead of adding the $k(q)$ extra bidders, we place a random reserve drawn from the distribution of the maximum value among the $k(q)$ extra bidders, and ran the Vickrey auction. Call the later $SP_n(R(q))$. If the winner of the auction $SP_{n+k(q)}$ is one among the original n bidders, clearly $SP_{n+k(q)}$ and $SP_n(R(q))$ will have the same expected revenue. Further, the expected revenue of $SP_{n+k(q)}$ conditioned on the winner being one among the original n bidders is no smaller than the expected revenue of $SP_{n+k(q)}$ conditioned on the winner being one among the newly added $k(q)$ bidders. Also, the probability that the winner comes from the newly added $k(q)$ bidders is at most 1/2. Thus $SP_n(R(q)) \geq \frac{1}{2}SP_{n+k(q)}$. Combining this with Equation (3.7), we have

$$\begin{aligned}
\mathcal{R}_M &\leq 2 \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_{SP_{n+k(q)}}(\mathbf{v})] \leq 4 \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_{SP_n(R(q))}(\mathbf{v})] \\
&= 4 \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[\sum_{t=1}^{k} R_{SP_n(R(q),t)}(\mathbf{v})] && (3.8) \\
&= \sum_{t=1}^{k} 4 \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_{SP_n(R(q),t)}(\mathbf{v})] \leq 4k \sum_{q\in[1..k]^n} p(q)\mathbf{E}_{\mathbf{v}\sim G(q)}[R_{SP_n(R(q),t^*)}(\mathbf{v})] && (3.9)
\end{aligned}$$

In equation (3.8), the revenue $R_{SP_n(R(q))}(\mathbf{v})$ is written as $\sum_{t=1}^{k} R_{SP_n(R(q),t)}(\mathbf{v})$, i.e., as the sum of contributions from each population group. Given this split, there exists a polulation group t^* that gets at least $\frac{1}{k}$ fraction of all groups together, and thus at least $\frac{1}{4k}$ fraction of the optimal mechanism, which is what is expressed through inequality (3.9).

Now the auction $SP_n(R(q))$ from the perspective of the group t^* is just the Vickrey auction run for group t^* alone with a single random reserve of max$\{R(q)$, Maximum value from groups other than $t^*\}$. However within the group t^* since we are in a i.i.d regular setting it is optimal to run Vickrey auction for the group t^* alone with the monopoly reserve price of that group. That is if we replace the single reserve of max$\{R(q)$, Maximum value from groups other than $t^*\}$ with the optimal (monopoly) reserve price for t^*, Vickrey auction for group t^* with such a reserve gives no lesser revenue, and this holds for every q! Finally, when we add in the agents from other groups, single-item Vickrey auction's revenue for the entire population with monopoly reserve price of group t^* is no smaller than the revenue of single-item Vickrey auction for group t^* alone with the monopoly reserve price of group t^*. Chaining the last two statements proves the theorem. ■

3.6 Deferred Proofs

Definition 3.12 *An auction M' is commensurate to M if:*

$$\mathbf{E}[\phi_{W'}(v_{W'})|W' \neq W] \geq \ 0 \tag{3.10}$$

$$\mathbf{E}[\phi_{W}(v_{W})|W' \neq W] \leq \ \mathbf{E}_{\mathbf{v}',\mathbf{v}}[p_{W'}|W' \neq W] \tag{3.11}$$

where W', W are the winners of M', M respectively and $p_{W'}$ is the price paid by the winner of M'.

The lemma also uses a theorem from [HR09] which we state below and provide a proof for completeness.

Theorem 3.13 (Hartline and Roughgarden [HR09]) *If mechanism M' is commensurate to M then $R_M \leq 2R_{M'}$ where $R_{M'}, R_M$ are the expected revenues of mechanisms M' and M.*

Proof: By Myerson's characterization [Mye81] we know that the expected revenue of any truthful auction is equal to its expected virtual surplus.

$$\begin{aligned} R_M = \mathbf{E}[\phi_W(v_W)] = &\ \mathbf{E}[\phi_W(v_W)|W' = W]Pr[W' = W] + \mathbf{E}[\phi_W(v_W)|W' \neq W]Pr[W' \neq W] \\ = &\ \mathbf{E}[\phi_{W'}(v_{W'})|W' = W]Pr[W' = W] + \mathbf{E}[\phi_W(v_W)|W' \neq W]Pr[W' \neq W] \end{aligned}$$

We lower bound each of the two terms in the RHS above by $R_{M'}$. By property (3.10) we have:

$$\begin{aligned}
&\mathbf{E}[\phi_{W'}(v_{W'})|W' = W]Pr[W' = W] \\
&\quad \leq \mathbf{E}[\phi_{W'}(v_{W'})|W' = W]Pr[W' = W] + \mathbf{E}[\phi_{W'}(v_{W'})|W' \neq W]Pr[W' \neq W] \\
&\quad = \mathbf{E}[\phi_{W'}(v_{W'})] = R_{M'}
\end{aligned}$$

By property (3.11) we have:

$$\begin{aligned}
&\mathbf{E}[\phi_{W}(v_{W})|W' \neq W]Pr[W' \neq W] \\
&\quad \leq \mathbf{E}[p_{W'}|W' \neq W]Pr[W' \neq W] \\
&\quad \leq \mathbf{E}[p_{W'}|W' \neq W]Pr[W' \neq W] + \mathbf{E}[p_{W'}|W' = W]Pr[W' = W] = R_{M'}
\end{aligned}$$

■

3.7 Conclusion

In this chapter, we introduced a way to parameterize irregular distributions as convex combinations of regular distributions, and showed that if we can target the mixtures that constitute the irregular distribution to recruit extra bidders, then the second price auction gets a good fraction of the optimal auction's revenue. An interesting open question is, for the case of i.i.d. irregular distributions, does the second price auction with one extra bidder from each constituent regular distribution outperform the optimal auction's revenue instead of just getting a constant factor of the optimal auction's revenue.

4 Prior Robust Mechanisms for Machine Scheduling

Organization. In this chapter, we design and analyze prior robust mechanisms for makespan minimization in machine scheduling. The chapter is organized as follows. In Section 4.1 we informally summarize our results, put them in context of related work with some additional discussion. In Section 4.2 we provide all the necessary preliminaries and the formal statements of all our results in this chapter. In Section 4.3 we design and analyze prior robust truthful machine scheduling mechanism when machines are a priori identical, but jobs need not be. In Section 4.4 we obtain better guarantees when both jobs and machines are a priori identically distributed. In Section 4.5 we provide the deferred proofs, and we conclude with some future research directions in Section 4.6.

4.1 Introduction & Summary of Results

We study the problem of scheduling jobs on machines to minimize makespan in a strategic context. The makespan is the longest time it takes any of the machines to complete the work assigned by the schedule. The running time or size of a job on a machine is drawn from a fixed distribution, and is a private input known to the machine but not to the optimizer. The machines are unrelated in the sense that the running time of a job on distinct machines may be distinct. A scheduling mechanism solicits job running times from the machines and determines a schedule as well as compensation for each of the machines. The machines are strategic and try to maximize the compensation they receive minus the work they perform. We are interested in understanding and quantifying the loss in performance due to the strategic incentives of the machines who may misreport the job running times.

A primary concern in the theory of mechanism design is to understand the compatibility of various objectives of the designer with the incentives of the participants. As an example, maximizing social welfare is incentive compatible; the Vickrey-Clarke-Groves (VCG) mechanism obtains this socially optimal outcome in equilibrium [Vic61, Cla71, Gro73]. For most other objectives, however, the optimal solution ignoring incentives (a.k.a. the *first-best* solution) cannot be implemented in an incentive compatible manner. This includes, for example, the objectives of revenue maximization, welfare maximization with budgets, and makespan minimization with unrelated machines. For these objectives there is no incentive compatible mechanism that is best on every input. The classical economic approach to mechanism design thus considers inputs drawn from a distribution (a.k.a. the *prior*) and looks for the mechanism that maximizes the objective in expectation over the distribution (a.k.a. the *second-best* solution).

The second-best solution is generally complex and, by definition, tailored to specific knowledge that the designer has on the distribution over the private information (i.e., the input) of the agents. The non-pointwise optimality, complexity, and distributional dependence of the second-best solution motivates a number of mechanism design and analysis questions.

price of anarchy: For any distribution over inputs, bound the gap between the first-best (optimal without incentives) and second-best (optimal with incentives) solutions (each in expectation over the input).

computational tractability: For any distribution over inputs, give a computationally tractable implementation of the second-best solution, or if the problem is intractable give a computationally tractable approximation mechanism.

simplicity: For any distribution over inputs, give a simple, practical mechanism that approximates the second-best solution.

prior independence: Give a single mechanism that, for all distributions over inputs, approximates the second-best solution.

These questions are inter-related. As the second-best mechanism is often complex, the price of anarchy can be bounded via a lower bound on the second-best mechanism as given by a simple approximation mechanism. Similarly, to show that a mechanism is a good approximation to second-best the upper bound given by the first-best solution can be used. Importantly though, if the first-best solution does not permit good approximation mechanisms then a better bound on the second-best solution should be sought. Each of the questions above can be further refined by consideration with respect to a large class of priors (e.g. identical distributions).

The prior-independence question gives a middle ground between worst-case mechanism design and Bayesian mechanism design. It attempts to achieve the best of both worlds in the tradeoff between informational efficiency and approximate optimality. Its minimal usage of information about the setting makes it robust. A typical side-effect of this robustness is simple and natural mechanisms; indeed, our prior-independent mechanisms will be simple, computationally tractable, and also enable a bound on the price of anarchy.

