UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

SAMPLE-DRIVEN ONLINE SELECTION

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN SISTEMAS DE INGENIERÍA

ANDRÉS IGNACIO CRISTI ESPINOSA

PROFESOR GUÍA:
JOSÉ CORREA HAEUSSLER

PROFESOR CO-GUÍA:
PAUL DÜTTING

MIEMBROS DE LA COMISIÓN:
JUAN ESCOBAR CASTRO
BRUNO ZILIOTTO

SANTIAGO DE CHILE
2023

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE DOCTOR EN SISTEMAS DE INGENIERÍA
POR: ANDRÉS IGNACIO CRISTI ESPINOSA
AÑO: 2023
PROF. GUÍA: JOSÉ CORREA HAEUSSLER
PROF. CO-GUÍA: PAUL DÜTTING

SELECCIÓN EN LÍNEA BASADA EN MUESTREO

En esta tesis, estudiamos desde una perspectiva basada en muestreo cuatro problemas de selección en línea que generalizan dos modelos clásicos: el problema del secretario y la desigualdad del profeta.

En el primer capítulo, proponemos un modelo llamado p-DOS, que generaliza tanto el problema del secretario como la desigualdad del profeta i.i.d. En este problema, un adversario elige un conjunto de números y cada uno de ellos es muestreado independientemente con probabilidad p y se nos revela de antemano. Los números restantes se revelan secuencialmente y tomamos decisiones irrevocables de parar/continuar, con el objetivo de maximizar la esperanza del número seleccionado. Caracterizamos el algoritmo óptimo para todos los valores de p, y mostramos que su garantía de rendimiento interpola continuamente entre $1/e$ y 0.745, las garantías óptimas en el problema del secretario y la desigualdad del profeta i.i.d.

En el segundo capítulo proponemos un modelo similar, llamado ROSp, donde el objetivo es maximizar la probabilidad de seleccionar el mejor número de los números no muestreados. Nuevamente, encontramos el rendimiento óptimo y mostramos que interpola suavemente entre $1/e$ y 0.5801, las garantías óptimas en el problema del secretario y el problema del secretario i.i.d.

El tercer capítulo estudia equidad y sesgo en el contexto del problema del secretario. Introducimos una nueva variante donde cada candidato pertenece a un cierto grupo. Asumimos que las comparaciones entre candidatos de diferentes grupos son sesgadas de manera arbitraria, así que analizamos el algoritmo óptimo que solo compara candidatos cuando son del mismo grupo. Mostramos que esta política óptima es equitativa en un sentido fundamental, logrando un equilibrio entre diferentes grupos. También proponemos una variante de este problema basada en muestreo, inspirada en el modelo ROSp.

En el cuarto capítulo estudiamos las subastas combinatoriales en línea, una importante generalización a selección múltiple de la desigualdad del profeta. En este problema, tenemos m elementos y establecemos un precio para cada uno de ellos. Luego, una secuencia de agentes que llegan uno a la vez compran su conjunto preferido de los elementos restantes, a los precios dados. Asumimos que conocemos las distribuciones a partir de las cuales se extraen las valoraciones de los compradores. Mostramos a través de un novedoso argumento de punto fijo que los precios óptimos logran una $(\mathrm{d}+1)$-aproximación al bienestar social óptimo a posteriori, donde d es el tamaño del conjunto más grande que cualquier comprador quisiera comprar. Probamos además que si solo tenemos muestras de los valores, en lugar de las distribuciones, todavía podemos calcular precios que logran una aproximación $(\mathrm{d}+1+\varepsilon)$ y, además, podemos hacerlo en tiempo polinomial.

ABSTRACT OF THE THESIS FOR THE DEGREE OF
DOCTOR EN SISTEMAS DE INGENIERÍA
BY: ANDRÉS IGNACIO CRISTI ESPINOSA
YEAR: 2023
ADVISOR: JOSÉ CORREA HAEUSSLER
CO-ADVISOR: PAUL DÜTTING

SAMPLE-DRIVEN ONLINE SELECTION

In this thesis, we study from a sample-based perspective four online selection problems that generalize two classic models: the secretary problem and the prophet inequality.

In the first chapter, we propose a model we call p-DOS, that generalizes both the secretary problem and the i.i.d. prophet inequality. In this problem, an adversary chooses a set of numbers, and each of them is sampled independently with probability p and revealed to us beforehand. The remaining numbers are revealed sequentially and we make irrevocable stop/continue decisions, with the objective of maximizing the expectation of the selected number. We characterize the optimal algorithm for all values of p, and show that its performance guarantee continuously interpolates between 1/e and 0.745, the optimal performances in the secretary problem and the i.i.d. prophet inequality.

In the second chapter we propose a similar model, called ROSp, where the objective is to maximize the probability of selecting the best number of the non-sampled numbers. Again, we find the optimal performance and show it gracefully interpolates between 1/e and 0.5801, the optimal guarantees in the secretary problem and the i.i.d. secretary problem.

The third chapter studies fairness and bias in the context of the secretary problem. We introduce a new variant where each candidate belongs to a certain group. We assume comparisons between candidates of different groups are arbitrarily biased, so we analyze the optimal algorithm that only compares candidates of the same group. We show that this optimal policy is fair in a fundamental sense, achieving a balance between different groups. We also propose a sample-based variant of this problem, inspired in the ROSp model.

In the fourth chapter we study Online Combinatorial Auctions, an important generalization of the prophet inequality to multiple selection. We are given m items and we set a price for each of them. Then, a sequence of agents that arrive one by one buy their preferred set from the remaining items at the given prices. We assume we know the distributions from which the valuations of the buyers are drawn. We show via a novel fixed-point argument that the optimal prices achieve a $(\mathrm{d}+1)$-approximation to the optimal social welfare in hindsight, where d is the size of the largest set any buyer would like to buy. We show that if we have only samples of the values instead of the distributions, we still can compute prices that achieve a $(\mathrm{d}+1+\varepsilon)$-approximation, and moreover, we can do this in polynomial time.

Acknowledgments

I am deeply grateful to my advisors, José Correa and Paul Dütting, for their continued support and encouragement during my PhD. With their help, not only in the form of collaboration and advice, but also by making me feel confident of myself and my abilities, I managed to accomplish so many great things I never even imagined. I thank José, in particular, for believing in me, for pushing me to improve, and for dedicating so many hours to patiently work with me and teach me how to do research since my Masters. I definitely owe to his optimistic and passionate approach to research that I had such a great time all these years. I am also grateful for his great advice in all aspects of academic life, and even beyond it. I thank Paul for inviting me to London, and then to work at Google, and for making me feel welcome there; these were amazing and enriching experiences. I also want to thank Paul for our motivating discussions about sample-based prophet inequalities, and for introducing me to online combinatorial auctions.

I thank all the great people I had the pleasure to meet and share a space together in the office, Kevin, Marc, Ruben, Víctor, Pato, Tim, Andy T., Fabio, Laura, Bastián, Natalie, Boris, Chino, Dana, Carlos, Martín, Raimundo, Krzysztof, Bastián, Benjamín, Gonzalo, Antoine, Alexandros, Andrew, Laurent. All the fun conversations, the jokes, the coffee, the lunches and dinners shared together, all the non-work activities, they were essential to enjoying the actual work (and I missed them so much during the pandemic!!).

A special thank to all my collaborators, in particular to those whose work is part of this thesis. Thanks to José Correa, Boris Epstein, José Soto, Laurent Feuilloley, Tim Oosterwijk, Alexandros Tsigonias-Dimitriadis, Paul Dütting, Ashkan Norouzi-Fard, Andrés Fielbaum, Tristan Pollner, and Matthew Weinberg. For me, research is a fundamentally collaborative activity, it's a conversation, and I become inspired about a problem when I talk with other people about it. So I owe them not only the effort they put on these projects, but also my own inspiration to work.

I also thank my friends outside academia, Simón, Ariel, Palama, Dani, Pablo, Celgadis, Bárbara, Degre. For being there to laugh, to talk, to have fun, and spend time together.

I thank my mother, for her love and support, and for always believing in me. I thank my father for transmitting to me the desire to learn, understand and discover. I thank my brother José Manuel for his peace and simplicity. I thank my brother Tomás and my aunt Monica for being there in the hardest moments.

I thank my wife Nicol for her tremendous love and support. For listening to me, for believing in me, for being my best friend and my partner. At the end of a long day, I feel happy and safe with her and our cats. Without her, I'm pretty sure this thesis wouldn't exist at all.

Table of Content

Introduction

Take the viewpoint of a seller that has a single item on sale and faces a sequence of customers that arrive one by one. For simplicity, imagine each customer makes a take-it-or-leave-it offer for the item. What would be a good strategy for the seller to maximize profit? It's probably not a good idea to sell too early, because there might be better offers later; but it's also not good to wait for too long. A similar problem is faced by a company interviewing candidates for an open position. Good candidates are likely to get other offers, so the company should make an offer soon after interviewing a good candidate and, crucially, before seeing all candidates. Again, we have a trade-off between accepting candidates that look good early but might not be the best, and waiting for better candidates and risking missing a good opportunity. The main challenge then in both problems is to decide when to stop the sequence. Two basic models capture the essence of this challenge: the *secretary problem* and the *prophet inequality*.

The secretary problem, the prophet inequality, and variants of them have been intensively studied in the last six decades. First studied by probabilists, more recently they have drawn the attention of economists and computer scientists because of their connections with problems faced by modern platforms, such as online advertisers and e-commerce marketplaces. Despite their importance, a drawback of these two models is that they take two extreme perspectives on prior information. On the one hand, in the secretary problem, we take a worst-case perspective and assume we have zero knowledge about the sequence. On the other hand, in the prophet inequality, we assume we have full distributional knowledge of the process that generates the sequence. In this thesis we take the perspective of a recent line of work that attempts to bridge these two extreme points by assuming we do not have full distributional knowledge, but we have access to a few independent samples of the process beforehand. In the first two chapters of the thesis we formulate and analyze two problems that model this intermediate form of prior knowledge, and interpolate between the secretary problem and the prophet inequality. While still taking a sample-based perspective, in the last two chapters we investigate two online selection problems that extend the classic models in two directions, fairness, and multi-item selection.

In what follows we describe the three basic models we consider in this thesis, the secretary problem, the prophet inequality, and a generalization of the latter, known as online combinatorial auctions. Then, we describe the specific problems and results of each chapter.

The Secretary Problem. In this problem, an adversary designs a collection of numbers $x_1, \ldots, x_n$, all different from each other, which are revealed one by one to a decision maker

(DM) according to a random permutation. The DM doesn't know which numbers were selected by the adversary until the moment they are revealed. At that time, she has to make an irrevocable continue/stop decision. Should she stop, she keeps the last revealed number and never observes the following ones. The goal of the DM is to maximize the probability of stopping when the largest of the n numbers is revealed.

The Secretary Problem is one of the most important problems in online decision-making [99, 54, 67, 69]. For large n, the optimal algorithm for this problem is to *not* stop at any of the first n/e numbers, and afterward stop at the first number which is larger than all of the numbers observed so far. This algorithm stops at the largest number in the sequence with a probability of at least $1/\mathrm{e}$, which is the best achievable probability in general. The optimal solution is the same if our objective is to maximize the expectation of the selected value.

The Prophet Inequality. In the Prophet Inequality, a sequence of non-negative, independent random variables $X_1, \ldots, X_n$ are presented one by one to a DM in a fixed order. The DM has full distributional knowledge about the random variables, but does not know their realizations in advance. The DM observes the realizations of the random variables one by one, and must make irrevocable continue/stop decisions before moving on to the next one. The goal of the DM is to maximize the expected value of the realized random variable in which she stops.

For this problem, there exist stopping rules that achieve a competitive ratio of $1/2$ against a prophet that knows the realization of all the random variables and always stops at the maximum, and $1/2$ is actually optimal [93, 94]. Plentiful variants of this problem have been studied and are being studied. Among them, an important special case is the i.i.d. prophet inequality, where the random variables $X_1, \ldots, X_n$ are sampled independently from a common distribution. In this setting, the optimal competitive ratio improves to $\alpha^* \approx 0.745$ [80, 84, 43, 101].

Online Combinatorial Auctions. If we think of the Prophet Inequality as the problem of selling an item to a sequence of agents with random valuations, a natural generalization is to assume we have multiple items on sale. In an Online Combinatorial Auction, we have a set M of items and a sequence of agents with independent random valuations $v_1, \ldots, v_n$ arrive one by one. Each valuation is a function $v_{\mathrm{i}} : 2^M \to \mathbb{R}_+$ that represents how much agent i is willing to pay for each subset of M. The DM has full distributional knowledge about the valuations but does not know the realizations in advance. After the arrival of an agent, the DM must irrevocably decide which items to allocate her (if any). The goal of the DM is to maximize the expected social welfare, i.e., the expectation of the sum over agents of the valuation of the agent for the set she receives.

In the general case poor approximation guarantees are possible, so most work on this problem focuses on special classes of valuation functions. One type of assumptions relates to the structure of complementarities of the valuations. For a class known as complement-free valuations, that satisfy that $v(A) + v(B) \leq v(A \cup B)$, an $O(\log\log|M|)$-approximation is possible [51]. For a special case known as fractionally subadditive valuations, the approximation factor can be improved to 2 [60]. In this work we take the approach of parameterizing the valuations by a number d, defined as the maximum size of a bundle any buyer might want to buy. Prior to our results, the best known factor in terms of d was $1/(4\mathrm{d}-2)$ [49].

Summary of the Chapters

In Chapter 1 we study the sample-based version of the i.i.d. Prophet Inequality. We take a unifying approach to single selection optimal stopping problems with random arrival order and independent sampling of items. In the problem we consider, a decision maker (DM) initially gets to sample each of N items independently with probability p, and can observe the relative rankings of these sampled items. Then, the DM faces the remaining items in an online fashion, observing the relative rankings of all revealed items. While scanning the sequence the DM makes irrevocable stop/continue decisions and her reward for stopping the sequence facing the item with rank i is Y_{i}. The goal of the DM is to maximize her reward. We start by studying the case in which the values Y_{i} are known to the DM, and then move to the case in which these values are adversarial. For the former case we are able to recover several classic results in the area, thus giving a unifying framework for single selection optimal stopping. For the latter, we pin down the optimal algorithm, obtaining the optimal competitive ratios for all values of p. The material in this chapter is based on joint work with José Correa, Boris Epstein, and José Soto [36].

In Chapter 2 we study the problem of selecting the maximum value of a sequence when we can observe beforehand some samples drawn from the same distribution. While in the classic secretary problem the values of upcoming elements are entirely unknown, in many realistic situations the decision-maker has access to some information, for example from past data. Here we take a sampling approach and assume that before starting the sequence each element is sampled independently with probability p. Our main result is to obtain the best possible algorithms for all values of p. As p grows to 1, the obtained guarantee converges to the optimal guarantee in the full information case. Notably, we characterize the best possible algorithm by a sequence of thresholds, dictating at which point in time we should accept a value. Surprisingly, this sequence is independent of p. We complement our theoretical results with numerical experiments on data from [70] of people playing the secretary problem repeatedly. Our results help explain some behavioral issues they raised and indicate that people play a strategy similar to our optimal algorithms from the start onwards, albeit slightly suboptimally. The material in this chapter is based on joint work with José Correa, Laurent Feuilloley, Tim Oosterwijk, and Alexandros Tsigonias-Dimitriadis [38].

In Chapter 3 we attempt to address the issues of fairness and bias in online selection by introducing the Multi-Color Secretary Problem. There is growing awareness and concern about fairness in machine learning and algorithm design. This is particularly true in online selection problems where decisions are often biased, for example, when assessing credit risks or hiring staff. In the Multi-Color Secretary Problem, candidates are partitioned into different groups or *colors*. The candidates arrive sequentially and upon arrival of a candidate we have to irrevocably decide whether we want to select the candidate or not. Candidates arrive in uniform random order and we can rank candidates within a group, but we cannot compare candidates across groups. There is also a prior probability that the best candidate from a group is the best candidate overall. The problem models situations in which different qualities of the candidates make them largely incomparable (this could arise in some form due to gender, race, social origin, type of education, etc.). The goal is to maximize the probability with which we stop at the best overall candidate and compare it with that for the *offline* optimum. Note that here the offline optimum simply picks the best candidate from the group of largest prior probability. Thus, it is extremely unfair. One may think that the best possible online algorithm is to mimick the offline optimum; namely to select the group of largest prior probability and then run the classic secretary algorithm on that group. We prove that this is not the case and indeed our main result is to obtain the best possible online algorithm for the problem and to establish that it satisfies very desirable fairness properties. Hence, for this variant on online selection, fairness follows as a consequence of being online optimal. Finally, we formulate a sample-driven version of our model and analyze a large class of online algorithms for the problem. The material in this chapter is based on joint work with José Correa, Paul Dütting, and Ashkan Norouzi-Fard [35].

In Chapter 4 we study online combinatorial auctions, focusing on posted prices mechanisms. We are given a set of indivisible items and a set of buyers with randomly drawn monotone valuations over subsets of items. The DM sets item prices and then the buyers make sequential purchasing decisions, taking their favorite set among the remaining items. We parametrize an instance by d, the size of the largest set a buyer may want. Our main result asserts that there exist prices such that the expected (over the random valuations) welfare of the allocation they induce is at least a factor $1/(\mathrm{d}+1)$ times the expected optimal welfare in hindsight. Moreover, we prove that this bound is tight. Thus, our result not only improves upon the $1/(4\mathrm{d}-2)$ bound of Dütting et al., but also settles the approximation that can be achieved by using item prices. The existence of these prices follows from the existence of a fixed point of a related mapping, and therefore, it is non-constructive. However, we show how to compute such a fixed point in polynomial time, even if we only have sample access to the valuation distributions. The material in this chapter is based on joint work with José Correa, Andrés Fielbaum, Tristan Pollner, and Matthew Weinberg [39].

Chapter 1

Sample-Driven Optimal Stopping: From the Secretary Problem to the i.i.d. Prophet Inequality

Two fundamental models in online decision making are that of competitive analysis and that of optimal stopping. In the former, the input is produced by an *adversary* whose goal is to make the algorithm perform poorly with respect to a certain benchmark. In the latter, the algorithm has full distributional knowledge of the input, making it much easier for the algorithm to achieve good approximation ratios. The area of optimal stopping has been very active in the last decade since many real-world situations, including several e-commerce platforms, often do not behave adversarially, and the distributional model of optimal stopping seems appropriate. Furthermore, the activity in the area has been boosted by the close connection between posted price mechanisms, attractive for their usability and simplicity, and prophet inequalities, a classic topic in optimal stopping theory [77, 27].

One of the most important problems in online decision making is the secretary problem [99, 54, 67, 69]. In this problem, an adversary designs a collection of numbers $x_1, \ldots, x_n$, all different from each other, which are revealed one by one to a decision maker (DM) according to a random permutation. The DM doesn't know which numbers were selected by the adversary until the moment they are revealed. At that time, she has to make an irrevocable continue/stop decision. Should she stop, she keeps the last revealed number and never observes the following ones. The goal of the DM is to maximize the probability of stopping when the largest of the n numbers is revealed. For large n, the optimal algorithm for this problem is to *not* stop at any of the first n/e numbers, and afterward stop at the first number which is larger than all of the numbers observed so far. This algorithm stops at the largest number in the sequence with a probability of at least $1/\mathrm{e}$, which is the best achievable probability in general.

A staple problem in optimal stopping is the classic prophet inequality. In this problem, a sequence of non-negative, independent random variables $X_1, \ldots, X_n$ are presented one by one to a DM in a fixed order. The DM has full distributional knowledge about the random variables, but does not know their realizations in advance. The DM observes the realizations of the random variables one by one, and must make irrevocable continue/stop decisions before moving on to the next one. The goal of the DM is to maximize the expected value of the realized random variable in which she stops. For this problem there exist stopping rules that achieve a competitive ratio of $1/2$ against a prophet that knows the realization of all the random variables and always stops at the maximum, and $1/2$ is actually optimal [93, 94]. Plentiful variants of this problem have been studied and are being studied. Among them, an important special case is the i.i.d. prophet inequality, where the random variables $X_1, \ldots, X_n$ are sampled independently from a common distribution. In this setting, the optimal competitive ratio improves to $\alpha^* \approx 0.745$ [80, 84, 43, 101].

Recently, data-driven versions of optimal stopping problems have been successfully studied. These constitute a bridge between the worst case model and the distributional model. A standard model, first described in Azar et al.'s [5] pioneering work, consists in replacing the full distributional knowledge with having access to one or more samples from each distribution. The model is very attractive both from a practical and theoretical perspective. On the one hand, full distributional knowledge is a strong assumption, while access to historical data is usually straightforward. And this historical data can be thought of as being samples from certain underlying distributions. On the other hand, the model gains back the combinatorial flavor of competitive analysis and thus becomes much more prone to be analyzed using standard algorithmic tools. A notable example of this is the recent result of Rubinstein et al. [110] for the classic prophet inequality. They study the setting in which the DM doesn't know the underlying distributions of the random variables, but instead has access to a single sample for each of them, and show that this amount of information is enough to guarantee the best possible factor in the full information case (with adversarial order), namely $1/2$. Inspired by Azar et al.'s model, Correa et al. [40] considered the variant of the i.i.d prophet inequality problem where the underlying distribution from which the random variables are sampled from is unknown to the DM. They establish that when the DM has no additional information the best she can do is to basically apply the classic algorithm for the secretary problem and thus obtain, in expectation, a fraction $1/e$ of the expected maximum value. On the other hand, if she has access to $O(n^2/\varepsilon)$ samples of the underlying distribution, then she can essentially learn the distribution and guarantee a factor of $\alpha^* - O(\varepsilon)$; where $\alpha^* \approx 0.745$ is the optimal factor for the i.i.d. prophet inequality with full distributional knowledge. This latter result was improved by Rubinstein et al. [110], who showed that $O(n/\varepsilon^6)$ samples are enough to guarantee a factor of $\alpha^* - O(\varepsilon)$. The sampling model from i.i.d. random variables [40] shares some aspects with the classic secretary problem, in which arbitrary non-negative numbers are presented to the DM in uniform random order. The former can be thought of as generating several i.i.d. samples from a common distribution, and shuffling them in a random order (as in the secretary problem). Given the random order, we can say that the first numbers are the 'samples' that can be observed but cannot be selected, and the remaining numbers are the actual instance of the optimal stopping problem. Along these lines, a particularly clean model [21, 82] is the *dependent* sampling model in which the instance, consisting of N items, is designed by an adversary. Then, the DM gets to sample a random subset of size $h = pN$ of these and scans the remaining items in random order. This model is very robust since it generalizes the sampling model from i.i.d. random variables while making no distributional assumptions. The name dependent sampling comes from the fact that the sampled set has a fixed size h. Thus, there is correlation between the events of each item being sampled. A closely related sampling model, and essentially equivalent for large values of N, is that with *independent* sampling [38]. Here, rather than sampling exactly $h = pN$ items, the DM samples each item independently with probability p.

In this chapter, we study a generic version of the classic single selection optimal stopping problem with sampling, which we call p-sample-driven optimal stopping problem (p-DOS). In this problem a collection of N items is shuffled in uniform random order. The decision maker gets to observe each item independently with probability $p \in [0, 1)$ and these items conform the *information set* or *history set*. The remaining items, conforming the *online set* are revealed sequentially to the DM. At any point, the DM observes the relative rankings of the items that have been revealed, and upon seeing an item, she must decide whether to take it and stop the sequence, or to drop it and continue with the next item. If the DM stops with the i-th ranked item she gets a reward of Y_i and her goal is to maximize the expected value with which she stops. While we do assume that the values are monotone, i.e., $Y_1 \geq \cdots \geq Y_N$, we do not assume that they are non-negative. The natural benchmark to measure the performance of an algorithm here is the expected (over the permutations) maximum value in the online set.

We study both, the cases when the values Y are fixed, (p-DOS with known values), and that when they are adversarial (p-DOS with adversarial values). The former, and already when p=0, models the most well known single selection optimal stopping problems. Indeed the classic secretary problem [99, 54] appears when $Y_1 = 1$ and the remaining values are 0, the 1-choice K-best secretary problem [76, 26] is recovered by $Y_1 = \cdots = Y_K = 1$ and filling zeros in the remaining values, while the problem of selecting an item of minimum ranking [32] is obtained by setting $Y_i = -i$. Still in the case $p = 0$, the problem with non-negative values was studied by Mucci [104]. By analyzing the underlying recursion he obtains a limiting ODE and established that the optimal algorithm takes the form of a sequence of thresholds such that starting at time t_i the DM should stop with an item currently ranked i or better. Bearden et al. [13] also consider this problem from an experimental viewpoint, while Mucci [105] studies the case in which all Y's are negative. The latter problem, p-DOS with adversarial values, generalizes the i.i.d. prophet inequality problem with samples. Indeed, a valid strategy for the adversary is to set the values of all items by generating i.i.d samples from a single distribution. This way, the problem for the adversary is not harder than selecting a worst case distribution to sample from. Specifically, for given p and N, p-DOS with adversarial values models the case when we play with $n = (1 - p)N$ i.i.d. values and we have access to $np/(1 - p)$ independent samples.[1]

1.1 Summary of Results and Overview of the Chapter

We derive the optimal algorithms for problem p-DOS with known values and for p-DOS with adversarial values, for all $p \in [0, 1)$.

After some preliminary definitions in Section 1.2, we start with the case in which Y is known to the DM. Here we take the, by now classic, linear programming approach of

[1]In p-DOS, the number of samples and the number of online values are random, while in the usual formulation of the prophet inequality these quantities are fixed. Thus, this reduction holds for large values of N to achieve concentration. We also examine the version of p-DOS with *dependent* sampling, where this quantities are fixed, and we show it is essentially equivalent to the independent model when N is large.

Buchbinder, Jain, and Singh [18] though slightly extending it to make it able to deal with arbitrary Y values and sampling probability p, and adding a term that forces the algorithm to stop.[2] We note that this LP exactly encodes the best possible algorithm for the problem and that its objective function value decreases with the number of items N (Section 1.3.1). This allows us to deduce that the hardest instances appear as $N \to \infty$. Thus, in Section 1.3.2, we derive the limit LP which shares some aspects with that of Chan, Chen, and Jiang [26]. In uncovering the structure of this limit LP, we provide our first main technical contribution in Section 1.3.3. By understanding monotonicity properties of the LP coefficients and by using mass moving arguments from the theory of optimal transport, we can deduce exactly which inequalities, and in what ranges, are tight in an optimal solution. This permits to bring down the problem of finding the optimal algorithm to that of solving certain, very simple, ODEs[3]. We find the explicit solution of these ODEs and thus bring the problem to a real optimization problem in which the variables are some t_i's determining the ranges where the solutions of the different ODEs should be used. These t_i's also have a natural algorithmic interpretation. They represent the *times* at which the DM should start accepting an item of rank i or higher (among the items seen son far). With this we can conclude that Mucci's structural result holds even if some (or all) Y values are negative and for arbitrary p.

In Section 1.3.4, pushing things a bit further, we prove that this optimization problem over t_i's is concave in each variable and relatively easy to solve, at least approximately. In particular we exemplify that its first order conditions quickly allow us to recover the known results for the secretary problem [99], the 1-choice 2-best secretary problem [26], and the minimum rank problem [32].

Then we move to our main contribution; the study of p-DOS with adversarial values, which we require to be non-negative. This essentially consists in adding a minimization over Y to the linear program for p-DOS with known values. However, to make the problem well posed we first need to normalize the objective function. This is done dividing the objective by the expected value of the optimal choice in the online set, namely $\sum_{i=1}^{\infty} Y_i(1-p)p^{i-1}$.[4] Equivalently, we may add a constraint to the LP imposing that this value is 1. In either way the resulting objective function represents the performance guarantee of an optimal online algorithm. With this formulation, von Neumann's Minmax Theorem allows us to rewrite the minmax problem as a new linear program in which the constraints take a stochastic dominance flavor (Section 1.4.1). We deal with this problem in an analogous way as in the case of known values and thus take the limit on N and apply our main structural theorem in Section 1.4.2. As the objective function of our problem encodes the ratio between the expected value the optimal algorithm gets and the expected maximum on the hindsight, we end up obtaining the best possible approximation guarantee for p-DOS with adversarial values, $\alpha(p)$, as a function of p, and for all values of N (Section 1.4.3). To this end we note that the optimal algorithm, which takes the form of a sequence of thresholds, can easily be implemented for finite values of N without losing in the approximation guarantee (Section 1.4.4).

The value $\alpha(p)$ we obtain in Section 1.4.3 improves upon the recent work of Kaplan et

[2]This is needed since some Y_i's may be negative.
[3]Which are very different to that of Mucci [104].
[4]Note that for $p = 0$ this is just Y_1.

al. [82] and that of Correa et al [42], for large values of N.[5] More importantly, it allows to draw interesting consequences as p varies. Before describing some of these let us note that by the Minmax Theorem p-DOS with adversarial values is equally hard (from an approximation guarantee perspective) if (1) the adversary chooses the Y values and then the DM picks the algorithm or if (2) the adversary chooses the Y values knowing the algorithm of the DM. In other words, for every value of p there is a sequence Y such that no algorithm for the p-DOS with independent sampling on this sequence can achieve an approximation better than $\alpha(p)$. Interestingly, for $p \leq 1/\mathrm{e}$ we prove that $\alpha(p) = 1/(\mathrm{e}(1-p))$. This result closes a small gap left by Kaplan et al. [82] in the dependent sampling model and matches the tight bound in the more restricted setting in which the values are i.i.d. samples from an unknown distribution [40]. Moreover, the minmax perspective above implies that for $p \leq 1/\mathrm{e}$ the secretary problem is the hardest single selection optimal stopping problem.

On the other end of the spectrum, as $p \to 1$, the optimal performance guarantee $\alpha(1) = \lim_{p\to 1} \alpha(p)$ equals $\alpha^* \approx 0.745$.[6] This is interesting since the model admits values that are not possible to capture by any instance of the i.i.d. prophet inequality[7] [82, Theorem 3.4] (so $\alpha(p) \leq \alpha^*$), where only recently it was proved that with an amount of samples linear in n one can approach α^*.[8] Indeed we can show that the approximation ratio of our algorithm, $\alpha(p)$, not only converges to 0.745 but also satisfies $\alpha(p) \geq p \cdot 0.745$, for all $p \in [0, 1)$. This in particular can be applied to improve upon the state of the art of the sampling complexity of the sample based i.i.d prophet inequality. Specifically, if we sample a fraction $(1-\varepsilon)$ of the values our algorithms guarantees a value that is at least $(1-\varepsilon)0.745$ times the expected maximum value of the last εN values. In other words, to guarantee an approximation factor of $\alpha^* - O(\varepsilon)$ we need $O(n/\varepsilon)$ samples, making a significant improvement in the dependence on ε when compared to the best known bound of $O(n/\varepsilon^6)$ by Rubinstein et al. [110].

Besides the extreme values of $p = 0$ and $p \to 1$, we obtain the best possible guarantee for all intermediate values of p. An interesting special case is that of $p = 1/2$, i.e., when the information set and the online set are of roughly the same size. For this special case (though with dependent sampling), Kaplan et al. [82] prove that a relatively simple algorithm, achieves a performance guarantee of $1 - 1/\mathrm{e}$, while the current best bound evaluates to 0.649 [42]. Here, we prove that the optimal algorithm for 1/2-DOS with adversarial values has an approximation guarantee of $\alpha(1/2) \approx 0.671$, thus improving upon the state of the art.

[5] Since the sampling models are only equivalent in the limit.

[6] Note that of course our problem is ill defined if $p = 1$ so the right way of thinking about p close to 1 is to first fix a value p and then make N grow large.

[7] Consider for instance the following particular case of our model where the values are correlated. With probability 1/2 the values are i.i.d. samples of Uniform$[0, 1]$ and with probability 1/2 they are i.i.d. samples of Uniform$[1, 2]$.

[8] Depending on the objective function, it is not always true that with a linear number of samples one can approximate the full information case, even in the i.i.d. model. A prominent case that has been extensively studied is revenue maximization [34, 74]. Consider the objective of maximizing the revenue using a single price, i.e., setting a threshold (or price) T in order to maximize T times the probability that at least one value is above T. If the variables are i.i.d. and equal to $n^2/(1-p)$ w.p. $1/n^2$, and $U[0, 1]$ w.p. $1 - 1/n^2$, a revenue of $\Omega(n)$ can be achieved in a set of $(1-p)n$ variables, taking $T = n^2/(1-p) - \varepsilon$. But if we have only access to pn samples, only with probability $O(1/n^2)$ we will see a high value in both sets. And most of the time we see only realizations of $U[0, 1]$, in which case we cannot differentiate the instance from only $U[0, 1]$ variables, where $T > 1$ gives 0 revenue, so the most we can get is $O(1)$.

To make the comparison between the dependent and independent sampling models more precise we prove, in Section 1.4.4, that the underlying optimal approximation factors in both models differ in an additive factor of at most $O(1/\sqrt{N})$, for fixed $p < 1$. In particular this says that the limit approximation factor $\alpha(p)$ applies to both settings. This connection is important since in much of our analysis we use the linear program for the dependent sampling model but then apply our results in the independent sampling model. It is worth mentioning here that although both models are very similar and essentially equivalent for large N, the independent sampling model is somewhat smoother than the dependent one. In particular, one can immediately define it for all values of $p \in [0, 1)$ and not just those for which pN is integral. Furthermore, as we prove in the chapter, the optimal approximation factor for the independent sampling model (and any Y) decreases with N so that the limit bounds apply for a finite number of items. On the contrary, monotonicity on the dependent sampling model seems very challenging.

We wrap up the chapter in Section 1.5 by considering versions of p-DOS under combinatorial constraints. In particular we consider the extension of the so called matroid secretary problem [10] to the case in which the DM has sampling capabilities. Using our machinery from the single selection case as a black box we are able to get a number of constant competitive algorithms for several special cases of matroids. Additionally, for general matroids, we observe that the existence of a constant competitive algorithm for p-DOS (for any p) implies the existence of a constant competitive algorithm for the matroid secretary problem, which is a notoriously hard open problem. In particular we note that if the optimal competitive ratio of p-DOS in this setting would converge to that in some variant of the i.i.d. case (as it does in the single selection case) then we could solve this open problem.

1.2 Preliminaries

1.2.1 p-DOS with Known Values.

We consider the following problem, which we call p-sample driven optimal stopping (p-DOS, for short). A decision maker (DM) is given list of N items with associated values $Y_1 \geq \cdots \geq Y_N$. Initially each item is independently sampled with probability p and conform the DM's information set (which we denote by H). The remaining items, which we call online set, are presented to the DM in an online fashion in random order. We call this way of conforming the information set *independent* or *binomial* sampling. Although the values $Y_1 \geq \cdots \geq Y_N$ are known to the DM from the beginning, upon seeing an item the DM only knows its relative ranking within the items revealed so far.[9] Thus, only after observing the last item the DM can certainly know which item is associated to each value. The DM has to select a single item with the goal of maximizing its expected value. To allow comparison between different values of N, we think about an infinite sequence Y. For instances of size N, the values are given by the first N components of Y, which we denote by $Y_{[N]}$.[10] In the unlikely event that the online set is empty (i.e., all N items are sampled), we give the DM a default reward of

[9] We are not assuming that all values Y_i are different, but we assume that there is an arbitrary tie-breaking rule that is consistent with the relative ranks revealed and selected before the process starts.

[10] This problem, with non-negative values, and without the sampling phase, was introduced by Mucci [104].

Y_{N+1}, the next value in the infinite sequence Y.[11] This is also the reward obtained if the decision maker selects no item, and since $Y_N \geq Y_{N+1}$ the decision maker is always better off selecting an item before the end of the process. When it is clear that we are working with an instance of N items, we drop the subscript $[N]$ for ease of notation.