The literature on prior-independent mechanism design has focused primarily on the objective of revenue maximization. Hartline and Roughgarden [HR09] show that with sufficient competition, the welfare maximizing (VCG) mechanism also attains good revenue. This result enables the prior-independent approximation mechanism for single-item auctions of Dhangwatnotai, Roughgarden, and Yan [DRY10] and the multi-item approximation mechanisms of Devanur et al. [DHKN11] and Roughgarden et al. [RTCY12]. Importantly, in single-item auctions the agents' private information is single-dimensional whereas in multi-item auctions it is multi-dimensional. There are several interesting and challenging directions in prior-independent mechanism design: (1) non-linear objectives, (2) general multi-parameter preferences of agents, (3) non-downwards-closed feasibility constraints, and (4) non-identically distributed types of agents. Our work addresses the first three of these four challenges.

We study the problem of scheduling jobs on machines where the runtime of a job on a machine is that machine's private information. The prior over runtimes is a product distribution that is symmetric with respect to the machines (but not necessarily symmetric with respect to the jobs). *Ex ante*, i.e., before the job sizes are instantiated, the machines appear identical; *ex post*, i.e., after the job sizes are realized, the machines are distinct and job runtimes are unrelated. The makespan objective is to schedule the jobs on machines so as to minimize the time at which the last machine completes all of its assigned jobs. Our goal is a prior-independent approximation of the second-best solution for the makespan objective.

To gain intuition for the makespan objective, consider why the simple and incentive compatible VCG mechanism fails to produce a good solution in expectation. The VCG mechanism for scheduling minimizes the total work done by all of the machines and accordingly places every job on its best machine. Note that because the machines are a priori identical, this is an i.i.d. uniformly random machine for every job. Therefore, in expectation, every machine gets an equal number of jobs. Furthermore, every job simultaneously has its smallest size possible. However, the maximum load in terms of the number of jobs per machine and so also the makespan can be quite large. The distribution of jobs across machines is akin to the distribution of balls into bins in the standard balls-in-bins experiment—when the number of balls and bins is equal, the maximum loaded bin contains $\Theta(\log n/\log\log n)$ balls with high probability even though the average load is 1.

Our designed mechanism must prevent the above balls-in-bins style behavior. Consider a variant of VCG that we call the *bounded overload* mechanism. The bounded overload mechanism minimizes the total work with the additional feasibility constraint that the load (i.e., number of jobs scheduled) of any machine is bounded to be at most a c factor more than the average load. This mechanism is "maximal in range", i.e., it is simply the VCG mechanism with a restricted space of feasible outcomes; it is therefore incentive compatible. Moreover, the bounded overload mechanism can be viewed as belonging to a class of "supply limiting" mechanisms (cf. the prior-independent supply-limiting approximation mechanism of [RTCY12] for multi-item revenue maximization).

While the bounded overload mechanism evens out the number of jobs per machine, an individual job may end up having a running time far larger than that on its best machine. The crux of our analysis is to show that this does not hurt the expected makespan of our schedule relative to an ideal setting where every job assumes its minimum size. Our analysis of job sizes has two components. First we show that every job with high probability gets assigned to one of its best machines. Second, we show that the running time of a job on its ith best machine can be related within a factor depending on i to its running time on its best machine. These components together imply that the bounded overload mechanism simultaneously obtains a schedule that is balanced in terms of the number of jobs per machine and where every job has a small size (in comparison to the best possible for that job). This is sufficient to imply a constant factor approximation to expected makespan when the number of jobs is proportional to the number of machines.

The second component of our analysis of job sizes in the bounded overload mechanism entails relating different order statistics of (arbitrary) i.i.d. distributions, a property that may have broader applications. In particular, letting $X[k{:}n]$ denote the kth minimum out of n independent draws from a distribution, we show that for any k and n, $X[k{:}n]$ is nearly stochastically dominated by an exponential function of k times $X[1{:}n/2]$. In simple terms, the minimum out of a certain number of draws cannot be arbitrarily smaller than the kth minimum out of twice as many draws.

As an intermediary step in our analysis we bound the performance of our approximation mechanism with respect to the first-best solution with half the machines (recall, machines are a priori identical). Within the literature on prior-independent revenue maximization this approach closely resembles the classical Bulow-Klemperer theorem [BK96]. For auctioning k units of a single-item to n agents (with values drawn i.i.d. from a "nice" distribution), the revenue from welfare maximization exceeds the optimal revenue from $n - k$ agents. In other words, a simple prior-independent mechanism with extra competition (namely, k extra agents) is better than the prior-optimal mechanism for expected revenue. Our result is similar: when the number of jobs is at most the number of machines and machines are a priori identical, we present a prior-independent mechanism that is a constant approximation to makespan with respect to the first-best (and therefore also with respect to the second-best) solution with half as many machines. Unlike the Bulow-Klemperer theorem we place no assumptions the distribution of jobs on machines besides symmetry with respect to machines.

To design scheduling mechanisms for the case where the number of jobs is large relative to the number of machines we can potentially take advantage of the law of large numbers. If there are many more large jobs (i.e., jobs for which the best of the machines' runtimes is significant) then assigning jobs to machines to minimize total work will produce a schedule where the maximum work on any machine is concentrated around its expectation; moreover, the expected load of any machine in the schedule that minimizes total work is at most the expected load of any machine in the schedule that minimizes makespan.

On the other hand, if there are a moderate number, e.g., proportional to the number of machines, of jobs with very large runtimes on all machines, both the minimum work mechanism and the bounded overload mechanism can fail to have good expected makespan. For the bounded overload mechanism, although the distribution of jobs across machines is more-or-less even, the distribution of the few "worst" jobs that contribute the most to the makespan may be highly uneven. Indeed, for a distribution where the expected number of large jobs is about the same as the number of machines, the bounded overload mechanism exhibits the same bad balls-in-bins behavior as the minimum work mechanism.

The problem above is that the existence of many small, but relatively easy to schedule jobs, prevents the bounded overload mechanism from working. To solve this problem we employ a two stage approach. The first stage acts as a sieve and schedules the small jobs to minimize total work and while leaving the large jobs unscheduled. Then in the second stage the bounded overload mechanism is run on the unscheduled jobs. With the proper parameter tunings (i.e., job size threshold for the sieve and partitioning of machines to the two stages) this mechanism gives a schedule with approximately optimal expected makespan. We give two parameter tunings and analyses, one which gives an $O(\sqrt{\log m})$ approximation and the other that gives an $O((\log\log m)^2)$ approximation under a certain tail condition on the distribution of job sizes (satisfied, for example, by all monotone hazard rate distributions).

The proper tuning of the parameters of the mechanism require knowledge of a single order statistic of the size distribution, namely the expected size of a job on its best out of k machines for an appropriate value of k, to decide which jobs get scheduled in which stage. This statistic can be easily estimated as the mechanism is running by using the reports of a small fraction of the machines as a "market analysis." To keep our exposition and analysis simple, we skip this detail and assume that the statistic is known.

Related Work

There is a large body of work on prior-free mechanism design for the makespan objective. This work does not assume a prior distribution, instead it looks at worst-case approximation of the first-best solution (i.e., the optimal makespan without incentive constraints). The problem was introduced by Nisan and Ronen [NR99] who showed that the minimum work (a.k.a. VCG) mechanism gives an m-approximation to makespan (where m is the number of machines). They gave a lower bound of two on the worst case approximation factor of any dominant strategy mechanism for unrelated machine scheduling. They conjectured that the best worst-case approximation is indeed $\Theta(m)$. Following this work, a series of papers presented better lower bounds for deterministic as well as randomized mechanisms [CKV07, CKK07, KV07, MS07]. Ashlagi, Dobzinski and Lavi [ADL09] recently proved a restricted version of the Nisan-Ronen conjecture by showing that no *anonymous* deterministic dominant-strategy incentive-compatible mechanism can achieve a factor better than m. This lower bound suggests that the makespan objective is fundamentally incompatible with incentives in prior-free settings. In this context, our work can be viewed as giving a meaningful approach for obtaining positive results that are close to prior-free for a problem for which most results are very negative.

Given these strong negative results, several special cases of the problem have been studied. Lavi and Swamy [LS07] give constant factor approximations when job sizes can take on only two different values. Lu and Yu [LY08b, LY08a, Lu09] consider the problem over two machines, and give approximation ratios strictly better than 2.

Related machine scheduling is the special case where the runtime of a job on a machine is the product of the machine's private speed and the job's public length. Importantly, the private information of each machine in a related machine scheduling problem is single-dimensional, and the total length of the jobs assigned to any given machine in the makespan minimizing schedule is monotone in the machine's speed. This monotonicity implies that the related machine makespan objective is incentive compatible (i.e., the price of anarchy is one). For this reason work on related machine scheduling has focused on computational tractability. Archer and Tardos [AT01] give a constant approximation mechanism and Dhangwotnotai et al. [DDDR08] give an incentive compatible polynomial time approximation scheme thereby matching the best approximation result absent incentives. There are no known approximation-preserving black-box reductions from mechanism design to algorithm design for related machine scheduling; moreover, in the Bayesian model Chawla, Immorlica, and Lucier [CIL12] recently showed that the makespan objective does not admit black-box reductions of the form that Hartline and Lucier [HL10] showed exist for the objective of social welfare maximization.

Another line of work studies the makespan objective subject to an envy-freedom constraint instead of the incentive-compatibility constraint. A schedule and payments (to the machines) are envy free if every machine prefers its own assignment and payment to that of any other machine. Mu'alem [Mu'09] introduced the envy-free scheduling problem for makespan. Cohen et al. [CFF$^+$10] gave a polynomial time algorithm for computing an envy-free schedule that is an $O(\log m)$ approximation to the first-best makespan (i.e., the optimal makespan absent envy-freedom constraints). Fiat and Levavi [FL12] complement this by showing that the optimal envy-free makespan (a.k.a. second-best makespan) can be an $\Omega(\log m)$ factor larger than the first-best makespan.

4.2 Preliminaries & Main Results

We consider the scheduling of n jobs on m unrelated machines where the running time of a job on a machine is drawn from a distribution. A *schedule* is an assignment of each job to exactly one machine. The *load* of a machine is the number of jobs assigned to it. The *load factor* is the average number of jobs per machine and is denoted $\eta = n/m$. The *work* of a machine is the sum of the runtimes of jobs assigned to it. The *total work* is the sum of the works of each machine. The *makespan* is the most work assigned to any machine.

The vector of running times for each of the jobs on a given machine is that machine's private information. A scheduling mechanism may solicit this information from the machines, may make payments to the machines, and must select a schedule of jobs on the machines. A scheduling mechanism is evaluated in the equilibrium of strategic behavior of the machines. A particularly robust equilibrium concept is dominant strategy equilibrium. A scheduling mechanism is *incentive compatible* if it is a dominant strategy for each machine to report its true processing time for each job.

We consider the following simple mechanisms:

minimum work The minimum work mechanism solicits the running times, selects the schedule to minimize the total work and pays each machine its externality, i.e., the difference between the minimum total work when the machine does nothing and the total work of all other machines in the selected schedule.

bounded overload The bounded overload mechanism is parameterized by an overload factor $c > 1$ and is identical to the minimum work mechanism except it optimizes subject to placing at most $c\eta$ jobs on any machine.

sieve / anonymous reserve The sieve mechanism, also known as the anonymous reserve mechanism, is parameterized by a reserve $\beta \geq 0$ and is identical to the minimum work mechanism except that there is a dummy machine added with runtime β for all jobs. Jobs assigned to the dummy machine are considered unscheduled.

sieve and bounded overload The sieve and bounded overload mechanism is parameterized by overload c, reserve β, and a partition parameter δ. It partitions the machines into two sets of sizes $(1-\delta)m$ and δm. It runs the sieve with reserve β on the first set of machines and runs the bounded overload mechanism with overload c on the unscheduled jobs and the second set of machines.

The above mechanisms are incentive compatible. The minimum work mechanism is incentive compatible as it is a special case of the well known Vickrey-Clarke-Groves (VCG) mechanism which is incentive compatible. The bounded overload mechanism is what is known as a "maximal in range" mechanism and is also incentive compatible (by the VCG argument). The sieve / anonymous reserve mechanism is incentive compatible because the incentives of the agents in the minimum work mechanism are unaffected by the addition of a dummy agent. Finally, the sieve and bounded overload mechanism is incentive compatible because from each machine's perspective it is either participating in the sieve mechanism or the bounded overload mechanism.

The runtimes of jobs on machines are drawn from a product distribution (a.k.a., the *prior*) that is symmetric with respect to the machines. (Therefore, the running times of a job on each machine are i.i.d. random variables.) The distribution of job j on any machine is denoted F_j; a draw from this distribution is denoted T_j. The *best runtime* of a job is its minimum runtime over all machines, this first order statistic of m random draws from F_j is denoted by $T_j[1{:}m]$.

Our goal is to exhibit a mechanism that is prior-independent and a good approximation to the expected makespan of the best incentive compatible mechanism for the prior, i.e., the second-best solution. Because both the second-best and the first-best expected makespans are difficult to analyze, we will give our approximation via one of the following two lower bounds on the first-best solution.

expected worst best runtime The expected worst best runtime is the expected value of the best runtime of the job with the longest best runtime, i.e., $\mathbf{E}[\max_j T_j[1{:}m]]$

expected average best runtime The expected average best runtime is the expected value of the sum of the best runtimes of each job averaged over all machines, i.e., $\mathbf{E}[\sum_j T_j[1{:}m]]/m$.

Intuitively, the former gives a good bound when the load factor is small, the latter when the load factor is large. We will refer to any of these bounds on the first-best makespan as OPT, with the assumption that which of the bounds is meant, if it is important, is clear from the context.

As an intermediary in our analysis of the makespan of our scheduling mechanisms with respect to OPT, we will give bicriteria results that compare our mechanism's makespan to the makespan of an optimal schedule with fewer machines. This restriction is well defined because the machines are a prior identical. For a given parameter δ, OPT_δ will denote the optimal schedule with δm machines (via bounds as described above). Much of our analysis will be with respect to $\mathrm{OPT}_{1/2}$, i.e., the optimal schedule with half the number of machines.

While it is possible to construct distributions where OPT is much smaller than $\mathrm{OPT}_{1/2}$, for many common distributions they are quite close. In fact, for the class of distributions that satisfy the monotone hazard rate (MHR) condition,[1] OPT and $\mathrm{OPT}_{1/2}$ are always within a factor of four; more generally OPT and OPT_δ are within a factor of $1/\delta^2$ for these distributions. (See proof in Section 4.5.)

Lemma 4.1 *When the distributions of job sizes have monotone hazard rates the expected worst best and average best runtimes on δm machines are no more than $1/\delta^2$ times the expected worst best and average best runtimes, respectively, on m machines.*

4.2.1 Main Results

Our main theorems are as follows. When the number of jobs is comparable to the number of machines, i.e., the load factor η is constant, then the bounded overload mechanism is a good approximation to the optimal makespan on $m/2$ machines.

Theorem 4.2 *For n jobs, m machines, load factor $\eta = n/m$, and runtimes distributed according to a machine-symmetric product distribution, the expected makespan of the bounded overload mechanism with overload $c = 7$ is a 200η approximation to the expected worst best runtime, and hence also to the optimal makespan, on $m/2$ machines.*

Corollary 4.3 *Under the assumptions of Theorem 4.2 where additionally the distributions of job sizes have monotone hazard rates, the expected makespan of the bounded overload mechanism with $c = 7$ is a 800η approximation to the expected optimal makespan.*

---

[1]The hazard rate of a distribution F is given by $h(x) = \frac{f(x)}{1-F(x)}$, where f is the probability density function for F; a distribution F satisfies the MHR condition if $h(x)$ is non-decreasing in x. Many natural distributions such as the uniform, Gaussian, and exponential distributions, satisfy the monotone hazard rate condition. Intuitively, these are distributions with tails no heavier than the exponential distribution.

When the load factor η is large and the job runtimes are identically distributed, the sieve and bounded overload mechanism is a good approximation to the optimal makespan. The following theorems and corollaries demonstrate the sieve and bounded overload mechanism under two relevant parameter settings.

Theorem 4.4 *For n jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 2/3$, and reserve $\beta = \frac{n}{m \log m} \mathbf{E}[T[1{:}\frac{\delta}{2}m]]$ is an $O(\sqrt{\log m})$ approximation to the larger of the expected worst best and average best runtime, and hence also to the optimal makespan, on $m/3$ machines. Here T denotes a draw from the distribution on job sizes.*

Corollary 4.5 *Under the assumptions of Theorem 4.4 where additionally the distribution of job sizes has monotone hazard rate, the expected makespan of the sieve and bounded overload mechanism is an $O(\sqrt{\log m})$ approximation to the expected optimal makespan.*

Theorem 4.6 *For $n \geq m \log m$ jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 1/\log\log m$, and reserve $\beta = \frac{2n}{m \log m} \mathbf{E}[T[1{:}\frac{\delta}{2}m]]$, is a constant approximation to the larger of the expected worst best and average best runtime, and hence also to the optimal makespan, on $\delta m/2$ machines. Here T denotes a draw from the distribution on job sizes.*

Corollary 4.7 *Under the assumptions of Theorem 4.6 where additionally the distribution of job sizes has monotone hazard rate the expected makespan of the sieve and bounded overload mechanism is a $O((\log\log m)^2)$ approximation to the expected optimal makespan.*

We prove Theorem 4.2 in Section 4.3 and Theorems 4.4 and 4.6 in Section 4.4.

4.2.2 Probabilistic Analysis

Our goal is to show that the simple processes described by the bounded overload and sieve mechanisms result in good makespan and our upper bound on makespan is given by the first order statistics of each job's runtime across the machines. The sieve's performance analysis is additionally governed by the law of large numbers. We describe here basic facts about order statistics and concentration bounds. Additionally we give a number of new bounds, proofs of which are in Section 4.5.

For random variable X and integer k, we consider the following basic constructions of k independent draws of the random variable. The ith order statistic, or the ith minimum of k draws, is denoted $X[i{:}k]$. The first order statistic, i.e., the minimum of the k draws, is denoted $X[1{:}k]$. The kth order statistic, i.e., the maximum of k draws, is denoted $X[k{:}k]$. Finally, the sum of k draws is denoted $X[\Sigma k]$. We include the possibility that i or k can be random variables. We also allow the notation to cascade, e.g., for the special case where the jobs are i.i.d. from F the lower bounds on OPT are $T[1{:}m][n{:}n]$ and $T[1{:}m][\Sigma n]/m$ for the expected worst best and average best runtime, respectively, and T drawn from F.