Note that we do not assume that the values are non-negative, and the sequence may even diverge to $-\infty$. This model, as simple as it is, turns out to be quite general. Indeed, even when $p = 0$, it manages to capture several problems that have been exhaustively studied in the literature, including:

- *Secretary problem [99].* In this classic problem, a decision maker is presented N values in an online fashion. The goal of the DM is to maximize the probability of selecting the item with the highest value. This is obtained by setting $Y_1 = 1$ and $Y_\mathrm{i} = 0$ for $\mathrm{i} \geq 2$.
- $(1, K)$*-secretary problem [76].* In this variant of the standard secretary problem, the goal of the decision maker is to maximize the probability of selecting one of the top K valued items. This is captured by the model by setting $Y_\mathrm{i} = 1$ for $\mathrm{i} = 1, \ldots, K$ and $Y_\mathrm{i} = 0$ for $\mathrm{i} \geq k + 1$.
- *Rank minimization problem [32].* In this problem, the goal of the decision maker is to minimize the expected rank of the selected item among the N items. This is captured by setting $Y_\mathrm{i} = -\mathrm{i}$ for $\mathrm{i} \geq 1$.

For any given p, we use ALG to refer to a specific (possibly randomized) algorithm or stopping rule. We use $\mathrm{ALG}(Y_{[N]})$ to denote the random variable that equals value of the the item selected by ALG on instance $Y_{[N]}$. For a given sequence Y and number of items N, our objective is to find an algorithm that maximizes $\mathbf{E}(\mathrm{ALG}(Y_{[N]}))$, where the expectation is taken over the randomness of the process and the inner randomness of the algorithm. A consequence of our results is that (for fixed Y) the decision to stop should only rely on the relative ranking of an item among those that have been revealed. Thus, whenever we see an item which is ranked ℓ among the i items seen so far, we say that that item is an ℓ-local maximum.

1.2.2 p-DOS with Adversarial Values.

We also study the the variant of p-DOS where the values Y_j are chosen by an adversary and are unknown to the decision maker. In this variant, our goal is to maximize the ratio of the reward obtained by the algorithm and the expected maximum in the online set. On an instance $Y_{[N]}$ we denote the expected maximum in the online set by $\mathbf{E}(\mathrm{OPT}(Y_{[N]}))$. As we are maximizing over a competitive ratio, we will restrict the adversary to select only non-negative values for the items. For instances of N items, we want to maximize $\beta_{N,p}$, defined as

$$\beta_{N,p} = \sup_{\mathrm{ALG} \in \mathcal{A}_N} \inf_{Y \text{ decreasing}} \frac{\mathbf{E}(\mathrm{ALG}(Y_{[N]}))}{\mathbf{E}(\mathrm{OPT}(Y_{[N]}))},$$

where $\mathcal{A}_N$ is the set of algorithms for p-DOS. A simple coupling argument verifies that for any $0 \leq p < 1$, $\beta_{N,p}$ is decreasing in N, so for any p the worst case will be when N is large.

[11] This is not very relevant, as it does not change our optimization problem: there is no decision to be made when the online set is empty. Also, we will focus primarily on the case where N is large, making this event highly unlikely.

With this in mind, we wish to find the value of

$$\beta(p) = \lim_{N\to\infty} \beta_{N,p}. \tag{1.1}$$

The guarantee of $\beta_{N,p}$ for p-DOS translates directly to the same guarantee for the i.i.d. prophet inequality with samples[12]. Indeed, for this purpose we can simply use an algorithm for p-DOS that relies only on relative rankings. Note that if we condition on the realizations of the values we obtain an instance of p-DOS and, since the algorithm does not change its behavior depending on the actual values, the guarantee holds realization by realization.

1.2.3 Dependent Sampling

In order to obtain our results for our independent sampling problem, we study the dependent sampling variant of p-DOS. This model was first introduced by Kaplan et al. [82]. In this problem, the information set consists of $h = \lfloor p \cdot N \rfloor$ items with probability 1, with each item being equally likely to be sampled. An equivalent way to think of this problem is that the N items are shuffled according to a random permutation, and the first h items belong to the information set. In addition, the order of the remaining $N - h$ items is determined by the permutation. For fixed p, we will focus on the limit of the problem as $N \to \infty$. Formally, we study

$$\alpha_{N,p} = \sup_{\mathrm{ALG}\in\bar{\mathcal{A}}_N} \inf_{Y \text{ decreasing}} \frac{\mathbf{E}(\mathrm{ALG}(Y_{[N]}))}{\mathbf{E}(\mathrm{OPT}(Y_{[N]}))},$$

where $\bar{\mathcal{A}}_N$ is the set of algorithms for the dependent sampling variant of p-DOS. Analogously as before, we define

$$\alpha(p) = \lim_{N\to\infty} \alpha_{N,p}$$

As we establish in Section 1.4.4, for all $0 \leq p < 1$ we have that $\alpha(p) = \beta(p)$.

1.3 Known Values

In this section we find the optimal algorithm for the p-DOS problem with known Y. In order to do this we present, for any amount of items in the information set, a linear program formulation whose optimal solution maps to an optimal algorithm. We then proceed to take the limit of this linear program as N goes to infinity, and reveal the structure of the limit problem. This structure allows us to rewrite the problem as that of optimizing a relatively simple real function. We conclude by showing that our approach is able to easily handle a number of classic optimal stopping problems.

[12]In the i.i.d. prophet inequality with samples we initially observe m independent samples from an unknown distribution. Then, we are given, one by one, a sequence of n values drawn independently from the same unknown distribution. After seeing each value, we must irrevocably decide whether to stop or continue. We say a stopping rule is a c-approximation if the expectation of the value it stops with is at least c times the expectation of the maximum value in the sequence.

1.3.1 Linear Programming Formulation

We present here a linear program formulation for our problem, inspired by Buchbinder et al. [18]. This linear program depends on the input instance Y and we denote it by $\mathrm{LP}_{h,N}(Y)$. Its objective function equals the expected value of an optimal algorithm for our problem, given that the information set H contains exactly h items. In the linear program, variable $x_{\mathrm{i},\ell}$ should be interpreted as the probability that the corresponding algorithm stops at step i *and* the item revealed at step i is ranked ℓ highest among the i items observed so far.

$$
\begin{aligned}
(\,\mathrm{LP}_{h,N}(Y)\,) \qquad \max_{x} \quad & Y_{N+1}\cdot\left(1-\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{\mathrm{i}} x_{\mathrm{i},\ell}\right)+\sum_{j=1}^{N} Y_j \sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{j}\frac{\mathrm{i}x_{\mathrm{i},\ell}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}} \\
\text{s.t.} \quad & \mathrm{i}x_{\mathrm{i},\ell}+\sum_{j=h+1}^{\mathrm{i}-1}\sum_{s=1}^{j} x_{j,s}\le 1 \qquad \forall \mathrm{i}\in[N]\setminus[h], \forall \ell\in[\mathrm{i}], \qquad (1.2)\\
& x_{\mathrm{i},\ell}\ge 0 \qquad \forall \mathrm{i}\in[N]\setminus[h], \forall \ell\in[\mathrm{i}].
\end{aligned}
$$

The idea behind this linear program is that constraint (1.2) forms a polyhedron rich enough to contain all relevant algorithms for the problem, and we can express the reward of the algorithm in terms of the LP variables. We call constraint (1.2) the feasibility constraint. This linear program presents three main differences with respect to that of Buchbinder et al. [18]. The first is that the objective function includes arbitrary values Y_{i}. In particular we include the additional term $Y_{N+1}\cdot\left(1-\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{\mathrm{i}} x_{\mathrm{i},\ell}\right)$ which forces the algorithm to stop, since in the event of not stopping an algorithm gets Y_{N+1} which is not better than having stopped in the last item. This additional term is important because values may be negative. The second is that linear program variables $x_{\mathrm{i},\ell}$ start at index $\mathrm{i}=h+1$. This difference reflects the fact that the first h items will conform the information set, and thus can not be selected. The third difference is that in Buchbinder et al.'s linear program, variables have the form $x_{\mathrm{i}|\ell}$, which represent instead the probability that the algorithm selects the i-th item *given* than the i-th item is ranked ℓ among the i items seen so far. This difference does not change the linear program as there exists a bijection between the solutions[13] given by $x_{\mathrm{i}|\ell}=\mathrm{i}x_{\mathrm{i},\ell}$.

The equivalence between solving the LP and finding an optimal algorithm is roughly as follows. Let us start by the inclusion of optimizing over algorithms in solving the LP. For any algorithm ALG, given that the information set contains h items, we can compute $x_{\mathrm{i},\ell}$: the probability that the algorithm stops at step i and the i-th item is ranked ℓ among the items seen so far. As the algorithm will only see ranks, this does not depend on the values Y_j. These probabilities $x_{\mathrm{i},\ell}$ will be feasible in the polyhedron. Moreover, we can write

$$\mathbb{P}(\mathrm{ALG}(Y)=Y_j)=\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{\mathrm{i}}\frac{\mathrm{i}x_{\mathrm{i},\ell}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}.$$

[13]This relation is obtained by simply conditioning on the event that the i-th item is ranked ℓ among the i items seen so far, which has probability 1/i.

This way, we can express the expected reward of the algorithm as a linear function of probabilities $x_{\mathrm{i},\ell}$.

For the other inclusion, we see that any feasible solution x can be converted into an algorithm that can be applied when the information set consists of h items. We call this algorithm ALG_x, and it works as follows. Let the first step be $h+1$ (representing that at the first step we have already seen h items from the history set). At each step i, stop with probability $\mathrm{i}x_{\mathrm{i},\ell}/(1-\sum_{j=1}^{\mathrm{i}-1}\sum_{\ell=1}^{j} x_{j,s})$ if the current item is ranked ℓ among the items seen so far. The probability that ALG_x stops at the i-th item and the i-th item is ranked ℓ among the i items seen so far is precisely $x_{\mathrm{i},\ell}$, which concludes the inclusion of solving the LP in finding an optimal algorithm.

Lemma 1.1 formalizes the previous discussion. The proof is essentially the same as the proofs in Buchbinder et al. [18], but for the sake of completeness we provide it here as well. This result says that the optimal algorithm for sequence Y with N items is to observe h and respond using ALG_x with x being the optimal solution of $\mathrm{LP}_{h,N}$.

Lemma 1.1. *Conditional on the information set containing exactly h items*

1. *For any algorithm* ALG, *denote by* $x_{\mathrm{i},\ell}$ *the probability that* ALG *stops at step* i *and the* i*-th item is ranked* ℓ *among the* i *items seen so far. Then* x *is feasible in* $\mathrm{LP}_{h,N}$ *and the objective function evaluated at* x *equals the expected reward of* ALG.

2. *The probability that* ALG_x *stops at the* i*-th item and the* i*-th item is ranked* ℓ *among the* i *items observed so far is given by* $x_{\mathrm{i},\ell}$. *The expected reward of* ALG_x *is equal to the objective value of* x.

Proof. If we condition on the information set containing exactly h items, then we can interpret the process as follows. At the beginning, values Y_j are shuffled according to a random permutation σ. That is, $\sigma(\mathrm{i}) = j$ means that the i-th item in the permutation has value Y_j. The items in the information set will be the first h items according to the permutation (i.e., $Y_{\sigma(1)}, \ldots, Y_{\sigma(h)}$). The online set will consist of the remaining items, which will be revealed according to the order of the permutation. That is, the order is $Y_{\sigma(h+1)}, Y_{\sigma(h+2),\ldots,Y_{\sigma(N)}}$.

For proving the first statement of the lemma, note that

$$
\begin{aligned}
x_{\mathrm{i}\ell} &= \mathbb{P}(\text{ALG stops at step i} \wedge\ Y_{\sigma(\mathrm{i})} \text{ is } \ell - \text{local maximum}) \\
&= \mathbb{P}(\text{ALG stops at step i}|\ Y_{\sigma(\mathrm{i})} \text{ is } \ell - \text{local maximum})\mathbb{P}(Y_{\sigma(\mathrm{i})} \text{ is } \ell - \text{local maximum}) \\
&\leq \mathbb{P}(\text{ALG does not stop before step i}|\ Y_{\sigma(\mathrm{i})} \text{ is } \ell - \text{local maximum})\mathbb{P}(Y_{\sigma(\mathrm{i})} \text{ is } \ell - \text{local maximum}) \\
&= \mathbb{P}(\text{ALG does not stop before step i})\mathbb{P}(Y_{\sigma(\mathrm{i})} \text{ is } \ell - \text{local maximum}) \\
&= \left(1 - \sum_{j=h+1}^{\mathrm{i}-1}\sum_{s=1}^{j} x_{j,s}\right)\frac{1}{\mathrm{i}}.
\end{aligned}
$$

Now, for any $1 \le j \le N$, we can write

$$\begin{aligned}\mathbb{P}(\text{ALG} = Y_j) &= \sum_{\mathrm{i}=h+1}^{N} \mathbb{P}(\text{ALG} = Y_j \wedge \text{ALG stops at step i}) \\ &= \sum_{\mathrm{i}=h+1}^{N} \mathbb{P}(Y_{\sigma(\mathrm{i})} = Y_j \wedge \text{ALG stops at step i}) \\ &= \sum_{\mathrm{i}=h+1}^{N} \mathbb{P}(\text{ALG}_x \text{ stops at step i} | Y_{\sigma(\mathrm{i})} = Y_j)\mathbb{P}(Y_{\sigma(\mathrm{i})} = Y_j).\end{aligned}$$

Since σ is a uniform random permutation, we have that $\mathbb{P}(Y_{\sigma(\mathrm{i})} = Y_j) = 1/N$. For computing $\mathbb{P}(\text{ALG stops at step i}|Y_{\sigma(\mathrm{i})} = Y_j)$ we rename the following events:

- $A_{\mathrm{i}} = \{\text{ALG stops at step i}\}$,
- $B_{\mathrm{i}\ell} = \{Y_{\sigma(\mathrm{i})} \text{ is } \ell\text{−local maximum}\}$, and
- $C_{\mathrm{i}j} = \{Y_{\sigma(\mathrm{i})} = Y_j\}$,

and write

$$\mathbb{P}(A_{\mathrm{i}}|C_{\mathrm{i}j}) = \sum_{\ell=1}^{\mathrm{i}} \mathbb{P}(A_{\mathrm{i}}|C_{\mathrm{i}j} \wedge B_{\mathrm{i}\ell})\mathbb{P}(B_{\mathrm{i}\ell}|C_{\mathrm{i}j}) = \sum_{\ell=1}^{\mathrm{i}} \mathbb{P}(A_{\mathrm{i}}|B_{\mathrm{i}\ell})\mathbb{P}(B_{\mathrm{i}\ell}|C_{\mathrm{i}j}) = \sum_{\ell=1}^{\mathrm{i}} \mathrm{i}x_{\mathrm{i},\ell}\mathbb{P}(B_{\mathrm{i}\ell}|C_{\mathrm{i}j}),$$

where the second equality holds because ALG_x decides whether to stop at step i based only on the relative order within the first i items. The third equality comes from the fact that $\mathbb{P}(A_{\mathrm{i}}|B_{\mathrm{i}\ell}) = \frac{\mathbb{P}(A_{\mathrm{i}} \wedge B_{\mathrm{i}\ell})}{\mathbb{P}(B_{\mathrm{i}\ell})} = \mathrm{i}x^{\text{ALG}}_{\mathrm{i},\ell}$, where $\mathbb{P}(B_{\mathrm{i}\ell}) = 1/\mathrm{i}$ because σ is a uniform random permutation.

To compute $\mathbb{P}(B_{\mathrm{i}\ell}|C_{\mathrm{i}j})$, notice this is the probability that Y_j is ℓ-local maximum conditional on $\sigma(j) = \mathrm{i}$. Now, this happens if out of the $j-1$ values that are larger than Y_j, exactly ℓ arrive within the first $\mathrm{i}-1$ positions. Since, conditional on $\sigma(j) = \mathrm{i}$, σ is a random permutation of the other $N-1$ items, we have that

$$\mathbb{P}(B_{\mathrm{i}\ell}|C_{\mathrm{i}j}) = \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}.$$

Putting together the computed probabilities we conclude that

$$\mathbb{P}(\text{ALG} = Y_j) = \sum_{\mathrm{i}=h+1}^{N} \sum_{\ell=1}^{\mathrm{i}} \frac{\mathrm{i}x_{\mathrm{i},\ell}}{N} \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}, \tag{1.3}$$

so the first statement follows.

To prove the second statement, first notice that as x satisfies the feasibility constraint, then ALG_x is well defined in the sense that $\frac{\mathrm{i}x_{\mathrm{i},\ell}}{1-\sum_{j=1}^{\mathrm{i}-1}\sum_{s=1}^{j} x_{j,s}}$ will always be between 0 and 1. We need to prove that the probability that ALG_x stops at step i and $Y_{\sigma(\mathrm{i})}$ is ℓ-local maximum is precisely $x_{\mathrm{i},\ell}$. This will be done by induction on i, defining the following events:

- $A_{\mathrm{i}} = \{\mathrm{ALG}_x \text{ stops at stage i}\}$,
- $B_{\mathrm{i},\ell} = \{Y_{\sigma(\mathrm{i})} \text{ is } \ell-\text{local maximum}\}$, and
- $R_{\mathrm{i}} = \{\mathrm{ALG}_x \text{ reaches stage i}\} = \{\mathrm{ALG}_x \text{ does not stop in steps } h+1, \dots, \mathrm{i}-1\}$.

The base case is $\mathrm{i} = h+1$ and any $1 \le \ell \le h+1$. Here, we have that $\mathbb{P}(R_{h+1}) = 1$, so

$$
\begin{aligned}
(h+1)x_{h+1,\ell} &= \mathbb{P}(A_{h+1} | R_{h+1} \wedge B_{h+1,\ell}) \\
&= \mathbb{P}(A_{h+1} | B_{h+1,\ell}) = \mathbb{P}(A_{h+1} \wedge B_{h+1,\ell}) / \mathbb{P}(B_{h+1,\ell}) \\
&= (h+1)\mathbb{P}(A_{h+1} \wedge B_{h+1,\ell})
\end{aligned}
$$

and we obtain the result by cancelling the $(h+1)$. For $\mathrm{i} > h+1$ and $1 \le \ell \le \mathrm{i}$ we have that

$$
\begin{aligned}
\mathbb{P}(A_{\mathrm{i}} \wedge B_{\mathrm{i},\ell}) &= \mathbb{P}(A_{\mathrm{i}} \wedge B_{\mathrm{i},\ell} \wedge R_{\mathrm{i}}) \\
&= \mathbb{P}(A_{\mathrm{i}} | B_{\mathrm{i},\ell} \wedge R_{\mathrm{i}}) \mathbb{P}(B_{\mathrm{i},\ell} \wedge R_{\mathrm{i}}) \\
&= \mathbb{P}(A_{\mathrm{i}} | B_{\mathrm{i},\ell} \wedge R_{\mathrm{i}}) \mathbb{P}(B_{\mathrm{i},\ell}) \mathbb{P}(R_{\mathrm{i}}),
\end{aligned}
$$

where the first equality comes from the fact that A_{i} is contained in R_{i} and the last equality comes from the fact that ALG_x cannot use the ranking of $Y_{\sigma(\mathrm{i})}$ to stop in a stage before i. By the construction of ALG_x, we have that $\mathbb{P}(A_{\mathrm{i}} | B_{\mathrm{i},\ell} \wedge R_{\mathrm{i}}) = \frac{\mathrm{i} x_{\mathrm{i}\ell}}{1 - \sum_{j=h+1}^{\mathrm{i}-1} \sum_{s=1}^{h} x_{j,s}}$. As σ is a random and uniform permutation, we have that $\mathbb{P}(B_{\mathrm{i},\ell}) = 1/\mathrm{i}$ for any $1 \le \ell \le \mathrm{i}$. The only thing left to conclude is computing $\mathbb{P}(R_{\mathrm{i}})$. For this we compute

$$
\begin{aligned}
\mathbb{P}(R_{\mathrm{i}}) = 1 &- \sum_{j=h+1}^{\mathrm{i}-1} \mathbb{P}(\text{Stop at step } j) \\
&= 1 - \sum_{j=h+1}^{\mathrm{i}-1} \sum_{s=1}^{j} \mathbb{P}(\text{Stop at step } j \wedge Y_{\sigma(j)} \text{ is } \ell-\text{local maximum}) = 1 - \sum_{j=h+1}^{\mathrm{i}-1} \sum_{s=1}^{j} x_{j,s}
\end{aligned}
$$

where the last equality holds because of our inductive hypothesis. It follows that the probability that ALG_x stops at step i and $Y_{\sigma(\mathrm{i})}$ is ℓ-local maximum is $x_{\mathrm{i},\ell}$. The second statement follows, as equation (1.3) holds for any algorithm, in particular for ALG_x. □

We have seen that in the objective function, coefficients accompanying Y_j are equal to the probability that the ALG_x selects the item with value Y_j. The following equivalent expression of the objective function is useful for establishing our results:

$$\begin{aligned}
\mathbf{E}\big(\mathrm{ALG}_x(Y) \;\big|\; |H|=h\big) &= \sum_{k=1}^{N+1} Y_k \mathbb{P}\big(\mathrm{ALG}_x(Y)=Y_k \;\big|\; |H|=h\big)\\
&= \sum_{k=1}^{N} (Y_k - Y_{k+1}) \mathbb{P}\big(\mathrm{ALG}_x(Y) \geq Y_k \;\big|\; |H|=h\big)\\
&\quad + Y_{N+1}\mathbb{P}\big(\mathrm{ALG}_x(Y) \geq Y_{N+1} \;\big|\; |H|=h\big)\\
&= \sum_{k=1}^{N} (Y_k - Y_{k+1}) \sum_{j=1}^{k}\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{j} \frac{\mathrm{i}x_{\mathrm{i},\ell}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}} + Y_{N+1}\\
&= Y_1 - \sum_{k=1}^{N} (Y_k - Y_{k+1}) \left(1 - \sum_{j=1}^{k}\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{j} \frac{\mathrm{i}x_{\mathrm{i},\ell}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}\right)\\
&= Y_1 - \sum_{k=1}^{N} (Y_k - Y_{k+1}) \left(1 - \sum_{\ell=1}^{k}\sum_{\mathrm{i}=h+1}^{N} x_{\mathrm{i},\ell}\sum_{j=\ell}^{k} \frac{\mathrm{i}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}\right),
\end{aligned} \tag{1.4}$$

where we use the fact that $\mathbb{P}\big(\mathrm{ALG}_x \geq Y_{N+1} \;\big|\; |H|=h\big) = 1$, and that $Y_{N+1} = Y_1 - \sum_{k=1}^{N}(Y_k - Y_{k+1})$.

1.3.2 Limit Problem

Consider an infinite sequence Y, a number of items $N+1$, and an algorithm ALG. This same algorithm can be implemented on the same infinite sequence Y with the only first N items by inserting a dummy item, ranked worst among all items, and running ALG on this artificial instance. If ALG would choose the dummy item, it simply does not stop in the real instance. The reward collected by applying this tweaked algorithm to $Y_{[N]}$ is not less than what ALG collects from $Y_{[N+1]}$. This simple coupling argument, formalized in Section 1.6.1, implies that

$$\max_{\mathrm{ALG}\in\mathcal{A}_N} \mathbf{E}\left(\mathrm{ALG}\left(Y_{[N]}\right)\right) \geq \max_{\mathrm{ALG}\in\mathcal{A}_{N+1}} \mathbf{E}\left(\mathrm{ALG}\left(Y_{[N+1]}\right)\right).$$

This means that as $N\to\infty$ the sequence of these maxima either converges or diverges to $-\infty$. We obtain the limit of the sequence analyzing the limit of the linear programs $\mathrm{LP}_{\lfloor pN\rfloor,N}$. This can be done by performing a Riemann sum analysis, which captures the cases where the limit value exists. Denote by $L^1([p,1]\times\mathbb{N})$ the space of measurable functions $q:[p,1]\times\mathbb{N}\to\mathbb{R}$ such that $\sum_{\ell=1}^{\infty}\int_p^1 |q(t,\ell)|\mathrm{d}t<\infty$. If for $q\in L^1([p,1]\times\mathbb{N})$, we define the function

$$F_k(q) = \sum_{\ell=1}^{k}\int_p^1 q(t,\ell)\sum_{j=\ell}^{k}\binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}\mathrm{d}t, \tag{1.5}$$

we can write the following limit problem, CLP_p, where we have dropped the dependency on Y for ease of notation.

$$
\begin{aligned}
(\mathrm{CLP}_p) \quad & \sup_{q \in L^1([p,1]\times\mathbb{N})} && Y_1 - \sum_{k\geq 1}(Y_k - Y_{k+1})\left(1 - F_k(q)\right) && \\
\text{s.t.} \quad & && tq(t,\ell) + \int_p^t \sum_{s\geq 1} q(\tau,s)\mathrm{d}\tau \leq 1 && \forall t \in [p,1], \forall \ell \geq 1 \\
& && q(t,\ell) \geq 0 && \forall t \in [p,1], \forall \ell \geq 1
\end{aligned}
$$

By standard arguments (see Section 1.6.2), for every $p \in [0,1)$ we can show that the limit of $\max_{\mathrm{ALG}\in\mathcal{A}_N} \mathbf{E}\left(\mathrm{ALG}\left(Y_{[N]}\right)\right)$, when $N \to \infty$, exists if and only if the optimal value of CLP_p is finite, and they are equal.

This limit problem has a natural interpretation as a continuous-time version of p-DOS. In this problem there are countably many items, each item has a uniform arrival time in the interval $[0,1]$, and each item is in the history set H if it arrives before time p and in the online set otherwise. We observe the items in H, then we scan the interval $[p,1]$ and when we reach the arrival time of an item, we irrevocably decide whether we should stop. The decision variables in CLP_p can be interpreted as encoding this decision. For a time τ and an integer s, $q(\tau,s)\mathrm{d}\tau$ is the probability that we stop with an s-local maximum in the interval $[\tau, \tau+\mathrm{d}\tau]$. In the objective function, $F_k(q)$ is the probability that we select an item with global rank k or better.

1.3.3 Structure of Optimal Solution

We show that CLP_p can be restricted to solutions with a very special structure. If we interpret CLP_p as the problem of selecting an item from an infinite set with uniform arrival times in $[0,1]$, we essentially prove that an optimal solution can be attained in the following class of algorithms. Given a non-decreasing sequence $\{t_\mathrm{i}\}_{\mathrm{i}\in\mathbb{N}} \subseteq [p,1]$, if at time τ we receive an item that is an ℓ-local maximum, we accept it if $t_\ell \leq \tau$. Thus, we reject everything arriving in $[p, t_1)$, then in $[t_1, t_2)$ we only accept a value that is the best so far, in $[t_2, t_3)$ we only accept a value that is best or second best, and so on. Formally we prove the following theorem.

Theorem 1.2. *For a fixed $p \in [0,1)$ and a given feasible solution q for CLP_p, there exists another feasible solution q^* such that $F_k(q^*) \geq F_k(q)$ for all $k \geq 1$, and there is a non-decreasing sequence of numbers $\{t_\mathrm{i}\}_{\mathrm{i}\in\mathbb{N}} \subseteq [p,1]$, with $t_0 = p$, that satisfies that for all $\ell \in \mathbb{N}, t \in [p,1]$,*

$$
tq^*(t,\ell) + \int_p^t \sum_{s\geq 1} q^*(\tau,s)\mathrm{d}\tau = 1, \quad \text{if } t \geq t_\ell \tag{1.6}
$$

$$
q^*(t,\ell) = 0, \quad \text{if } t < t_\ell. \tag{1.7}
$$

Moreover, for all $t \in [p,1]$, we have that

$$
q^*(t,\ell) = \begin{cases} \dfrac{T_\mathrm{i}}{t^{\mathrm{i}+1}} & \text{if } t \in [t_\mathrm{i}, t_{\mathrm{i}+1}), \ell \leq \mathrm{i} \\ 0 & \text{else,} \end{cases} \tag{1.8}
$$

where $T_{\mathrm{i}} = \prod_{j=1}^{\mathrm{i}} t_j$.

Proof. The proof is done in two steps. The first is to show that we can modify q without decreasing $F_k(q)$ to obtain a solution that satisfies Equations (1.6) and (1.7). The second is to prove that if a solution satisfies Equations (1.6) and (1.7), then it is actually as in Equation (1.8).

A key ingredient is to study the term accompanying $q(t,\ell)$ in $F_k(q)$. Note that the term is either 0, if $\ell > k$, or it is

$$\sum_{j=\ell}^{k} \binom{j-1}{\ell-1} (1-t)^{j-\ell} t^{\ell},$$

if $\ell \leq k$. The property that we will extensively use is that this term is increasing in t and decreasing in ℓ. This is implied by the fact that it corresponds to the probability that a NegativeBinomial(ℓ,t)[14] random variable is at most k. For completeness, an arithmetic proof of this fact can be found in Section 1.6.3. Then, we use these facts to argue that if we take a solution that is not as in the Theorem, we can modify it without reducing the objective value.

We recursively define a sequence of solutions $(q_n)_{n\geq 0}$ as follows. We start with an arbitrary feasible solution $q_0 = q$ for CLP$_p$. If q_{n-1} is a feasible solution, we have that

$$q_{n-1}(t,\ell) \leq \frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_{n-1}(\tau,s)\,\mathrm{d}\tau\right), \quad \forall t \in [p,1], \ell \geq 1.$$

Note also that $\frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_{n-1}(\tau,s)\,\mathrm{d}\tau\right)$ is non-negative for all t,ℓ so there must exist a value $t_\ell(n) \in [p,1]$ such that

$$\int_p^1 q_{n-1}(t,\ell)\,\mathrm{d}t = \int_{t_\ell(n)}^1 \frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_{n-1}(\tau,s)\,\mathrm{d}\tau\right)\mathrm{d}t.$$

Thus, we define q_n as

$$q_n(t,\ell) = \begin{cases} \frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_{n-1}(\tau,s)\,\mathrm{d}\tau\right) & \text{if } t \geq t_\ell(n) \\ 0 & \text{if } t < t_\ell(n). \end{cases}$$

Now we prove a few facts about q_n. First, note that for all $\ell \geq 1$, $\int_p^1 q_n(t,\ell)\mathrm{d}t = \int_p^1 q_{n-1}(t,\ell)\mathrm{d}t$. Also note that we are only moving mass to the right, and therefore,

$$\frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_{n-1}(\tau,s)\,\mathrm{d}\tau\right) \leq \frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_{n}(\tau,s)\,\mathrm{d}\tau\right), \quad \forall t \in [p,1]. \tag{1.9}$$

This implies that if q_{n-1} is feasible, q_n is also feasible. Also since we are only moving mass to the right, and since the term accompanying $q(t,\ell)$ in $F_k(q)$ is increasing in t, necessarily

[14]The number of coin tosses necessary to obtain ℓ heads, if the coin lands heads with probability t.

$F_k(q_{n-1}) \leq F_k(q_n)$ for all $k \geq 1$. Moreover, notice Equation (1.9) also implies that $t_\ell(n) \leq t_\ell(n+1)$ for all $\ell \geq 1, n \geq 1$. Since these numbers are upper bounded by 1, they must converge to some values $t_\ell(\infty) \in [p, 1]$.

We now prove that for each $\ell \geq 1$ the sequence $(q_n(\cdot, \ell))_{n\geq 1}$ has a pointwise limit $q_\infty(\cdot, \ell)$, to which it also converges under the L^1 norm. Note first that if $q_0 = q_1$, the sequence is constant and therefore it trivially converges. If $q_0 \neq q_1$, then Equation (1.9) for $n = 1$ holds with strict inequality in some interval $[\tau_1, \tau_2] \subseteq [p, 1]$, and then, if $t_\ell(1) < \tau_2$ for some $\ell \geq 1$, necessarily $t_\ell(1) < t_\ell(2)$. By evaluating the feasibility constraint in $t = 1$, we have that $\sum_{s\geq 1} \int_p^1 q_0(\tau, s) \mathrm{d}\tau \leq 1$, so there is some s^* such that $\int_p^1 q_0(\tau, s^*) \mathrm{d}\tau = \max_{s\geq 1} \int_p^1 q_0(\tau, s) \mathrm{d}\tau$. By the definition of $t_\ell(n)$, we have that $t_{s^*}(n) = \min_{s\geq 1} t_s(n)$ for all $n \geq 1$. Then, if $q_0 \neq q_1$, $0 \leq t_{s^*}(1) < t_{s^*}(2) \leq t_\ell(n)$ for all $\ell \geq 1, n \geq 2$. Now, the sequence of functions

$$G_n(t) = \frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_n(\tau, s)\, \mathrm{d}\tau\right)$$

is monotone in n, so it has a pointwise limit. Now, from the definition of $q_n(t, \ell)$, this also implies that $q_n(t, \ell)$ has a pointwise limit $q_\infty(t, \ell)$ when $n \to \infty$. Indeed, for $t < t_\ell(\infty)$, eventually $t < t_\ell(n)$ because $t_\ell(n) \nearrow t_\ell(\infty)$, and then $q_n(t, \ell)$ becomes 0; and for $t \geq t_\ell(\infty)$, $q_n(t, \ell) = G_n(t)$, which has a pointwise limit. Since $t_\ell(\infty) \geq t_{s^*}(2) > 0$, there is some n_0 such that $t_\ell(n) \geq t_{s^*}(2)/2$ for all $n \geq n_0$ and then $q_n(t, \ell)$ is dominated by the constant function equal to $2/t_{s^*}(2)$, which is integrable, so by the dominated convergence theorem, it converges to $q_\infty(t, \ell)$ in $L^1([p, 1])$. This is sufficient to conclude that $F_k(q_\infty) \geq F_k(q_0)$ for all $k \geq 1$, because F_k is a continuous function of q and involves only the first k components of q.

We have now that

$$q_\infty(t, \ell) = \begin{cases} \frac{1}{t}\left(1 - \int_p^t \sum_{s\geq 1} q_\infty(\tau, s)\, \mathrm{d}\tau\right) & \text{if } t \geq t_\ell(\infty) \\ 0 & \text{if } t < t_\ell(\infty). \end{cases}$$

The only missing piece is the monotonicity of $t_\ell(\infty)$. In fact, they are not necessarily monotone. However, note that swapping components of q_∞ does not affect its feasibility. Since the term accompanying $q(t, \ell)$ in $F_k(q)$ is decreasing in ℓ, for all $k \geq 1$, we can swap components of q_∞ to obtain a function q^* and a sequence $(t_\ell)_{\ell\geq 1}$ such that $t_\ell \leq t_{\ell+1}$, that satisfies Equations (1.6) and (1.7).

For the second part of the proof of the theorem we first prove that, given the sequence $(t_\ell)_{\ell\geq 1}$, Equations (1.6) and (1.7) admit a unique solution. Then we prove that they are satisfied by the one given in Equation (1.8). In fact, notice that for any ℓ, in the interval $[t_\ell, t_{\ell+1}]$ all functions $q(t, \ell')$ with $\ell' \leq \ell$ are equal, and the rest are 0. Thus, denoting this function by $y_\ell(t)$, we can rewrite Equation (1.6) as follows.

$$y_\ell'(t) = -\frac{(\mathrm{i}+1)}{t} \cdot y_\ell(t), \quad \forall t \in [t_\ell, t_{\ell+1}]. \tag{1.10}$$

Again by Equation (1.6), we have that the function has to satisfy a continuity constraint $y_\ell(t_\ell) = \frac{1}{t_\ell}\left(1 - \int_p^{t_\ell} \sum_{s\geq 1} q^*(\tau, s) \mathrm{d}\tau\right)$, which depends only on previous intervals, and for $\ell = 1$

it evaluates as 0. This determines the initial value in the interval. Therefore, by the Cauchy-Lipschitz theorem, Equations (1.6) and (1.7) admit a unique solution.