We will use the following forms of Chernoff-Hoeffding bounds in this paper. Let $X = \sum_i X_i$, where $X_i \in [0, B]$ are independent random variables. Then, for all $\epsilon \geq 1$,

$$\mathbf{Pr}[X > (1+\epsilon)\,\mathbf{E}[X]] < \exp\left(\tfrac{-\epsilon\,\mathbf{E}[X]}{3B}\right) < \exp\left(\tfrac{-(1+\epsilon)\,\mathbf{E}[X]}{6B}\right)$$

Our analysis often involves relating different order statistics of a random variable (e.g. how does the size of a job on its best machine compare to that on its second best machine). We relate these different order statistics via the stochastic dominance relation. This is useful in our analysis because stochastic dominance is preserved by the max and sum operators. We say that a random variable X is stochastically dominated by another random variable Y if for all t, $\mathbf{Pr}[X \leq t] \geq \mathbf{Pr}[Y \leq t]$. Stochastic dominance is equivalent to being able to couple the two random variables X and Y so that X is always smaller than Y.

Below, the first lemma relates the ith order statistic over some number of draws to the first order statistic over half the draws. The second relates the minimum over several draws of a random variable to a single draw of that variable. The third relates the maximum over multiple draws of a random variable to an appropriate sum over those draws. These lemmas are proved in Section 4.5.

Lemma 4.8 *Let X be any nonnegative random variable and m and $i \leq m$ be arbitrary integers. Let α be defined such that $\mathbf{Pr}[X \leq \alpha] = 1/m$ (or for discontinuous distributions, $\alpha = \sup\{z : \mathbf{Pr}[X \leq z] < 1/m\}$). Then $X[i{:}m]$ is stochastically dominated by* $\max(\alpha, X[1{:}m/2][4^i{:}4^i])$.

Lemma 4.9 *For a random variable X whose distribution satisfies the monotone hazard rate condition, X is stochastically dominated by $rX[1{:}r]$.*

Lemma 4.10 *Let $K_1, \cdots, K_n$ be independent and identically distributed integer random variables such that for some constant $c > 1$, we have $K_j \geq c$, and let $W_1, \cdots, W_n$ be arbitrary independent nonnegative variables. Then,*

$$\mathbf{E}\left[\max_j W_j[K_j{:}K_j]\right] \leq \tfrac{c}{c-1}\,\mathbf{E}\left[K_1\right]\mathbf{E}\left[\max_j W_j\right].$$

We will analyze the expected makespan of a mechanism as the maximum over a number of correlated real-valued random variables. The correlation among these variables makes it difficult to understand and bound the makespan. Our approach will be to replace these random variables with an ensemble of independent random variables that have the same marginal distributions. Fortunately, this operation does not change the expected maximum by too much. Our next lemma relates the expected maximum over an arbitrary set of random variables to the expected maximum over a set of independent variables with the same marginal distributions. It is a simple extension of the correlation gap results of Aggarwal et al. [ADSY10], Yan[Yan11], and Chawla et al. [CHMS10].

Lemma 4.11 *Let $X_1, \cdots, X_n$ be arbitrary correlated real-valued random variables. Let $Y_1, \cdots, Y_n$ be independent random variables defined so that the distribution of Y_i is identical to that of X_i for all i. Then, $\mathbf{E}[\max_j X_j] \leq \frac{e}{e-1}\,\mathbf{E}[\max_j Y_j]$.*

4.3 The Bounded Overload Mechanism

Recall that the bounded overload mechanism minimizes the total work subject to the additional feasibility constraint that every machine is assigned at most $c\eta$ jobs. In this section we prove that the expected makespan of the bounded overload mechanism, with the overload set to $c = 7$, is a 200η factor approximation to the expected best worst runtime and thus to the optimal makespan.

Intuitively the bounded overload mechanism tries to achieve two objectives simultaneously: (1) keep the size of every job on the machine its schedule to be close to its size on its best machine, but also (2) evenly distribute the jobs across all the machines. Recall, that the minimum work mechanism achieves the first objective exactly, but fails on the second objective. Due to the independence between jobs, the number of jobs on each machine may be quite unevenly distributed. In contrast, the bounded overload mechanism explicitly disallows uneven assignments of jobs and therefore the main issue to address in its analysis is whether it satisfies the first objective, i.e., that the sizes of the jobs are close to what they are in the minimum work mechanism.

To setup for the proof of Theorem 4.2 consider the following definitions that describe the outcome of the bounded overload mechanism and the worst best runtime on $m/2$ machines (which bounds the optimal makespan on $m/2$ machines). Let T_j denote a random variable drawn according to job j's distribution of runtimes F_j. Let B_j denote the job's best runtime out of $m/2$ machines, i.e., $B_j = T_j[1{:}m/2]$, the first order statistic of $m/2$ draws. The expected worst best runtime on $m/2$ machines is $\mathbf{E}[\max_j B_j]$. The bounded overload mechanism considers placing each job on one of m machines. These runtimes of job j drawn i.i.d. from F_j impose a (uniformly random) ordering over the machines starting from the machine that is "best" for j to the one that is "worst"; this is j's preference list. Let $T_j[r{:}m]$ denote the size of job j on the rth machine in this ordering (also called the job's rth favorite machine). Let R_j be a random variable to denote the rank of the machine that job j is placed on by the bounded overload mechanism. As each machine is constrained to receive at most $c\eta$ jobs, the expected makespan of bounded overload is $c\eta\,\mathbf{E}[\max_j T_j[R_j{:}m]]$. We will bound this quantity in terms of $\mathbf{E}[\max_j B_j]$.

There are three main parts to our argument. First, we note that the R_js are correlated across different j's, and so are the $T_j[R_j{:}m]$s. This makes it challenging to directly analyze $\mathbf{E}[\max_j T_j[R_j{:}m]]$. We use Lemma 4.11 to replace the R_js in this expression by independent random variables with the same marginal distributions. We then show that the marginal distributions can be bounded by simple geometric random variables $\tilde{R}_j$. To do so, we introduce another procedure for assigning jobs to machines that we call the last entry procedure. The assignment of each job under the last entry procedure is no better than its assignment under bounded overload. On the other hand, the ranks of the machines to which jobs are allocated in the last entry procedure are geometric random variables with a bounded failure rate. Finally, we relate the runtimes $T_j[\tilde{R}_j{:}m]$ to the optimal runtimes B_j using Lemma 4.8.

We begin by describing the last entry procedure.

last entry In order to schedule job j, we first apply the bounded overload mechanism BO_c to all jobs other than j. We then place j on the first machine in its preference list that has fewer than $c\eta$ jobs. Let L_j denote the rank of the machine to which j gets allocated.

We now make a few observations about the ranks L_j realized by the last entry procedure.

Lemma 4.12 *The runtime of any job j in bounded overload is no worse than its runtime in the last entry procedure. That is, $R_j \leq L_j$.*

Proof: Fix any instantiation of jobs' runtimes over machines. Consider the assignment of job j in the last entry procedure, and let LE(j) denote the schedule where all of the jobs but j are scheduled according to bounded overload and j is scheduled according to the last entry procedure. Since the bounded overload mechanism minimizes total work, the total runtime of all of the jobs in BO_c is no more than the total runtime of all of the jobs in LE(j). On the other hand, the total runtime of all jobs except j in LE(j) is no more than the total runtime of all jobs except j in BO_c. This immediately implies that j's runtime in bounded overload is no more than its runtime in last entry. Since this holds for any fixed instantiation of runtimes, we have $R_j \leq L_j$. ■

Next, we show that the rank L_j of a job j in last entry is stochastically dominated by a geometric random variable $\tilde{R}_j$ that is capped at $\lceil \frac{m}{c} \rceil$. Note that L_j is at most $\lceil \frac{m}{c} \rceil$ since $\lceil \frac{m}{c} \rceil$ machines can accommodate $\lceil \frac{m}{c} \rceil c\eta \geq n$ jobs and therefore last entry will never have to send a job to anything worse than its $\lceil \frac{m}{c} \rceil$th favorite machine. The random variable $\tilde{R}_j$ also lives in $\{1, \ldots, \lceil \frac{m}{c} \rceil\}$, and is drawn independently for all j as follows: for $i \in \{1, \ldots, \lceil \frac{m}{c} \rceil - 1\}$, we have $\mathbf{Pr}[\tilde{R}_j = i] = \frac{1-1/c}{c^{i-1}}$; and the remaining probability mass is on $\lceil \frac{m}{c} \rceil$.

Lemma 4.13 *The rank L_j of a job j in last entry is stochastically dominated by $\tilde{R}_j$, and so the runtime of job j in last entry is stochastically dominated by $T_j[\tilde{R}_j{:}m]$.*

Proof: We use the principle of deferred decisions. In order to schedule j, the last entry procedure first runs bounded overload on all of the jobs other than j. This produces a schedule in which at most a $\frac{1}{c}$ fraction of the machines have all of their slots occupied. Conditioned on this schedule, job j's preference list over machines is a uniformly random permutation. So the probability (over the draw of j's runtimes) that job j's favorite machine is fully occupied is at most $1/c$. Likewise, the probability that the job's two most favorite machines are both occupied is at most $1/c^2$, and so on.

Therefore, the rank of the machine on which j is eventually scheduled is dominated by a geometric random variable with failure rate $1/c$. ∎

Lemmas 4.12 and 4.13 yield the following corollary.