We are ready now to check that the function defined by Equation (1.8) satisfies our equations. In fact, it is easy to check the continuity, by noticing that $T_{\mathrm{i}}/t_{\mathrm{i}+1}^{\mathrm{i}+1} = T_{\mathrm{i}+1}/t_{\mathrm{i}+1}^{\mathrm{i}+2}$. Replacing in Equation (1.10) we obtain

$$-(\ell+1)\frac{T_\ell}{t^{\ell+2}} = -\frac{\ell+1}{t}\cdot\frac{T_\ell}{t^{\ell+1}} \quad \forall t \in [t_\ell, t_{\ell+1}],$$

which clearly holds. □

Let us now apply Theorem 1.2 to simplify problem CLP_p. Noting that the differences $Y_k - Y_{k+1}$ are non-negative we can reduce the feasible set in CLP_p to just solutions satisfying Equation (1.8). These solutions automatically satisfy the constraints in CLP_p and therefore the problem reduces to one in which the optimization is done only over the t_{i}'s for $\mathrm{i} \geq 1$. To explicitly write this reduced problem, and slightly abusing notation, consider the functions $F_k : [0,1]^{\mathbb{N}} \to \mathbb{R}$ given by

$$F_k(t) = \sum_{j=1}^{k}\sum_{\mathrm{i}=1}^{\infty}\int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}}\sum_{\ell=1}^{j\wedge\mathrm{i}}\frac{T_{\mathrm{i}}}{\tau^{\mathrm{i}+1}}\binom{j-1}{\ell-1}(1-\tau)^{j-\ell}\tau^{\ell}\mathrm{d}\tau.$$

Note that $F_k(t) = F_k(q^*)$ where q^* satisfies (1.8). We obtain that the value of CLP_p equals that of the reduced problem:

$$\begin{aligned} (\mathrm{RP}_p) \qquad & \sup_{t=(t_{\mathrm{i}})_{\mathrm{i}\in\mathbb{N}}} \quad Y_1 - \sum_{k\geq 1}(Y_k - Y_{k+1})\,(1 - F_k(t)) \\ & \text{s.t.} \quad p \leq t_{\mathrm{i}} \leq t_{\mathrm{i}+1} \leq 1 \qquad\qquad \forall \mathrm{i} \geq 1. \end{aligned}$$

Straightforward (but tedious) calculations show that $F_k(t)$ is increasing in t_{i} for all $\mathrm{i} > k$, and also concave in each t_{i} (see Section 1.6.4). Unfortunately though, these $F_k(\cdot)$ are not jointly concave. Therefore, the reduced problem RP_p is a real optimization problem, which is concave on each t_{i}.

1.3.4 Finding the Optimal Thresholds

As mentioned earlier, RP_p can be interpreted as the problem of finding the optimal algorithm for a continuous version of p-DOS with an infinite sequence known values that arrive continuously in the interval $[0,1]$. A solution $\{t_{\mathrm{i}}\}_{\mathrm{i}\in\mathbb{N}}$ corresponds to the algorithm that, upon receiving at time τ an item that is an ℓ-local maximum, stops if $t_\ell \leq \tau$. The implementation of this algorithm for p-DOS with known values and N items is standard. After solving the corresponding RP_p and finding the implied optimal thresholds $\{t_{\mathrm{i}}\}_{\mathrm{i}\in\mathbb{N}}$, we sample one *arrival time*, i.e., a uniform random variable in $[0,1]$, for each of the N items. The items corresponding to arrival times that landed in $[0,p]$ are included in the information set while the remaining form the online set.[15] These are inspected in increasing order of their arrival times

[15]If the information set is already sampled and contains h items, then the procedure would be to sample $N-h$ arrival times uniform in $(p,1)$ for the items in the online set.

and the sequence $\{t_{\mathrm{i}}\}_{\mathrm{i}\in\mathbb{N}}$ dictates the stopping time as before. In Section 1.4.4 we prove that the expected reward is at least as large and converges to the objective value of RP_p as N tends to infinity. One last thing to notice is that the algorithm we just described might not stop, although this can be easily fixed by selecting the last item if no item was to be selected.

Note that this formulation RP_p already establishes a number of facts. The first interesting consequence is that, quite naturally, the optimal algorithm for p-DOS with known values is given by a sequence of thresholds $t_1 \le t_2 \le \dots$ so that after time t_{i} we accept any item whose current ranking is i or better. This fact was previously shown in some special cases by Mucci [104] and Chan et al [26]. Moreover, by exploiting properties of the objective function we can show how it leads to relatively simple real optimization problems that solve various classic single selection optimal stopping problems.

Note first that if only finitely many Y's are different –as often happens in classic optimal stopping problems– then RP_p is a finite dimensional real optimization problem. Indeed, let us assume $Y_1 \ge \dots \ge Y_m > Y_{m+1} = \dots$. Thus the objective function in RP_p becomes $\sum_{k=1}^{m}(Y_k - Y_{k+1})F_k(t) - Y_{m+1}$. Additionally, since the $F_k(t)$ are increasing in t_{i} for $\mathrm{i} > k$, all terms in the objective function are increasing in t_{i} for $\mathrm{i} > m$, so that we may set these variables to be 1. With this RP_p becomes the finite dimensional optimization problem of maximizing, over $t \in [p,1]^m$ the function

$$\sum_{k=1}^{m}(Y_k - Y_{k+1})\sum_{j=1}^{k}\sum_{\mathrm{i}=1}^{m}\sum_{\ell=1}^{j\wedge \mathrm{i}} T_{\mathrm{i}}\binom{j-1}{\ell-1}\int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}}(1-\tau)^{j-\ell}\tau^{\ell-\mathrm{i}-1}\mathrm{d}\tau.$$

This problem is concave in each variable t_{i}, since it is a non-negative linear combination of concave functions. For the problem of maximizing the probability of selecting the best item, Correa et al. [38] establish a similar characterization as a continuous optimization problem, which they prove is concave. We suspect our problem also has a unique local maximizer, so we expect that it can be solved using gradient descent methods. [16] In particular, this holds in the following examples that recover some classical results in optimal stopping.

- *Secretary problem.* Recall that the secretary problem is recovered by setting $Y_1 = 1$ and $Y_{\mathrm{i}} = 0$ for $\mathrm{i} > 1$. With this, the problem simplifies to

$$\max_{0\le t_{\mathrm{i}}<1}\sum_{\mathrm{i}=1}^{\infty}\int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}}\frac{T_{\mathrm{i}}}{\tau^{\mathrm{i}}}\mathrm{d}\tau = \max_{0\le t_1<1}\int_{t_1}^{1}\frac{t_1}{\tau}\mathrm{d}\tau = \max_{0\le t_1\le 1} -t_1\ln(t_1),$$

  where the first equality follows since, by the monotonicity property of $F_k(t)$, $t_2, t_3, \dots$ approach 1 in the supremum. The problem to the right is easily solved by taking first order conditions, so we recover the classic result that $t_1 = 1/\mathrm{e}$ and that the optimal value is $1/\mathrm{e}$.

[16] A consequence of this discussion is that given an instance of p-DOS with its corresponding Y one can, in time $O(1)$, find an sequence of thresholds leading to an arbitrarily close to optimal online algorithm. To see this first note that restricting to the first $K = O(1)$ terms in the sequence of Y's is enough. Then we can restrict to the finite version of RP_p in which only the variables $t_1, \dots, t_k$ are present. Now, for these variables we evaluate the objective function in all values belonging to a fine grid of $[0,1]^K$ and keep the best value found.

- *(1,2)-Secretary.* Here, we have that $Y_1 = Y_2 = 1$ and $Y_{\mathrm{i}} = 0$ for $\mathrm{i} > 2$. So the problem is

$$\max_{0 \le t_1 \le t_2 \le 1} t_1^2 + 2t_1(\ln(t_2/t_1) + 1) - 3t_1 t_2.$$

  First order conditions give that $t_1 \approx 0.347$ and $t_2 = 2/3$.[17] The optimal value is approximately 0.5737, which matches the bound of Gusein-Zade and Chan et al. [76, 26].

- *Minimum rank.* In this problem we seek to minimize the expected rank of the selected value, which is modeled by taking $Y_k = -k$, so that $Y_k - Y_{k+1} = 1$. Thus RP_p becomes[18]

$$\sup_{0 \le t_{\mathrm{i}} \le 1,\, \mathrm{i} \ge 1} \quad -1 + \sum_{k=1}^{\infty} (F_k(t) - 1) \quad = \sup_{0 \le t_{\mathrm{i}} \le 1,\, \mathrm{i} \ge 1} \quad -\sum_{\mathrm{i}=1}^{\infty} T_{\mathrm{i}} \frac{\mathrm{i}}{2} \left(\frac{1}{t_{\mathrm{i}}^{\mathrm{i}+1}} - \frac{1}{t_{\mathrm{i}+1}^{\mathrm{i}+1}} \right),$$

  where the equality follows by using the identity $\sum_{j=\ell}^{\infty} \binom{j}{\ell} (1-t)^j = (1-t)^{\ell} t^{-(\ell+1)}$. Again the first order optimality conditions are enough to solve the problem. Indeed, they solve for $t_{\mathrm{i}} = \prod_{m=\mathrm{i}}^{\infty} \left(\frac{m}{m+2}\right)^{1/(m+1)}$, which evaluates for an expected rank of $\prod_{m=1}^{\infty} \left(\frac{m+2}{m}\right)^{1/(m+1)} \approx 3.8695$, recovering the result of Chow et al. [32].

The case of $p > 0$

Although the examples we have recovered are all for the case $p = 0$ we note that our results hold for general p. The "right" way of taking this limit is by first normalizing the objective function value. To see this note that in RP_p (and also in p-DOS with known values) the values of Y can be scaled without affecting the optimization problem. Thus, for instance, if $p = 0$ we could scale these values to have $Y_1 = 1$ (so long as $Y_1 > 0$). This makes sense since in this situation an optimal clairvoyant algorithm will always pick Y_1 so that the objective of RP_p after this normalization represents the relative performance of the best online algorithm when compared to the optimal offline algorithm. For $p > 0$ the expected value of the optimal offline algorithm is given by $\sum_{\mathrm{i}=1}^{\infty} Y_{\mathrm{i}} p^{\mathrm{i}-1}(1-p)$. Therefore, when all Y's are non-negative, the right normalization of the objective in RP_p is to divide it by this quantity. This leads to measuring the performance of the algorithm as the ratio between the expectation of the selected value and the expectation of the highest eligible value (the maximum value among the items in the online set). For instance, in the case of the secretary problem, for $p > 1/\mathrm{e}$, the ratio equals $p \ln(1/p)/(1-p)$.

An important remark is that this normalization does not change the optimization problem, as the denominator in the ratio depends solely on the values of Y_{i} and p. However, in the next section, we consider p-DOS with adversarial values and therefore the Y_{i}'s become variables selected by an adversary. In this setting, the normalization is needed to appropriately measure the competitive ratio of an algorithm.

[17] t_1 is the only solution of equation $x - \ln(x) = 1 + \ln(3/2)$ in (0,1).

[18] To interpret the expression on the left, recall that $F_k(t)$ is the probability that the algorithm stops with an item whose rank is k or better. Thus the objective simply represents the negative of the expected rank.

1.4 Adversarial Values

Up to this point we have considered that the vector of values Y is known to the decision maker from the beginning. In what follows we will relax this assumption, and instead we will let the values to be chosen by an adversary. Our objective function will thus become a competitive ratio, as suggested at the end of the previous section. Consequently, we will restrict the adversary to select a decreasing sequence of non-negative values for the items. The analysis in this section will initially rely on the dependent sampling variant, where the information set is conformed of h items with probability 1, and each item has equal probability of belonging to it. This model leads to a cleaner linear program and its limit naturally coincides with that for the independent sampling variant.

We start by presenting the adversary's optimization problem and use von Neumann's Minmax Theorem to derive a factor revealing LP. We take the limit of this problem as $N \to \infty$ and find that our structural results of Section 1.3.3 also hold for this limit problem. Using this structural result we reduce the limit problem to finding an optimal sequence of optimal time thresholds $(t_{\mathrm{i}})_{\mathrm{i}\in\mathbb{N}}$. We solve this reduced problem, putting special emphasis on values of p within 0 and $1/\mathrm{e}$, on $p = 1/2$, and on the limit as $p \to 1$. We close the section by connecting the dependent and independent sampling models. In particular, we show that our obtained guarantees also hold for finite N in the independent sampling model, while in the dependent sampling model they hold approximately with an error $\tilde{O}(1/\sqrt{N})$ (for fixed $p < 1$).

1.4.1 Factor Revealing LP

In this subsection we present a factor-revealing linear program, whose optimal value equals the optimal competitive ratio for instances with N items and history set of size h. We start by stating our objective function, which consists of the competitive ratio just mentioned. The benchmark we will be comparing the performance of our algorithms will be the highest value among the items of the online set. Formally, our benchmark is the expectation of random variable $\mathrm{OPT}(Y_{[N]})$, defined as the highest value among the items in the online set. This way, for given integers $0 \le h < N$, we want to find the largest ratio between $\mathbf{E}(\mathrm{ALG}(Y_{[N]}))$ and $\mathbf{E}(\mathrm{OPT}(Y_{[N]}))$, for all instances $Y_{[N]}$ of N items.

The following lemma establishes the distribution of $\mathrm{OPT}(Y_{[N]})$, which will be useful for formulating $\mathrm{SDLP}_{h,N}$.

Lemma 1.3. *Consider an instance $Y_{[N]}$. Then:*

$$\mathbb{P}(\mathrm{OPT}(Y_{[N]}) = Y_j) = \begin{cases} \frac{N-h}{N-j+1} \prod_{s=0}^{j-2} \frac{h-s}{N-s} & 1 \le j \le h+1 \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. For $OPT(Y) = Y_j$ we need that all Y_{i} with $\mathrm{i} < j$ belong to the history set. The first observation is that numbers smaller than Y_{h+1} cannot be the optimum, because we would need the largest $h+1$ numbers to be in the history set, which has only h items.

For $j \le h$, as the construction of the history and the online sets are based on a random permutation, we can simulate it by sequentially inserting the numbers in N slots of which h will correspond to the history set and the remaining $N-h$ correspond to the online set. The probability that $OPT = Y_1$ is simply the probability that Y_1 lands on the online slots, i.e., $\frac{N-h}{N} = 1-p$. For Y_j with $1 < j \leq h$, we need that the largest $j-1$ values appear in H. Conditional on the largest s values are in H, the probability that Y_{s+1} is also in H is that it lands on the $h-s$ slots of H remaining among the $N-s$ remaining slots: $\frac{h-s}{N-s}$. Once all the $j-1$ largest values landed on H, then we need Y_j to land on the $N-h$ slots of O, which happens with probability $\frac{N-h}{N-j+1}$.

□

We proceed to present a factor revealing linear program for p-DOS with adversarial values and dependent sampling. For a given N, let $\mathcal{Y}_N = \{Y_{[N]} \in \mathbb{R}^N : Y_1 \geq Y_2 \geq \cdots \geq Y_N \geq 0\}$ be the set of relevant feasible values that the adversary may choose.[19] The problem for the adversary can be stated as follows:

$$\begin{aligned}
\min_{Y_{[N]} \in \mathcal{Y}_N} \max_{x} \quad & \frac{\mathbf{E}(\mathrm{ALG}_x(Y_{[N]}))}{\mathbf{E}(\mathrm{OPT}(Y_{[N]}))} \\
s.t. \quad & \mathrm{i} x_{\mathrm{i},\ell} + \sum_{j=1}^{\mathrm{i}-1} \sum_{s=1}^{j} x_{j,s} \leq 1 && \forall \mathrm{i} \in [N] \setminus [h+1], \forall \ell \in [\mathrm{i}] \\
& x_{\mathrm{i},\ell} \geq 0 && \forall \mathrm{i} \in [N] \setminus [h+1], \forall \ell \in [\mathrm{i}].
\end{aligned}$$

Since we may assume $Y_{N+1} = 0$, the expression in Section 1.3 for $\mathbf{E}(\mathrm{ALG}_x(Y_{[N]}))$ becomes

$$\mathbf{E}(\mathrm{ALG}_x(Y_{[N]})) = \sum_{j=1}^{N} Y_j \sum_{\mathrm{i}=h+1}^{N} \sum_{\ell=1}^{j} \frac{\mathrm{i} x_{\mathrm{i},\ell}}{N} \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}},$$

and for the dependent sampling variant we have $\mathbf{E}(\mathrm{OPT}(Y_{[N]})) = \sum_{j=1}^{N} Y_j \mathbb{P}(\mathrm{OPT}(Y_{[N]}) = Y_j)$. This problem is not linear, as the denominator of the objective function, $\mathbf{E}(\mathrm{OPT}(Y_{[N]}))$, depends on variables Y_j. However, note that we can arbitrarily scale Y since the scaling will cancel out in the ratio $\mathbf{E}(\mathrm{ALG}_x(Y_{[N]}))/\mathbf{E}(\mathrm{OPT}(Y_{[N]}))$. Thus, without loss of generality, we can restrict the adversary to select values such that $\mathbf{E}(\mathrm{OPT}(Y_{[N]})) = 1$. Now the objective function is linear in x and linear in $Y_{[N]}$, and also the corresponding feasible sets are convex and compact. The compactness follows from the fact that every coordinate of x must be in $[0,1]$; and that $0 \leq Y_j \leq Y_1$ for all $j \in [N]$, and by Lemma 1.3 $\frac{N-h}{N} \cdot Y_1 \leq \mathbf{E}(\mathrm{OPT}(Y_{[N]})) = 1$. Therefore, we can use von Neumann's Minmax Theorem to change the order of the minimization and the maximization. We obtain the following problem:

$$\max_{\substack{\mathrm{i} x_{\mathrm{i},\ell} + \sum_{j=1}^{\mathrm{i}-1} \sum_{s=1}^{j} x_{j,s} \leq 1, \forall \mathrm{i} \in [N] \setminus [h], \ell \in [\mathrm{i}], \\ x \geq 0}} \quad \min_{\substack{Y \in \mathcal{Y}_N \\ \mathbf{E}(\mathrm{OPT}(Y_{[N]}))=1}} \quad \mathbf{E}(\mathrm{ALG}_x(Y_{[N]})),$$

[19] We say relevant because for N items, only the first $N+1$ of sequence Y will affect the outcome for instances of N items (recall Y_{N+1} is the reward obtained if the DM makes no selection). Now the online set cannot be empty, so $\mathrm{OPT}(Y_{[N]})$ is independent of Y_{N+1}. This way, setting $Y_{N+1} = 0$ will always be optimal for an adversary minimizing the competitive ratio.

Through a stochastic dominance argument (presented in Section 1.7.1) we finally derive our factor revealing linear program which we denote by $\mathrm{SDLP}_{h,N}$, short for "Stochastic Dominance Linear Program":

$$
\begin{aligned}
(\mathrm{SDLP}_{h,N}) \qquad & \max_{x,\alpha} \quad \alpha \\
\text{s.t.} \qquad & \mathrm{i}x_{\mathrm{i},\ell} + \sum_{j=h+1}^{\mathrm{i}-1} \sum_{s=1}^{j} x_{j,s} \leq 1 && \forall \mathrm{i} \in [N] \setminus [h], \forall \ell \in [\mathrm{i}] \\
& \alpha - \frac{\sum_{j=1}^{k} \sum_{\mathrm{i}=h+1}^{N} \sum_{\ell=1}^{j} \frac{\mathrm{i}x_{\mathrm{i},\ell}}{N} \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}}{\sum_{j=1}^{k} \frac{N-h}{N-j+1} \prod_{s=0}^{j-2} \frac{h-s}{N-s}} \leq 0 && \forall k \in [h+1] \\
& x_{\mathrm{i},\ell} \geq 0 && \forall \mathrm{i} \in [N] \setminus [h], \forall \ell \in [I].
\end{aligned}
$$

The stochastic dominance argument says that for a given x, in the inner minimization problem we can focus our attention on instances of the form $Y_1 = \cdots = Y_k = 1$, $Y_j = 0$ for $j \geq k+1$, for all $k \in [N]$ (each one of them normalized so that $\mathbf{E}(\mathrm{OPT}(Y)) = 1$).[20]

The first step is to see the second set of constraints as stochastic dominance constraints of the form

$$\alpha - \frac{\mathbb{P}(\mathrm{ALG}_x(Y_{[N]}) \geq Y_j)}{\mathbb{P}(\mathrm{OPT}(Y_{[N]}) \geq Y_j)} \leq 0 \quad \forall j \in [h+1].$$

Consequently, if α is feasible we can write the inequality as $\mathbb{P}(\mathrm{ALG}_x(Y_{[N]}) \geq Y_j) \geq \alpha \mathbb{P}(\mathrm{OPT}(Y_{[N]}) \geq Y_j)$, integrate both sides and obtain the same bound but for the expectations instead of the probabilities. The bound in the expectations will be tight if α is feasible and the stochastic dominance constraint is binding for some index k. To see this, consider an instance Y^k with $Y^k_{\mathrm{i}} = 1$ for $\mathrm{i} \leq k$ and $Y^k_{\mathrm{i}} = 0$ for $\mathrm{i} > k$. This way $\mathbf{E}(\mathrm{ALG}_x(Y^k_{[N]})) = \mathbb{P}(\mathrm{ALG}_x(Y^k_{[N]}) \geq Y_k)$ and $\mathbf{E}(\mathrm{OPT}(Y^k_{[N]})) = \mathbb{P}(\mathrm{OPT}(Y^k_{[N]}) \geq Y_k)$. With this analysis, we conclude that the optimal value of $\mathrm{SDLP}_{h,N}$ equals the optimal worst case competitive ratio for the dependent sampling variant of p-DOS with fixed h and N. Moreover, we can recover an optimal algorithm from its optimal solution.

1.4.2 The Limit Problem and its Solution

Similarly as in Section 1.3.2, we obtain the limit problem of $\mathrm{SDLP}_{\lfloor pN \rfloor, N}$:

$$
\begin{aligned}
(\mathrm{SDCLP}_p) \qquad & \sup_{q \in L^1([p,1]\times\mathbb{N}),\, \alpha \in [0,1]} \quad \alpha \\
\text{s.t.} \qquad & tq(t,\ell) + \int_p^t \sum_{s \geq 1} q(\tau, s) \mathrm{d}\tau \leq 1 && \forall t \in [p,1], \forall \ell \geq 1 && (1.11) \\
& \alpha \leq \frac{F_k(q)}{1-p^k} && \forall k \geq 1 && (1.12) \\
& q(t,\ell) \geq 0 && \forall t \in [p,1], \forall \ell \geq 1.
\end{aligned}
$$

[20]Perhaps the easiest way to see this is that every feasible instance for the adversary is a convex combination of these instances.

Now we can directly apply Theorem 1.2 to SDCLP_p. For a solution q, consider a solution q^* as in the theorem. By definition, q^* satisfies Equation (1.11); and from the fact that $F_k(q) \le F_k(q^*)$ for all $k \ge 1$, q^* also satisfies Equation (1.12) for the same α as q. We obtain the following reduced problem analogous to RP_p, by noticing that for the thresholds $(t_{\mathrm{i}})_{\mathrm{i}\in\mathbb{N}}$ that correspond to q^* we have that $F_k(t) = F_k(q^*)$.

$$
\begin{aligned}
(\mathrm{SDRP}_p) \quad & \sup_{t=(t_{\mathrm{i}})_{\mathrm{i}\in\mathbb{N}}} \min_{k\ge 1} \frac{F_k(t)}{1-p^k} & \\
\text{s.t.} \quad & p \le t_{\mathrm{i}} \le t_{\mathrm{i}+1} \le 1 & \forall \mathrm{i} \ge 1.
\end{aligned}
$$

Recall that we defined $\alpha(p)$ as the limit of ratios $\alpha_{N,p}$, whose values correspond to the optimal value of $\mathrm{SDLP}_{\lfloor pN \rfloor, N}$. Consequently, $\alpha(p)$ equals the optimal value of SDRP_p.

1.4.3 Solving for Different Values of p

We proceed to obtain values of $\alpha(p)$ for $p \in [0, 1)$. We start by briefly discussing the case where $0 \le p < 1/\mathrm{e}$ and then study the limit as $p \to 1$. For intermediate values of p, we present (almost) matching numerical bounds. Note that $\alpha(p)$ is an increasing function, as we establish, in a more general setting, with Lemma 1.14 in Section 1.5. As a consequence, the limit of $\alpha(p)$ as p tends to 1 is well-defined.

The case $0 \le p < 1/\mathrm{e}$

For this range of p, we establish that $\alpha(p) = (\mathrm{e}(1-p))^{-1}$. This closes the gap in Kaplan et al. [82], where they obtain the same upper bound but a slightly weaker lower bound.[21] Our upper bound, which works for any $p \in [0, 1)$ is shown in Lemma 1.16 on a more general setting and with a simpler analysis than the one presented in Kaplan et al [82]. We obtain the lower bound by evaluating $t_1 = 1/\mathrm{e}$ and $t_{\mathrm{i}} = 1$ for $\mathrm{i} \ge 2$ in SDRP_p (i.e., the classic secretary problem algorithm). This means that the optimal algorithm will wait until seeing in total (counting both the online set and the history set) a fraction $1/\mathrm{e}$ of N, and from that point on it will stop whenever we find an item whose value is larger than what has been observed so far. Our results also reveal that the hardest single selection optimal stopping problem for this range of p is the secretary problem ($Y_1 = 1$ and the remaining values are 0). Indeed, the fact that the optimal value of SDRP_p is $(\mathrm{e}(1-p))^{-1}$, together with von Neumann's Minmax Theorem tells us that for any sequence Y, we can obtain a competitive ratio of at least $(\mathrm{e}(1-p))^{-1}$. Details about this case are presented in Section 1.7.2.

Limit as p goes to 1

We now turn our attention to the case where p is close to 1. In order to show that $\lim_{p\to 1} \alpha(p) = \alpha^*$, we will explicitly construct for each $p \in (0, 1)$, a feasible solution $(\tilde{q}, \tilde{\alpha}(p))$ for SDCLP_p, and then we will show that $\lim_{p\to 1} \tilde{\alpha}(p) = \alpha^*$. Since for every p, $\tilde{\alpha}(p) \le \alpha(p) \le \alpha^*$, this would prove the result.

[21] This value of $\alpha(p)$ was essentially known in a more restricted model with i.i.d. samples from an unknown distribution [42].

Fix $p \in (0,1)$ for now and recall from equation (1.8) that we can restrict to solutions q for SDCLP_p with the form

$$q(t,\ell) = \begin{cases} \frac{T_{\mathrm{i}}}{t^{\mathrm{i}+1}} & \text{if } t \in [t_{\mathrm{i}}, t_{\mathrm{i}+1}], \ell \leq \mathrm{i} \\ 0 & \text{otherwise,} \end{cases} \tag{1.13}$$

where $p \leq t_1 \leq t_2 \leq \cdots$, and $T_{\mathrm{i}} = \prod_{j=1}^{\mathrm{i}} t_j$. Note that for fixed i and $t \in [t_{\mathrm{i}}, t_{\mathrm{i}+1}]$, the function $f(\ell) = q(t,\ell)$ is positive and constant for $\ell \leq \mathrm{i}$, and 0 for $\ell > \mathrm{i}$. In particular, the function $q(t,\ell)$ is non-decreasing in ℓ. The last property is important because of the following lemma.

Lemma 1.4. *Let (q,α) be a feasible solution for* SDCLP_p *with $q(t,\ell)$ non-increasing in ℓ, for all $t \in [p,1]$ and α maximal (i.e., such that (q,c) is infeasible for any $c > \alpha$). Then we must have:*

$$\alpha \geq \inf_{k\geq 1} \frac{1}{1-p^k} \sum_{j=1}^{k} \int_p^1 tq(t,j)\mathrm{d}t\,. \tag{1.14}$$

Proof. By the maximality of α,

$$\alpha = \inf_{k\geq 1} \frac{1}{1-p^k} \sum_{j=1}^{k} \int_p^1 \sum_{\ell=1}^{j} q(t,\ell)\binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}\mathrm{d}t\,.$$

Since $q(t,\ell)$ is non-increasing in ℓ we can replace $q(t,\ell)$ by $q(t,j)$ in the inner sum to obtain

$$\alpha \geq \inf_{k\geq 1} \frac{1}{1-p^k} \sum_{j=1}^{k} \int_p^1 \sum_{\ell=1}^{j} q(t,j)\binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}\mathrm{d}t \geq \inf_{k\geq 1} \frac{1}{1-p^k} \sum_{j=1}^{k} \int_p^1 tq(t,j)\,\mathrm{d}t\,,$$

where we have used that for all $j \geq 1$ and $t \in [0,1]$, $\sum_{\ell=1}^{j} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell} = t$. $\square$

The idea behind the construction of our explicit feasible solution for SDCLP_p is to enforce that the infimum in the lower bound of Lemma 1.4 is attained for every k simultaneously. The following lemma gives us a characterization for all such solutions.

Lemma 1.5. *Let q be a function of the form* (1.13) *for some parameters $p = t_1 \leq t_2 \leq \cdots \leq 1$. The system of equations*

$$\alpha = \frac{1}{1-p^k} \sum_{j=1}^{k} \int_p^1 tq(t,j)\mathrm{d}t, \quad \forall k \geq 1 \tag{1.15}$$

is equivalent to

$$\alpha(1-p) = p \ln \frac{t_2}{p} + p - \mu_3 \tag{1.16}$$

$$\alpha(1-p)p^{k-1} = \frac{1}{k-1} \cdot \frac{T_k}{t_k^{k-1}} - \mu_{k+1}, \qquad \forall k \geq 2, \tag{1.17}$$

$$\textit{where} \qquad \mu_k = \sum_{\mathrm{i}=k}^{\infty} \frac{T_{\mathrm{i}}}{t_{\mathrm{i}}^{\mathrm{i}-1}} \cdot \frac{1}{(\mathrm{i}-2)(\mathrm{i}-1)}.$$

Proof. Observe that (1.15) is equivalent to (i) $\int_p^1 tq(t,1)\mathrm{d}t = \alpha(1-p)$ and (ii) for $k \geq 2$, $\int_p^1 tq(t,k)\mathrm{d}t = \alpha(1-p^k) - \alpha(1-p^{k-1}) = \alpha(1-p)p^{k-1}$. So, we only need to check that the right hand side of (1.16) and (1.17) are $\int_p^1 tq(t,1)\mathrm{d}t$ and $\int_p^1 tq(t,k)\mathrm{d}t$ respectively. Indeed, for $k \geq 2$:

$$\int_p^1 tq(t,k)\mathrm{d}t = \sum_{\mathrm{i}=k}^{\infty} \int_{t_\mathrm{i}}^{t_{\mathrm{i}+1}} \frac{T_\mathrm{i}}{t^\mathrm{i}}\mathrm{d}t = \sum_{\mathrm{i}=k}^{\infty} \frac{1}{\mathrm{i}-1}\left(\frac{T_\mathrm{i}}{t_\mathrm{i}^{\mathrm{i}-1}} - \frac{T_\mathrm{i}}{t_{\mathrm{i}+1}^{\mathrm{i}-1}}\right) = \sum_{\mathrm{i}=k}^{\infty} \frac{1}{\mathrm{i}-1}\left(\frac{T_\mathrm{i}}{t_\mathrm{i}^{\mathrm{i}-1}} - \frac{T_{\mathrm{i}+1}}{t_{\mathrm{i}+1}^{\mathrm{i}}}\right)$$
$$= \frac{1}{k-1}\cdot\frac{T_k}{t_k^{k-1}} + \sum_{\mathrm{i}=k+1}^{\infty} \frac{T_\mathrm{i}}{t_\mathrm{i}^{\mathrm{i}-1}}\left(\frac{1}{\mathrm{i}-1} - \frac{1}{\mathrm{i}-2}\right) = \frac{1}{k-1}\cdot\frac{T_k}{t_k^{k-1}} - \mu_{k+1}\,.$$

Similarly, for $k = 1$ we have

$$\int_p^1 tq(t,1)\mathrm{d}t = \sum_{\mathrm{i}=1}^{\infty} \int_{t_\mathrm{i}}^{t_{\mathrm{i}+1}} tq(t,1)\mathrm{d}t = \int_{t_1}^{t_2} \frac{t_1}{t}\mathrm{d}t + \sum_{\mathrm{i}=2}^{\infty} \int_{t_\mathrm{i}}^{t_{\mathrm{i}+1}} \frac{T_\mathrm{i}}{t^\mathrm{i}}\mathrm{d}t$$
$$= t_1 \ln\frac{t_2}{t_1} + \frac{T_2}{t_2} - \mu_3 = p\ln\frac{t_2}{p} + p - \mu_3\,. \qquad \square$$

Thanks to the previous lemma, we can restrict our search to pairs (q,α) satisfying (1.13), (1.16) and (1.17). The following lemma gives us one such solution.

Lemma 1.6. *Let $p, \alpha \in (0,1)$ be arbitrary numbers. Define for each $k \geq 1$, the quantity*

$$\gamma_k = 1 - \alpha + \alpha[kp^{k-1} - (k-1)p^k].$$

Define also the sequence of times $t_1 = p$, $t_2 = p\exp(\alpha(1-p)^2/p)$, and inductively for $k \geq 2$ define t_{k+1} as the real number satisfying

$$\left(\frac{t_k}{t_{k+1}}\right)^{k-1} = \frac{\gamma_k}{\gamma_{k-1}}\,. \tag{1.18}$$

This sequence has the following properties.

(i) $(t_k)_{k\geq 1}$ *is increasing.*
(ii) $\lim_{k\to\infty} t_k \leq 1$ *if and only if*

$$\ln p + \frac{\alpha(1-p)^2}{p} \leq \sum_{\mathrm{i}=1}^{\infty} \frac{\ln(\gamma_{\mathrm{i}+1})}{\mathrm{i}(\mathrm{i}+1)}\,. \tag{1.19}$$

and $\lim_{k\to\infty} t_k = 1$ *when equality holds in* (1.19).
(iii) Let q be the function defined from the sequence $(t_k)_{k\geq 1}$ as in (1.13). *Then (q,α) is feasible in* SDCLP_p.

Proof. We clearly have $t_1 \leq t_2$. Furthermore, the denominator minus the numerator of the right hand side of (1.18) is $(k-1)\alpha p^{k-2}(p-1)^2 \geq 0$, implying that $t_{k+1} \geq t_k$ for all $k \geq 2$. This proves (i).

Since the sequence (t_k) is increasing, it has a (possibly unbounded) limit. To compute it, we first take logarithm on both sides of (1.18) and rearrange terms to obtain that for $k \geq 2$,

$$\ln(t_{k+1}) = \ln(t_k) - \ln(\gamma_k^{1/(k-1)}) + \ln(\gamma_{k-1}^{1/(k-1)})$$

iterating this formula we obtain

$$\ln(t_{k+1}) = \ln(t_2) - \sum_{\mathrm{i}=1}^{k-1} \ln(\gamma_{\mathrm{i}+1}^{1/\mathrm{i}}) + \sum_{\mathrm{i}=1}^{k-1} \ln(\gamma_{\mathrm{i}}^{1/\mathrm{i}}) = \ln(t_2) - \sum_{\mathrm{i}=1}^{k-1} \ln(\gamma_{\mathrm{i}+1}^{1/\mathrm{i}}) + \sum_{\mathrm{i}=0}^{k-2} \ln(\gamma_{\mathrm{i}+1}^{1/(\mathrm{i}+1)})$$

and since $\ln(\gamma_1) = \ln(1) = 0$, and $\ln(t_2) = \ln p + \alpha(1-p)^2/p$, we get

$$\ln(t_{k+1}) = \ln p + \frac{\alpha(1-p)^2}{p} - \frac{1}{k+1}\ln(\gamma_k) - \sum_{\mathrm{i}=1}^{k-1} \frac{\ln(\gamma_{\mathrm{i}+1})}{\mathrm{i}(\mathrm{i}+1)}.$$

Observe that $\lim_{k\to\infty} \gamma_k = 1-\alpha$. Thus, taking the limit on the previous expression we have

$$\lim_{k\to\infty} \ln(t_k) = \ln p + \frac{\alpha(1-p)^2}{p} - \sum_{\mathrm{i}=1}^{\infty} \frac{\ln(\gamma_{\mathrm{i}+1})}{\mathrm{i}(\mathrm{i}+1)}$$

Note that (ii) follows directly from here.