Corollary 4.14 *For all j, the runtime $T_j[R_j{:}m]$ of job j in bounded overload is stochastically dominated by $T_j[\tilde{R}_j{:}m]$.*

The benefit of relating $T_j[R_j{:}m]$s with $T_j[\tilde{R}_j{:}m]$s is that while the former are correlated random variables, the latter are independent, because the $\tilde{R}_j$'s are picked independently. Corollary 4.14 implies that we can replace the former with the latter, gaining independence, while losing only a constant factor in expected makespan.

Corollary 4.15 $\mathbf{E}[\max_j T_j[R_j{:}m]]$ *is no more than $e/(e-1)$ times* $\mathbf{E}[\max_j T_j[\tilde{R}_j{:}m]]$.

The final part of our analysis relates the $T_j[\tilde{R}_j{:}m]$s to the B_js. A natural inequality to aim for is to bound $\mathbf{E}[T_j[\tilde{R}_j{:}m]]$ from above by a constant times $\mathbf{E}[B_j]$ for each j. Unfortunately, this is not enough for our purposes: note that our goal is to upper bound $\mathbf{E}[\max_j T_j[\tilde{R}_j{:}m]]$ in terms of $\mathbf{E}[\max_j B_j]$. Thus we proceed to show that $T_j[\tilde{R}_j{:}m]$ is stochastically dominated by a maximum among some number of copies of B_j. We apply Lemma 4.8 (stated in Section 4.2 and proved in Section 4.5) to the random variable $T_j[i{:}m]$ for this purpose. Define $\alpha_j = \sup\{t : F_j(t) < 1/m\}$. Then the lemma shows that $T_j[i{:}m]$ is stochastically dominated by $\max(\alpha_j, B_j[4^i{:}4^i])$.

Let D_j be defined as $4^{\tilde{R}_j}$. Note that $\mathbf{E}[D_j]$ can be bounded by a constant whenever $c > 4$ (this upper bound is obtained by treating $\tilde{R}_j$ as a geometric random variable without being capped at $\lceil \frac{m}{c} \rceil$). Then Lemma 4.8 implies the following corollary.

Lemma 4.16 *$T_j[\tilde{R}_j{:}m]$ is stochastically dominated by* $\max(\alpha_j, B_j[D_j{:}D_j])$.

We are now ready to prove the main theorem of this section.

Theorem 4.2 *For n jobs, m machines, load factor $\eta = n/m$, and runtimes distributed according to a machine-symmetric product distribution, the expected makespan of the bounded overload mechanism with overload $c = 7$ is a 200η approximation to the expected worst best runtime, and hence also to the optimal makespan, on $m/2$ machines.*

Proof: The proof follows from the following series of inequalities that we explain below. First we have Makespan(BO$_c$) $\leq c\eta\, \mathbf{E}\left[\max_j T_j[R_j{:}m]\right]$ by the fact that BO$_c$ schedules at most $c\eta$ jobs per machine

$$\begin{aligned}
\frac{e-1}{e}\,\mathbf{E}\left[\max_j T_j[R_j{:}m]\right] &\leq \mathbf{E}\left[\max_j T_j[\tilde{R}_j{:}m]\right] \\
&\leq \mathbf{E}\left[\max_j(\max(\alpha_j, B_j[D_j{:}D_j]))\right] \\
&\leq \mathbf{E}\left[\max_j\left(\alpha_j + B_j[D_j{:}D_j]\right)\right] \\
&\leq \max_j \alpha_j + \mathbf{E}\left[\max_j B_j[D_j{:}D_j]\right] \\
&\leq 2\mathrm{OPT}_{1/2} + \tfrac{4}{4-2}\,\mathbf{E}[D_j]\,\mathbf{E}[\max_j B_j] \\
&\leq \left(2 + 8\frac{c-1}{c-4}\right)\mathrm{OPT}_{1/2}.
\end{aligned}$$

The first of the inequalities follows from Lemma 4.15, the second from Lemma 4.16, the third from noting that the maximum of non-negative random variables is upper bounded by their sum, and the last by the definition of $\mathrm{OPT}_{1/2}$, along with the fact that $\mathbf{E}[D_j] \leq 4\frac{c-1}{c-4}$. For the fifth inequality we use Lemma 4.10 to bound the second term. For the first term in that inequality consider the job j that has the largest α_j. For this job, the probability that its size on all of the $m/2$ machines in $\mathrm{OPT}_{1/2}$ is at least α_j is $(1 - F_j(\alpha_j))^{m/2} \geq (1 - 1/m)^{m/2} \geq 1/2$ by the definition of α_j. So $\mathrm{OPT}_{1/2} \geq \max_j \alpha_j/2$.

The final approximation factor therefore is $c\eta\frac{e}{e-1}\left(2 + 8\frac{c-1}{c-4}\right)$ for all $c > 4$. At $c = 7$, this evaluates to a factor 200η approximation. ■

4.4 The Sieve and Bounded Overload Mechanism

We will now analyze the performance of the sieve and bounded overload mechanisms under the assumption that the jobs are a priori identical. Let us consider the sieve mechanism first. Recall that this is essentially the minimum work mechanism where every job is assigned to its best machine, except that jobs with a size larger than β on every machine are left unscheduled. The bound of β on the size of scheduled jobs allows us to employ concentration results to bound the expected makespan of the mechanism. Changing the value of β allows us to tradeoff the makespan of the mechanism with the number of unscheduled jobs.

Lemma 4.17 *For $k < \log m$, the expected makespan of the sieve mechanism with $\beta = \frac{n\,\mathbf{E}[T[1{:}m]]}{km}$ is no more than $O(\log m/k)$ times the expected average best runtime, and hence also the expected optimal makespan. The expected number of jobs left unscheduled by the mechanism is km.*

Proof: Let us first consider the expected total work of any single machine, that is the expected total size of jobs scheduled on that machine. Let Y_{ij} be a random variable that takes on the value 0 if job j is not scheduled on machine i, and takes on the size of j on machine i if the job is scheduled on that machine. The probability that j is scheduled on i is no more than $1/m$; its expected size on i conditioned on being scheduled is at most $\tau = \mathbf{E}[T[1{:}m]]$. Therefore, $\mathbf{E}[\sum_j Y_{ij}] \le \frac{n\tau}{m}$, which in turn is at most the average best runtime.

Note that the Y_{ij}'s are independent and bounded random variables. So we can apply Chernoff-Hoeffding bounds and use $\beta = \frac{n\tau}{km}$ to get

$$\begin{aligned}\mathbf{Pr}\left[\sum\nolimits_j Y_{ij} > \tfrac{7\log m}{k}\mathrm{OPT}\right] &\le \mathbf{Pr}\left[\sum\nolimits_j Y_{ij} > \tfrac{7\log m}{k}\,\mathbf{E}\left[\sum\nolimits_j Y_{ij}\right]\right] \\ &< \exp\left(-\tfrac{1}{3}\tfrac{6\log m}{k}\tfrac{n\tau}{\beta m}\right) = \tfrac{1}{m^2}.\end{aligned}$$

Taking the union bound over the m machines, we get that with probability $1 - 1/m$, the makespan of the sieve mechanism is at most $O(\log m/k)$ times OPT.

We will now convert this tail probability into a bound on the expected makespan. Let γ denote the factor by which the expected makespan of the mechanism exceeds OPT. Remove all jobs with best runtimes greater than β from consideration and consider creating sieve's schedule by assigning each of the leftover jobs to their best machine (minimizing total work) one-by-one in decreasing order of best runtime, until the makespan exceeds $\frac{7}{k}\log m$ times OPT. This event happens with a probability at most $1/m$. When this event happens, we are left with a smaller set of jobs; conditioned on being left over at this point, these jobs have a smaller best runtime than the average over all scheduled jobs. Thus the expected makespan for scheduling them will be at most γOPT. So we get $\gamma \le 7\log m/k + \gamma/m$, i.e., $\gamma = O(\log m/k)$. This implies the first part of the lemma.

We now prove the second part of the lemma, i.e., the expected number of jobs left unscheduled is km. Note that β exceeds a job's expected best runtime by a factor of n/km. Thus by applying Markov's inequality, we get the probability of a job's best runtime being larger than β to be at most km/n. Hence the expected number of jobs with best runtime larger than β is km. ■

Next we will combine the sieve mechanism with the bounded overload mechanism. We consider two different choices of parameters. Note that if in expectation the sieve mechanism leaves km jobs unscheduled, using the bounded overload mechanism to schedule these jobs over a set of $\Omega(m)$ machines gives us an expected makespan that is at most $O(k)$ larger than the expected optimal makespan on that number of machines. In order to balance this with the makespan achieved by sieve, we pick $k = \sqrt{\log m}$. This gives us Theorem 4.4.

Theorem 4.4 *For n jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 2/3$, and reserve $\beta = \frac{n}{m \log m}\,\mathbf{E}[T[1{:}\frac{\delta}{2}m]]$, is an $O(\sqrt{\log m})$ approximation to the larger of the worst best runtime and the average best runtime, and hence also to the optimal makespan, on $m/3$ machines. Here T denotes a draw from the distribution on job sizes.*

Proof: For the choice of parameters in the theorem statement, we use $m/3$ of the m machines for the sieve mechanism, and the remainder for the bounded overload mechanism. The expected makespan of the overall mechanism is no more than the sum of the expected makespans of the two constituent mechanisms. Lemma 4.17 implies that the expected makespan of the sieve mechanism is $O(\sqrt{\log m})$ times $\mathrm{OPT}_{1/3}$, and the load factor for the bounded overload mechanism is also $O(\sqrt{\log m})$. Theorem 4.2 then implies that the expected makespan of the bounded overload mechanism is also $O(\sqrt{\log m})$ times $\mathrm{OPT}_{1/3}$. ■

If we partition the machines across the sieve and the bounded overload mechanisms roughly equally, then Theorem 4.4 gives us the optimal choice for the parameter β. A different possibility is to perform a more aggressive screening of jobs by using a smaller β, while comparing our performance against a more heavily penalized optimal mechanism – one that is allowed to use only a δ fraction of the machines.