To finish the proof we use Lemma 1.5, and so we only need to show (1.16) and (1.17). For all $\mathrm{i} \geq 2$, we have

$$\frac{T_{\mathrm{i}}}{t_{\mathrm{i}}^{\mathrm{i}-1}} = t_1 \prod_{j=2}^{\mathrm{i}} \frac{t_j}{t_{\mathrm{i}}} = t_1 \prod_{j=2}^{\mathrm{i}} \prod_{\ell=j}^{\mathrm{i}-1} \frac{t_\ell}{t_{\ell+1}} = p \prod_{\ell=2}^{\mathrm{i}-1} \frac{\gamma_\ell}{\gamma_{\ell-1}} = p\gamma_{\mathrm{i}-1}.$$

Therefore, using formulas for geometric series we get that for $k \geq 3$

$$\begin{aligned}
\mu_k := \sum_{\mathrm{i}=k}^{\infty} \frac{T_{\mathrm{i}}}{t_{\mathrm{i}}^{\mathrm{i}-1}} \cdot \frac{1}{(\mathrm{i}-2)(\mathrm{i}-1)} &= p \sum_{\mathrm{i}=k}^{\infty} \frac{1-\alpha+\alpha[(\mathrm{i}-1)p^{\mathrm{i}-2} - (\mathrm{i}-2)p^{\mathrm{i}-1}]}{(\mathrm{i}-1)(\mathrm{i}-2)} \\
&= p(1-\alpha)\sum_{\mathrm{i}=k}^{\infty}\left(\frac{1}{\mathrm{i}-2} - \frac{1}{\mathrm{i}-1}\right) + \alpha \sum_{\mathrm{i}=k}^{\infty} \frac{p^{\mathrm{i}-2}}{(\mathrm{i}-2)} - \alpha \sum_{\mathrm{i}=k}^{\infty} \frac{p^{\mathrm{i}-1}}{(\mathrm{i}-1)} \\
&= \frac{p(1-\alpha+\alpha p^{k-2})}{k-2}.
\end{aligned}$$

To see (1.16), we write

$$p \ln \frac{t_2}{p} + p - \mu_3 = \alpha(1-p)^2 + p - p(1-\alpha+\alpha p) = \alpha(1-p).$$

And to get (1.17) we let $k \geq 2$ and write

$$\begin{aligned}
\frac{1}{k-1}\frac{T_k}{t_k^{k-1}} - \mu_{k+1} &= \frac{p\gamma_{k-1} - p(1-\alpha+\alpha p^{k-1})}{k-1} \\
&= \frac{p(1-\alpha+\alpha[(k-1)p^{k-2} - (k-2)p^{k-1}] - p(1-\alpha+\alpha p^{k-1})}{k-1} \\
&= p\alpha(p^{k-2} - p^{k-1}).
\end{aligned}$$

□

Thanks to the previous lemma, as long as (1.19) holds for values $p, \alpha \in (0,1)$, we obtain a solution for CLP_p of value α. The following lemma shows that such pair of values always exists.

Lemma 1.7. *For $p \in (0,1)$, there is a unique $\tilde{\alpha} \in (0,1)$ that satisfies*

$$\ln p + \frac{\tilde{\alpha}(1-p)^2}{p} = \sum_{\mathrm{i}=1}^{\infty} \frac{\ln(1-\tilde{\alpha}+\tilde{\alpha}[(\mathrm{i}+1)p^{\mathrm{i}} - \mathrm{i}p^{\mathrm{i}+1}])}{\mathrm{i}(\mathrm{i}+1)}. \tag{1.20}$$

Furthermore, the map $p \mapsto \tilde{\alpha}(p)$ is continuous.

Proof. Define the following functions as the left and right hand sides of the previous expression

$$\begin{aligned} f(p,\alpha) &= \ln p + \frac{\alpha(1-p)^2}{p} \\ g(p,\alpha) &= \sum_{\mathrm{i}=1}^{\infty} \frac{\ln(1+\alpha(p^{\mathrm{i}}(1+\mathrm{i}(1-p))-1))}{\mathrm{i}(\mathrm{i}+1)}. \end{aligned}$$

Both f and g are continuous functions of their domains. Furthermore, $f(p,\alpha)$ is increasing in α. On the other hand, by Bernoulli inequality, $p^{-\mathrm{i}} = (1-(1-p))^{-\mathrm{i}} \geq 1+\mathrm{i}(1-p)$. Therefore $p^{\mathrm{i}}(1+\mathrm{i}(1-p)) - 1 \leq 0$. From here it is easy to see that $g(p,\alpha)$ is decreasing in α. We now evaluate these two functions in $\alpha = 0$

$$f(p,0) = \ln p < 0 \qquad \text{and} \qquad g(p,0) = 0\,.$$

For the case $\alpha = 1$, observe that $f(p,1) = \ln p + \frac{(1-p)^2}{p}$ is a convex function in $p \in (0,1)$ and it is minimized on $p = (\sqrt{5}-1)/2$. Therefore, there exists a universal constant c such that $f(p,1) \geq c$ for all $p \in (0,1)$.

On the other hand, we have that as α increases, there is a vertical asymptote on some $\alpha_0 \leq 1$ in which the function $g(p,\alpha)$ decreases to $-\infty$. Indeed if this was not the case, the formula for $g(p,1)$ would be well-defined, but simply replacing 1 on its expression yields

$$\begin{aligned} g(p,1) &= \sum_{\mathrm{i}=1}^{\infty} \frac{\ln\left(p^{\mathrm{i}}(1+\mathrm{i}(1-p))\right)}{\mathrm{i}(\mathrm{i}+1)} = \sum_{\mathrm{i}=1}^{\infty} \frac{\mathrm{i}\ln p + \ln(1+\mathrm{i}(1-p)}{\mathrm{i}(\mathrm{i}+1))} \\ &\leq \sum_{\mathrm{i}=1}^{\infty} \frac{\mathrm{i}\ln p + \mathrm{i}(1-p)}{\mathrm{i}(\mathrm{i}+1))} = (\ln p + 1 - p) \sum_{\mathrm{i}=1}^{\infty} \frac{1}{\mathrm{i}+1} = -\infty. \end{aligned}$$

Summarizing, for every fixed value $p \in (0,1)$, the functions $f(p,\alpha)$ and $g(p,\alpha)$ are continuous, the former is increasing in α, and the latter is decreasing in α, and we also have that $f(p,0) < g(p,0)$ and there exists some value $\alpha' \in (0,1)$ such that $f(p,\alpha') > c > g(p,\alpha')$. By the intermediate value theorem there must be some value $\tilde{\alpha}(p)$ for which $f(p,\tilde{\alpha}) = g(p,\tilde{\alpha})$, and by monotonicity and continuity of both functions, this value is unique and the map $p \mapsto \tilde{\alpha}(p)$ is continuous. ☐

We are now ready to prove the main theorem of this section. In the next statement, $\tilde{\alpha}(p)$ is the map defined in the previous lemma, $\alpha(p)$ is the optimal value of SDCLP$_p$ and $\alpha^*(\approx 0.745)$ is the unique solution of $\int_0^1 \frac{1}{y(1-\ln y)+1/\alpha^*-1}\mathrm{d}y = 1$.

Theorem 1.8. *For every $p \in (0,1)$, $0 \leq \tilde{\alpha}(p) \leq \alpha(p) \leq \alpha^*$. Furthermore, if we define by continuity $\tilde{\alpha}(1) := \lim_{p\to 1} \tilde{\alpha}(p)$, then $\tilde{\alpha}(1) = \alpha(1) = \alpha^*$.*

Proof. By Lemmas 1.6 and 1.7, we conclude that there is a feasible solution of SDCLP$_p$ with value $\tilde{\alpha}(p)$. Therefore, $0 \leq \tilde{\alpha}(p) \leq \alpha(p)$. From Theorem 3.4 in [82] we know that $\alpha(p) \leq \alpha^*$.[22] Thus, we only need to show that $\tilde{\alpha}(1) = \alpha^*$, for that, define the function

$$h(p,\eta) = \frac{\sum_{\mathrm{i}=1}^{\infty} \frac{\ln(1-\eta+\eta[(\mathrm{i}+1)p^{\mathrm{i}}-\mathrm{i}p^{\mathrm{i}+1}])}{\mathrm{i}(\mathrm{i}+1)}}{\ln p + \frac{\eta(1-p)^2}{p}}. \tag{1.21}$$

and note that by definition of $\tilde{\alpha}(p)$, $h(p,\tilde{\alpha}(p)) = 1$

Let us study $h(p,\eta)$ as $p \to 1$. As both the numerator and the denominator go to 0 as $p \to 1$, we use l'Hôpital's rule to find the limit

$$\lim_{p\to 1} h(p,\eta) = \lim_{p\to 1} \frac{\sum_{\mathrm{i}=1}^{\infty} \frac{\eta(p^{\mathrm{i}-1}-p^{\mathrm{i}})}{\eta((\mathrm{i}+1)p^{\mathrm{i}}-\mathrm{i}p^{\mathrm{i}+1})+1-\tilde{\eta}}}{\frac{1}{p} + \eta\frac{-2p(1-p)-(1-p)^2}{p^2}} = \lim_{p\to 1} \frac{\sum_{\mathrm{i}=1}^{\infty} \frac{p^{\mathrm{i}-1}-p^{\mathrm{i}}}{(\mathrm{i}+1)p^{\mathrm{i}}-\mathrm{i}p^{\mathrm{i}+1}+\frac{1}{\eta}-1}}{\frac{1}{p} + \eta\left(1-\frac{1}{p^2}\right)}.$$

As $p \to 1$, the denominator in the last expression goes to 1. For the numerator, we will analyze the limit through a Riemann's integral analysis. For this we define $x_{\mathrm{i}} = p^{\mathrm{i}}$ (therefore $\mathrm{i} = \ln x_{\mathrm{i}}/\ln p$), so that intervals $(x_{\mathrm{i}+1}, x_{\mathrm{i}}]$ for $\mathrm{i} \geq 1$ form a partition of the interval $(0,1]$, resulting in

$$\begin{aligned}
\lim_{p\to 1} \sum_{\mathrm{i}=1}^{\infty} \frac{p^{\mathrm{i}-1}-p^{\mathrm{i}}}{(\mathrm{i}+1)p^{\mathrm{i}}-\mathrm{i}p^{\mathrm{i}+1}+\frac{1}{\eta}-1} &= \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\infty} \frac{x_{\mathrm{i}-1}-x_{\mathrm{i}}}{(\mathrm{i}+1)x_{\mathrm{i}}-\mathrm{i}x_{\mathrm{i}}p+\frac{1}{\eta}-1} \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\infty} \frac{x_{\mathrm{i}-1}-x_{\mathrm{i}}}{x_{\mathrm{i}}(1-\ln x_{\mathrm{i}}\frac{p-1}{\ln p})+\frac{1}{\eta}-1} \\
&= \int_0^1 \frac{1}{y(1-\ln y)+\frac{1}{\eta}-1}\mathrm{d}y.
\end{aligned}$$

In the last equality we first replaced $\lim_{p\to 1} \frac{p-1}{\ln p} = 1$ and then the limit of the Riemann sum. This can be justified by the fact that the sum is monotone in the term $\frac{p-1}{\ln p}$. So, for p close enough to 1 we can bound by replacing with $1-\varepsilon \leq \frac{p-1}{\ln p} \leq 1+\varepsilon$. Since the integral is continuous in the factor that accompanies $\ln y$, both bounds converge when $\varepsilon \to 0$. Replacing η by α^* finishes the proof. □

[22] The idea behind this result is that any algorithm for adversarial p-DOS with dependent sampling can be applied to the i.i.d. prophet inequality without loss in performance. If the result was not true it would contradict the fact that α^* is the optimal competitive ratio for the latter problem.

Linear lower bound for p close to 1

For $p \in (0,1)$, we have just designed a stopping rule $\tilde{q}$ that has a competitive ratio of at least $\tilde{\alpha}(p)$. We proceed to prove that $\tilde{\alpha}(p)$ lies above the line that connects 0 and α^*, which has implications for problems related to $p-$DOS. Numerically, it appears that $\tilde{\alpha}(p)$ is actually concave, which would suffice for this purpose. Unfortunately, we have not been able to prove this so we rely on the following result.

Theorem 1.9. *For $p \in (0,1)$, $\tilde{\alpha}(p) \geq \alpha^* p$.*

Proof. To prove this result we define $f(p) = \tilde{\alpha}(p)/p$. What we would like to prove is that $f(p) \geq \alpha^*$. For this, we replace $\tilde{\alpha}(p) = f(p)p$ in equation (1.20):

$$\ln p + f(p)(1-p)^2 = \sum_{\mathrm{i}=1}^{\infty} \frac{\ln(1 - f(p)p + f(p)p[(\mathrm{i}+1)p^{\mathrm{i}} - \mathrm{i}p^{\mathrm{i}+1}])}{\mathrm{i}(\mathrm{i}+1)} .$$

Note that the left-hand side of the equation is increasing in $f(p)$ and the right-hand side of the equation is decreasing in $f(p)$. Thus, to prove that $f(p) \geq \alpha^*$, we need to prove that

$$\ln p + \alpha^*(1-p)^2 \leq \sum_{\mathrm{i}=1}^{\infty} \frac{\ln(1 - \alpha^* p + \alpha^* p[(\mathrm{i}+1)p^{\mathrm{i}} - \mathrm{i}p^{\mathrm{i}+1}])}{\mathrm{i}(\mathrm{i}+1)} .$$

By subtracting $\ln p$ the latter is equivalent to proving

$$\alpha^* \leq \sum_{\mathrm{i}=1}^{\infty} \frac{\ln(\frac{1}{p} - \alpha^*(1 - p^{\mathrm{i}}(1 + \mathrm{i}(1-p))))}{\mathrm{i}(\mathrm{i}+1)} + \alpha^*(2p - p^2) .$$

To prove the inequality let us call its right hand side $a(p)$ and note that by definition of $\tilde{\alpha}(1)$, $a(1) = \alpha^*$, i.e., the inequality is tight for $p = 1$. Therefore, to conclude we show that $a(p)$ is decreasing in p, so that the inequality holds for all $p \in (0,1)$. Indeed,

$$\frac{\mathrm{d}}{\mathrm{d}p} a(p) = \sum_{\mathrm{i}=1}^{\infty} \frac{-\frac{1}{p^2} + \mathrm{i}(\mathrm{i}+1)\alpha^*(p^{\mathrm{i}-1} - p^{\mathrm{i}})}{\mathrm{i}(\mathrm{i}+1)(\frac{1}{p} - \alpha^*(1 - p^{\mathrm{i}}(1 + \mathrm{i}(1-p))))} + 2\alpha^*(1-p) .$$

Letting

$$b(p) = \frac{1}{p^2} \sum_{\mathrm{i}=1}^{\infty} \frac{1}{\mathrm{i}(\mathrm{i}+1)(\frac{1}{p} - \alpha^*(1 - p^{\mathrm{i}}(1 + \mathrm{i}(1-p))))} \quad \text{and} \quad c(p) = \sum_{\mathrm{i}=1}^{\infty} \frac{p^{\mathrm{i}-1} - p^{\mathrm{i}}}{\frac{1}{p\alpha^*} - 1 + p^{\mathrm{i}}(1 + \mathrm{i}(1-p))},$$

we have $\frac{\mathrm{d}}{\mathrm{d}p} a(p) = 2\alpha^*(1-p) - b(p) + c(p)$. Now, as as $\alpha^*(1 - p^{\mathrm{i}}(1 + \mathrm{i}(1-p)))$ lies between 0 and $\alpha^* < 1$, we have that

$$b(p) \geq \frac{1}{p^2} \sum_{\mathrm{i}=1}^{\infty} \frac{1}{\mathrm{i}(\mathrm{i}+1)\frac{1}{p}} = \frac{1}{p} .$$

We now show that $c(p) \leq 1/p - \alpha^*(1-p) - \alpha^* \frac{1-p}{p}$. For this define $x_{\mathrm{i}} = p^{\mathrm{i}}$ and note that

$$
\begin{aligned}
c(p) &= \frac{1}{p} \sum_{\mathrm{i}=1}^{\infty} \frac{x_{\mathrm{i}} - x_{\mathrm{i}+1}}{\frac{1}{p\alpha^*} - 1 + x_{\mathrm{i}}(1 + \mathrm{i}(1-p))} \\
&= \frac{1}{p} \sum_{\mathrm{i}=1}^{\infty} \frac{x_{\mathrm{i}} - x_{\mathrm{i}+1}}{\frac{1}{p\alpha^*} - 1 + x_{\mathrm{i}} \left(1 + \ln x_{\mathrm{i}} \frac{(1-p)}{\ln p}\right)} \\
&\leq \frac{1}{p} \sum_{\mathrm{i}=1}^{\infty} \frac{x_{\mathrm{i}} - x_{\mathrm{i}+1}}{\frac{1}{p\alpha^*} - 1 + x_{\mathrm{i}}\left(1 - p \ln x_{\mathrm{i}}\right)} \\
&\leq \frac{1}{p} \int_0^p \frac{1}{\frac{1}{p\alpha^*} - 1 + y(1 - p\ln y)} \mathrm{d}y \\
&= \frac{1}{p} \int_0^1 \frac{1}{\frac{1}{p\alpha^*} - 1 + y(1 - p\ln y)} \mathrm{d}y - \frac{1}{p} \int_p^1 \frac{1}{\frac{1}{p\alpha^*} - 1 + y(1 - p\ln y)} \mathrm{d}y \\
&\leq \frac{1}{p} \int_0^1 \frac{1}{\frac{1}{p\alpha^*} - 1 + y(1 - p\ln y)} \mathrm{d}y - \alpha^*(1-p)\,.
\end{aligned}
$$

The first inequality comes from $(1-p)/\ln p \leq -p$. The second inequality follows because $x_{\mathrm{i}} > x_{\mathrm{i}+1}$ and the function $((1/(p\alpha^*) - 1 + y(1 - p\ln y)))^{-1}$ is decreasing in y. The last inequality comes from the fact that $1 - y(1 - p\ln(y)) \in [0,1]$ when $y, p \in [0,1]$. Now, the integral in the last step can be rewritten as

$$
\begin{aligned}
&\frac{1}{p} \int_0^1 \frac{1}{\frac{1}{\alpha^*} - 1 + y(1 - \ln y)} \mathrm{d}y - \frac{1-p}{p} \int_0^1 \frac{\frac{1}{p\alpha^*} + y\ln y}{\left(\frac{1}{\alpha^*} - 1 + y(1-\ln y)\right)\left(\frac{1}{p\alpha^*} - 1 + y(1 - p\ln y)\right)} \mathrm{d}y \\
&\leq \frac{1}{p} - \frac{1-p}{p} \int_0^1 \frac{1}{\frac{1}{\alpha^*} - 1 + y(1-\ln y)} \cdot \frac{\frac{1}{p\alpha^*} - \frac{1}{\mathrm{e}}}{\frac{1}{p\alpha^*} - 1 + y + \frac{p}{\mathrm{e}}} \mathrm{d}y \\
&\leq \frac{1}{p} - \frac{1-p}{p} \int_0^1 \frac{\frac{1}{p\alpha^*} - \frac{1}{\mathrm{e}}}{\frac{1}{p\alpha^*} - 1 + y + \frac{p}{\mathrm{e}}} \mathrm{d}y = \frac{1}{p} - \frac{1-p}{p} \left(\frac{1}{p\alpha^*} - \frac{1}{\mathrm{e}}\right) \ln \left(\frac{\frac{1}{p\alpha^*} + \frac{p}{\mathrm{e}}}{\frac{1}{p\alpha^*} + \frac{p}{\mathrm{e}} - 1}\right).
\end{aligned}
$$

Here, the first inequality follows from the definition of α^* and fact that $-y\ln y \in [0, 1/\mathrm{e}]$ when $y \in (0,1)$. The second inequality comes from observing that $1/\left(\frac{1}{\alpha^*} - 1 + y(1 - \ln y\right)$ is decreasing, non-negative and integrates 1, and $\left(\frac{1}{p\alpha^*} - \frac{1}{\mathrm{e}}\right) / \left(\frac{1}{p\alpha^*} - 1 + y + \frac{p}{\mathrm{e}}\right)$ is non-negative and decreasing. Finally, it can be checked numerically that if $\alpha^* \in [0.74, 0.75]$, the term $\left(\frac{1}{p\alpha^*} - \frac{1}{\mathrm{e}}\right) \ln \left(\frac{\frac{1}{p\alpha^*} + \frac{p}{\mathrm{e}}}{\frac{1}{p\alpha^*} + \frac{p}{\mathrm{e}} - 1}\right)$ is at least 0.8 for all $p \in (0,1)$. Then, since we know that $\alpha^* \approx 0.745$, we can conclude that $c(p) \geq \frac{1}{p} - \alpha^*(1-p)\left(1 + \frac{1}{p}\right)$. Therefore,

$$
\frac{\mathrm{d}}{\mathrm{d}p} a(p) = 2\alpha^*(1-p) - b(p) + c(p) \leq 2\alpha^*(1-p) - \frac{1}{p} + \frac{1}{p} - \alpha^*(1-p)\left(1 + \frac{1}{p}\right) \leq 0\,.
$$

The result follows. $\square$

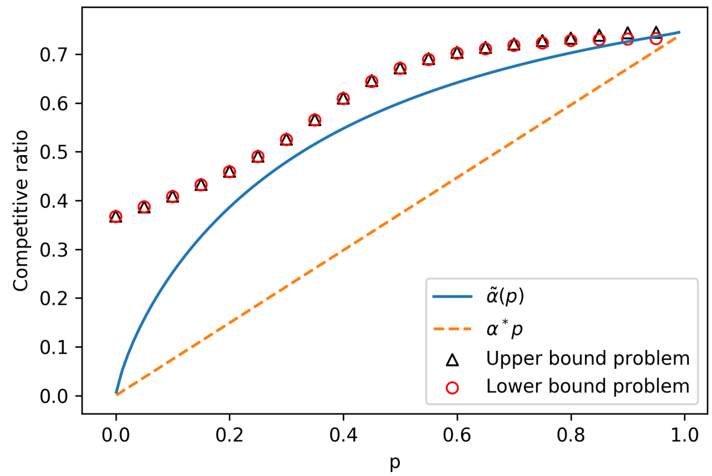


Figure 1.1: Plot of the numerical values of $\mathrm{UBP}_{p,N,k_{\max}}$ (black triangles) and $\mathrm{LBP}_{p,k_{\max}}$ (red circles). The blue line is $\tilde{\alpha}(p)$, the lower bound on $\alpha(p)$ given by Theorem 1.8, while the orange line is $\alpha^* p$, the lower bound given by Theorem 1.9.

It is worth contrasting the latter result with recent results of Correa et al. [40] and Rubinstein et al. [110]. They consider a more restricted model than p-DOS with dependent sampling, in which the decision maker sequentially observes i.i.d. values taken from a distribution F. Furthermore, the decision maker has, beforehand, access to a number of samples from F. Correa et al. [40] show that if she has access to $O(n^2/\varepsilon)$ samples then she can essentially learn F and guarantee a factor of $\alpha^* - O(\varepsilon)$. Rubinstein et al. [110] improve this result by showing that $O(n/\varepsilon^6)$ samples are enough to guarantee a factor of $\alpha^* - O(\varepsilon)$. Since p-DOS is more general than the latter setting, Theorem 1.9 can be interpreted as a further improvement in this direction.[23] Indeed if we take $p = 1 - \varepsilon$ in Theorem 1.9 the online set is of size $n = \varepsilon N$ so that our information set is of size $(1-\varepsilon)N = n(1-\varepsilon)/\varepsilon$. Thus with $O(n/\varepsilon)$ samples we guarantee a factor of $\alpha^* - O(\varepsilon)$.

Numerical bounds for $0 \le p < 1$

To close this subsection we present numerical bounds for SDCLP_p for different values of p. For the upper bound we solve an optimization problem based on SDCLP_p, which we call $\mathrm{UBP}_{p,N,k_{\max}}$. For the lower bound we solve a truncation of SDRP_p, which we call LBP_p. Details about these optimization problems can be found in Section 1.7.3.

In Figure 1.1 we plot the obtained upper and lower bounds together with the lower bound $\tilde{\alpha}(p)$ and the linear lower bound $\alpha^* p$. It is worth noting that $\tilde{\alpha}(p)$ is apparently concave but unfortunately we have not been able to prove this.

We pay special attention to the case when $p = 1/2$, which corresponds to one sample for each item in the online set. In this case we obtain a lower bound of 0.671, improving upon 0.649, the best known bound [42]. The thresholds for the algorithm are shown in Table 1.1.

[23] Certainly, our improvement only holds when n is large compared to $1/\varepsilon$, as we are analyzing the value of the limit problem.

Table 1.1: Best found solution for $p = 1/2$, rounded to the third decimal.

i	1	2	3	4	5	6	7	8	9	10
t_{i}	0.500	0.836	0.903	0.941	0.957	0.985	0.994	0.994	0.994	0.994

1.4.4 Connection Between the Sampling Models

Recall that we have defined $\alpha(p)$ and $\beta(p)$ as the limit optimal competitive ratios in the dependent and independent sampling models, respectively. So far, we have established that for any $p \in [0, 1)$, $\alpha(p)$ equals SDRP_p, which describes an algorithm parameterized by time thresholds t. We now proceed to show that $\beta(p)$ also equals to the value of SDRP_p, and that this value is actually a lower bound of $\beta_{N,p}$ when N is finite.

We start by relating solutions of SDRP_p with algorithms. As in Section 1.3.4, given an increasing sequence $(t_{\mathrm{i}})_{\mathrm{i}\in\mathbb{N}}$, we interpret the arrival order as uniform in $[0, 1]$ arrival times, and accept any ℓ-local maximum from t_ℓ onwards. Let us denote this algorithm by ALG_t and its competitive ratio by

$$\beta_{N,p}(t) = \inf_{Y \text{ decreasing}} \frac{\mathbf{E}(\mathrm{ALG}_t(Y_{[N]}))}{\mathbf{E}(\mathrm{OPT}(Y_{[N]}))}.$$

Certainly, $\beta_{N,p} \geq \beta_{N,p}(t)$, for any sequence $t = (t_{\mathrm{i}})_{\mathrm{i}\in\mathbb{N}}$. In the following two lemmas, we establish that in fact, for any feasible solution t for SDRP_p, $\beta_{N,p}(t)$ is decreasing and converges to the corresponding value of the objective function in SDRP_p.

Lemma 1.10. *For all $N \geq 1$, $\beta_{N,p}(t) \geq \beta_{N+1,p}(t)$.*

Proof. For an instance $Y_{[N]}$ of size N, denote by $Y_{[N]}^{+0}$ the instance of size $N + 1$ that results from appending a 0 to $Y_{[N]}$. We prove that for all instances $Y_{[N]}$ it holds that

$$\frac{\mathbf{E}\left(\mathrm{ALG}_t(Y_{[N]})\right)}{\mathbf{E}(\mathrm{OPT}(Y_{[N]}))} \geq \frac{\mathbf{E}\left(\mathrm{ALG}_t\left(Y_{[N]}^{+0}\right)\right)}{\mathbf{E}\left(\mathrm{OPT}\left(Y_{[N]}^{+0}\right)\right)},$$

which immediately implies the result. Clearly, $\mathbf{E}(\mathrm{OPT}(Y_{[N]})) = \mathbf{E}(\mathrm{OPT}(Y_{[N]}^{+0}))$, as the arrival time of the added 0 is independent of the other arrival times. We conclude by proving that $\mathbf{E}\left(\mathrm{ALG}_t\left(Y_{[N]}\right)\right) \geq \mathbf{E}\left(\mathrm{ALG}_t\left(Y_{[N]}^{+0}\right)\right)$. In fact, we can couple the arrival times of the values of $Y_{[N]}$ with the corresponding ones in $Y_{[N]}^{+0}$, and for the latter, add an independent arrival time for 0. Since the 0 is the smallest element, the relative rank of all other values is the same in both instances. Therefore, every time ALG_t selects a positive element in $Y_{[N]}^{+0}$, it selects the same element in $Y_{[N]}$. When ALG_t selects the 0 in $Y_{[N]}^{+0}$, it may select a positive element in $Y_{[N]}$ or not stop at all. Thus, with this coupling we get that $\mathrm{ALG}_t(Y_{[N]}) \geq \mathrm{ALG}_t\left(Y_{[N]}^{+0}\right)$. $\square$

Lemma 1.11. *Fix vector t of non-decreasing time thresholds. For any instance Y, it holds that*

$$\lim_{N\to\infty} \mathbb{P}(\mathrm{ALG}_t(Y_{[N]}) = Y_j) = \sum_{\mathrm{i}=1}^{\infty} \int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}} \sum_{\ell=1}^{j\wedge \mathrm{i}} \frac{T_{\mathrm{i}}}{\tau^{\mathrm{i}}} \binom{j-1}{\ell-1} (1-\tau)^{j-\ell} \tau^{\ell-1} \, \mathrm{d}\tau.$$

Proof. For ease of notation, in what follows we write ALG_t instead of $\mathrm{ALG}_t(Y_{[N]})$. We have that

$$\begin{aligned}
\mathbb{P}(\mathrm{ALG}_t = Y_j) &= \int_p^1 \mathbb{P}(\mathrm{ALG}_t = Y_j \,|\, Y_j \text{ arrives at time } \tau)\,\mathrm{d}\tau \\
&= \sum_{\mathrm{i}=1}^{\infty} \int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}} \mathbb{P}(\mathrm{ALG}_t = Y_j \,|\, Y_j \text{ arrives at time } \tau)\,\mathrm{d}\tau \\
&= \sum_{\mathrm{i}=1}^{\infty} \int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}} \sum_{\ell=1}^{j\wedge \mathrm{i}} \mathbb{P}(\mathrm{ALG}_t \text{ does not stop before } \tau \,|\, Y_j \text{ is } \ell\text{-local and arrives at } \tau) \\
&\qquad \cdot \mathbb{P}(Y_j \text{ is } \ell\text{-local} \,|\, Y_j \text{ arrives at } \tau)\,\mathrm{d}\tau \\
&= \sum_{\mathrm{i}=1}^{\infty} \int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}} \sum_{\ell=1}^{j\wedge \mathrm{i}} \mathbb{P}(\mathrm{ALG}_t \text{ does not stop before } \tau \,|\, Y_j \text{ is } \ell\text{-local and arrives at } \tau) \\
&\qquad \cdot \binom{j-1}{\ell-1} (1-\tau)^{j-\ell}\tau^{\ell-1}\,\mathrm{d}\tau.
\end{aligned}$$

The last equality comes from the fact that Y_j is ℓ-local if exactly $\ell-1$ items from $Y_1, ..., Y_{j-1}$ arrive before Y_j. Now, note that the event that ALG_t stops before τ does not depend on what elements arrive after τ and what are their relative rankings, but only on the relative rankings of the items that arrive before τ. Also, note that when N is large, the probability that at least i items arrive before a given time $\tau > 0$ tends to 1. Therefore,

$$\begin{aligned}
&\mathbb{P}(\mathrm{ALG}_t \text{ does not stop before } \tau \,|\, Y_j \text{ is } \ell\text{-local and arrives at } \tau) \\
&= \mathbb{P}(\mathrm{ALG}_t \text{ does not stop before } \tau \,|\, \text{at least i items arrive before } \tau) + o(N) \\
&= \prod_{r=1}^{\mathrm{i}} \mathbb{P}(r\text{-th largest item before } \tau \text{ arrives before } t_r) + o(N) \\
&= \prod_{r=1}^{\mathrm{i}} \frac{t_r}{\tau} + o(N) = \frac{T_{\mathrm{i}}}{\tau^{\mathrm{i}}} + o(N).
\end{aligned}$$

Taking limit when N tends to infinity we conclude the proof of the lemma. □

Lemma 1.11 implies that $\lim_{N\to\infty} \mathbb{P}(\mathrm{ALG}_t(Y_{[N]}) \geq Y_j) = F_k(t)$. This, together with the fact that the guarantee of ALG_t in instances of size N, as for any algorithm, is given by

$$\beta_{N,p}(t) = \min_{1\leq j\leq N} \frac{\mathbb{P}(\mathrm{ALG}_t(Y_{[N]}) \geq Y_j)}{\mathbb{P}(\mathrm{OPT}(Y_{[N]}) \geq Y_j)} = \min_{1\leq j\leq N} \frac{\mathbb{P}(\mathrm{ALG}_t(Y_{[N]}) \geq Y_j)}{1-p^j},$$

implies that the limit guarantee is the one given by SDRP_p. This means that taking t^* as the optimal solution of SDRP_p, $\beta_{N,p} \geq \beta_{N,p}(t^*) \geq \alpha(p)$, and therefore $\beta(p) \geq \alpha(p)$.

To prove that $\beta(p) \leq \alpha(p)$, assume there is $p \in [0,1)$ such that $\beta(p) \geq \alpha(p)+\varepsilon$, for some $\varepsilon > 0$. Fix N, and consider the viewpoint where each item has an independent $U[0,1]$ arrival time and is in H if it arrives before p. Since $\beta_{N,p} \geq \beta(p)$, it is clear that there is a sufficiently small $\delta > 0$ such that there is an algorithm A that does not stop in $[p, p+\delta]$, that obtains at least an $(\alpha(p)+\varepsilon/2)$ fraction of the optimal offline algorithm in the independent sampling model for any instance with N elements[24]. We derive from A an algorithm for the dependent sampling model in the following way: let H be the history set for the dependent sampling model, which always has size $|H| = pN$. We draw N independent $U[0,1]$ arrival times, randomly assign the smallest pN times to the items of H, and the rest to the items of the online set, so that the order of arrival and order of the uniform times agree (notice that we can always do this on the fly). We obtain a new history set H' defined as the items with arrival time in $[0,p]$. The set H' has a random size, and when $|H'| < pN$, $H' \subsetneq H$, and otherwise $H \subseteq H'$. We run A as if we were in the independent sampling model with history set H', i.e., we pass it the elements not in H' one by one. If A stops with an item in H, we declare failure and do not stop. Otherwise, we stop whenever A stops. Note that by the definition of A, failure can only occur when we assign to an element of H an arrival time larger than $p+\delta$; or equivalently, when out of the N arrival times, less than pN arrive in the interval $[0, p+\delta]$. By increasing N we can make this event occur with arbitrarily small probability, say smaller than $(1-p)\varepsilon/4$. Thus, if we upper bound by Y_1 the value of the item A selects when we fail, since $(1-p)Y_1 \leq \mathbf{E}(\mathrm{OPT})$, our new algorithm gets in expectation at least $\mathbf{E}(A) - \frac{\varepsilon}{4}\mathbf{E}(\mathrm{OPT})$. Therefore, for large enough N, we have an algorithm for the dependent sampling model with a guarantee of at least $\alpha(p) + \varepsilon/4$, which is a contradiction. We conclude the following theorem.

Theorem 1.12. *Let t^* be an optimal solution for SDRP_p. We have that as N tends to infinity, $\beta_{N,p}(t^*) \searrow \beta(p) = \alpha(p)$.*

The situation for dependent sampling is a bit trickier, and it is unclear whether $\alpha_{N,p}$ is a decreasing sequence. However, we can establish that $\alpha_{N,p}$ is still close to $\alpha(p)$.

Theorem 1.13. *For any $p \in [0,1)$ we have that*

$$\alpha_{N,p} = \alpha(p) + O\left(\frac{(\log N)^2}{(1-p)^2\sqrt{N}}\right).$$

Summarizing the previous discussion, we obtain that for any fixed value of N the guarantee obtained by our algorithm ALG_{t^*}, $\alpha(p)$ applies to both sampling models. In particular, for independent sampling we have that $\beta_{N,p} \geq \alpha(p)$, while for dependent sampling we have that $\alpha_{N,p} \geq \alpha(p) - \tilde{O}(1/((1-p)^2\sqrt{N}))$.

1.5 On Multiple-Choice p-DOS Problems

Until now we have focused on single selection problems. It is natural to ask whether our techniques can be used for selecting multiple items from a list subject to some combinatorial constraints, such as cardinality constraints, knapsack constraints or selecting edges that form a matching in a graph. It is possible to extend some of the linear programming machinery to tackle simple constraints such as cardinality bounds using quotas (see, [18, 26] for particular examples), but adding more complex constraints seems difficult. Nevertheless, our resulting

[24] A simple way of obtaining such an algorithm is by simply ignoring elements that arrive in $[p, p+\delta]$, and act as if we skipped a uniformly random interval $[x, x+\delta] \subseteq [p,1]$. Since the arrival times are uniformly distributed, the set of items that arrive in $[p+\delta, 1]$ and their ordering have the same distribution as those that arrive in $[p,1] \setminus [x, x+\delta]$.

algorithms can be used as black boxes to obtain new results for certain multiple selection problems.