Theorem 4.6 *For $n \geq m \log m$ jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 1/\log\log m$, and reserve $\beta = \frac{2n}{m \log m} \mathbf{E}[T[1{:}\frac{\delta}{2}m]]$, is a constant approximation to the larger of the worst best runtime and the average best runtime, and hence also to the optimal makespan, on $\delta m/2$ machines. Here T denotes a draw from the distribution on job sizes.*

Proof: We will show that the expected makespan of the sieve mechanism is at most a constant times the average best runtime on $\delta m/2$ machines, and the expected number of unscheduled jobs is $O(\delta m)$. The current theorem then follows by applying Theorem 4.2.

Let us analyze the expected makespan of the sieve mechanism first. Let $\tau = \mathbf{E}[T[1{:}\frac{\delta}{2}m]]$. Then we can bound $\mathrm{OPT}_{\delta/2}$ as $\mathrm{OPT}_{\delta/2} \geq \frac{2n\tau}{\delta m}$. As in the proof of Lemma 4.17, let Y_{ij} be a random variable that takes on the value 0 if job j is not scheduled on machine i, and takes on the size of j on machine i if the job is scheduled on that machine. Then,

$$\mathbf{E}\left[\sum\nolimits_j Y_{ij}\right] \leq \tfrac{n}{(1-\delta)m} \mathbf{E}[T[1{:}(1-\delta)m]] \leq \tfrac{n\tau}{(1-\delta)m} \leq \tfrac{\delta}{2(1-\delta)} \mathrm{OPT}_{\delta/2}.$$

Applying Chernoff-Hoeffding bounds we get

$$\begin{aligned}\mathbf{Pr}\left[\sum\nolimits_j Y_{ij} > 2\mathrm{OPT}_{\delta/2}\right] &\leq \mathbf{Pr}\left[\sum\nolimits_j Y_{ij} > 4(1/\delta - 1)\,\mathbf{E}\left[\sum\nolimits_j Y_{ij}\right]\right] \\ &< \exp\left(-\tfrac{1}{\delta}\tfrac{n\tau}{(1-\delta)m}\tfrac{1}{\beta}\right) \leq m^{-1/2\delta}.\end{aligned}$$

Here we used $\beta = 2n\tau/m \log m$. Taking the union bound over the m machines, we get that with probability $o(1)$, the makespan of the sieve mechanism is at most twice $\mathrm{OPT}_{\delta/2}$. Once again, as in the proof of Lemma 4.17 we can convert this tail bound into a constant factor bound on the expected makespan.

Now let us consider the jobs left unscheduled. For any given job, we will compute the probability that its runtime on all of the $(1-\delta)m$ machines is larger than β. Because β is defined in terms of $T[1{:}\frac{\delta}{2}m]$, we will consider the machines in batches of size $\delta m/2$ at a time. Using Markov's inequality, the probability that the job's runtime exceeds β on all machines in a single batch is at most $m \log m/2n$. There are $2(1/\delta - 1)$ batches in all, so the probability that a job remains unscheduled is at most $(m \log m/n)(2^{2(1-1/\delta)})$, which by our choice of δ is $O(\delta m/n)$. ■

4.5 Deferred Proofs

In this section we prove the bounds for random variables and order statistics from Section 4.2.2.

Lemma 4.8 *Let X be any nonnegative random variable, and m, $i \leq m$ be arbitrary integers. Let α be defined such that $\mathbf{Pr}[X \leq \alpha] = 1/m$ (or for discontinuous distributions, $\alpha = \sup\{z : \mathbf{Pr}[X \leq z] < 1/m\}$). Then $X[i{:}m]$ is stochastically dominated by* $\max(\alpha, X[1{:}m/2][4^i{:}4^i])$.

Proof: Let F be the cumulative distribution function of X. We prove this by showing that $X[i{:}m]$ is "almost" stochastically dominated by $X[1{:}m/2][4^i{:}4^i]$; specifically, we show that for all $t \geq \alpha$,

$$\mathbf{Pr}\left[X[i{:}m] > t\right] \leq \mathbf{Pr}\left[X[1{:}m/2][4^i{:}4^i] > t\right].$$

To prove this inequality, we will define a process for instantiating the variables $X[i{:}m]$ and $X[1{:}m/2][4^i{:}4^i]$ in a correlated fashion such that the former is always larger than the other.

$X[1{:}m/2][4^i{:}4^i]$ is a statistic based on $4^i m/2$ independent draws of the random variable X. Consider partitioning these draws into $4^i/2$ groups of size m each. We then randomly split each group into two smaller groups, which we will refer to as blocks, of size $m/2$ each. Define a good event $\mathcal{G}$ to be the event that at least one of these $4^i/2$ groups get split such that the i smallest runtimes in it all fall into the same block. If event $\mathcal{G}$ occurs, arbitrarily choose one group which caused event $\mathcal{G}$, and for all k define $X[k{:}m]$ to be the kth min from this group. Otherwise, select an arbitrary group to define the $X[k{:}m]$. Note that since we split the groups into blocks randomly, and this is independent of the drawn runtimes in the groups, $X[k{:}m]$ has the correct distribution, both when $\mathcal{G}$ occurs and does not occur. Define the minimum from each of the 4^i blocks to be a draw of $X[1{:}m/2]$. Thus, whenever $\mathcal{G}$ occurs, the probability that the $X[1{:}m/2][4^i{:}4^i] > t$ is at least the probability that $X[i+1{:}m] > t$. We have that

$$\begin{aligned}
\mathbf{Pr}\left[X[1{:}m/2][4^i{:}4^i] > t\right] &\geq \mathbf{Pr}\left[\mathcal{G}\right] \cdot \mathbf{Pr}\left[X[i+1{:}m] > t\right] \\
&= \left(\mathbf{Pr}\left[\mathcal{G}\right] \cdot \frac{\mathbf{Pr}\left[X[i+1{:}m] > t\right]}{\mathbf{Pr}\left[X[i{:}m] > t\right]}\right) \cdot \mathbf{Pr}\left[X[i{:}m] > t\right].
\end{aligned}$$

We now show that $\left(\mathbf{Pr}\left[\mathcal{G}\right] \cdot \frac{\mathbf{Pr}[X[i+1:m]>t]}{\mathbf{Pr}[X[i:m]>t]}\right) \geq 1$ whenever $F(t) \geq 1/m$, which completes our proof of the lemma. Note that

$$\frac{\mathbf{Pr}\left[X[i+1:m] > t\right]}{\mathbf{Pr}\left[X[i:m] > t\right]} = \frac{\sum_{k=0}^{i} \binom{m}{k} F(t)^k (1-F(t))^{m-k}}{\sum_{k=0}^{i-1} \binom{m}{k} F(t)^k (1-F(t))^{m-k}}$$
$$= 1 + \frac{\binom{m}{i} F(t)^i (1-F(t))^{m-i}}{\sum_{k=0}^{i-1} \binom{m}{k} F(t)^k (1-F(t))^{m-k}},$$

which we can see is an increasing function of $F(t)$. Thus in the range $F(t) \geq 1/m$, it attains its minimum precisely at $F(t) = 1/m$. Substituting $F(t) = 1/m$ into the above, and using standard approximations for $\binom{m}{k}$ (namely $\left(\frac{m}{k}\right)^k \leq \binom{m}{k} \leq \left(\frac{me}{k}\right)^k$, we have

$$\frac{\mathbf{Pr}\left[X[i+1:m] > t\right]}{\mathbf{Pr}\left[X[i:m] > t\right]} \geq 1 + \frac{\left(\frac{m}{i}\right)^i \left(\frac{1}{m}\right)^i \left(1-\frac{1}{m}\right)^{m-i}}{\left(1-\frac{1}{m}\right)^m + \sum_{k=1}^{i-1} \left(\frac{me}{k}\right)^k \left(\frac{1}{m}\right)^k \left(1-\frac{1}{m}\right)^{m-k}}$$
$$\geq 1 + \frac{i^{-i}}{1 + (i-1)\cdot \max_k (\frac{e}{k})^k} \geq 1 + \frac{i^{-i}}{1+(i-1)e}.$$

It suffices to show that this last quantity, when multiplied with $\mathbf{Pr}[\mathcal{G}]$, is at least 1. We consider the complement of event $\mathcal{G}$, call it even $\mathcal{B}$. The event $\mathcal{B}$ occurs only when none of the $4^i/2$ groups split favorably. The probability that a group splits favorably (for $i \geq 1$) is $2 \cdot \binom{m-i}{m/2-i} \Big/ \binom{m}{m/2} \geq 2^{-(i-1)}$. So we can see that $\mathbf{Pr}[\mathcal{B}] \leq (1-2^{-(i-1)})^{4^i/2} \leq e^{-(4/2)^i}$, and thus $\mathbf{Pr}[\mathcal{G}] \geq 1 - e^{-(4/2)^i}$. It can be verified that $(1-e^{-(4/2)^i}) \cdot \left(1 + \frac{i^{-i}}{1+(i-1)e}\right) \geq 1$. ■