To cast the problem more precisely, we consider the following version of p-DOS with adversarial values. A DM is given a value $p \in [0,1)$ and an independence system $(S, \mathcal{I})$.[25] An adversary assigns a non-negative weight $Y(\mathrm{e})$ to every element e of S. Every element is then independently placed on the information set with probability p and in the online set otherwise. As in the single selection case, the DM observes all the elements in the information set and the relative rankings of their Y-weights (assuming a universal tie-breaking rule). Then, the online set is revealed one by one in uniform random order. Every time an element is revealed the DM needs to irrevocably decide whether to add it or not to the solution set, while making sure that the solution set is at all times independent in $(S, \mathcal{I})$. An algorithm for this problem is ρ-competitive if the expected weight of the elements in the solution set is at least ρ times the expected weight of a maximum weight independent subset of the online set. An alternative way to state this is the following: for any q, let $S[q]$ be a random subset of S obtained by adding each element of S to it with probability q independently. The online set of our problem behaves like $S[1-p]$. Let also $\mathrm{OPT}(\mathcal{I}, q, Y)$ be the expectation of the maximum Y-weight independent subset of $S[q]$ in $\mathcal{I}$. An algorithm for p-DOS on $(S, \mathcal{I})$ is ρ-competitive if for any instance the expected Y-weight of its output is at least $\rho\mathrm{OPT}(\mathcal{I}, 1-p, Y)$.

Denote by $\beta_{S,\mathcal{I}}(p)$ to the maximum competitive ratio ρ achievable by an algorithm for p-DOS on $(S, \mathcal{I})$. In general, we need to analyze entire classes of independence system at once. We tackle this in the following way. If $\mathcal{C}$ is a collection of independence systems, we define $\beta_{\mathcal{C}}(p)$ as the infimum over all $(S, \mathcal{I})$ in $\mathcal{C}$ of $\beta_{S,\mathcal{I}}(p)$. For instance, by setting $\mathcal{C}$ to be the class of all matroids of rank 1 (where S can have any number of elements), we recover the single-selection p-DOS problem and we get $\beta_{\mathcal{C}}(p) = \beta(p) = \alpha(p)$.

When $p = 0$ the p-DOS problem just described coincides with the generalized secretary problem by Babaioff et al. [10]. There is a long line of work for that problem for different independence systems, most notably for knapsack [7, 88], matchings [91, 86] and many classes of matroids (see [115] for a recent comprehensive list). Optimal competitive ratios, again for $p = 0$, are only known for the classes of uniform and transversal matroids [86], and constant competitive ratios are known for several other cases. An important open question, known as the matroid secretary conjecture, [10, 8] is to decide whether the class $\mathcal{M}$ of all matroids admits an constant competitive algorithm (in our notation, whether $\beta_{\mathcal{M}}(0) > 0$). The best ratio so far is parameterized on the rank r of the matroid. In our notation, if $\mathcal{M}_r$ is the class of matroids of rank r, then $\beta_{\mathcal{M}_r}(0) = \Omega(1/\log\log r)$ [97, 64].

The problem on general independence systems has not been studied yet for the case $p > 0$, however we show in the next sections that the lower bounds on the guarantees for $p = 0$ transfer directly to any $p < 1$. In fact, we show that for a certain natural class of independence systems, we can further improve the guarantees for large p via a reduction to the single selection case p-DOS problem.

[25] An independence system is a pair $(S, \mathcal{I})$, where S is a finite ground set, and $\mathcal{I}$ is a family of subsets of S, called the independent sets of the system. The system must satisfy that the empty set is independent and that every subset of an independent set is independent.

1.5.1 Relation Among Guarantees for Different p on a Given Independence System $(S, \mathcal{I})$

The following lemma shows that for any class $\mathcal{C}$ of independence systems, $\beta_{\mathcal{C}}(p)$ is increasing in p.

Lemma 1.14. *Let $p_1, p_2 \in [0, 1)$ with $p_1 < p_2$. For any ρ-competitive algorithm for p_1-DOS on $(S, \mathcal{I})$ we can construct a ρ-competitive algorithm for p_2-DOS. Therefore, for any class $\mathcal{C}$ of independence systems, $\beta_{\mathcal{C}}(p_1) \leq \beta_{\mathcal{C}}(p_2)$.*

Proof. Fix $(S, \mathcal{I})$, p_1 and p_2 and let A_1 be any ρ-competitive algorithm for p_1-DOS on $(S, \mathcal{I})$. Let Y be any instance (that is, a map $Y : S \to \mathbb{R}_+$). To simplify the exposition, we assume that every e in S selects an arrival time $t(\mathrm{e})$ uniformly on $[0, 1]$ at random, that the elements arrive in that order and furthermore, that the arrival times are also revealed to the algorithm A_1 upon arrival. Consider the algorithm A_2 that does the following on the instance I. Let X be the set of elements e with arrival time $t(\mathrm{e}) < f := (p_2 - p_1)/(1 - p_1)$. Note that $f \leq p_2$, so X is a subset of A_2's history set. The algorithm will create a new instance Y', on the same system, with weight assignment $Y'(\mathrm{e}) = 0$ for all $\mathrm{e} \in X$ and $Y'(\mathrm{e}) = Y(\mathrm{e})$ for the elements outside X. Now, it simulates A_1 on Y' in the following way. The simulation receives all elements of $S \setminus X$ in their arrival order as before, but all elements in X will be inserted at random times uniformly. More precisely, for every $\mathrm{e} \in X$, the algorithm selects $t'(\mathrm{e})$ uniformly at random on the interval $[f, 1]$, and for every $\mathrm{e} \in S \setminus X$, it sets $t'(\mathrm{e}) = t(\mathrm{e})$. The simulation will consider every element that has $t'(\mathrm{e}) \leq p_2$ as its history set and the rest as the online set (note that some elements from X may fall in the history set and some may fall in the online set, but every element in A_2's online set will also be in the simulation's online set), using Y' as their values. Whenever the simulation accepts an element $\mathrm{e} \in S \setminus X$, it puts e on the solution set ALG. The elements from X that the simulation accepts are discarded. The solution set ALG is independent in $(S, \mathcal{I})$ because it is a subset of the simulation's answer.

To analyze the algorithm, from this point onward let us condition on the set X. Observe that the simulated A_1 receives the elements of the instance given by Y' in a uniform random order. Furthermore, every element e is in the simulation's history set as long as $t'(\mathrm{e}) < p_2$, which happens with probability $(p_2 - f)/(1 - f) = p_1$, so for all purposes, the instance behaves in the same way as in the p_1-DOS problem. For any realization of the times t', let $\mathrm{OPT}_{t'}$ be an optimum Y'-weight set of $\{\mathrm{e} \in S : t'(\mathrm{e}) \geq p_2\}$, and and let OPT_t be an optimum Y-weight set of $\{\mathrm{e} \in S : t(\mathrm{e}) \geq p_2\}$. Since the elements of X have Y'-weight 0, both $\mathrm{OPT}_{t'}$ and OPT_t have the same Y-weight.

Now, since A_1 is ρ-competitive for p_1-DOS, the total Y'-weight of the simulation solution (which is equal to the Y-weight of ALG) is at least ρ times the expected Y'-weight of $\mathrm{OPT}_{t'}$, which in turn equals the expected Y-weight of OPT_t. Removing the condition on X, we obtain that A_2 is ρ-competitive for p_2-DOS.

From here we deduce that $\beta_{S,\mathcal{I}}(p_1) \leq \beta_{S,\mathcal{I}}(p_2)$. Taking the infimum over all systems $(S, \mathcal{I})$ in $\mathcal{C}$ we conclude that $\beta_{\mathcal{C}}(p_1) \leq \beta_{\mathcal{C}}(p_2)$. $\square$

The previous lemma has some nice consequences. If we apply it to the class $\mathcal{M}_1$ of unit rank matroids we recover that for the single-selection p-DOS problem $\alpha(p)$ is increasing in p. Furthermore, it shows that any ρ-competitive algorithm for the generalized secretary problem (the 0-DOS) on a particular class $\mathcal{C}$ can be adapted to the p-DOS problem without decreasing its competitive ratio. To name a few examples: for any p, we get a $1-\Theta(1/\sqrt{k})$-algorithm for p-DOS on k-uniform matroids (adapting Kleinberg's multiple choice secretary algorithm), we get a 1/e-competitive algorithm for p-DOS on transversal matroids (adapting Kesselheim's et al.'s algorithm [86]) and a 1/4-competitive for p-DOS on graphical matroids (adapting Soto et al.'s algorithm [115]), and these are the current best algorithms for all three classes.

1.5.2 Better Guarantees for p-DOS on Special Types of Independence Systems

Babaioff et al. [6] introduced a powerful technique to obtain algorithms for generalized secretary problems by randomly reducing them to a collection of independent parallel single-choice secretary problems. This works on any independence system satisfying a property known as the γ-partition property. [26] If an independence system has the γ-partition property it is easy to create an algorithm for the associated secretary problem (the 0-DOS case) that has competitive ratio γ/e.

Below, we extend this construction to the p-DOS case using a stronger property that we call the γ-sample partition property. We will show that if a system has this particular property then one can easily obtain a $\gamma\alpha(p)$-competitive algorithm for the associated p-DOS problem for every p (Babaioff et al.'s reduction is the special case for $p = 0$). Here $\alpha(p)$ is the optimal guarantee for single-selection p-DOS.

Sample partition property

A *unitary partition matroid* $(S, \mathcal{P})$ is an independence system whose ground set is partitioned into color classes $(S_0, S_1, \ldots, S_m)$, where only S_0 may be empty, so that a set $X \subseteq S$ is independent if and only if X does not contain elements from S_0, and X contains at most 1 element restricted from each other color class. We say that an independence system $(S, \mathcal{I})$ has the γ sample partition property if we can (randomly) define a unitary partition matroid $(S, \mathcal{P})$ on the same ground set so that

1. Every set X independent in $\mathcal{P}$ is also independent in $\mathcal{I}$
2. For any $q \in [0, 1]$, and any assignment of nonnegative weights to S.

$$\mathbb{E}_{\mathcal{P}}[\mathrm{OPT}(\mathcal{P}, q)] \geq \gamma \mathrm{OPT}(\mathcal{I}, q).$$

The notion of γ-partition property of Babaioff et al. [6] is recovered if we only require property (2) to hold for $q = 1$.

Algorithm for p-DOS on a system $(S, \mathcal{I})$ with the γ sample partition property.

On a given instance Y our algorithm does the following:

[26] Rigorously, Babaioff et al. use $1/\gamma$ instead off γ to define this notion, but we prefer to use values smaller than one to be consistent with the presentation of the rest of this chapter.

- Construct the random unit partition matroid $\mathcal{P}$ given by the γ sample partition property, and let $S_1, \ldots, S_m$ be the parts that have allowed size 1.
- Let $H = S[p]$ be the information set of S.
- Run in parallel m instances of the optimal asymptotic algorithm ALG_{t^*} for single-selection p-DOS, one for each part S_{i}. Use $S_{\mathrm{i}} \cap H$ and $S_{\mathrm{i}} \setminus H$ as the history set and online set respectively on the i-th instance. Use the arrival times defined above on each online element. Whenever a copy of ALG_{t^*} selects an element, our algorithm also selects it.

Let ALG be the output set of our algorithm and $Y(\mathrm{ALG})$ be its weight. By construction ALG is independent in the unit partition matroid $\mathcal{P}$ and therefore also in the original independence system. So, our algorithm is correct. The following theorem gives us a bound on its competitive ratio.

Theorem 1.15. *The expected weight of* ALG *is at least* $\alpha(p) \cdot \gamma$ *times* $\mathrm{OPT}(\mathcal{I}, 1-p, Y)$. *Therefore our algorithm for p-DOS on an independence system with* γ *sample partition property is* $\alpha(p) \cdot \gamma$*-competitive, where* $\alpha(p)$ *is the optimal guarantee for single-selection p-DOS.*

Proof. Let us fix $\mathcal{P}$ (recall that it is allowed to be random). Since ALG_{t^*} is $\alpha(p)$-competitive for single-selection, the expected weight of $\mathrm{ALG} \cap S_{\mathrm{i}}$ is at least $\alpha(p)$ times the expected maximum weight of $S_{\mathrm{i}} \setminus H$. Summing over all i we get that the expected weight of ALG (given $\mathcal{P}$) is at least $\alpha(p)$ times $\mathrm{OPT}(\mathcal{P}, 1-p, Y)$. Taking the expectation over $\mathcal{P}$ and using the γ-unit partition property, we obtain

$$\mathbb{E}_{\mathcal{P}}[Y(\mathrm{ALG})] \geq \alpha(p) \cdot \mathbb{E}_{\mathcal{P}}[\mathrm{OPT}(\mathcal{P}, 1-p, Y)] \geq \alpha(p) \cdot \gamma \cdot \mathrm{OPT}(\mathcal{I}, 1-p, Y). \qquad \square$$

We can use Theorem 1.15 above to obtain better guarantees for some classes of independence systems. First of all we observe that our notion of γ sample partition property, although stronger than the γ partition property, is not really that restrictive. In fact, most (if not all) proofs that a particular system satisfy the weaker notion of γ partition, can be adapted to the stronger version directly.

We mentioned that this theorem can be used to get lower bounds for $\beta_{\mathcal{C}}(p)$ that are strictly larger than the ones available for $\beta_{\mathcal{C}}(0)$ for certain classes $\mathcal{C}$. A particularly interesting example is the class $\mathcal{G}$ of all graphic matroids. Babaioff et al. [6] showed that graphic matroids have the partition property for $\gamma = 1/3$, and thus they got a 1/(3e)-competitive algorithm for graphic matroids. Korula and Pál [91] improved this by showing that this class admits the partition property for $\gamma = 1/2$, obtaining a 1/(2e)-competitive algorithm. The current best algorithm by Soto et al. [115] is 1/4-competitive and uses a different technique that does not reduce to the single-choice secretary problem. Using the monotonicity of $\beta_{\mathcal{G}}$, we know that $\beta_{\mathcal{G}}(p)$ is at least $1/4$ for every p. However, it is quite simple to modify Korula and Pál's proof to show that graphic matroids have the stronger 1/2 sample partition property. Using the algorithm given by Theorem 1.15 we obtain that $\beta_{\mathcal{G}}(p) \geq \alpha(p)/2$. We note that $\alpha(p)/2$ grows from 1/(2e) when $p = 0$ to $\alpha^*/2 \approx 0.3725$, when $p = 1$. So, for sufficiently large p, $\alpha(p)/2$ beats $1/4$.

By adapting the proofs in [6] and [114] we get a few other classes of matroids with constant

γ sample partition property such as uniform matroids with $\gamma = 1 - 1/\mathrm{e}$, cographic matroids ($\gamma = 1/3$), k-column sparse matroids ($\gamma = 1/k$) and matroids of density d ($\gamma = 1/\mathrm{d}$).

1.5.3 Limiting Problem as $p \to 1$ and Consequences for the Matroid Secretary Problem (MSP)

In Lemma 1.14 we showed that for any class $\mathcal{C}$, the function $\alpha_C(p)$ is increasing, our next lemma shows that this function cannot grow extremely fast.

Lemma 1.16. *Let $p_1, p_2 \in [0, 1)$ with $p_1 < p_2$. For any ρ-competitive algorithm for p_2-DOS on $(S, \mathcal{I})$ we can construct a $\rho(1-p_2)/(1-p_1)$-competitive algorithm for p_1-DOS. As a corollary, for any class $\mathcal{C}$ of independence systems, $\beta_{\mathcal{C}}(p_1) \geq \beta_{\mathcal{C}}(p_2) \cdot (1-p_2)/(1-p_1)$. Applying this to the single-selection problem we conclude that $\alpha(0) \geq \alpha(p)(1-p)$.*

Proof. Fix $(S, \mathcal{I})$, p_1 and p_2 and let A_2 be any ρ-competitive algorithm for p_2-DOS on $(S, \mathcal{I})$. We will use the same random arrival time interpretation of the elements of the system. Consider a new algorithm A_1 that on any instance Y for p_1-DOS it simply mimics what A_2 would do on the same instance and arrival times (note that all the elements that A_2 accepts arrive after time p_2 so they also belong to the online set of A_1). The set ALG that A_1 returns is independent in $(S, \mathcal{I})$. To analyze its performance, we need a simple observation. Let $S[t_1, t_2]$ denote the elements arriving between times t_1 and t_2. If X is the maximum weight independent set of $S[p_1, 1]$ then because of the random arrival, $X \cap S[p_2, 1]$ has expected weight $Y(X) \cdot (1-p_2)/(1-p_1)$, therefore, the maximum weight independent set of $S[p_2, 1]$ has at least that expected weight. Using that A_2 is ρ-competitive for p_2-DOS

$$\rho(1-p_2)/(1-p_1)\mathrm{OPT}(\mathcal{I}, 1-p_1, Y) \leq \rho\mathrm{OPT}(\mathcal{I}, 1-p_2, Y) \leq Y(\mathrm{ALG}).$$

From here we conclude that A_1 is $\rho(1-p_2)/(1-p_1)$ competitive for p_1-DOS, and we deduce that $\beta_{S,\mathcal{I}}(p_1) \geq (1-p_2)/(1-p_1)\beta_{S,\mathcal{I}}(p_2)$. We finish the proof taking infimum on the previous inequality over all systems $(S, \mathcal{I})$ in $\mathcal{C}$. $\square$

Recall now that for the single-selection p-DOS problem the limit $\lim_{p\to 1} \alpha(p)$ coincides with the factor α^* associated to the single-selection i.i.d. prophet inequality with known distribution. An interesting question is whether something similar occurs for other classes of independence systems different than matroids of rank 1. For example, denote again $\mathcal{M}$ and $\mathcal{M}_r$ to denote the classes of all matroids and that of all matroids of rank r respectively. Let $L = \lim_{p\to 1} \beta_{\mathcal{M}}(p)$ and $L_r = \lim_{p\to 1} \beta_{\mathcal{M}_r}(p)$ so that $L_1 = \alpha^*$. It would be natural to ask whether there is an analog of the i.i.d. prophet inequality on matroids whose optimal competitive ratio equals L.

There are many candidates one could study, for example in the i.i.d. MSP, every element of a known matroid is assigned independently a value from a known distribution, and the values are later revealed to the DM. Soto [114] studied a generalization of the i.i.d. case known as the random-assignment MSP in which an adversary selects a list of non-negative values which are then randomly assigned to the elements to the matroid, which in turn is presented in random order to the DM. Another alternative is the prophet secretary model on matroids, studied by Ehsani et al. [56] in which every element from the matroid receives independently a value from a known distribution, which may be different for every element.

Proving that any of this problems behaves like the limit of p-DOS as $p \to 1$ on all matroids may be, in fact, a very difficult task. For if we are able to show that then we would have, indirectly, solved the matroid secretary conjecture. Indeed, for all the i.i.d., the random-assignment and the prophet secretary problem on matroids, constant competitive algorithms are known [114, 56], so if any of those cases holds then $L > 0$. However, since $L = \lim_{p\to 1} \beta_{\mathcal{M}}(p)$, then there exists a sufficiently small $\varepsilon > 0$, so that $\beta_{\mathcal{M}}(1-\varepsilon) \geq L/2$. But then, by Lemma 1.14, $\beta_{\mathcal{M}}(0) \geq \varepsilon L/2 > 0$, meaning that every matroid admits a constant competitive algorithm for the matroid secretary problem.

In any case, it is likely that neither the random-assignment nor the prophet secretary problem are the correct candidates, because if one restricts the former problem to the class $\mathcal{M}_1$ we recover the classic secretary problem whose optimal competitive ratio is $1/\mathrm{e} \neq \alpha^*$, and the latter becomes the single-selection prophet secretary problem with known distribution for which an upper bound of $0.732 < \alpha^*$ is known [45].

1.6 Proofs of Section 1.3

1.6.1 Coupling Argument for Monotonicity

We take an algorithm ALG for $Y_{[N+1]}$ and obtain an algorithm for $Y_{[N]}$ with at least as much reward as for $Y_{[N+1]}$. Indeed, we define ALG$'$ for $Y_{[N]}$ in the following way. We insert a dummy item with the smallest rank in a random position, and run ALG on the sequence of $N+1$ resulting items. If ALG attempts to select the dummy item, ALG$'$ simply does not stop and obtains a reward of Y_{N+1}. We couple both algorithms by taking the position of the dummy item to be the same as Y_{N+1}. Then, every time ALG selects an item in $Y_{[N+1]}$ greater than Y_{N+1}, ALG$'$ selects the same item in $Y_{[N]}$. When ALG selects Y_{N+1}, ALG$'$ does not stop, in which case the reward is defined as Y_{N+1}. If ALG does not stop, its reward is $Y_{N+2} \leq Y_{N+1}$. In all cases ALG$'$ obtains more than ALG.

1.6.2 Convergence of $\mathbf{E}(\mathbf{ALG}^*_N(Y))$ to CLP_p

Denote by $\mathbf{E}(\mathrm{ALG}^*_N(Y))$ the expected reward of the optimal algorithm for a given sequence Y, and $N \geq 1$. We start by relaxing the problem. Given a value $Z \in (-\infty, Y_1)$, we consider the problem where we get a reward of Z if the algorithm does not stop. This means we replace with Z in the sequence Y all values $Y_j < Z$. We denote this modified sequence by Y^Z. We then proceed in three main steps. First, we prove that for fixed Z, when $p = h/N$ the difference between the optimal values of $\mathrm{LP}_{h,N}(Y^Z)$ and $\mathrm{CLP}_p(Y^Z)$[27] tends to 0 when $N \to \infty$. Second, we prove that the optimal value of $\mathrm{CLP}_p(Y^Z)$ is a continuous function of p and use a concentration bound to show that the expectation of the optimal algorithm $\mathbf{E}(\mathrm{ALG}^*_N(Y^Z))$ tends to the optimal value of $\mathrm{CLP}_p(Y^Z)$ when N tends to ∞. And third, we conclude by making Z tend to $\lim_{\mathrm{i}\to\infty} Y_{\mathrm{i}}$.

For the first step, notice that for any $Z > \lim_{\mathrm{i}\to\infty} Y_{\mathrm{i}}$, we only care about finitely many Y_j, so we can argue about the convergence of each element in the summations of the objective functions. Note also that for any $k \geq \ell$, if $\mathrm{i}/N = t$,

$$\sum_{j=\ell}^{k} \frac{\mathrm{i}}{N} \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}} \underset{N\to\infty}{\longrightarrow} \sum_{j=\ell}^{k} \binom{j-1}{\ell-1} (1-t)^{j-\ell} t^{\ell}, \tag{1.22}$$

simply because they represent the probabilities of drawing samples with or without replacement. Indeed, they correspond to the probability that we need to draw at most k random elements from a total of N to get at least ℓ from a given subset of i elements. Now, for an optimal solution q of $\mathrm{CLP}_p(Y^Z)$, we define a solution for $\mathrm{LP}_{h,N}(Y^Z)$ given by

$$x_{\mathrm{i},\ell} = \int_{\frac{\mathrm{i}-1}{N}}^{\frac{\mathrm{i}}{N}} q(t,\ell)\mathrm{d}t.$$

From the feasibility of q one can easily show that x is feasible for $\mathrm{LP}_{h,N}(Y^Z)$. This, together with Equation (1.22), implies that the limit of the optimal value of $\mathrm{LP}_{h,N}(Y^Z)$ is at least the optimal value of $\mathrm{CLP}_p(Y^Z)$. For the opposite inequality, from an optimal solution x^* of

[27] Here we make explicit the dependence of CLP_p on the sequence Y.

$\mathrm{LP}_{h,N}(Y^Z)$ and a given $\varepsilon > 0$, define

$$q(t,\ell) = \begin{cases} N x^*_{\mathrm{i},\ell}(1-\varepsilon) & \text{if } \ell \le \mathrm{i} \text{ and } \mathrm{i} = \lceil t \cdot N \rceil \\ 0 & \text{otherwise.} \end{cases}$$

For a certain ε that tends to 0 with N, q is feasible for $\mathrm{CLP}_p(Y^Z)$. This, together with Equation (1.22) implies that the optimal value of CLP_p is at least the limit optimal value of $\mathrm{LP}_{h,N}(Y^Z)$.

Now, we show the optimal value of $\mathrm{CLP}_p(Y^Z)$ is continuous. In fact, note on the one hand it is decreasing, since we can take a solution q for a given $p \in (0,1)$ and extend it to $[p',1]$ for $p' < p$ setting it equal to 0 for $t \in [p',p]$. On the other hand, from a solution for p' we can obtain a solution for p by simply truncating it. Since $F_k(q)$ is continuous in p, if p' is close to p, then the truncated solution is close to the solution for p'. Although the number of items in H is random, when $N \to \infty$, $|H|/N$ converges to p, so the continuity of the value of $\mathrm{CLP}_p(Y^Z)$ implies that if we use the optimal solution of $\mathrm{LP}_{|H|,N}$, the expected reward converges to $\mathrm{CLP}_p(Y^Z)$.

Finally, when we make Z tend to $\lim_{\mathrm{i}\to\infty} Y_{\mathrm{i}}$, the optimal value of $\mathrm{CLP}_p(Y^Z)$ tends to the optimal solution of $\mathrm{CLP}(Y)$, and the limit (when N tends to infinity) of $\mathbf{E}(\mathrm{ALG}^*(Y^Z))$ tends to $\mathbf{E}(\mathrm{ALG}^*(Y))$, so we conclude that if they exist they must be equal.

1.6.3 Monotonicity of $\sum_{j=\ell}^{k} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^\ell$

Lemma 1.17. *For any fixed $k \ge 1, \ell \le k$, the term $\sum_{j=\ell}^{k} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^\ell$ as a function of $t \in [0,1]$ is increasing.*

Proof. The derivative of the function with respect to t is

$$\begin{aligned}
&\sum_{j=\ell}^{k} \binom{j-1}{\ell-1}\Big(\ell(1-t)^{j-\ell}t^{\ell-1} - (j-\ell)(1-t)^{j-\ell-1}t^\ell\Big) \\
&= \sum_{j=\ell}^{k} \binom{j-1}{\ell-1}\Big(\ell(1-t) - (j-\ell)t\Big) \cdot (1-t)^{j-\ell-1}t^{\ell-1} \\
&= \sum_{j=\ell}^{k} \binom{j-1}{\ell-1}\Big(j(1-t) - (j-\ell)\Big) \cdot (1-t)^{j-\ell-1}t^{\ell-1} \\
&= t^{\ell-1}\sum_{j=\ell}^{k} \left(\binom{j}{\ell-1}(j-\ell+1)(1-t)^{j-\ell} - \binom{j-1}{\ell-1}(j-\ell)(1-t)^{j-\ell-1}\right) \\
&= t^{\ell-1}\binom{k}{\ell-1}(k-\ell+1)(1-t)^{k-\ell} \ge 0\,,
\end{aligned}$$

where in the second last equality we used the identity $\binom{j-1}{\ell-1}j = \binom{j}{\ell-1}(j-\ell+1)$, and in the last equality we reduced the telescopic sum. $\square$

Lemma 1.18. *For any fixed $k \geq 1$ and $t \in [0,1]$, the term $\sum_{j=\ell}^{k} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}$ as a function of ℓ is decreasing.*

Proof. We want to prove that for $\ell \leq k-1$,

$$\sum_{j=\ell}^{k} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell} \geq \sum_{j=\ell+1}^{k} \binom{j-1}{\ell}(1-t)^{j-\ell-1}t^{\ell+1}\,. \tag{1.23}$$

If we compare term by term in the sum (with the same value for j), we have that

$$\begin{aligned}\frac{\binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}}{\binom{j-1}{\ell}(1-t)^{j-\ell-1}t^{\ell+1}} &= \frac{\ell(1-t)}{(j-\ell)t} \\ &= \frac{\ell - t\ell}{tj - t\ell}\,,\end{aligned}$$

which is larger than 1 whenever $j \leq \ell/t$. Thus, we can safely conclude that Equation (1.23) is true when $k \leq \ell/t$.

On the other hand, we make use of the fact that for any $y \in (-1,1)$ and $\ell \in \mathbb{N}$, the identity $\sum_{j=\ell}^{\infty} \binom{j}{\ell} y^j = \frac{y^\ell}{(1-y)^{\ell+1}}$ holds true. From this it is easy to see that when k tends to ∞, the term tends to 1, so we can rewrite it as

$$\sum_{j=\ell}^{k} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell} = 1 - \sum_{j=k+1}^{\infty} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}\,. \tag{1.24}$$

Therefore, we can rewrite Equation (1.23) as

$$\sum_{j=k+1}^{\infty} \binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell} \leq \sum_{j=k+1}^{\infty} \binom{j-1}{\ell}(1-t)^{j-\ell-1}t^{\ell+1}\,,$$

and then, whenever $k > \ell/t$ we can conclude that the inequality is true by comparing term by term here. □

1.6.4 Concavity of $F_k(t)$ in Each Variable

Proof. We start by rearranging the sums in the definition of $F_k(t)$:

$$\begin{aligned}F_k(t) &= \sum_{j=1}^{k}\sum_{\mathrm{i}=1}^{\infty}\int_{t_\mathrm{i}}^{t_{\mathrm{i}+1}}\sum_{\ell=1}^{j\wedge \mathrm{i}}\frac{T_\mathrm{i}}{\tau^{\mathrm{i}+1}}\binom{j-1}{\ell-1}(1-\tau)^{j-\ell}\tau^{\ell}\mathrm{d}\tau. \\ &= \sum_{\ell=1}^{k}\sum_{\mathrm{i}=\ell}^{\infty}\int_{t_\mathrm{i}}^{t_{\mathrm{i}+1}}\frac{T_\mathrm{i}}{\tau^{\mathrm{i}+1}}\sum_{j=\ell}^{k}\binom{j-1}{\ell-1}(1-\tau)^{j-\ell}\tau^{\ell}\,\mathrm{d}\tau.\end{aligned}$$

We now calculate the second derivative with respect to t_s, for some $s \geq 1$. Recall that we defined $T_\mathrm{i} = \prod_{j=1}^{\mathrm{i}} t_j$. Observe that in the sum indexed by i the terms with $\mathrm{i} < s-1$ do not depend of t_s, and the terms with i $> s$ are linear in t_s, so neither of them affect the second derivative. Thus, if we denote $H(\tau,\ell,k) = \sum_{j=\ell}^{k}\binom{j-1}{\ell-1}(1-\tau)^{j-\ell}\tau^{\ell}$, we have that

$$\begin{aligned}
\frac{\partial^2}{\partial t_s^2}F_k(t) &= \sum_{\ell=1}^{k}\frac{\partial^2}{\partial t_s^2}\left(\mathbb{1}_{s-1\geq\ell}\int_{t_{s-1}}^{t_s}\frac{T_{s-1}}{\tau^s}H(\tau,\ell,k)\,\mathrm{d}\tau + \mathbb{1}_{s\geq\ell}\int_{t_s}^{t_{s+1}}\frac{T_s}{\tau^{s+1}}H(\tau,\ell,k)\,\mathrm{d}\tau\right)\\
&= \sum_{\ell=1}^{k}\frac{\partial}{\partial t_s}\left(\mathbb{1}_{s-1\geq\ell}\frac{T_{s-1}}{t_s^s}H(t_s,\ell,k) - \mathbb{1}_{s\geq\ell}\frac{T_s}{t_s^{s+1}}H(t_s,\ell,k) + \mathbb{1}_{s\geq\ell}\int_{t_s}^{t_{s+1}}\frac{T_{s-1}}{\tau^{s+1}}H(\tau,\ell,k)\mathrm{d}\tau\right).
\end{aligned}$$

Now, notice that $\frac{T_{s-1}}{t_s^s} = \frac{T_s}{t_s^{s+1}}$, so,

$$\begin{aligned}
\frac{\partial^2}{\partial t_s^2}F_k(t) =& -\mathbb{1}_{s\leq k}\frac{\partial}{\partial t_s}\frac{T_{s-1}}{t_s^s}H(t_s,s,k) + \sum_{\ell=1}^{\min\{k,s\}}\frac{\partial}{\partial t_s}\int_{t_s}^{t_{s+1}}\frac{T_{s-1}}{\tau^{s+1}}H(\tau,\ell,k)\mathrm{d}\tau\\
=& -\mathbb{1}_{s\leq k}\frac{\partial}{\partial t_s}\frac{T_{s-1}}{t_s^s}H(t_s,s,k) - \sum_{\ell=1}^{\min\{k,s\}}\frac{T_{s-1}}{t_s^{s+1}}H(t_s,\ell,k).
\end{aligned}$$

At this point it is already clear that for $s > k$ the second derivative is negative. So from now on we assume $s \leq k$. Let us expand $H(t_s,s,k)$ to calculate the last derivative.

$$\begin{aligned}
\frac{\partial^2}{\partial t_s^2}F_k(t) =& -\frac{\partial}{\partial t_s}\frac{T_{s-1}}{t_s^s}\sum_{j=s}^{k}\binom{j-1}{s-1}(1-t_s)^{j-s}t_s^s - \sum_{\ell=1}^{s}\frac{T_{s-1}}{t_s^{s+1}}H(t_s,\ell,k)\\
&= T_{s-1}\left(\sum_{j=s+1}^{k}\binom{j-1}{s-1}(j-s)(1-t_s)^{j-s-1} - \sum_{\ell=1}^{s}\frac{1}{t_s^{s+1}}H(t_s,\ell,k)\right)\\
&= T_{s-1}\left(\sum_{j=s+1}^{k}\binom{j-1}{s}s(1-t_s)^{j-s-1} - \sum_{\ell=1}^{s}\frac{1}{t_s^{s+1}}H(t_s,\ell,k)\right)\\
&= T_{s-1}\left(\frac{s}{t_s^{s+1}}H(t_s,s+1,k) - \sum_{\ell=1}^{s}\frac{1}{t_s^{s+1}}H(t_s,\ell,k)\right)\\
&= s\frac{T_{s-1}}{t_s^{s+1}}\left(H(t_s,s+1,k) - \sum_{\ell=1}^{s}\frac{1}{s}H(t_s,\ell,k)\right).
\end{aligned}$$

To conclude, note that $H(\tau,\ell,k)$ is the probability that a NegativeBinomial(τ,ℓ) is at most k, i.e., the probability that at most k independent coin tosses are necessary to obtain ℓ heads, if the coin comes up head with probability τ. Therefore, $H(t_s,\ell,k) \geq H(t_s,s+1,k)$ for all $\ell \leq s$, so we get that $\frac{\partial^2}{\partial t_s^2}F_k(t) \leq 0$. This implies that $F_k(t)$ is concave as a function of t_s, for all $s \geq 1$. □

1.7 Proofs of Section 1.4

1.7.1 Derivation of $\mathrm{SDLP}_{h,N}$

To establish the equivalence between both problems we show that for any x feasible in the maximization problem, the optimal values of inner problems

$$(A) \quad \min_{\substack{Y\in\mathcal{Y}_N\\ E(\mathrm{OPT}(Y))=1}} \quad \mathbf{E}(\mathrm{ALG}_x(Y))$$

and

$$(B)\quad \max_{\alpha} \quad \alpha$$
$$\text{s.t.}\quad \alpha - \frac{\sum_{j=1}^{k}\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{j}\frac{\mathrm{i}x_{\mathrm{i},\ell}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}}{\sum_{j=1}^{k}\frac{N-h}{N-j+1}\prod_{s=0}^{j-2}\frac{h-s}{N-s}} \leq 0 \quad \forall k \in [h+1]. \tag{1.25}$$

are equal. From Lemma 1.1 we know that

$$\mathbb{P}(\mathrm{ALG}_x(Y) = Y_j) = \sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{\mathrm{i}}\frac{\mathrm{i}x_{\mathrm{i},\ell}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}.$$

From Lemma 1.3 we know that

$$\mathbb{P}(\mathrm{OPT}(Y) = Y_j) = \begin{cases} \frac{N-h}{N-j+1}\prod_{s=0}^{j-2}\frac{h-s}{N-s} & 1 \leq j \leq h+1 \\ 0 & \text{otherwise.}\end{cases}$$

That way, constraint (1.25) can be read as

$$\mathbb{P}(\mathrm{ALG}_x(Y) \geq Y_k) \geq \alpha\mathbb{P}(\mathrm{OPT}(Y) \geq Y_k) \quad \forall k \in [h+1]. \tag{1.26}$$

If α is feasible, it will hold that $\mathbf{E}(\mathrm{ALG}_x(Y)) \geq \alpha\mathbf{E}(\mathrm{OPT}(Y))$ for any instance Y of N items. Indeed, we can integrate $\mathbb{P}(\mathrm{ALG}_x(Y) \geq z)$ and $\mathbb{P}(\mathrm{OPT}(Y) \geq z)$ at both sides of (1.26) to obtain the bound, as both random variables can only equal values of items. Restricting the first $h+1$ items is enough, as $\mathbb{P}(\mathrm{OPT}(Y) \geq Y_{h+1}) = 1$, and $\mathbb{P}(\mathrm{ALG}_x(Y) \geq Y_k)$ is non-decreasing in k. In particular, if we restrict to Y such that $\mathbf{E}(\mathrm{OPT}(Y)) = 1$, we get that $\mathbf{E}(\mathrm{ALG}_x(Y)) \geq \alpha$. This holds for feasible α, so it holds for the optimal solution α^* and we get the optimal value of problem A is at least α^*.

Now consider an optimal solution for problem B, α^*. It must be the case that constraint (1.25) is binding for some k^*. Consider the following instance Y^{k^*}, where we set $Y_1 = \cdots = Y_{k^*} = \lambda_{k^*}$, and $Y_j = 0$ if $j > k^*$. Here, $\lambda_{k^*} > 0$ is such that $\mathbf{E}(\mathrm{OPT}(Y^{k^*})) = 1$. We have that k^* is binding, so

$$\mathbf{E}(\mathrm{ALG}_x(Y_{k^*})) = \frac{\mathbf{E}(\mathrm{ALG}_x(Y^{k^*}))}{\mathbf{E}(\mathrm{OPT}(Y^{k^*}))} = \frac{\lambda_{k^*}\mathbb{P}(\mathrm{ALG}_x(Y^{k^*}) \geq Y^{k^*})}{\lambda_{k^*}\mathbb{P}(\mathrm{OPT}(Y^{k^*}) \geq Y^{k^*})} = \alpha^*.$$

Now, Y^{k^*} is feasible in problem (A), concluding that the optimal value of problem A is at most $\mathbf{E}(\mathrm{ALG}_x(Y^{k^*})) = \alpha^*$. The equivalence between the two problems follows by replacing the inner problems.

1.7.2 Solution of SDRP_p for $p < 1/\mathrm{e}$

Proof. The upper bound follows immediately from Lemma 1.16 (see also, Kaplan et al. [82, Theorem 3.8], [40]). To prove that the bound is tight we find a feasible solution of SDRP_p

attaining this value. Take then $t_1 = 1/\mathrm{e}$, $t_{\mathrm{i}} = 1$ for $\mathrm{i} \geq 2$, we prove that the objective value of this solution is at least $1/(\mathrm{e}(1-p))$

To this end first observe that the following inequalities hold for all $0 \leq p \leq 1/\mathrm{e}$.

$$\int_{1/\mathrm{e}}^{1} \frac{(1-\tau)^{j-1}}{\tau} \mathrm{d}\tau \geq \frac{1}{\mathrm{e}^{j-1}} \geq p^{j-1} .$$

Indeed the second inequality is direct. Note that the first is actually an equality for $j = 1$ and $j = 2$. Also for $j \geq 5$ the inequality follows since $\int_{1/\mathrm{e}}^{1}(1-\tau)^{j-1}/\tau \mathrm{d}\tau \geq \int_{1/\mathrm{e}}^{1}(1-\tau)^{j-1}\mathrm{d}\tau = (1-1/\mathrm{e})^j/j \geq 1/\mathrm{e}^{j-1}$. Finally, for $j = 3, 4$ it follows from a straightforward calculation.

Replacing our solution in $F_k(t)$ and using the previous inequalities we get

$$F_k(t) = \frac{1}{\mathrm{e}} \sum_{j=1}^{k} \int_{1/\mathrm{e}}^{1} \frac{(1-\tau)^{j-1}}{\tau} \mathrm{d}\tau \geq \frac{1}{\mathrm{e}} \sum_{j=1}^{k} p^{j-1} = \frac{1-p^k}{\mathrm{e}(1-p)} .$$

If we replace these values of $F_k(t)$ in the inner minimization of SDRP_p, we get that all ratios equal $1/(\mathrm{e}(1-p))$, as $1-p^k$ cancel out. We conclude that the considered solution is feasible and therefore the optimal value of SDRP_p (which is $\alpha(p)$) is at least $1/(\mathrm{e}(1-p))$. □

1.7.3 Details on Numerical Bounds

We now develop the optimization problems used for obtaining upper and lower bounds of $\alpha(p)$ when $p \in (0, 1)$. For the upper bound, we construct a linear program based on SDCLP_p. In this linear program we partition interval $(p, 1)$ into $N(1-p)$ intervals of equal length. Inside of interval $(\frac{\mathrm{i}-1}{N}, \frac{\mathrm{i}}{N}]$ we restrict variables $q(t, \ell)$ to be constant for every $\ell \geq 1$ and rename them $x_{\mathrm{i},\ell}$. We modify the feasibility constraints for making them slightly less restrictive (and equivalent as $N \to \infty$). In the minmax constraint we replace the term $(1-t)^{j-\ell}t^{\ell}$ by its upper bound $\left(1-\frac{\mathrm{i}-1}{N}\right)^{j-\ell}\left(\frac{\mathrm{i}}{N}\right)^{\ell}$. To deal with the infinite number of variables and constraints, we introduce the parameter $k_{\max}$, which indicates that only the first $k_{\max}$ terms of the stochastic dominance constraint will be considered in the maximization. As only the first $k_{\max}$ amount of variables are considered in the objective function, we can consider only variables $x_{\mathrm{i},\ell}$ with $\ell \leq k_{\max}$. We call this problem $\mathrm{UBP}_{p,N,k_{\max}}$ (for Upper Bound Problem).

$$(\mathrm{UBP}_{p,N,k_{\max}}) \qquad \max_{x,\alpha} \quad \alpha$$

$$\text{s.t.} \qquad \mathrm{i}x_{\mathrm{i},\ell} + \sum_{j=h+1}^{\mathrm{i}-1} \sum_{s=1}^{k_{\max}} x_{j,s} \leq 1 \qquad \forall \mathrm{i} \in [N] \setminus [h], \forall \ell \in [k_{\max}]$$

$$\alpha - \frac{\sum_{j=1}^{k} \sum_{\mathrm{i}=h+1}^{N} \sum_{\ell=1}^{j} x_{\mathrm{i},\ell} \frac{\binom{j-1}{\ell-1}\left(\frac{\mathrm{i}}{N}\right)^{\ell}}{\left(1-\frac{\mathrm{i}-1}{N}\right)^{\ell-j}}}{1-p^k} \leq 0 \qquad \forall k \in [k_{\max}]$$

$$x_{\mathrm{i},\ell} \geq 0 \qquad \forall \mathrm{i} \in [N] \setminus [h], \forall \ell \in [k_{\max}]$$

For the lower bound, we numerically solve a truncated version of SDRP_p, in which we use the parameter $k_{\max}$ to limit the amount of terms to be considered in the stochastic

Table 1.2: Upper and lower bounds obtained for multiples of 0.1. Parameters used for $\mathrm{UBP}_{p,N,k_{\max}}$ were $N = 1000$ and $k_{\max} = \ln(N/(1-p))$. For $\mathrm{LBP}_{p,k_{\max}}$, $k_{\max} = \ln 0.001/\ln p$ was used. For values of p up to $1/\mathrm{e}$ the bounds are exact and thus the difference simply comes from rounding.

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Lower bound	0.408	0.459	0.525	0.609	0.671	0.702	0.718	0.728	0.730
Upper bound	0.409	0.460	0.526	0.610	0.672	0.704	0.721	0.733	0.744

dominance constraint. As the solution must be a lower bound, we replace the denominator of the last term of the min-max problem by 1. This makes the objective function to be lower than SDRP_p by at most $p^{k_{\max}}$. As in the upper bound, reducing the number of stochastic dominance constraints also reduces the amount of variables to be considered, only needing to consider t_{i} with $\mathrm{i} \leq k_{\max}$. For simplicity, we fix $t_{k_{\max}+1} = 1$ as a parameter. We call this problem $\mathrm{LBP}_{p,k_{\max}}$ (for Lower Bound Problem).

$$
\begin{aligned}
(\mathrm{LBP}_p) \quad & \max_{t,\,\alpha\in[0,1]} \quad \alpha \\
\text{s.t.} \quad & \alpha \leq \frac{1}{1-p^k} \sum_{j=1}^{k} \sum_{\mathrm{i}=1}^{k_{\max}} \int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}} \sum_{\ell=1}^{j\wedge\mathrm{i}} \frac{T_{\mathrm{i}}}{\tau^{\mathrm{i}+1}} \binom{j-1}{\ell-1} (1-\tau)^{j-\ell}\tau^{\ell} \mathrm{d}\tau \qquad \forall k \in [k_{\max}-1] \\
& \alpha \leq \sum_{j=1}^{k_{\max}} \sum_{\mathrm{i}=1}^{k_{\max}} \int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}} \sum_{\ell=1}^{j\wedge\mathrm{i}} \frac{T_{\mathrm{i}}}{\tau^{\mathrm{i}+1}} \binom{j-1}{\ell-1} (1-\tau)^{j-\ell}\tau^{\ell} \mathrm{d}\tau \\
& p \leq t_{\mathrm{i}} \leq t_{\mathrm{i}+1} \leq 1 \qquad \forall \mathrm{i} \in [k_{\max}]
\end{aligned}
$$

The bounds obtained for some values of p are shown in Table 1.2. Note that as p gets closer to 1 we need more variables and therefore our upper and lower bounds are slightly off. This can definitely be improved by just considering more variables when solving $\mathrm{UBP}_{p,N,k_{\max}}$ and $\mathrm{LBP}_{p,k_{\max}}$ since they converge to each other.

1.7.4 Proof of Theorem 1.13

We first introduce two lemmas that bound the ratio between the coefficients of the linear programs. Then, to bound the difference between the values, we produce a solution for one problem from a solution to the other, and vice versa.

Lemma 1.19. *For integers N, h, k, such that $p = \frac{h}{N} \in (0,1)$, $N \geq \frac{1}{(1-p)} + 1$, and $1 \leq k \leq h+1$, we have that*

$$
1 \leq \frac{\sum_{j=1}^{k} \frac{N-h}{N-j+1} \prod_{s=0}^{j-2} \frac{h-s}{N-s}}{1-p^k} \leq 1 + \frac{\mathrm{e}}{(1-p)(N-1)}. \tag{1.27}
$$

Proof. Denote $A_j = \frac{N-h}{N-j+1} \prod_{s=0}^{j-2} \frac{h-s}{N-s}$ and $B_j = (1-p)p^{j-1}$. In order to bound the ratio $\frac{\sum_{j=1}^{k} A_j}{\sum_{j=1}^{k} B_j}$ for $1 \leq k \leq h+1$, we find a uniform bound on $\frac{A_j}{B_j}$ for $1 \leq j \leq h+1$.

Recall that by definition $h = p \cdot N$. It is easy to see that $\frac{A_1}{B_1} = 1$, and that $\frac{A_{j+1}}{B_{j+1}} = \frac{A_j}{B_j} \cdot \frac{h-j+1}{(N-j) \cdot p}$. Therefore, $\frac{A_{j+1}}{B_{j+1}} \geq \frac{A_j}{B_j}$ if and only if $h - j + 1 \geq h - pj$, which is equivalent to $j \leq \frac{1}{1-p}$. Then we can conclude that for all $j \leq h + 1$,

$$
\begin{aligned}
\frac{A_j}{B_j} &= \frac{A_1}{B_1} \cdot \prod_{\mathrm{i}=1}^{j-1} \left(\frac{h - \mathrm{i} + 1}{h - p\mathrm{i}} \right) \\
&\leq \prod_{\mathrm{i}=1}^{\lfloor 1/(1-p) \rfloor} \left(\frac{h - \mathrm{i} + 1}{h - p\mathrm{i}} \right) \\
&\leq \left(\frac{h}{h - p} \right)^{\lfloor 1/(1-p) \rfloor} \\
&= \left(1 + \frac{1}{N - 1} \right)^{\lfloor 1/(1-p) \rfloor} \\
&\leq 1 + \frac{1}{N - 1} \cdot \frac{1}{1 - p} \left(1 + \frac{1}{1/(1-p)} \right)^{1/(1-p)} \\
&\leq 1 + \frac{\mathrm{e}}{(1 - p)(N - 1)},
\end{aligned}
$$

where the second last inequality comes from doing a first-order approximation of a convex function.

For the lower bound of 1, it is enough to note that $\sum_{j=1}^{h+1} A_j = 1$ and that $\sum_{j=1}^{h+1} B_j \leq \sum_{j=1}^{\infty} B_j = 1$, together with the already mentioned fact that $A_j \geq B_j$ if and only if $j \leq \frac{1}{1-p}$. $\square$

Lemma 1.20. *For positive integers N, i, j, ℓ such that $N \geq 32$, $\frac{\sqrt{N} \log N}{1-p} \leq \mathrm{i} \leq N - \frac{\sqrt{N} \log N}{1-p}$, $j \leq \frac{\log N}{1-p}$, and $\ell \leq j$, and for a real $t \in \left[\frac{\mathrm{i}-1}{N}, \frac{\mathrm{i}}{N}\right]$, we have that*

$$
1 - \frac{3 \log N}{(1-p)\sqrt{N}} \leq \frac{\frac{\mathrm{i}}{N} \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}}{\binom{j-1}{\ell-1}(1-t)^{j-\ell} t^{\ell}} \leq 1 + \frac{5 \log N}{(1-p)\sqrt{N}} \tag{1.28}
$$

Proof. We start by rewriting the expression in the middle of Equation (1.28).

$$
\begin{aligned}
\frac{\frac{\mathrm{i}}{N} \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}}{\binom{j-1}{\ell-1}(1-t)^{j-\ell} t^{\ell}} &= \frac{\frac{\mathrm{i}}{N} \cdot \frac{(N-\mathrm{i})!}{(N-\mathrm{i}-j+\ell)!} \cdot \frac{(\mathrm{i}-1)!}{(\mathrm{i}-\ell)!} \cdot \frac{(N-j)!}{(N-1)!}}{(1-t)^{j-\ell} t^{\ell}} &\quad (1.29) \\
&= \frac{\prod_{k=0}^{j-\ell-1} \frac{N-\mathrm{i}-k}{N-k} \cdot \prod_{k=0}^{\ell-1} \frac{\mathrm{i}-k}{N-j+\ell-k}}{(1-t)^{j-\ell} t^{\ell}}. &\quad (1.30)
\end{aligned}
$$

Now, the expression in Equation (1.30) is clearly at most

$$\frac{\prod_{k=0}^{j-\ell-1}\frac{N-\mathrm{i}-k}{N-k}\cdot\prod_{k=0}^{\ell-1}\frac{\mathrm{i}-k}{N-j+\ell-k}}{\left(\frac{N-\mathrm{i}}{N}\right)^{j-\ell}\left(\frac{\mathrm{i}-1}{N}\right)^{\ell}} = \prod_{k=0}^{j-\ell-1}\left(\frac{N-\mathrm{i}-k}{N-\mathrm{i}}\cdot\frac{N}{N-k}\right)\cdot\prod_{k=0}^{\ell-1}\left(\frac{\mathrm{i}-k}{\mathrm{i}-1}\cdot\frac{N}{N-j+\ell-k}\right)$$
$$\leq \frac{\mathrm{i}}{\mathrm{i}-1}\cdot\left(\frac{N}{N-j}\right)^{j}$$
$$= \left(1+\frac{1}{\mathrm{i}-1}\right)\cdot\left(1+\frac{j}{N-j}\right)^{j}$$
$$\leq \left(1+\frac{1}{\sqrt{N}\log_2 N-1}\right)\cdot\left(1+\frac{j^2\mathrm{e}}{N-j}\right)$$
$$\leq 1+\frac{5\log N}{(1-p)\sqrt{N}}$$

And is also at least

$$\frac{(N-\mathrm{i}-j+\ell)^{j-\ell}(\mathrm{i}-\ell)^{\ell}\frac{1}{N^j}}{\left(\frac{N-\mathrm{i}+1}{N}\right)^{j-\ell}\left(\frac{\mathrm{i}}{N}\right)^{\ell}} = \frac{(N-\mathrm{i}-j+\ell)^{j-\ell}(\mathrm{i}-\ell)^{\ell}}{(N-\mathrm{i}+1)^{j-\ell}\mathrm{i}^{\ell}}$$
$$= \left(1-\frac{j-\ell+1}{N-\mathrm{i}+1}\right)^{j-\ell}\left(1-\frac{\ell}{\mathrm{i}}\right)^{\ell}$$
$$\geq 1-\frac{(j-\ell)(j-\ell+1)}{N-\mathrm{i}+1}-\frac{\ell^2}{\mathrm{i}}$$
$$\geq 1-3\frac{\log_2 N}{(1-p)\sqrt{N}}\,.$$

$\square$

Proof of Theorem 1.13. To prove the theorem we take a solution to one problem and transform it into a solution of the other. Let N, h be integers and $0 < p < 1$ a scalar such that $h = p\cdot N$. We start with an optimal solution $(q^*, \alpha(p))$ for SDCLP_p, and define for a given $N \geq 1$ a solution (x, α') as follows.

$$x_{\mathrm{i},\ell} = \int_{\frac{\mathrm{i}-1}{N}}^{\frac{\mathrm{i}}{N}} q^*(t,\ell)\mathrm{d}t, \quad \text{for } \mathrm{i}\in[N]\setminus[h], \ell\in[\mathrm{i}]$$
$$\alpha' = \min_{k\in[h+1]} \frac{\sum_{j=1}^{k}\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{j}\frac{\mathrm{i}x_{\mathrm{i},\ell}}{N}\frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}}{\sum_{j=1}^{k}\frac{N-h}{N-j+1}\prod_{s=0}^{j-2}\frac{h-s}{N-s}}$$

We prove first that (x, α') is a feasible solution for $\mathrm{SDLP}_{pN,N})$. Note that for given $\mathrm{i}\in[N]\setminus[h]$,

$\ell \in [\mathrm{i}]$, and $t \in [\frac{\mathrm{i}-1}{N}, \frac{\mathrm{i}}{N}]$ we have from the feasibility of q^* that

$$\begin{aligned} tq^*(t,\ell) + \int_{\frac{\mathrm{i}-1}{N}}^{t} q^*(\tau,\ell)\mathrm{d}\tau &\le 1 - \int_{p}^{\frac{\mathrm{i}-1}{N}} \sum_{s\ge 1} q^*(\tau,s)\mathrm{d}\tau \\ &\le 1 - \sum_{j=h+1}^{\mathrm{i}-1}\sum_{s=1}^{j} x_{j,s} \,. \end{aligned}$$

Integrating on both sides we obtain that

$$\begin{aligned} \int_{\frac{\mathrm{i}-1}{N}}^{\frac{\mathrm{i}}{N}} \left(tq^*(t,\ell) + \int_{\frac{\mathrm{i}-1}{N}}^{t} q^*(\tau,\ell)\mathrm{d}\tau \right) \mathrm{d}t &\le \int_{\frac{\mathrm{i}-1}{N}}^{\frac{\mathrm{i}}{N}} \left(1 - \sum_{j=h+1}^{\mathrm{i}-1}\sum_{s=1}^{j} x_{j,s}\right) \mathrm{d}t \\ \Leftrightarrow \qquad t\int_{\frac{\mathrm{i}-1}{N}}^{t} q^*(\tau,\ell)\mathrm{d}\tau \Bigg|_{t=\frac{\mathrm{i}-1}{N}}^{t=\frac{\mathrm{i}}{N}} &\le \frac{1}{N}\left(1 - \sum_{j=h+1}^{\mathrm{i}-1}\sum_{s=1}^{j} x_{j,s}\right) \\ \Leftrightarrow \qquad \mathrm{i}\cdot x_{\mathrm{i},\ell} &\le 1 - \sum_{j=h+1}^{\mathrm{i}-1}\sum_{s=1}^{j} x_{j,s} \,, \end{aligned}$$

where in the second inequality we applied integration by parts on the left-hand side. Therefore, x is a feasible solution. We now give an upper bound for $\alpha(p) - \alpha'$. From the definition of α' and Lemma 1.19, together with the fact that $1/(1+y) \ge 1-y$ for all $y \ge 0$, we obtain that

$$\alpha' \ge \min_{k\in[h+1]} \frac{\sum_{j=1}^{k}\sum_{\mathrm{i}=h+1}^{N}\sum_{\ell=1}^{j} \frac{\mathrm{i}x_{\mathrm{i},\ell}}{N} \frac{\binom{j-1}{\ell-1}\binom{N-j}{\mathrm{i}-\ell}}{\binom{N-1}{\mathrm{i}-1}}}{1-p^k} \cdot \left(1 - \frac{\mathrm{e}}{(1-p)(N-1)}\right).$$

Now, if $k > \frac{\log N}{1-p} \ge \log_p(1/N)$, then $p^k \le 1/N$, so we can take in the minimization $k \le \frac{\log N}{1-p}$ and lose a factor $(1-1/N)$. Denote $\mathrm{i}^* = \frac{\sqrt{N}\log N}{1-p}$. Since $j \le k$, after replacing $x_{\mathrm{i},\ell}$ with the integral that defines it, we can apply Lemma 1.20 to obtain that

$$\begin{aligned} \alpha' &\ge \min_{1\le k\le \frac{\log N}{1-p}} \frac{\sum_{j=1}^{k}\sum_{\mathrm{i}=(h+1)\vee \mathrm{i}^*}^{N-\mathrm{i}^*}\sum_{\ell=1}^{j}\int_{\frac{\mathrm{i}-1}{N}}^{\frac{\mathrm{i}}{N}} q^*(t,\ell)\binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}\mathrm{d}t}{1-p^k} \cdot \left(1 - \frac{7\log N}{(1-p)\sqrt{N}}\right) \\ &= \min_{1\le k\le \frac{\log N}{1-p}} \frac{\sum_{j=1}^{k}\int_{p\vee\frac{\mathrm{i}^*}{N}}^{1-\frac{\mathrm{i}^*}{N}}\sum_{\ell=1}^{j} q^*(t,\ell)\binom{j-1}{\ell-1}(1-t)^{j-\ell}t^{\ell}\mathrm{d}t}{1-p^k} \cdot \left(1 - \frac{7\log N}{(1-p)\sqrt{N}}\right). \end{aligned}$$

Now, since $t \cdot q^*(t,\ell) \le 1$ for all ℓ, t and $\sum_{\ell=1}^{j} \binom{j-1}{\ell-1}(1-t)^{j-\ell} t^{\ell-1} = 1$ for all $j \ge 1$, we get that

$$\begin{aligned}
\alpha' &\ge \min_{1 \le k \le \frac{\log N}{1-p}} \frac{\sum_{j=1}^{k} \int_p^1 \sum_{\ell=1}^{j} q^*(t,\ell) \binom{j-1}{\ell-1} (1-t)^{j-\ell} t^\ell \mathrm{d}t}{1-p^k} \cdot \left(1 - \frac{7\log N}{(1-p)\sqrt{N}} - \frac{2\mathrm{i}^* k}{N}\right) \\
&\ge \alpha(p) \cdot \left(1 - \frac{9(\log N)^2}{(1-p)^2\sqrt{N}}\right) \\
&\ge \alpha(p) - \frac{9(\log N)^2}{(1-p)^2\sqrt{N}}\,.
\end{aligned}$$

We prove now the other side of the inequality. Let $(x^*, \alpha_{N,p})$ be an optimal solution for $\mathrm{SDLP}_{pN,N}$. We construct a solution (q, α'') as follows.

$$q(t,\ell) = \begin{cases} N x^*_{\mathrm{i},\ell} \cdot \left(1 - \frac{\log N}{(1-p)\sqrt{N}}\right)\,, & \text{for } t \in [p,1], \text{ if } \mathrm{i} = \lceil t \cdot N \rceil \ge \sqrt{N} \text{ and } \ell \le \mathrm{i} \wedge \frac{\log N}{1-p} \\ 0 & \text{for } t \in [p,1], \text{ if } \mathrm{i} = \lceil t \cdot N \rceil < \sqrt{N} \text{ or } \ell > \mathrm{i} \wedge \frac{\log N}{1-p} \end{cases}$$

$$\alpha'' = \min_{k \ge 1} \frac{\sum_{j=1}^{k} \int_p^1 \sum_{\ell=1}^{j} q(t,\ell) \binom{j-1}{\ell-1} (1-t)^{j-\ell} t^\ell \mathrm{d}t}{1-p^k}\,.$$

Now, we can check this solution is feasible in the continuous problem. In fact, for $t < \frac{1}{\sqrt{N}}$, it is trivially satisfied because $q(t,\ell) = 0$ for all ℓ. For $t \ge p \vee \frac{1}{\sqrt{N}}$, $\mathrm{i} = \lceil t \cdot N \rceil$, and any $\ell \ge 1$,

$$\begin{aligned}
tq(t,\ell) + \int_p^t \sum_{s \ge 1} q(\tau, s) \mathrm{d}\tau &\le \left(\mathrm{i} x^*_{\mathrm{i},\ell} + \sum_{j=h+1}^{\mathrm{i}-1} \sum_{s=1}^{j} x^*_{j,s} + \int_{\frac{\mathrm{i}-1}{N}}^{t} \sum_{s=1}^{\frac{\log N}{1-p}} N x^*_{\mathrm{i},s} \mathrm{d}\tau \right) \cdot \left(1 - \frac{\log N}{(1-p)\sqrt{N}}\right) \\
&\le \left(1 + \frac{\log N}{\mathrm{i}(1-p)}\right) \cdot \left(1 - \frac{\log N}{(1-p)\sqrt{N}}\right) \\
&\le \left(1 + \frac{\log N}{(1-p)\sqrt{N}}\right) \cdot \left(1 - \frac{\log N}{(1-p)\sqrt{N}}\right) \\
&\le 1\,,
\end{aligned}$$

where the first inequality comes from replacing with the definition of q, and the third one comes from the fact that $\mathrm{i} \ge \sqrt{N}$.

We argue similarly to the lower bound for α', using lemmas 1.19 and 1.20, together with the extra factor $\left(1 - \frac{\log N}{(1-p)\sqrt{N}}\right)$ that was necessary for the feasibility constraint. This yields the inequality

$$\begin{aligned}
\alpha'' &\ge \alpha_{N,p} - \frac{5(\log N)^2}{(1-p)^2\sqrt{N}} - \frac{\log N}{(1-p)\sqrt{N}} \\
&\ge \alpha_{N,p} - \frac{6(\log N)^2}{(1-p)^2\sqrt{N}}\,.
\end{aligned}$$

□

Chapter 2

Selecting the Best with Samples

In the same spirit as in the previous chapter, we study optimal stopping with samples. We now consider the objective of maximizing the probability of selecting the best element, which is the objective in the classic formulation of the secretary problem.

Mathematically, in the secretary problem, we are faced with a randomly permuted sequence of n elements with arbitrary values. The elements' values are revealed one at a time. Upon receiving an element, we need to make an irrevocable decision of whether we keep the value and stop the sequence or drop the value forever and continue observing the next. The goal is to maximize the probability of stopping with the largest value. For this problem the best possible success guarantee has long been known to be 1/e. The optimal algorithm is remarkably simple: Look at the first n/e values without taking any of them, and then stop with the first value larger than all values seen so far [54, 99, 67]. In the last decades, the secretary problem, its variants, and related basic optimal stopping problems such as the prophet inequality and the Pandora's box problem have been considered fundamental building blocks of online selection problems [93, 94, 118, 48, 14].

An essential limitation of the secretary problem for modeling real-world situations is the assumption that the values of the elements that have not yet been revealed are completely unknown. This is a very pessimistic assumption, as in realistic situations one would expect to have some available information, coming, for instance, from the context or past data. As a consequence, the best possible 1/e success probability for the secretary problem can be substantially improved in many settings. This gives rise to the following natural question: what is a reasonable model to take into account this additional available information? A first approach is to assume that the numbers originate from a distribution that is known to the algorithm. This assumption is relevant when the process at hand has been repeated many times, and past data can be aggregated into a distribution. Along these lines, already in the sixties, [69] considered the so-called full information secretary problem in which we additionally know that the elements' values are i.i.d. random variables from a known distribution. For this variant, they showed how to compute the optimal stopping rule by dynamic programming and were able to conclude, numerically, that the best possible success probability is $\gamma \approx 0.5801$. In subsequent work, [113] finds an explicit expression for this quantity. [57] relaxed the i.i.d.-ness assumption, considering the problem when the elements' values are arbitrary independent random variables. They show that one can guarantee a success probability of 0.517, which, quite surprisingly, was very recently improved to γ by [107]. Interestingly, in this full information model with independent but not necessarily identical values, [1] showed that if the order is not random but adversarial, the optimal stopping rule guarantees a success probability of 1/e.[1]

While assuming no knowledge about the values seems too pessimistic, assuming that the full distribution is known might be too optimistic for most scenarios. Indeed, a typical situation would be that we have access to past data, but not enough to safely reconstruct a distribution. These informational issues in optimal stopping have given rise to a stream of research aiming at understanding the relationship between the amount of information available and the success probabilities that can be derived. In this context, [5] pioneered the study of data-driven versions of optimal stopping problems. Recently, [110] established a notable result in this direction for the classic prophet inequality.[2] They prove that a single sample from each distribution, rather than its full knowledge, is enough to achieve the optimal guarantee. Also, for the prophet secretary problem, the variant of the prophet inequality when the elements come in random order, one sample has been proved to be quite effective [37, 108].

However, this *sampling* approach still assumes that there is an underlying distribution from which we can effectively sample. In many situations this assumption may be strong, and ideally we would like to combine the idea of having samples representing past data with having arbitrary values chosen adversarially, to ensure maximum robustness while requiring no distributional assumption. Recently, [82] study such a model.[3] In their model, there are n arbitrary values, and they sample a fraction p of them at random. Then the non-sampled values are presented to the decision maker in either random order or adversarial order. [82] design algorithms for maximizing the expectation, rather than the probability of picking the maximum, that translate into algorithms for data-driven versions of prophet inequalities.

In this chapter, we consider an alternative sampling model, inspired by that of [21] and [82]. The main difference is that in our model, the sampling of each element is performed independently with the same fixed probability p. In other words, roughly a p-fraction of the elements are considered to be samples and revealed to the player upfront, before the non-sampled elements are revealed one by one, whose maximum value the player aims to obtain. Such data-driven versions are well-motivated from several perspectives. First, in many applications, the decision maker has access to historical data that give some insight into the distribution of future values. In our model, this information is captured in the form of samples that the decision maker knows a priori. Second, the model is robust in the sense that only minimal knowledge of the involved data is needed. And, third, the general idea is closely related to machine learning methods that use predictors to learn the distribution (see e.g. [71]). The insight here is that for problems that can be modeled as data-driven versions of the secretary problem, these learning procedures are overly complicated: The

[1] [57] also obtained this result.

[2] The classic prophet inequality asserts that when faced with a sequence of n independent random variables, $X_1, \ldots, X_n$, a decision maker who knows their distributions and is allowed to stop the sequence at any time, can obtain, in expectation, at least half the reward of a prophet who knows the values of each realization.

[3] Interestingly, the model was proposed much earlier by [21] and recently rediscovered.

simple combinatorial model presented in this paper already makes it possible to increase the solution quality even with modest sampling.

Of course, for large n our model is essentially equivalent to the model of [21] and [82]. However, our independent sampling has two crucial advantages. On the one hand, independence makes many mathematical calculations a lot simpler and thus allows to obtain simpler expressions. it allows dealing with instances of unknown size, which is often the case in practical applications. In particular, several of our results hold if we do not know n. A slight disadvantage of the independent sampling model is that we may end up sampling all n elements. For consistency in this case, we assume, by vacuity, that we *win* (i.e., pick the maximum). However, this is not very restrictive since, as we will see, the difficult instances involve large values of n for a fixed value of p.

We call our model the *random order secretary problem with p-sampling* (ROSp). In ROSp, we are given n elements with values $\alpha_1, \ldots, \alpha_n$, which are unknown to us, and a uniformly random order $\sigma : [n] \to [n]$ is drawn. Each element is sampled independently with probability p. Let S be the (random) set of sampled elements and V be the remaining elements, also referred to as the online set or the set of online elements. First, the set S of sampled elements are revealed to us. Then the elements in V are presented to us in the order dictated by σ. Once an element is revealed we either pick it and stop the sequence or drop it forever and continue. The goal is to maximize the probability of picking the maximum valued element in V. In particular, it is not allowed to pick an element of S, which is justified by the fact that we consider S to represent past data.

Given n and an algorithm we define its *success probability* as the infimum over all values $\alpha_1, \ldots, \alpha_n$ of the probability that the algorithm stops with the maximum $\alpha_i \in V$. Moreover, the *success guarantee* of an algorithm is the infimum over all values of n of its success probability.

All algorithms considered in this paper are *ordinal*, i.e., algorithms whose decision to stop at a given point depend only on the relative rankings of the values seen so far, and not on the actual values that have been observed, plus, possibly, on some external randomness. We observe that this is without loss of generality general algorithms cannot perform better than ordinal algorithms. Indeed, as noted by [82, Theorem 2.3], a result of [102] implies the existence of an infinite subset of the natural numbers where general algorithms behave like ordinal algorithms (for single selection ordinal objective functions such as ours). Therefore, and because the worst-case performance of our algorithms is attained as $n \to \infty$, our bounds apply to general algorithms; see also Theorem 2.2.

2.1 Summary of Results

For ROSp we obtain a randomized algorithm with best possible success guarantee that works as follows. First, we assign to each of the n elements a uniformly random arrival time in the interval $[0, 1]$, which implies that the elements arrive in uniform random order. All elements whose arrival time is less than p are placed in the sample set S. Then we find a sequence of time thresholds $0 < t_1 < t_2 < \cdots < 1$, dictating that if an element's arrival time is between t_i and t_{i+1}, the algorithm stops if its value is the maximum among elements arriving after p and

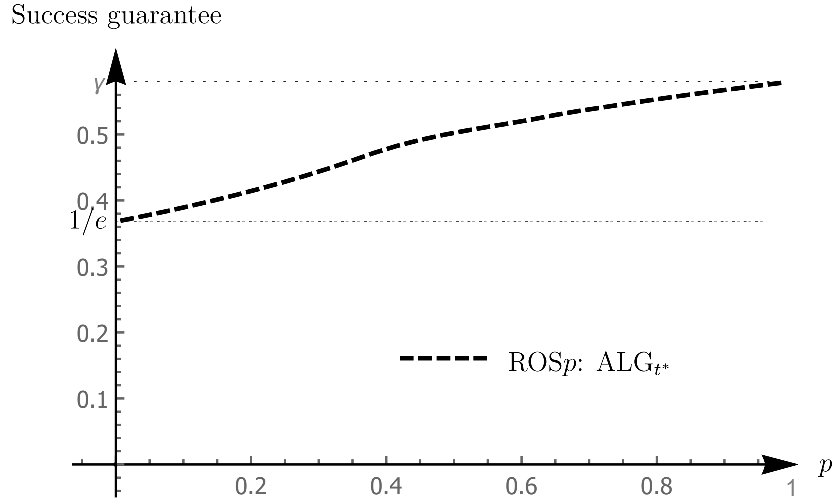


Figure 2.1: The best possible success guarantee for ROSp as a function of p.

it is among the i largest values of all elements seen so far. To obtain the success guarantee of this algorithm we first prove that for a fixed sequence $0 < t_1 < t_2 < \cdots < 1$, the success guarantee of the algorithm decreases with n. Then we write the optimization problem over the time thresholds, and interestingly, this turns out to be a separable concave optimization problem with a fairly simple solution. Moreover, the solution is universal in the sense that it does not depend on p. The resulting guarantee is thus easily computed and grows from 1/e when $p = 0$ to $\gamma \approx 0.58$ as $p \to 1$.[4] We also prove that this is a best possible algorithm. To this end we first argue that ordinal algorithms in our model are essentially equivalent to a ranking function that determines what *global* ranking an element, which is a *local* maximum, should have in order to accept it. Here, by global ranking we mean the ranking an element has among all samples and values revealed so far, and local ranking refers only to the values revealed and not to the samples. Finally, as n grows, this ranking function converges to a sequence of time thresholds as we defined them.