Lemma 4.9 *For a random variable X whose distribution satisfies the monotone hazard rate condition, X is stochastically dominated by $rX[1:r]$.*

Proof: The hazard rate function is related to the cumulative distribution function as $\mathbf{Pr}[X \geq t] = e^{-\int_0^t h(z)\,dz}$. Likewise, we can write:

$$\mathbf{Pr}[rX[1:r] \geq t] = \mathbf{Pr}[X[1:r] \geq t/r] = (e^{-\int_0^{t/r} h(z)\,dz})^r = e^{-r\int_0^{t/r} h(z)\,dz}.$$

In order to prove the lemma, we need only show that $\int_0^t h(z)\,dz \geq r \cdot \int_0^{t/r} h(z)\,dz$. Since the hazard rate function $h(z)$ is monotone, the function $\int_0^t h(z)\,dz$ is a convex function of t. The required inequality follows from the definition of convexity. ■

Lemma 4.10 *Let $K_1, \cdots, K_n$ be independent and identically distributed integer random variables such that for some constant $c > 1$, we have $K_j \geq c$ for all j, and let $W_1, \cdots, W_n$ be arbitrary independent nonnegative variables. Then,*

$$\mathbf{E}\left[\max_j W_j[K_j{:}K_j]\right] \leq \tfrac{c}{c-1}\,\mathbf{E}\left[K_1\right]\mathbf{E}\left[\max_j W_j\right].$$

Proof: We consider the following process for generating correlated samples for $\max_j W_j$ and $\max_j W_j[K_j{:}K_j]$. We first independently instantiate K_j for every j; recall that these are identically distributed variables. Let $k = \sum_j K_j \geq cn$. Then we consider all possible $n!$ permutations of these instantiated values. For each permutation σ, we make the corresponding number of independent draws of the random variable W_j for all j; call this set of draws X_σ. In all, we get $kn!$ draws from the distributions, that is, $|\cup_\sigma X_\sigma| = kn!$. Exactly $k(n-1)!$ of these draws belong to any particular j; denote these by Y_j.

Now, the maximum element out of each of the X_σ sets is an independent draw from the same distribution $\max_j W_j[K_j{:}K_j]$ is drawn from. We get $n!$ independent samples from that distribution. Call this set of samples X.

Next note that each set Y_j contains $k(n-1)!$ independent draws from the distribution corresponding to W_j. We construct a uniformly random n-dimensional matching over the sets Y_j, and from each n-tuple in this matching we pick the maximum. Each such maximum is an independent draw from the distribution corresponding to $\max_j W_j$, and we get $k(n-1)!$ such samples; call this set of samples Y.

Finally, we claim that $\mathbf{E}[\sum_{y\in Y} y] \geq (1-1/c)\,\mathbf{E}[\sum_{x\in X} x]$, with the expectation taken over the randomness in generating the n-dimensional matching across the Y_js. The lemma follows, since we have $\mathbf{E}[\sum_{x\in X} x] = n!\,\mathbf{E}[\max_j W_j[K_j{:}K_j]]$ as well as

$$\mathbf{E}[\sum_{y\in Y} y] = \mathop{\mathbf{E}}_{\{K_j\}}[k(n-1)!\,\mathbf{E}[\max_j W_j]] = n!\,\mathbf{E}[K_j]\,\mathbf{E}[\max_j W_j].$$

To prove the claim, we call an $x \in X$ "good" if the n-tuple in the matching over $\{Y_j\}$ that it belongs to does not contain any other element of X. Then, $\mathbf{E}[\sum_{y\in Y} y] \geq \mathbf{E}[\sum_{x\in X} x\,\mathbf{Pr}[x \text{ is "good"}]]$.

Let us compute the probability that some x is "good". Without loss of generality, suppose that $x \in Y_1$. In order for x to be good, it's n-tuple must not contain any of the other elements of X from the other Y_j's. If we define $x_j = |X \cap Y_j|$, then $\mathbf{Pr}[x \text{ is "good"}]$ is at least $\prod_{j\neq 1}(1 - \frac{x_j}{k(n-1)!})$ where $\sum x_j \leq n!$. This product is minimized when we set one of the x_js to $n!$ and the rest to 0, and takes on a minimum value of $1 - n/k \geq 1 - 1/c$. ■

Lemma 4.11 *Let $X_1, \cdots, X_n$ be arbitrary correlated real-valued random variables. Let $Y_1, \cdots, Y_n$ be independent random variables defined so that the distribution of Y_i is identical to that of X_i for all i. Then, $\mathbf{E}[\max_j X_j] \leq \frac{e}{e-1} \mathbf{E}[\max_j Y_j]$.*

Proof: We use the following result from [ADSY10] (also implicit in [CHMS10]). Let U be a universe of n elements, f a monotone increasing submodular function over subsets of this universe, and D a distribution over subsets of U. Let $\tilde{D}$ be a product distribution (that is, every element is picked independently to draw a set from this distribution) such that $\mathbf{Pr}_{S\sim D}[i \in S] = \mathbf{Pr}_{S\sim\tilde{D}}[i \in S]$. Then $\mathbf{E}_{S\sim D}[f(S)] \leq \frac{e}{e-1} \mathbf{E}_{S\sim\tilde{D}}[f(S)]$.

To apply this theorem, let us first assume that the variables X_i are discrete random variables over a finite domain. The universe U will then have one element for each possible instantiation of each variable X_i with a value equal to that instantiation. Then any joint instantiation of the variables $X_1, \cdots, X_n$ corresponds to a subset of U; let D denote the corresponding distribution over subsets. Let f be the max function over the instantiated subset. Then $\mathbf{E}[\max_j X_j]$ is exactly equal to $\mathbf{E}_{S\sim D}[f(S)]$. As before, let $\tilde{D}$ denote the distribution over subsets of U where each element is picked independently. Likewise, the random variables $Y_1, \cdots, Y_n$ define a distribution, say D', over subsets of U. Note that under D' the memberships of elements of U in the instantiated subset are negatively correlated – for two elements that correspond to instantiations of the same variable, including one in the subset implies that the other is not included. This raises the expected maximum. In other words, $\mathbf{E}_{S\sim D'}[f(S)] \geq \mathbf{E}_{S\sim\tilde{D}}[f(S)]$. Therefore, we get $\mathbf{E}[\max_j X_j] = \mathbf{E}_{S\sim D}[f(S)] \leq (e/e-1)\, \mathbf{E}_{S\sim D'}[f(S)] = (e/e-1)\, \mathbf{E}[\max_j Y_j]$.

When the variables X_j are defined over a continuous but bounded domain, we can apply the above argument to an arbitrarily fine discretization of the variables. Our claim then follows from taking the limit as the granularity of the discretization goes to zero.

Finally, let us address the boundedness assumption. For some $\epsilon < 1/n^2$, let B be defined so that for all i, $\mathbf{Pr}[X_i > B] \leq \epsilon$. Then the contribution to the expected maximum from values above B is similar for the Xs and the Ys: the probability that some variable X_i attains the maximum value $b > B$ is at most $\mathbf{Pr}[X_i = b]$ whereas the probability that the variable Y_i attains the maximum value $b > B$ is at least $(1-\epsilon)^{n-1}\,\mathbf{Pr}[Y_i = b]$. Therefore, $\mathbf{E}[\max_j X_j] \leq (1+o(\epsilon))(e/e-1)\,\mathbf{E}[\max_j Y_j]$. Taking the limit as ϵ goes to zero implies the theorem. ■

Comparing OPT **and** OPT_δ We now prove Lemma 4.1. The key intuition behind the lemma is that it can be viewed as the result of scaling both sides of the stochastic dominance relation of Lemma 4.9 up by a constant, and as we shall see, the monotone hazard rate condition is retained by the minimum among multiple draws from a probability distribution.

Lemma 4.1 *When the distributions of job sizes have monotone hazard rates the expected worst best and average best runtimes on δm machines are no more than $1/\delta^2$ times the expected worst best and average best runtimes respectively on m machines.*

Proof: We will show that the random variable $T_j[1{:}\delta m]$ is stochastically dominated by $\frac{1}{\delta}T_j[1{:}m]$. Then, the expected worst best runtime with δm machines is no more than $1/\delta$ times the expected worst best runtime with m machines. Likewise, the expected average best runtime with δm machines is no more than $1/\delta^2$ times the expected average best runtime with m machines. (The extra $1/\delta$ factor comes about because we average over δm machines for the former, versus over m machines for for the latter.)

Our desired stochastic dominance relation is precisely of the form given by Lemma 4.9. In particular, observe that taking a minimum among m draws is exactly the same as first splitting the m draws into $1/\delta$ groups, selecting the minimum from each group of δm draws, and then taking the minimum from this collection of $1/\delta$ values. Thus, we can see that $(1/\delta)T_j[1{:}m] = (1/\delta)T_j[1{:}\delta m][1{:}1/\delta]$, and so the claim follows immediately from Lemma 4.9 as long as the distribution of $T_j[1{:}\delta m]$ has a monotone hazard rate. We show in Claim 4.18 below that the first order statistic of i.i.d. monotone hazard rate distributions also has a monotone hazard rate. ■

Claim 4.18 *A distribution F has a monotone hazard rate if and only if the distribution of the minimum among k draws from F has a monotone hazard rate.*

Proof: Let F_k denote the cdf for minimum among n draws from F. Then we have $F_k(x) = 1 - (1 - F(x))^k$, and the corresponding $f_k(x) = k(1 - F(x))^{k-1} f(x)$. Thus the hazard rate function is:

$$h_k(x) = \frac{f_k(x)}{1 - F_k(x)} = \frac{k(1 - F(x))^{k-1} f(x)}{(1 - F(x))^k} = k \frac{f(x)}{1 - F(x)}.$$

This is precisely k times the hazard rate function $h(x)$, and therefore, $h_k(x)$ is monotone increasing if and only if $h(x)$ is. ■

4.6 Conclusion

Non-linear objectives coupled with multi-dimensional preferences present a significant challenge in mechanism design. Our work shows that this challenge can be overcome for the makespan objective when agents (machines) are a priori identical. This suggests a number of interesting directions for follow-up. Is the gap between the first-best and second-best solutions (i.e. the cost of incentive compatibility) still small when agents are not identical? Does knowledge of the prior help? Note that this question is meaningful even if we ignore computational efficiency. On the other hand, even if the gap is small, the optimal incentive compatible mechanism may be too complex to find or implement. In that case, can we approximate the optimal incentive compatible mechanism in polynomial time?

Similar questions can be asked for other non-linear objectives. One particularly interesting objective is max-min fairness, or in the context of scheduling, maximizing the running time of the least loaded machine. Unlike for makespan, in this case we cannot simply “discard” a machine (that is, schedule no jobs on it) without hurting the objective. This necessitates techniques different from the ones developed in this paper.

Bibliography

[ADL09] Itai Ashlagi, Shahar Dobzinski, and Ron Lavi. An optimal lower bound for anonymous scheduling mechanisms. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 169–176, New York, NY, USA, 2009. ACM.

[ADSY10] Shipra Agrawal, Yichuan Ding, Amin Saberi, and Yinyu Ye. Correlation robust stochastic optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1087–1096, 2010.

[AHK05] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta algorithm and applications. Technical report, 2005.

[AHL12] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *ACM Conference on Electronic Commerce*, pages 18–35, 2012.

[AT01] Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *FOCS*, pages 482–491, 2001.

[AWY09] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. arXiv:0911.2974v1, 2009.

[BJN07] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA'07: Proceedings of the 15th annual European conference on Algorithms*, pages 253–264, Berlin, Heidelberg, 2007. Springer-Verlag.

[BK96] Jeremy Bulow and Paul Klemperer. Auctions vs negotiations. *American Economic Review*, 86(1):180–194, 1996.

[BK10] Bahman Bahmani and Michael Kapralov. Improved bounds for online stochastic matching. In *ESA*, pages 170–181, 2010.

[CCD+10] Denis Xavier Charles, Max Chickering, Nikhil R. Devanur, Kamal Jain, and Manan Sanghi. Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads. In *ACM Conference on Electronic Commerce*, pages 121–128, 2010.

[CFF+10] Edith Cohen, Michal Feldman, Amos Fiat, Haim Kaplan, and Svetlana Olonetsky. Envy-free makespan approximation: extended abstract. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, pages 159–166, 2010.

[CHK07] Shuchi Chawla, Jason D. Hartline, and Robert D. Kleinberg. Algorithmic pricing via virtual valuations. In *ACM Conference on Electronic Commerce*, pages 243–251, 2007.

[CHMS10] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 311–320, 2010.

[CHMS13] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Prior-independent mechanisms for scheduling. In *STOC*, pages 51–60, 2013.

[CIL12] Shuchi Chawla, Nicole Immorlica, and Brendan Lucier. On the limits of black-box reductions in mechanism design. In *STOC*, pages 435–448, 2012.

[CKK07] George Christodoulou, Elias Koutsoupias, and Annamária Kovács. Mechanism design for fractional scheduling on unrelated machines. In *ICALP*, pages 40–52, 2007.

[CKV07] George Christodoulou, Elias Koutsoupias, and Angelina Vidali. A lower bound for scheduling mechanisms. In *SODA*, pages 1163–1170, 2007.

[Cla71] E. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[DDDR08] Peerapong Dhangwatnotai, Shahar Dobzinski, Shaddin Dughmi, and Tim Roughgarden. Truthful approximation schemes for single-parameter agents. In *FOCS*, pages 15–24, 2008.

[DH09] Nikhil R. Devenur and Thomas P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 71–78, 2009.

[DHKN11] Nikhil R. Devanur, Jason D. Hartline, Anna R. Karlin, and C. Thach Nguyen. Prior-independent multi-parameter mechanism design. In *WINE*, pages 122–133, 2011.

[DJSW11] Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, EC '11, pages 29–38, New York, NY, USA, 2011. ACM.

[DRY10] Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, pages 129–138, 2010.

[DSA12] Nikhil R. Devanur, Balasubramanian Sivan, and Yossi Azar. Asymptotically optimal algorithm for stochastic adwords. In *ACM Conference on Electronic Commerce*, pages 388–404, 2012.

[FHK+10] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Clifford Stein. Online stochastic packing applied to display ad allocation. In *ESA*, pages 182–194, 2010.

[FL12] Amos Fiat and Ariel Levavi. Tight lower bounds on envy-free makespan approximation. *CoRR*, abs/1205.1786, 2012.

[Fle00] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discret. Math.*, 13(4):505–520, 2000.

[FMMM09] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *FOCS '09: Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126, Washington, DC, USA, 2009. IEEE Computer Society.

[GK98] Naveen Garg and Jochen Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 300, Washington, DC, USA, 1998. IEEE Computer Society.

[GM08] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 982–991, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[Gro73] Theodore Francis Groves. Incentives in teams. *Econometrica*, 41(4):617–31, 1973.

[GS11] Bernardo Guimaraes and Kevin D. Sheedy. Sales and monetary policy. *American Economic Review*, 101(2):844–76, September 2011.

[HL10] J. Hartline and B. Lucier. Bayesian algorithmic mechanism design. In *STOC*, 2010.

[HMZ11] Bernhard Haeupler, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*, pages 170–181, 2011.

[HR09] Jason D. Hartline and Tim Roughgarden. Simple versus optimal mechanisms. In *ACM Conference on Electronic Commerce*, pages 225–234, 2009.

[JM03] Justin P. Johnson and David P. Myatt. Multiproduct quality competition: Fighting brands and product line pruning. *The American Economic Review*, 93(3):pp. 748–774, 2003.

[Kir11] Rene Kirkegaard. Ranking asymmetric auctions using the dispersive order. *Manuscript*, 2011.

[KMT11] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *STOC*, 2011.

[KPP96] Anil Kamath, Omri Palmon, and Serge Plotkin. Routing and admission control in general topology networks with poisson arrivals. In *SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 269–278, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

[KV07] Elias Koutsoupias and Angelina Vidali. A lower bound of 1+*phi* for truthful scheduling mechanisms. In *MFCS*, pages 454–464, 2007.

[LP03] Carlos A. León and François Perron. Extremal properties of sums of bernoulli random variables. *Statistics & Probability Letters*, 62(4):345–354, May 2003.

[LS07] Ron Lavi and Chaitanya Swamy. Truthful mechanism design for multidimensional scheduling via cycle monotonicity. In *ACM Conference on Electronic Commerce*, pages 252–261, 2007.

[Lu09] Pinyan Lu. On 2-player randomized mechanisms for scheduling. In *WINE*, pages 30–41, 2009.

[LY08a] Pinyan Lu and Changyuan Yu. An improved randomized truthful mechanism for scheduling unrelated machines. In *STACS*, pages 527–538, 2008.

[LY08b] Pinyan Lu and Changyuan Yu. Randomized truthful mechanisms for scheduling unrelated machines. In *WINE*, pages 402–413, 2008.

[MGS11] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. In *SODA*, pages 1285–1294, 2011.

[MS07] Ahuva Mu'alem and Michael Schapira. Setting lower bounds on truthfulness: extended abstract. In *SODA*, pages 1143–1152, 2007.

[MSVV05] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized on-line matching. In *In FOCS 05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 264–273. IEEE Computer Society, 2005.

[Mu'09] Ahuva Mu'Alem. On multi-dimensional envy-free mechanisms. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, ADT '09, pages 120–131, 2009.

[MY11] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *STOC*, 2011.

[Mye81] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

[Nee03] Zvika Neeman. The effectiveness of english auctions. *Games and Economic Behavior*, 43(2):214–238, May 2003.

[NR99] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, pages 129–140, 1999.

[PST91] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. In *SFCS '91: Proceedings of the 32nd annual symposium on Foundations of computer science*, pages 495–504, Washington, DC, USA, 1991. IEEE Computer Society.

[Rob33] Joan Robinson. *The Economics of Imperfect Competition.* McMillan and Co., 1933.

[Ron01] Amir Ronen. On approximating optimal auctions. In *ACM Conference on Electronic Commerce*, pages 11–17, 2001.

[RTCY12] Tim Roughgarden, Inbal Talgam-Cohen, and Qiqi Yan. Supply-limiting mechanisms. In *ACM Conference on Electronic Commerce*, pages 844–861, 2012.

[SS13] Balasubramanian Sivan and Vasilis Syrgkanis. Auctions vs negotiations in irregular markets. *CoRR*, abs/1306.4022, 2013.

[Vaz01] Vijay V. Vazirani. *Approximation algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[Vic61] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, pages 8–37, 1961.

[WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[Yan11] Qiqi Yan. Mechanism design via correlation gap. In *SODA*, pages 710–719, 2011.

[You95] Neal E. Young. Randomized rounding without solving the linear program. In *SODA '95: Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 170–178, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[You01] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, FOCS '01, pages 538–547, Washington, DC, USA, 2001. IEEE Computer Society.