We demonstrate some of the strengths of our algorithm in practice, by evaluating it on the real-world data set of [70]. We show that our algorithm for ROSp can help explain some behavioral issues raised by [70]. [70] set up an experiment in which people play repeated secretary problems. The values come from any of three possible distributions, unknown to the players (the distribution is fixed for all games played by a person). They analyze a total of 48,336 games played by 6,537 players. Among other issues, [70] study how close to optimal people play. However, they find difficulty in establishing what optimal means in their context, since for the first game that players played optimal means simply the secretary algorithm, while after playing many games optimal should mean something close to the dynamic program of [69]. They thus consider several candidate models for the players' behavior and conclude that the closest to actual play is a multi-threshold algorithm that is very much in the spirit of that of [69]. Interestingly, they find that by the fifth game, players have essentially learned

[4]We should note that we recently became aware of the work of [21] who obtain very similar results. Indeed they consider the dependent sampling version described earlier and obtain that the optimal success guarantee converges to γ as the fraction sampled grows to 1. Their methods however are very different from ours and are significantly more complicated.

the optimal thresholds [70, Figure 9]. However, they also find an apparent dichotomy between the strategy players use in the first few games and that used later on. Indeed they state that: *"One possible explanation for the apparent change in strategy is that players spent the first few games primarily collecting information about the distribution and then switched to trying to actually win the game only in later games: that is, they spent the first few games exploring and then switched to exploiting only later."*

Our results for ROSp, being optimal in a closely related model, sheds more light on the players' strategies and strengthens existing insights in the work of [70]. To see this, observe that the first game the players face is just the normal secretary problem, or ROS0, the second closely corresponds to ROS$\frac{1}{2}$, the third to ROS$\frac{2}{3}$, and so on. With this observation we are able to directly compare the performance of the players' strategies with that of our algorithms. Our experimental evaluation leads to two main conclusions: (a) the players' strategies are strongly correlated with our algorithm (i.e., players use similar, suboptimal series of thresholds) and (b) a good fraction plays near-optimally from the start; the players in any case improve their performance in the first few games by learning how to optimally use the information they gain (as [70] also observe) and at some point their success rate stabilizes.

2.2 Further related literature

An interesting connection arises between our model and results when p is close to 1, and the so-called full information case. First, recall that [69] obtained the optimal algorithm with worst case performance γ (see also [112, 113]), in the secretary problem where the elements' values are taken as i.i.d. random variables from a known distribution. It may thus seem natural that our guarantee matches this quantity as $p \to 1$. However, this is far from obvious. Indeed, for the prophet inequality with i.i.d. values from an unknown distribution (a model that arguably gives more information than ours) [41] proved that with $O(n^2)$ samples, one can achieve the best possible performance guarantee of the case with known distribution, and only very recently [110] improved this to $O(n)$ samples. This is in line with our result here since for p close to, but strictly less than 1, the size of the sample set is linear in the size of V.

Still in the random order case, [36] study the same sampling model we discuss in this paper but with the objective of maximizing the expected value of the chosen element. They obtain best possible ordinal algorithms for all values of p and the implied guarantees grow from 1/e when $p = 0$ to 0.745 when p tends to 1, which is the optimal guarantee for the i.i.d. prophet inequality ([80, 44]), since in both cases the optimal guarantees converge to those of the full information case.

The line of research exploring the use of data to improve solutions to problems has gained momentum over the past years. Our sampling approach can be considered in this setting as a case of the secretary problem with advice based on past data. [52] study this problem from a more general perspective and use a factor revealing LP to gain structural insight into the optimal policy, depending on the type of advice the algorithm is given.

Another very recent line of work studies robust or semi-random versions of the classical secretary problem [85, 16]. The main idea is that the problem input should be a mix of stochastic and adversarial parts. More specifically, in their (similar) models, some of the elements arrive at adversarially chosen times, and the rest at times uniformly randomly drawn from $[0, 1]$. Their objective functions (and in some cases also the benchmarks) are quite different from ours. [85] consider the knapsack secretary problem in this mixed model, while [16] design algorithms for selecting k items or maximizing the expectation under various matroid or knapsack constraints.

2.3 Model and definitions

Let $p \in [0, 1]$. We consider the following game between a player and an adversary. The game takes place in several phases.

- Phase 1: The adversary chooses an integer n and a set U of n integers.
- Phase 2: U is ordered according to a uniformly random permutation σ.
- Phase 3: Every element of U is added to the sample set S independently with probability p, and otherwise it is added to the set $U \setminus S$ of online elements.
- Phase 4: The sample set is revealed to the player.
- Phase 5: The elements of the online set are presented to the player one after the other, according to the order σ. The player chooses at each step to continue or to stop the game. If the player continues, the next element is presented to her.

An instance of the game can be denoted as (U, σ, S), where U is the set of n values, σ is the permutation and S is the result of the sampling process. In particular, it is a sequence of values from U, presented in order σ, where the first $|S|$ values are considered to be sampled elements. We will identify an instance of the problem with such a sequence of values.

We say that a player *wins* or an algorithm *succeeds* in an instance of the problem if it stops on the largest element of the online set $U \setminus S$. This allows us to formally define the success guarantee of an algorithm.

Definition 2.1. The *success guarantee* of a deterministic algorithm A for ROS_p it is defined as

$$\inf_n \min_U \mathbb{P}_{S,\sigma}(A \text{ succeeds on } (U, \sigma, S)),$$

where $\mathbb{P}_{S,\sigma}$ takes the probability over the sampling phase as well as the permutation.

Observe that the success probability is unconditional on the sample set S.

Note that we use a minimum for U, σ even if there is potentially an infinite number of possible sets U. This is because of the following result. This theorem is the analogue of Theorem 2.3 in [82], and the proof is essentially the same. For completeness, and because this statement is central to the paper, we sketch the approach below.

Theorem 2.2. *For negative results, we can restrict the study to algorithms that are ordinal, that is, that do not use the values of the numbers, but only their relative ordering.*

Proof. Any algorithm ALG for $ROSp$ can be described as a series of n functions $f_t : \mathbb{N}^{2n} \mapsto \{0,1\}$. Indeed, we can encode both the sample set and revealed values of the online set as sets of at most n reals (completing with zeros for example), and the decision (stop or continue) as a bit. Then f_t represents the function used for the decision at the t-th step. Ramsey theory (in particular Corollary 3.4 in [102]) ensures that there exists an infinite set $\mathcal{V} \subseteq \mathbb{N}$ such that the functions f_t are all ordinal on $\mathcal{V}$. As a consequence, negative results on ordinal algorithms also apply to general algorithms. $\square$

We will see that our positive results are actually also ordinal, and that they match the negative bounds. In particular, once we restrict to ordinal algorithms, we can assume that the input sequence is a permutation of $\{1, \ldots, n\}$.

2.4 The Optimal Algorithm

In this section we derive the optimal strategy for ROSp, for any $p \in [0, 1)$. For the analysis it is useful to have the following equivalent setting.

Continuous time arrival model. We are given the values $\alpha_1, \ldots, \alpha_n$, and nature samples n uniformly random and independent arrival times $(\tau_i)_{i=1}^n$ in the interval $[0, 1]$. Now S contains all elements α_i such that $\tau_i < p$ and V contains all other elements. We get to observe all elements in S beforehand. Then, we observe one by one the elements in V in the order given by the τ_i's. We show equivalence between ROSp and the continuous time arrival model by showing that any algorithm A for the original setting can be applied to the continuous time model, obtaining the same success guarantee, and vice versa. Consider an algorithm for ROSp. It is easy to see that in the continuous time model each element is in S independently with probability p, and that the elements in V are revealed in uniformly random order, as in ROSp. Therefore, we can use the algorithm A and simply ignore the arrival times. Consider now an algorithm A' for the continuous time model. There are no arrival times in ROSp, but we can simulate them: we can sample $|S|$ uniform arrival times in the interval $[0, p]$ and assign them to the elements of S in an arbitrary way, and sample $|V| = n - |S|$ uniform arrival times in $[p, 1]$ and assign them to each observed element in V. Notice that since in ROSp each element is in S independently with probability p and the elements in V are revealed in uniformly random order, the simulated arrival times distribute exactly as n uniform arrival times in $[0, 1]$. Thus, if the algorithm A' requires observing the arrival times, we can simply pass it the simulated arrival times and we obtain a randomized algorithm for ROSp with the same success probability as if we were applying it to the continuous time model.

From now on, we consider the continuous time arrival model. Consider the family of algorithms ALG_t, described in Algorithm 1. The algorithm is parameterized by a sequence $t = (t_i)_{i \in \mathbb{N}}$ such that $0 \leq t_1 < t_2 < \cdots < 1$. The algorithm ALG_t accepts an element α_i if it is the largest element of V seen so far and it is larger than the k-th largest element in S, where k is such that $t_k \leq \tau_i < t_{k+1}$. In other words, for each k, between times t_k and t_{k+1} the algorithm sets as threshold the k-th largest element of S. We prove that the best possible success guarantee is attained in this family of decreasing threshold algorithms. Notice that even though these algorithms are designed for the continuous time model, because of the aforementioned equivalence they are also optimal in the original setting.

Algorithm 1 Time-threshold algorithm ALG_t for ROS_p.

```
for i = 1, . . . , |S| do
    s_i ← the i-th largest element in S.
end for
s_i ← −∞ if i > |S|.
for j = 1, . . . , |V| do
    σ(j) ← the index of the j-th observed element of V.
    if τ_σ(j) < t_1 then
        Discard the value
    else
        ℓ ← max{ℓ' : t_ℓ' ≤ τ_σ(j)}.
        if α_σ(j) > s_ℓ and α_σ(j) is the largest element of V seen so far then
            Accept the value and stop the game
        else
            Discard the value
        end if
    end if
end for
```

Theorem 2.3. *There exists a universal sequence t, independent of p and n, such that ALG_t obtains the best possible success guarantee for ROSp. Furthermore, when $p = 0$ this guarantee is equal to $1/e$, and when p tends to 1, the guarantee tends to $\gamma \approx 0.58$, the optimal success guarantee in the full-information secretary problem.*[5]

We prove this theorem in two main steps. First, we find the sequence t^* that maximizes the success guarantee of ALG_t. Then, we find an expression for the optimal success probability when p and n are given, and prove that for fixed p it converges to the success guarantee of ALG_{t^*} when n tends to infinity.

In order to find the optimal sequence t^* we start by studying the success probability of algorithm ALG_t, for any sequence t, sample rate p and instance size n. We prove that in fact the worst case for this class of algorithms is when n is very large. The approach of approximating the problem when n is large by a continuous time problem was pioneered by [17] and has been used for different optimal stopping problems (see e.g. [26, 81]).

Lemma 2.4. *For any sequence t and sampling probability p, the success probability of ALG_t in ROSp decreases with n.*

Proof. Fix a sequence t and a sampling probability p. We use a coupling argument between realizations of the arrival times in instances with n and $n+1$ values. We start with an instance $\alpha_1, \ldots, \alpha_{n+1}$, and assume the values are indexed in decreasing order. Consider a realization of the arrival times $\tau_1 = \tau'_1, \ldots, \tau_{n+1} = \tau'_{n+1}$ and couple it with the corresponding realization $\tau_1 = \tau'_1, \ldots, \tau_n = \tau'_n$ in the instance $\alpha_1, \ldots, \alpha_n$. Assume that in the instance with n values and

[5] The optimal guarantee $\gamma \approx 0.58$ was first obtained numerically by [69]. An explicit formula for γ was later found by [112, 113].

for this particular realization of the arrival times, ALG_t fails. This means that $V \setminus \{\alpha_{n+1}\}$ is non-empty and either ALG_t never stops or it accepts a value that is not the maximum of $V \setminus \{\alpha_{n+1}\}$. Note that regardless of τ'_{n+1}, the rankings of the values in $V \setminus \{\alpha_{n+1}\}$ are the same in both instances because α_{n+1} is smaller than all other values. Thus, if $\tau'_{n+1} < p$, ALG_t does not succeed either when applied in the instance of $n+1$ values. On the other hand, if $\tau'_{n+1} > p$, we have to distinguish between two cases. If ALG_t accepts α_{n+1}, it fails, because $V \setminus \{\alpha_{n+1}\}$ is non-empty and then α_{n+1} cannot be the largest in V. If ALG_t does not accept α_{n+1}, then the behavior of ALG_t in the rest of the variables is the same as in the instance with n values and then it fails.

Since the distribution of $\tau_1, \ldots, \tau_n$ is the same in both instances, we conclude with this argument that the probability that ALG_t fails in the instance with $n+1$ values is at least as large as in the instance with n values. $\square$

By Lemma 2.4 the success guarantee of ALG_t is simply the limit of its success probability when n grows to infinity. We calculate these probabilities and obtain an explicit formula for the limit in the following lemma. Interestingly, the formula turns out to be fairly simple.

Lemma 2.5. *Fix a sequence t and a sampling probability p. The success guarantee of ALG_t in ROSp is given by*

$$\sum_{\mathrm{i}=1}^{\infty} p^{\mathrm{i}-1}\left(1 - \max\{p, t_\mathrm{i}\} - \int_{\max\{p,t_\mathrm{i}\}}^{1} \sum_{j=1}^{\mathrm{i}} \frac{t - \max\{p, t_\mathrm{i}\}}{t^j}\, \mathrm{d}t\right). \tag{2.1}$$

Proof. We first calculate the success probability of ALG_t for fixed p and n and then take the limit when n tends to infinity.

We say a value α_i is *acceptable* for ALG_t (for a particular realization of the arrival times) if $p < \tau_\mathrm{i}$, for some $j \in \mathbb{N}$ we have that $t_j \le \tau_\mathrm{i} < t_{j+1}$, and α_i is larger than the j-th largest value in S. Now, note that if $\max V$ is not acceptable for ALG_t, then ALG_t does not stop. This is because we restricted the sequence t to be increasing, so values that arrive before $\max V$ are not acceptable, and values arriving after $\max V$ will not be the best seen so far from V. We use this to decompose the success probability as follows.

$$\mathbb{P}(ALG_t \text{ succeeds}) = \mathbb{P}(\max V \text{ is acceptable}) - \mathbb{P}(ALG_t \text{ stops before seeing } \max V)\,. \tag{2.2}$$

In this definition, if V is empty we also say $\max V$ is acceptable. We first calculate the probability that $\max V$ is acceptable. Assume that the values are indexed in decreasing order, i.e., that $\alpha_1 > \cdots > \alpha_n$.

$$\begin{aligned}
\mathbb{P}(\max V \text{ is acceptable}) &= \mathbb{P}(V = \emptyset) + \sum_{\mathrm{i}=1}^{n} \mathbb{P}(\max V = \alpha_\mathrm{i}) \cdot \mathbb{P}(t_\mathrm{i} \le \tau_\mathrm{i} \mid \max V = \alpha_\mathrm{i}) \\
&= p^n + \sum_{\mathrm{i}=1}^{n} p^{\mathrm{i}-1}(1-p) \cdot \frac{1 - \max\{p, t_\mathrm{i}\}}{1-p} \\
&= p^n + \sum_{\mathrm{i}=1}^{n} p^{\mathrm{i}-1}\left(1 - \max\{p, t_\mathrm{i}\}\right)\,. \qquad (2.3)
\end{aligned}$$

By the same argument, ALG_t stops before seeing $\max V$ if and only if at least one value arrives after p and before the arrival time of $\max V$, and the maximum such value is acceptable.

$$
\begin{aligned}
&\mathbb{P}(ALG_t \text{ stops before seeing } \max V) \\
&= \sum_{j=1}^{n} \mathbb{P}(\max V = \alpha_j) \cdot \mathbb{P}(\text{maximum before } \max V \text{ is acceptable} \mid \max V = \alpha_j) \\
&= \sum_{j=1}^{n} \mathbb{P}(\max V = \alpha_{\mathrm{i}}) \sum_{\mathrm{i}=j}^{n-1} \mathbb{P}\Big(\text{max. in } [p, \tau_j) \text{ has rank i among elements that arrive in } [0, \tau_j), \\
&\qquad \text{and arrives in } [t_{\mathrm{i}}, \tau_j) \;\Big|\; \max V = \alpha_j\Big) \\
&= \sum_{j=1}^{n} p^{j-1}(1-p) \sum_{\mathrm{i}=j}^{n-1} \frac{1}{1-p} \int_p^1 \mathbb{P}\Big(\text{max. in } [p, t) \text{ has rank i among elements that arrive in} \\
&\qquad [0, t), \text{ and arrives in } [t_{\mathrm{i}}, t) \;\Big|\; \max V = \alpha_j, \tau_j = t\Big) \, \mathrm{d}t \\
&= \sum_{j=1}^{n} p^{j-1}(1-p) \sum_{\mathrm{i}=j}^{n-1} \frac{1}{1-p} \int_{\max\{p, t_{\mathrm{i}}\}}^1 \left(\frac{p}{t}\right)^{\mathrm{i}-j} \cdot \frac{(t - \max\{p, t_{\mathrm{i}}\})}{t} \\
&\qquad \cdot \mathbb{P}(\text{at least i values arrive before } t \mid \max V = \alpha_j, \tau_j = t) \, \mathrm{d}t \\
&= \sum_{j=1}^{n} p^{j-1} \sum_{\mathrm{i}=j}^{n-1} \int_{\max\{p, t_{\mathrm{i}}\}}^1 \left(\frac{p}{t}\right)^{\mathrm{i}-j} \cdot \frac{(t - \max\{p, t_{\mathrm{i}}\})}{t} \Big(1 - B_{t, n-j}(\mathrm{i} - j + 1)\Big) \, \mathrm{d}t \\
&= \sum_{\mathrm{i}=1}^{n-1} p^{\mathrm{i}-1} \int_{\max\{p, t_{\mathrm{i}}\}}^1 \sum_{j=1}^{\mathrm{i}} \frac{t - \max\{p, t_{\mathrm{i}}\}}{t^j} \Big(1 - B_{t, n-j}(\mathrm{i} - j + 1)\Big) \, \mathrm{d}t \,, \qquad (2.4)
\end{aligned}
$$

where $B_{p,n}(x) = \sum_{\mathrm{i}=0}^{x} \binom{n}{\mathrm{i}} p^{\mathrm{i}} (1-p)^{n-\mathrm{i}}$ is the CDF of a Binomial distribution of parameters p and n. Note that for any fixed integers i and j, and time $t \in (0,1)$, $B_{t,n-j}(\mathrm{i} - j + 1)$ converges to 0 when n tends to infinty. Therefore, replacing Equation (2.3) and Equation (2.4) in Equation (2.2), and taking the limit when n tends to infinity, we conclude the proof of the lemma. □

We then focus our attention on optimizing this success guarantee. Surprisingly, it turns out the problem of maximizing Equation (2.1) is separable and concave, so we can simply impose the first-order conditions to obtain the optimum. Perhaps even more surprising is that these first-order conditions are independent of p, and therefore, the optimal sequence t^* is also independent of p, as the following lemma shows.

Lemma 2.6. *Fix a sampling probability p. The sequence t^* defined as the unique solution of the equations*

$$
\ln\left(\frac{1}{t^*_{\mathrm{i}}}\right) + \sum_{j=1}^{\mathrm{i}-1} \frac{(1/t^*_{\mathrm{i}})^j - 1}{j} = 1, \quad \textit{for all } \mathrm{i} \in \mathbb{N}\,, \qquad (2.5)
$$

maximizes Equation (2.1). In particular, t^* does not depend on p.

Proof. First, we relax the monotonicity constraint on the sequence of t_{i}'s. The resulting relaxed optimization problem is separable, i.e., optimizing over the entire sequence is equivalent to optimizing over each variable independently. For each t_{i} we get the following equivalent problem.

$$\max_{t_{\mathrm{i}}\in[0,1]} p^{\mathrm{i}-1}\left(1-\max\{p,t_{\mathrm{i}}\}-\int_{\max\{p,t_{\mathrm{i}}\}}^{1}\sum_{j=1}^{\mathrm{i}}\frac{t-\max\{p,t_{\mathrm{i}}\}}{t^j}\,\mathrm{d}t\right).$$

Equivalently, we can remove the factor $p^{\mathrm{i}-1}$ and restrict t_{i} to be in $[p,1]$, obtaining

$$\max_{t_{\mathrm{i}}\in[p,1]} 1-t_{\mathrm{i}}-\int_{t_{\mathrm{i}}}^{1}\sum_{j=1}^{\mathrm{i}}\frac{t-t_{\mathrm{i}}}{t^j}\,\mathrm{d}t\,.$$

Denoting by $G_{\mathrm{i}}(t_{\mathrm{i}})$ this objective function, we get that

$$\frac{\mathrm{d}}{\mathrm{d}t_{\mathrm{i}}}G_{\mathrm{i}}(t_{\mathrm{i}})=-1+\int_{t_{\mathrm{i}}}^{1}\sum_{j=1}^{\mathrm{i}}\frac{1}{t^j}\,\mathrm{d}t\,,\text{ and }\quad\frac{\mathrm{d}^2}{\mathrm{d}t_{\mathrm{i}}^2}G_{\mathrm{i}}(t_{\mathrm{i}})=-\sum_{j=1}^{\mathrm{i}}\frac{1}{t_{\mathrm{i}}^j}\,.$$

Therefore, $G_{\mathrm{i}}(t_{\mathrm{i}})$ is a concave function and then the optimum is $\max\{p,t^*_{\mathrm{i}}\}$, where t^*_{i} is the solution of $\frac{\mathrm{d}}{\mathrm{d}t_{\mathrm{i}}}G_{\mathrm{i}}(t_{\mathrm{i}})=0$. In the original objective function t_{i} appears always as $\max\{p,t_{\mathrm{i}}\}$ so there we can simply take t^*_{i} as the solution. Now we prove that t^*_{i} is actually increasing in i, so it is also the optimal solution before doing the relaxation. In fact, t^*_{i} satisfies

$$\int_{t^*_{\mathrm{i}}}^{1}\sum_{j=1}^{\mathrm{i}}\frac{1}{t^j}\,\mathrm{d}t=1\,.$$

Note that the left-hand side of this equation is decreasing in t^*_{i}, and is increasing in i. Thus, necesarily $t^*_{\mathrm{i}}\le t^*_{\mathrm{i}+1}$, for all $\mathrm{i}\ge 1$. We conclude that t^*_{i} satisfies Equation (2.5) by simply integrating on the left-hand side of the last equation. □

Now that we have the best algorithm in the family, we prove that its success guarantee is actually the best possible. To do this, we first characterize the algorithm that achieves the highest success probability for fixed sampling probability p and instance size n.

For a non-decreasing function $\ell:[n]\to[n]$, we define the *sequential-ℓ-max algorithm* in the following way.

Definition 2.7. Let $\ell:[n]\to[n]$. The *sequential-ℓ-max algorithm* accepts the i-th observed value (considering the values from S and the ones that have been revealed from V) if it is the largest seen so far from V and it is larger that the $\ell(\mathrm{i})$-th largest value from S.

We prove that the optimal algorithm is in this class.

Lemma 2.8. *Fix a sampling probability p and an instance size n. There is a function ℓ such that the sequential-ℓ-max algorithm obtains the best possible success probability for instances of size n of ROSp.*

Proof. We study the optimal ordinal policy obtained with backward induction, and prove that it is in fact a sequential-ℓ-max algorithm for certain ℓ. Recall that we can assume the optimal policy is ordinal, so this algorithm will be optimal not only among ordinal algorithms.

Denote by $X_{\mathrm{i}} = \alpha_{\pi(\mathrm{i})}$ the i-th value, in the order of increasing arrival times. Denote by $R(X_1, \ldots, X_j)$ the relative ranks of values $X_1, \ldots, X_j$. In what comes, we use the notation $R(X_1, \ldots, X_j) = x$ to condition on a particular realization x of the relative ranks. Let x be a realization of the ranks such that X_j is the maximum in V so far, i.e., $X_j = \max V \cap \{X_1, \cdots, X_j\}$, and has rank r among $X_1, \cdots, X_j$. Then,

$$\begin{aligned}
&\mathbb{P}\Big(X_j = \max V \;\Big|\; R(X_1, \ldots, X_j) = x\Big) \\
&= \mathbb{P}\Big(X_{j+1}, \ldots, X_n \text{ have overall rank at most } r+1 \;\Big|\; R(X_1, \ldots, X_j) = x\Big) \\
&= \mathbb{P}\Big(X_{j+1}, \ldots, X_n \text{ have overall rank at most } r+1\Big) \\
&= \prod_{s=0}^{r-1} \frac{j-s}{n-s}\,.
\end{aligned}$$

The optimal policy is to accept X_j if this probability is larger or equal than the probability of picking $\max V$ after rejecting X_j if from $j+1$ onwards we use the optimal policy, conditional on $R(X_1, \ldots, X_j) = x$.

Let now x' be a realization of $R(X_1, \ldots, X_{j+1})$ such that the relative rank of the best of V up to step $j+1$ is r. Suppose that conditional on $R(X_1, \ldots, X_{j+1}) = x'$, the probability of winning if we use the optimal strategy from $j+2$ onwards depends solely of n, $j+1$ and the relative rank r, for all possible ranks r. Denote this conditional probability by $W(n, j+1, r)$. We want to inductively prove that this is in fact true for all n, j and r. It is of course true in the last step, when $j+1 = n$, so we do induction on j. Let x'' be a realization of $R(X_1, \ldots, X_j)$ such that the relative rank of the best of V up to step j is r. We have that

$$\begin{aligned}
&\mathbb{P}\Big(\text{win after } j \;\Big|\; R(X_1, \ldots, X_j) = x''\Big) \\
&= \mathbb{P}\Big(X_{j+1} \text{ has relative rank } \geq r+1 \;\Big|\; R(X_1, \ldots, X_j) = x''\Big) \cdot W(n, j+1, r) \\
&\quad + \sum_{r'=1}^{r} \mathbb{P}\Big(X_{j+1} \text{ has relative rank } r' \;\Big|\; R(X_1, \ldots, X_j) = x''\Big) \\
&\qquad \cdot \max\left\{W(n, j+1, r'), \prod_{s=0}^{r'-1} \frac{j+1-s}{n-s}\right\}. \qquad (2.6)
\end{aligned}$$

But for all x,

$$\mathbb{P}\Big(X_{j+1} \text{ has relative rank } r' \;\Big|\; R(X_1, \ldots, X_j) = x\Big) = \frac{1}{j+1}\,.$$

This proves the inductive step. Therefore, $W(n, j, r)$ is well defined for all n, j and r, and the optimal policy accepts X_j that has relative rank r and is the maximum so far in V if and only if

$$\prod_{s=0}^{r-1} \frac{j-s}{n-s} \geq W(n, j, r)\,.$$

From Equation (2.6) it is easy to check that $W(n,j,r)$ is decreasing in j for fixed n, r and increasing in r for fixed n, j.[6] Therefore the optimal policy is the sequential-ℓ-max algorithm, for ℓ defined as

$$\ell(j) = \max\left\{ r : \prod_{s=0}^{r-1} \frac{j-s}{n-s} \geq W(n,j,r) \right\}.$$

This concludes the proof of the lemma. □

To conclude the optimality of ALG_{t^*} we show that the success probability of the best sequential-ℓ-max algorithm for each n converges to Equation (2.1) for some sequence t, when n grows to infinity. To this end we first calculate the success probability of a sequential-ℓ-max algorithm.

Lemma 2.9. *Fix n, p and a non-decreasing function ℓ. Consider an integer h such that $0 \leq h < n$, and define $\hat{\ell}(\mathrm{i}) = \min\{\ell(\mathrm{i}), h+1\}$ for all $\mathrm{i} \in [n]$. The success probability of the sequential-ℓ-max algorithm, conditional on $|S| = h$, is given by*

$$\begin{aligned} &\frac{1}{n-h}\left(1 - \prod_{j=0}^{\hat{\ell}(h+1)-1} \frac{h-j}{n-j}\right) \\ &+ \sum_{\mathrm{i}=h+1}^{n-1}\left(\sum_{r=h+1}^{\mathrm{i}} \frac{1}{n-\mathrm{i}}\left(\frac{1}{\mathrm{i}-h}\prod_{j=0}^{\hat{\ell}(r)-1}\frac{h-j}{\mathrm{i}-j} - \frac{1}{n-h}\prod_{j=0}^{\hat{\ell}(r)-1}\frac{h-j}{n-j}\right) - \frac{1}{n-h}\prod_{j=0}^{\hat{\ell}(\mathrm{i}+1)-1}\frac{h-j}{n-j}\right). \end{aligned} \tag{2.7}$$

Proof. We calculate first the probability of some events. For $\mathrm{i} \in \{h+1, \ldots, n\}$, denote by A_i the event that the i-th element is the largest of V and the algorithm never stops. Notice that A_i is equivalent to the event that the overall largest $\hat{\ell}(\mathrm{i})$ elements are in S, and the i-th element is the largest of V (for this equivalence it is necessary that ℓ is non-decreasing). Therefore, we have that

$$\mathbb{P}(A_\mathrm{i}) = \frac{1}{n-h} \prod_{j=0}^{\hat{\ell(\mathrm{i})}-1} \frac{h-j}{n-j}.$$

Note that this is 0 if $\hat{\ell}(\mathrm{i}) = h+1$. Now, for $h+1 \leq r \leq \mathrm{i} \leq n$, define $B_{r,\mathrm{i}}$ the event that the r-th element is the largest among positions $\{h+1, \ldots, \mathrm{i}\}$ and the algorithm does not stop before $\mathrm{i}+1$. This is equivalent to the event that the r-th element is the largest among positions $\{h+1, \ldots, \mathrm{i}\}$ and the largest $\hat{\ell}(r)$ elements among positions $\{1, \ldots, \mathrm{i}\}$ are in S. Thus,

$$\mathbb{P}(B_{r,\mathrm{i}}) = \frac{1}{\mathrm{i}-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{\mathrm{i}-j}.$$

[6] At an intuitive level it is also easy to be convinced of this: as time passes it is harder to win, and if only low values (with large rank) have appeared, it is easier to win in the future.

Now, note that $B_{r,\mathrm{i}} \setminus A_r$ is the event that the r-th element is the largest among positions $\{h+1, \ldots, \mathrm{i}\}$, but not of V, and the algorithm does not stop before $\mathrm{i}+1$. Note also that $A_r \subseteq B_{r,\mathrm{i}}$. Therefore, the probability that the algorithm does not stop before $\mathrm{i}+1$ and the maximum of V is among positions $\{\mathrm{i}+1, \ldots, n\}$ is

$$\sum_{r=h+1}^{\mathrm{i}} \mathbb{P}(B_{r,\mathrm{i}}) - \mathbb{P}(A_r) = \sum_{r=h+1}^{\mathrm{i}} \frac{1}{\mathrm{i}-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{\mathrm{i}-j} - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j}.$$

Conditional on this event, the probability that the number in the i+1-th position is the largest of V is $1/(n-\mathrm{i})$, because the relative order within positions $\{\mathrm{i}+1, \ldots, n\}$ is independent of this event. Thus, we obtained the probability that the $\mathrm{i}+1$-th element is the largest of V and the algorithm does not stop before $\mathrm{i}+1$. To obtain the probability of winning in step $\mathrm{i}+1$, we have to subtract the probability that the $\mathrm{i}+1$-th element is the largest of V, but the algorithm never stops, i.e., $\mathbb{P}(A_{\mathrm{i}+1})$. Therefore, the probability of winning at step $\mathrm{i}+1$ is

$$\frac{1}{n-\mathrm{i}} \sum_{r=h+1}^{\mathrm{i}} \left(\frac{1}{\mathrm{i}-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{\mathrm{i}-j} - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j} \right) - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(\mathrm{i}+1)-1} \frac{h-j}{n-j}.$$

The probability of winning at step $h+1$ is slightly different, because the algorithm never stops before it. In that case the probability of winning is

$$\frac{1}{n-h} \left(1 - \prod_{j=0}^{\hat{\ell}(h+1)-1} \frac{h-j}{n-j} \right).$$

Adding these expressions concludes the proof of the lemma. □

We then show that there is a limit for the optimal ℓ in a continuous space, and use a Riemann sum analysis to obtain Equation (2.1) in the limit, proving Theorem 2.3.

Lemma 2.10. *Fix a sampling probability p. For each $n \in \mathbb{N}$, choose $\ell_{p,n}$ so that the sequential-$\ell_{p,n}$-max algorithm achieves the best possible success probability for fixed p and n. There exists a sequence t such that the success probability of the sequential-$\ell_{p,n}$-max algorithm converges to Equation (2.1) when n grows to infinity.*

Proof. First we show that the function ℓ that maximizes Equation (2.7), in a certain sense converges to a function $\tilde{\ell} : (0,1) \to \mathbb{N}$. Then, we do a Riemann sum analysis to show that the success probability of the sequential-ℓ-max algorithm converges to an expression in terms of $\tilde{\ell}$, and then we show that this can be equivalently expressed as Equation (2.1) for some sequence t.

Except for terms that vanish when n tends to infinity, Equation (2.7) can be rewritten as

$$\sum_{r=h+1}^{n} \left(\sum_{\mathrm{i}=r}^{n} \frac{1}{n-\mathrm{i}} \left(\frac{1}{\mathrm{i}-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{\mathrm{i}-j} - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j} \right) - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j} \right). \quad (2.8)$$

To find the optimal $\ell(r)$ we simply maximize the following term as a function of s.

$$F_n(r,s) = \sum_{\mathrm{i}=r}^{n} \frac{1}{n-\mathrm{i}} \left(\frac{1}{\mathrm{i}-h} \prod_{j=0}^{s-1} \frac{h-j}{\mathrm{i}-j} - \frac{1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \right) - \frac{1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} .$$

Between s and $s+1$ the change is

$$\begin{aligned}
&F_n(r,s+1) - F_n(r,s) \\
&= \sum_{\mathrm{i}=r}^{n} \frac{1}{n-\mathrm{i}} \left(\frac{\frac{h-s}{\mathrm{i}-s} - 1}{\mathrm{i}-h} \prod_{j=0}^{s-1} \frac{h-j}{\mathrm{i}-j} - \frac{\frac{h-s}{n-s} - 1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \right) - \frac{\frac{h-s}{n-s} - 1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \\
&= \sum_{\mathrm{i}=r}^{n} \frac{1}{n-\mathrm{i}} \left(-\frac{1}{\mathrm{i}-s} \prod_{j=0}^{s-1} \frac{h-j}{\mathrm{i}-j} + \frac{1}{n-s} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \right) + \frac{1}{n-s} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \\
&= \beta(n,s,h) \left(\sum_{\mathrm{i}=r}^{n} \frac{1}{n-\mathrm{i}} \left(1 - \frac{n-s}{\mathrm{i}-s} \prod_{j=0}^{s-1} \frac{n-j}{\mathrm{i}-j} \right) + 1 \right) ,
\end{aligned}$$

where $\beta(n,s,h)$ is a positive term, so the sign of this difference is not affected by it. The other term is decreasing in s, so $F_n(r,s)$ is maximized when this differences changes sign. In other words, it is maximized in

$$\ell_n^*(\mathrm{i}) = \min \left\{ s \in [n] : \sum_{\mathrm{i}=r}^{n} \frac{1}{n-\mathrm{i}} \left(1 - \prod_{j=0}^{s} \frac{n-j}{\mathrm{i}-j} \right) + 1 \leq 0 \right\} .$$

Now, doing a Riemann sum analysis, we have that $\tilde{\ell}(\tau) = \lim_{n\to\infty} \ell_n^*(\lfloor \tau n \rfloor)$ satisfies

$$\tilde{\ell}(\tau) = \min \left\{ s \in \mathbb{N} : \int_{\tau}^{1} \left(\frac{1}{1-t} \left(1 - \frac{1}{t^{s+1}} \right) + 1 \right) \mathrm{d}t \leq 0 \right\} . \tag{2.9}$$

Thus, interpreting Equation (2.8) as a Riemann sum, and noting that $|S|/n$ converges to p almost surely, we have that the success guarantee of the optimal policy converges to

$$\int_{p}^{1} \int_{\tau}^{1} \frac{1}{1-t} \left(\frac{1}{t-p} \left(\frac{p}{t} \right)^{\tilde{\ell}(\tau)} - \frac{1}{1-p} p^{\tilde{\ell}(\tau)} \right) \mathrm{d}t - \frac{1}{1-p} p^{\tilde{\ell}(\tau)} \, \mathrm{d}\tau .$$

From Equation (2.9) it is clear that $\tilde{\ell}$ is non-decreasing, so we can define the sequence $t_{\mathrm{i}} = \inf \left\{ \tau \in [p,1] : \tilde{\ell}(\tau) \geq \mathrm{i} \right\}$ and rewrite the limiting success guarantee in terms of it. Thus, we obtain

$$\sum_{\mathrm{i}=0}^{\infty} \left(\int_{t_{\mathrm{i}}}^{t_{\mathrm{i}+1}} \int_{\tau}^{1} \frac{1}{1-t} \left(\frac{1}{t-p} \left(\frac{p}{t} \right)^{\mathrm{i}} - \frac{1}{1-p} p^{\mathrm{i}} \right) \mathrm{d}t \; \mathrm{d}\tau - \frac{t_{\mathrm{i}+1}^{\mathrm{i}} - t_{\mathrm{i}}^{\mathrm{i}}}{1-p} \right) .$$

If we rearrange the terms, turning the integral from t_{i} to $t_{\mathrm{i}+1}$ into the difference between the integral from t_i to 1 and the integral from $t_{\mathrm{i}+1}$ to 1, we obtain

$$\begin{aligned}
&\int_p^1 \int_\tau^1 \frac{1}{(t-p)(1-p)}\,\mathrm{d}t\,\mathrm{d}\tau - \frac{p}{1-p}\\
&+\sum_{\mathrm{i}=1}^{\infty}\left(\int_{t_\mathrm{i}}^1\int_\tau^1 \frac{1}{1-t}\left(\frac{\left(\frac{p}{t}\right)^{\mathrm{i}}-\left(\frac{p}{t}\right)^{\mathrm{i}-1}}{t-p}-\frac{p^{\mathrm{i}}-p^{\mathrm{i}-1}}{1-p}\right)\mathrm{d}t\,\mathrm{d}\tau+\frac{t_\mathrm{i}\left(p^{\mathrm{i}}-p^{\mathrm{i}-1}\right)}{1-p}\right)\\
=&\frac{1}{1-p}-\sum_{\mathrm{i}=1}^{\infty}p^{\mathrm{i}-1}\left(\int_{t_\mathrm{i}}^1\int_\tau^1\frac{1}{1-t}\left(\frac{t-p}{t^{\mathrm{i}}(t-p)}-\frac{1-p}{1-p}\right)\mathrm{d}t\,\mathrm{d}\tau+t_\mathrm{i}\frac{1-p}{1-p}\right)\\
=&\frac{1}{1-p}-\sum_{\mathrm{i}=1}^{\infty}p^{\mathrm{i}-1}\left(\int_{t_\mathrm{i}}^1\int_\tau^1\frac{1}{t^{\mathrm{i}}(1-t)}\left(1-t^{\mathrm{i}}\right)\mathrm{d}t\,\mathrm{d}\tau+t_\mathrm{i}\right)\\
=&\frac{1}{1-p}-\sum_{\mathrm{i}=1}^{\infty}p^{\mathrm{i}-1}\left(\int_{t_\mathrm{i}}^1\int_\tau^1\sum_{j=0}^{\mathrm{i}-1}\frac{t^j}{t^{\mathrm{i}}}\,\mathrm{d}t\,\mathrm{d}\tau+t_\mathrm{i}\right)\\
=&\sum_{\mathrm{i}=1}^{\infty}p^{\mathrm{i}-1}\left(1-t_\mathrm{i}-\int_{t_\mathrm{i}}^1\int_\tau^1\sum_{j=1}^{\mathrm{i}}\frac{1}{t^j}\,\mathrm{d}t\,\mathrm{d}\tau\right)\\
=&\sum_{\mathrm{i}=1}^{\infty}p^{\mathrm{i}-1}\left(1-t_\mathrm{i}-\int_{t_\mathrm{i}}^1\sum_{j=1}^{\mathrm{i}}\frac{t-t_\mathrm{i}}{t^j}\,\mathrm{d}t\right).
\end{aligned}$$

This concludes the proof, since we defined the t_i's in a way that they satisfy $t_\mathrm{i}=\max\{p,t_\mathrm{i}\}$. □

Finally, we study the success guarantee of ALG_{t^*} in the border values of p, and show that it actually becomes equal to the best possible among all algorithms. It is easy to see that the success guarantee is $1/\mathrm{e}$ when $p=0$. Note that when $p=0$, Equation (2.1) simplifies to $t_1\ln(1/t_1)$, and that Equation (2.5) yields $t_1^*=1/\mathrm{e}$. Substitution gives the success guarantee of $1/\mathrm{e}$. The case when p tends to 1 is a bit more involved and requires some tedious calculations. We evaluate Equation (2.1) with the first order approximation $t_\mathrm{i}^*\approx t_\mathrm{i}':=1-c/\mathrm{i}$, where c is a constant. To fix c we impose that (t_i') satisfies Equation (2.5) in the limit when $\mathrm{i}\to\infty$. More precisely, we take c such that

$$\begin{aligned}
1&=\lim_{\mathrm{i}\to\infty}\ln\left(\frac{1}{1-c/\mathrm{i}}\right)+\sum_{j=1}^{\mathrm{i}-1}\frac{(1-c/\mathrm{i})^{-j}-1}{j}\\
&=\int_0^1\frac{\mathrm{e}^{cx}-1}{x}\,\mathrm{d}x\,.
\end{aligned}$$

With this in hand we use a Riemann sum analysis to show the next lemma, which states that when p tends to 1, this approximation converges to the explicit expression of [112, 113] for γ.

Lemma 2.11. *Let $t_\mathrm{i}'=1-c/\mathrm{i}$, where c is the solution of $\int_0^1\frac{\mathrm{e}^{cx}-1}{x}\,\mathrm{d}x=1$. When evaluated in t', Equation (2.1) tends to*

$$\gamma=\mathrm{e}^{-c}+(\mathrm{e}^{-c}-1-c)\int_1^\infty x^{-1}\mathrm{e}^{-cx}\,\mathrm{d}x\approx 0.5801\,, \tag{2.10}$$

when p tends to 1.

PROOF. We analyze separately the sum when $p = \max\{p, t'_{\mathrm{i}}\}$ and when $t'_{\mathrm{i}} = \{p, t'_{\mathrm{i}}\}$. We call the first part V_1, which includes the terms up to $\mathrm{i} = \lfloor \frac{c}{1-p} \rfloor$, and V_2 the rest.

$$
\begin{aligned}
V_1 &= \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\lfloor \frac{c}{1-p} \rfloor} p^{\mathrm{i}-1} \left(1 - p - \int_p^1 \sum_{j=1}^{\mathrm{i}} \frac{t-p}{t^j} \, \mathrm{d}t \right) \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\lfloor \frac{c}{1-p} \rfloor} p^{\mathrm{i}-1} \left(1 - p - \int_p^1 \mathrm{d}t + \int_p^1 \frac{\mathrm{d}t}{t^{\mathrm{i}}} - \int_p^1 \sum_{j=1}^{\mathrm{i}} \frac{1-p}{t^j} \, \mathrm{d}t \right) \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\lfloor \frac{c}{1-p} \rfloor} p^{\mathrm{i}-1} \left(\frac{p^{-(\mathrm{i}-1)} - 1}{\mathrm{i}-1} - (1-p)\ln(1/p) - (1-p) \sum_{j=2}^{\mathrm{i}} \frac{p^{-(j-1)} - 1}{j-1} \right) \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\lfloor \frac{c}{1-p} \rfloor} \frac{1 - p^{\mathrm{i}-1}}{\mathrm{i}-1} - \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\lfloor \frac{c}{1-p} \rfloor} (p^{\mathrm{i}-1} - p^{\mathrm{i}}) \sum_{j=2}^{\mathrm{i}} \frac{\mathrm{e}^{-(j-1)\ln p} - 1}{j-1} \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\lfloor \frac{c}{1-p} \rfloor} \frac{1 - (p^{\frac{1}{1-p}})^{(\mathrm{i}-1)(1-p)}}{(\mathrm{i}-1)(1-p)} (1-p) - \lim_{p\to 1} \sum_{\mathrm{i}=1}^{\lfloor \frac{c}{1-p} \rfloor} (p^{\mathrm{i}-1} - p^{\mathrm{i}}) \sum_{j=2}^{\mathrm{i}} \frac{\mathrm{e}^{-\frac{(j-1)}{\mathrm{i}} \mathrm{i} \ln p} - 1}{(j-1)/\mathrm{i}} \cdot \frac{1}{\mathrm{i}}
\end{aligned}
$$

Interpreting these two sums as Riemann sums, we obtain

$$
\begin{aligned}
V_1 &= \int_0^c \frac{1 - \mathrm{e}^{-x}}{x} \, \mathrm{d}x - \int_{\mathrm{e}^{-c}}^1 \int_0^1 \frac{\mathrm{e}^{-x\ln y} - 1}{x} \, \mathrm{d}x \, \mathrm{d}y \\
&= \int_0^c \frac{1 - \mathrm{e}^{-x}}{x} \, \mathrm{d}x - \int_{\mathrm{e}^{-c}}^1 \int_0^1 \frac{\mathrm{e}^{-x\ln y} - 1}{-x\ln y} (-\ln y) \, \mathrm{d}x \, \mathrm{d}y \\
&= \int_0^c \frac{1 - \mathrm{e}^{-x}}{x} \, \mathrm{d}x - \int_{\mathrm{e}^{-c}}^1 \int_0^{-\ln y} \frac{\mathrm{e}^{x} - 1}{x} \, \mathrm{d}x \, \mathrm{d}y \\
&= \int_0^c \frac{1 - \mathrm{e}^{-x}}{x} \, \mathrm{d}x - \int_0^c \int_{\mathrm{e}^{-c}}^{\mathrm{e}^{-x}} \frac{\mathrm{e}^{x} - 1}{x} \, \mathrm{d}y \, \mathrm{d}x \\
&= \int_0^c \frac{1 - \mathrm{e}^{-x} - (\mathrm{e}^{-x} - \mathrm{e}^{-c})(\mathrm{e}^{x} - 1)}{x} \, \mathrm{d}x \\
&= \mathrm{e}^{-c} \int_0^c \frac{\mathrm{e}^{x} - 1}{x} \, \mathrm{d}x \\
&= \mathrm{e}^{-c} \int_0^1 \frac{\mathrm{e}^{cx} - 1}{x} \, \mathrm{d}x \\
&= \mathrm{e}^{-c},
\end{aligned}
$$

where the last step comes from the definition of c. On the other hand, we have that

$$
\begin{aligned}
V_2 &= \lim_{p\to 1} \sum_{\mathrm{i}=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} p^{\mathrm{i}-1} \left(\frac{c}{\mathrm{i}} - \int_{1-\frac{c}{\mathrm{i}}}^{1} \sum_{j=1}^{\mathrm{i}} \frac{t-1+c/\mathrm{i}}{t^j} \,\mathrm{d}t \right) \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} p^{\mathrm{i}-1} \left(\frac{c}{\mathrm{i}} - \int_{1-c/\mathrm{i}}^{1} \mathrm{d}t + \int_{1-c/\mathrm{i}}^{1} \frac{1}{t^{\mathrm{i}}} \,\mathrm{d}t - \int_{1-c/\mathrm{i}}^{1} \sum_{j=1}^{\mathrm{i}} \frac{c/\mathrm{i}}{t^j} \,\mathrm{d}t \right) \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} p^{\mathrm{i}-1} \left(\frac{(1-c/\mathrm{i})^{-(\mathrm{i}-1)}-1}{\mathrm{i}-1} + \frac{c}{\mathrm{i}} \ln(1-c/\mathrm{i}) - \sum_{j=2}^{\mathrm{i}} c \frac{(1-c/\mathrm{i})^{-(j-1)}-1}{\mathrm{i}(j-1)} \right) \\
&= \lim_{p\to 1} \sum_{\mathrm{i}=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} (p^{\mathrm{i}-1} - p^{\mathrm{i}}) \frac{(1-c/\mathrm{i})^{-(\mathrm{i}-1)}-1}{\frac{1-p}{-\ln p}(\mathrm{i}-1)(-\ln p)} \\
&\quad - \lim_{p\to 1} \sum_{\mathrm{i}=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} \frac{p^{\mathrm{i}-1} - p^{\mathrm{i}}}{\frac{1-p}{-\ln p}\mathrm{i}(-\ln p)} \sum_{j=2}^{\mathrm{i}} \frac{c\left((1-c/\mathrm{i})^{-\mathrm{i}\frac{j-1}{\mathrm{i}}} - 1 \right)}{j/\mathrm{i}} \cdot \frac{1}{\mathrm{i}} \,,
\end{aligned}
$$

where in the last equality we omitted a term that vanishes when p tends to 1. We again interpret the sums as Riemann sums.

$$
\begin{aligned}
V_2 &= \int_0^{\mathrm{e}^{-c}} \frac{\mathrm{e}^c - 1}{\ln(1/x)} \,\mathrm{d}x - c \int_0^{\mathrm{e}^{-c}} \frac{1}{\ln(1/x)} \int_0^1 \frac{\mathrm{e}^{cy}-1}{y} \,\mathrm{d}y \,\mathrm{d}x \\
&= (\mathrm{e}^c - 1 - c) \int_0^{\mathrm{e}^{-c}} \frac{1}{\ln(1/x)} \,\mathrm{d}x \\
&= (\mathrm{e}^{-c} - 1 - c) \int_1^{\infty} x^{-1} \mathrm{e}^{-cx} \,\mathrm{d}x \,.
\end{aligned}
$$

In the second equality we used the definition of c and in the third one we performed a change of variables. Summing V_1 and V_2 we get Equation (2.10). □

2.5 Computation of the time thresholds

In this section we discuss how to compute the optimal time thresholds t^*. Notice that by Lemma 2.6, t^* does not depend on p, nor in n, and therefore it is enough to compute them once. However, since t^* is an infinite sequence, a reasonable question is how well we can do if we compute only finitely many of these thresholds.

Denote by $\gamma(p)$ the optimal success guarantee for a given $p \in [0, 1)$. To achieve a success guarantee of $\gamma(p) - \varepsilon$ for a given $\varepsilon > 0$, it is sufficient to compute $O\left(\frac{1}{\varepsilon\cdot(1-p)}\right)$ many thresholds within an $O(\varepsilon^2(1-p)^2)$ margin of error each. The reason for this is that our algorithm can fail (compared to ALG_{t^*}) if the best element of the interval $[p, 1]$ falls too close to the thresholds (closer than the margin of error), or after the last threshold we computed, which by the first-order approximation is $1 - O(\varepsilon(1-p))$. But the best element of the interval $[p, 1]$ falls in a set of measure $\varepsilon(1-p)$ with probability ε and therefore we can ignore this event while

Table 2.1: Approximation of the first 10 optimal time thresholds within an error of 10^{-7}.

$t_1^* \approx 0.3678794$	$t_6^* \approx 0.8709762$
$t_2^* \approx 0.6422006$	$t_7^* \approx 0.8887973$
$t_3^* \approx 0.7518116$	$t_8^* \approx 0.9022956$
$t_4^* \approx 0.8101810$	$t_9^* \approx 0.9128731$
$t_5^* \approx 0.8463645$	$t_{10}^* \approx 0.9213851$

only losing ε in the success probability. Since the thresholds satisfy Equation (2.5), and the left-hand side is monotone, we can compute them using binary search in the interval $[0,1]$. If we assume that the left-hand side of Equation (2.5) can be computed in $O(\mathrm{i})$ time for a given i (because it has i terms), we obtain the following.

Lemma 2.12. *For any given $p \in [0,1]$, instance size n, and $\varepsilon > 0$, we can compute thresholds $\tilde{t}$ that approximate t^* and such that $ALG_{\tilde{t}}$ has a success probability of at least $\gamma(p) - \varepsilon$ in time $O\left(\frac{1}{\varepsilon^2(1-p)^2}\log\left(\frac{1}{\varepsilon(1-p)}\right)\right)$.*

Table 2.1 provides the first ten optimal time thresholds.

2.6 Numerical experiments

In this section, we implement our algorithms and evaluate them on the data set of [70]. In their paper, they design a large-scale online experiment in which people repeatedly play a secretary problem. The values each player faces are drawn i.i.d. from a distribution unknown to them and their total number is the same in every game. The main goal is to study experimentally the evolution of the players' stopping behavior. In particular, the main research question posed is whether the players progressively learn a near-optimal stopping strategy as they gain more experience. [70] use a Bayesian comparison framework to model the players' behavior and conclude that the estimated thresholds are indeed very close to the ones of [69] (i.e., the optimal ones for i.i.d. values from a known distribution) after only a few games.

We start by observing that our independent sampling model can be applied to their repeated secretary problem in a straightforward way; the first game is precisely the classic secretary problem or ROS0. Now, note that the second game closely corresponds to ROS$\frac{1}{2}$; we can imagine the values of the first game as our samples and the values of the second game as the online values. All values are i.i.d. and the two sets have equal sizes. Our independent sampling model with $p = \frac{1}{2}$ and the values of both games as the n input values $\alpha_1, \alpha_2, \cdots, \alpha_n$, would also result in splitting them into two sets of roughly equal size. Applying the same reasoning, the third game closely corresponds to ROS$\frac{2}{3}$, and so on; the i-th game corresponds to ROS$\frac{\mathrm{i}-1}{\mathrm{i}}$. In general, our model and this repeated secretary problem would be equivalent in the limit as $n \to \infty$, but for small values of n, we essentially ignore the variance of the independent sampling process. Since our algorithms are guaranteed to be optimal in a very similar model, they can serve as a meaningful benchmark for studying the players' strategies across games. By doing so, we hope to provide new insights and strengthen the existing ones regarding the players' behavior. In particular, and as mentioned in Section 2.2, our new comparison can help explain up to which extent players optimally use the information they have at each game and, as a result, resolve some behavioral issues raised by [70]. Since the values are i.i.d. (thus, closer to random order than adversarial) and players could choose the order of inspecting the elements (of course without knowing their values or the distribution over them), ALG_t is the natural candidate to use as the main benchmark, especially since its guarantee converges to the algorithm of [69].

First, we describe in more detail the behavioral experiment, the methodology, and the results obtained in [70]. The participants of the online experiment were directed to a simple interface, in which they were presented with a number of boxes containing money (i.e., hidden values). They could open the boxes in any order they wanted and decide when to stop. The interface would not let them terminate the game if the value with which they stopped was not a local maximum. After the end of each game, they were told if they won, how far away they were from the maximum value, and they would observe all the values of that game. They were also incentivized to play at least six games. The values for each player were drawn independently from an identical unknown distribution across all games. Each player was randomly assigned to one of three candidate distributions with the same support but with very different density functions; the first had high negative skew, the second was the uniform distribution, and the third had high positive skew. Each player was also randomly assigned to play games of 7 boxes or 15 boxes (recall that the game becomes harder as the number of boxes increases). In the end, 6,637 people participated and were distributed across the aforementioned random choices. They played a total of 48,836 games, for an average of 7.39 games per player.

The data set of the experiments contains the following information for each player: in which distribution and to which number of boxes they were assigned, how many games they played, the number of each game (if it was the first game, the second, and so on), the value and the rank of the box with which they stopped at each game, and the values of all *opened* boxes. What is not included in the data set of [70], although the players observed them at the end of each game, are the values of the remaining closed boxes. Nevertheless, this is not a major issue; since we know the distribution from which the values were drawn, we can complement each instance with random values that would be very similar to the realized ones.

Their data shows that the players rapidly improve their performance in early games, thus exhibiting substantial learning. The effects of learning are not as strong in later games, and the players eventually converge to a probability of success that is five to ten percentage points below the theoretical optima (that is, the guarantee given by the optimal algorithm of [69] with perfect knowledge of the distribution). The authors focus on modeling the players' strategies and attempt to fit several models they define to the data. In the first game, they observe that the thresholds belong to the class of what they call "value oblivious" strategies, which also contains the solution to the classical secretary problem. After the first game, the authors conclude that the players play according to a multi-threshold strategy. Therefore, there is a switch in strategies from the first to the second game. Note that the optimal algorithm of [69] and our optimal algorithm for ROSp also compute a series of (decreasing) thresholds. To investigate at which rate players learn to play optimally, [70] compute the players' estimated thresholds for the seven box games; they show that these converge quickly to the optimal ones of [69] and are very close to them already in the fourth game. Finally, they give some potential explanations for the strategy switch from the first game to the second (see also Section 2.2). One important takeaway from that work is that for such fundamental optimal stopping problems, their results indicate that *"the optimal procedure is likely to give a close approximation of human behavior. This is in contrast to many other areas of economics."*. Our experimental results, which we present next, will provide strong further support to this insight.

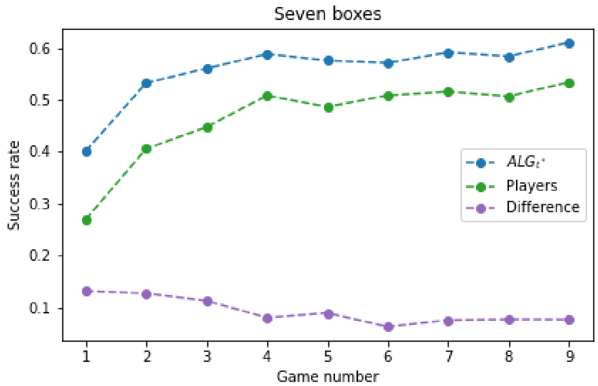
2.6.1 Experimental setup

For our experiments, the first thing to do is to complete the missing data of the real-world data set. Since for each game the rank and the value of the box the player stopped with, the value distribution, and the values of the opened boxes are known, we generate the values of the unopened boxes using rejection sampling. This is a necessary step for ALG_t since it uses the whole sample set to fix the thresholds. We generate 100 such complete instances by independently generating the missing data 100 times and take the average success rate of ALG_t (i.e., the fraction of instances in which the algorithm stopped with the maximum value) for each game as our final result. For a fair comparison with the work of [70], we also perform no cleaning on the data by excluding, e.g., players who did not put any effort into winning. After the end of each game, we add the values of the boxes to our sample set S. Thus, we know that e.g., in the third game the samples are twice as many as the online values. In this case, as we explained before, we run our algorithm with $p = \frac{2}{3}$. All our results are plotted for the first 9 games. We made this choice so that, on the one hand, we are consistent with some of the important figures of [70] (e.g., Figures 8 and 9) and, on the other hand, because less than a quarter of the players played more than nine games.

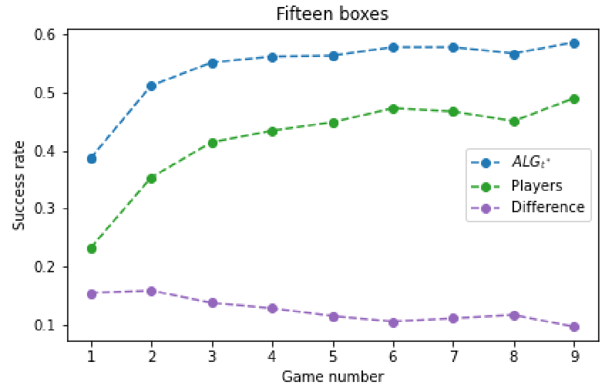
2.6.2 Experimental results

We present in Figure 2.2 the results of evaluating ALG_t on the data of [70] complemented with the generated values of the unopened boxes. To calculate the series of decreasing thresholds of ALG_t we have to solve Equation (2.5); it is faster to search for the solution for each threshold using binary search and stop when we are ε-close to the solution of the equation (here, we set $\varepsilon = 10^{-7}$).

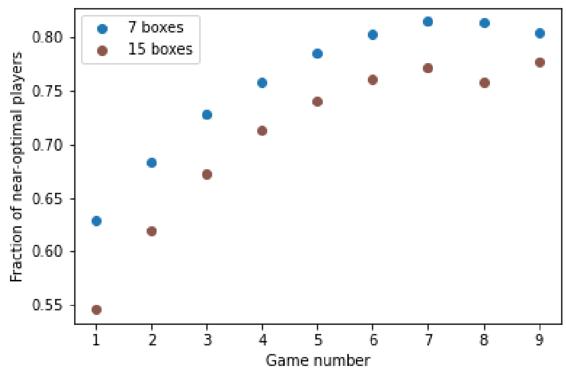
As we observe from Figures 2.2a and 2.2b, the evolution of the success rates of the players and ALG_t exhibit a very strong correlation, both for seven and fifteen boxes. This suggests that the players play according to some strategy that is similar to ALG_t all along (i.e., a series of decreasing thresholds), but (slightly) suboptimally. Moreover, the difference in the success rates is similar across games, and in particular, it slightly decreases in the first few games, and from game 4 onward it remains stable. These results strengthen the belief that for this type of simple online selection problems the optimal algorithm provides good insights into how players behave. Figure 2.2c provides additional supporting evidence to the claim that the players quickly learn how to play strategies that are close to the optimal. The figure shows the percentage of players that played close to optimal in the sense that the ranking of the chosen value of a near-optimal player and that of ALG_t differ by at most one. We observe that there is a strong learning effect in the first approximately four games. Starting at game 6 it remains relatively stable, and it even decreases slightly in the subsequent games.



(a) The success rate of the players and ALG_t for the first nine games and their difference for seven boxes.



(b) The success rate of the players and ALG_t for the first nine games and their difference for fifteen boxes.



(c) Percentage of players who picked a value with ranking at most one away from the value ALG_t picks.

Figure 2.2: Comparison of the players' behavior with the optimal algorithm for ROSp for seven and fifteen boxes.

Nonetheless, around 80% of the players (and a bit less for the harder case of 15 boxes) learn how to play close to optimal after only a few games.

Chapter 3

Fairness in Online Selection: The Multi-Color Secretary Problem

The sharp growth in data availability that characterizes modern society challenges our processing capabilities, not only because of its massiveness, but also because of the increasing strict social norms that society seeks in the algorithms processing it. For instance, machine learning algorithms are now used to make credit and lending decisions, to estimate the success of a kidney transplant, to inform hiring decisions, to recommend schools to pupils, among others. Therefore there is a founded concern over the use of algorithms that may violate social norms. Two basic such norms, that are receiving significant attention are fairness and privacy, and while a formalization of the latter is relatively well established through the notion of differential privacy [53], the former is much more unexplored from an algorithmic perspective [83].

In this chapter, we are particularly interested in the study of fairness in online selection. Not only has the area seen many recent theoretical developments, but also it naturally encompasses many real-world decision-making processes where biased evaluations should be avoided, such as those mentioned earlier.

Specifically, we consider the basic single-item selection model given by the secretary problem, in which items are classified into different groups. In our secretary model, two candidates from different groups are incomparable. A precursor of the study of fairness in the secretary problem is the work of [19], who studied, among other things, an incentive-compatible version of the secretary problem in which the selection probability does not depend on the arrival position of a candidate. More recently, [75] have studied machine learning algorithms for biased versions of the secretary problem, whereas [22] studies similar issues from the perspective of online learning. The term fairness has been used for various concepts in the machine learning community. We adopt here the common notion used in various previous works [78, 23, 24, 25, 31, 30], where we ask that the solution obtained is balanced with respect to some sensitive attribute (e.g., race, gender).

3.1 Summary of Results

We propose a fundamental problem in fair online selection, concerned with selecting a single candidate. Candidates are partitioned into different groups or *colors*. The candidates arrive sequentially and upon arrival of a candidate we have to irrevocably decide whether we want to select the candidate or not. In this setting, which we call *multi-color secretary problem*, candidates arrive in uniform random order and we can rank candidates within a group, but we cannot compare candidates across groups. There is also a prior probability that the best candidate from a group is the best candidate overall. The problem models situations in which different qualities of the candidates make them largely incomparable (this could arise in some form due to gender, race, social origin, type of education, etc.). The goal is to maximize the probability with which we stop at the best overall candidate and compare it with that for the *offline* optimum. Note that here the offline optimum simply picks the best candidate from the group of largest prior probability. Thus, it is extremely unfair. One may think that the best possible online algorithm is to mimic the offline optimum; namely, to select the group of largest prior probability and then run the classic secretary algorithm on that group. We prove that this is not the case, and indeed our main result is to obtain the best possible online algorithm for the problem and to establish that it satisfies very desirable fairness properties. Hence, for this variant of online selection, fairness follows as a consequence of being online optimal.

More specifically, our main result for the multi-color secretary problem (Theorem 3.3) characterizes the optimal online algorithm and gives a closed formula for its competitive ratio. In the case where there are k groups and the maximum is equally likely to come from any of these groups, the competitive ratio is $k^{\frac{1}{k-1}}$ (Corollary 3.4). This is 2 for $k = 2$, $\sqrt{3}$ for $k = 3$, and $1 + O(\frac{\log k}{k})$ as $k \to \infty$. In regard to fairness, we show that for equal priors over k groups and arbitrary group sizes, the optimal online algorithm does not choose from all groups with equal probability (a property we coin 1-fairness), but approaches this property exponentially fast in the minimum group size (Theorem 3.10). For general priors over k groups, we show that when two groups j, j' have a similar prior $p_j > p_{j'} > (1-\varepsilon)p_j$, then the probability that the optimal online algorithm selects color j and the probability that it selects color j' are within ε of each other (Theorem 3.11). To exemplify this bound, consider the case where there are two groups, men and women, and the prior is such that the top candidate is a woman with probability 60% and a man with probability 40%. This translates into having $\varepsilon = 1/3$ in the theorem statement, and thus the optimal online algorithm will pick a woman at most 33% more often than a man.

We also present a set of experiments to illustrate the effectiveness of the proposed algorithm. We consider both synthetic and real world data sets, and compare to natural benchmark algorithms: (a) running the secretary algorithm ignoring the colors, and (b) choosing a color according to the prior and then running the secretary algorithm on that color. We consider a data set from a Portuguese banking institution. The goal of this experiment is to select a client and contact them and ask for their feedback. In order to achieve high quality feedback, we want to maximize the phone call duration while being fair with respect to the age of the interviewee. We consider five age groups, and that the best candidate of each group is the best overall with equal probability. In this example, the algorithm that ignores the colors is very unfair. It picks from the fourth age group in 80% of the runs. Both the fair benchmark algorithm and our optimal online algorithm pick from the five groups with roughly equal probability, but our algorithm selects more elements (+34.7%) and also more often the maximum element of a color (+76%).

Finally, we propose and analyze a sample-driven version of the multi-color secretary problem, along the lines of the model presented in Chapter 2. We propose a large class of algorithms that contains the optimal algorithms for the case of one color, and the case of many colors and no samples; and we find a closed formula for their success probability (Theorem 3.13).

3.2 Illustrative Example

Suppose you are hiring a professional for a position and have four candidates to fill it. Two come from school A and two from school B. You can compare candidates coming from the same school, but you cannot compare across schools. Suppose, in addition, that from previous experience, you know that 60% of the time the best candidate comes from school A. If the process is offline, and you can see all candidates simultaneously, the best strategy is to pick the best candidate from school A. This guarantees you a probability of picking the overall best of 0.6. On the contrary, if the process is online as in the secretary problem, and candidates come in random order, the situation changes dramatically. A natural idea would be to simply ignore the candidates from school B and run the secretary algorithm on the candidates from school A. For $n = 2$ the secretary algorithm selects the best with probability $1/2$, so you end up selecting the overall best candidate with probability 0.3. You could do instead something that is fairer to the candidates from school B: wait until you see the second candidate from any of the two schools. If they are the best from their school, select them. If not, select a candidate from the other school. One can easily show that with this policy you end up selecting the best candidate with probability 0.375. Indeed, the probability that the second candidate of a school arrives before the second of the other school is $1/2$, and this candidate is the best of their school with probability $1/2$, so the probability of selecting the overall best is

$$\frac{1}{2}\left(\frac{1}{2}0.6 + \frac{1}{2}\cdot\frac{1}{2}0.4\right) + \frac{1}{2}\left(\frac{1}{2}0.4 + \frac{1}{2}\cdot\frac{1}{2}0.6\right) = \frac{3}{8} = 0.375.$$

Thus, in the online setting, we can take advantage of the fact that in some realizations we observe both candidates from B before the second candidate from A. Our general result leverages this idea, by skipping a fraction of the candidates of each color, where the fraction depends continuously on the prior probability that the overall best candidate is of that color.

3.3 Related Work

An important precursor to our work is the aforementioned paper by [19]. Their starting point is the observation that the optimal policy for the classic secretary problem introduces incentives for candidates to arrive late. Indeed, that optimal policy skips the first 1/e fraction of candidates, and then selects the first candidate that is best-so-far. With this in mind, [19] look for incentive-compatible policies. That is, policies in which the acceptance probabilities for each of the arrival positions have to be the same. [89] use this paradigm to study fairness in the online allocation of tasks to workers. They model this problem through the weighted secretary problem with k positions, and design incentive-compatible algorithms for this problem.

A different approach to modeling fairness in online allocation problems is that of [98]. They consider the online allocation of a scarce resource, but not from a revenue optimization viewpoint. Rather they study the fill rate —the ratio of the allocated amount to observed demand— and seek algorithms maximizing the minimum fill rate.

Other works have considered different types of bias in online selection. [2] study a variant of the classic secretary problem in which candidates are evaluated independently by two committee members with different objectives. In their model each candidate is described by a pair of scores (x, y), where x and y are distributed uniformly and independently on $[0, 1]$ and observable by both selectors. One committee member is interested in x, and the other in y. The utility that the firm derives from a candidate is assumed to be $(x + y)/2$. Unanimous hiring decisions are respected, while candidates with a split decision are hired with probability p. A consensus cost c is deducted from the utility of a selector who has rejected a candidate that is nevertheless hired. The main result is that each stage game has a unique (symmetric) equilibrium which involves setting two thresholds $z < v$. A selector recommends hiring if her value is at least v, or her value is at least z and the other value is at least v. They then examine how p and c affect the utilities of the selectors and the firm at equilibrium.

[75] study the k-secretary problem, where the goal is to maximize the sum of weights of the candidates hired, when the algorithm observes biased evaluations, and propose algorithms for mitigating these biases. Their first model assumes that the candidates are partitioned into g groups $G_1, \ldots, G_g$; and that for a candidate $\mathrm{e} \in G_\mathrm{i}$ the algorithm observes $\widetilde{w}(\mathrm{e}) = w(\mathrm{e})/\beta_\mathrm{i}$ where $w(\mathrm{e}) \geq 0$ is the candidate's true value and $\beta_\mathrm{i} \geq 1$ is a bias factor shared by all members of that group. They seek algorithms that satisfy ranked demographic parity. This requires that (a) candidates of different groups that share the same rank within their group should be accepted with the same probability and (b) a candidate's acceptance probability should be monotonically increasing in the candidate's rank within her group. For adversarial group assignment and adversarial weights, they show a lower bound of $\Omega(g)$ on the competitive ratio of any online algorithm, and give a $((g + 1)\mathrm{e}^2)$-competitive algorithm. They also consider a model, where the candidates are partially ordered, and adjust their notion of ranked demographic parity to that situation. For this model they show a lower bound of $\Omega(\omega)$ on the competitive ratio of any online algorithm, where ω is the width of the poset; and they give an online algorithm that achieves a competitive ratio of $2\mathrm{e}^3(2\omega + 1)(1 + o(1))$ as the number of candidates tends to infinity.

[96] and [66] don't take a fairness perspective, but study secretary problems which share features of our model. [96] consider the problem of selecting a maximal secretary from a partially ordered set of candidates. Their algorithm skips the first $\tau(k)$ elements where k is the number of maximal secretaries. Afterwards, it takes any undominated candidate, provided that among the candidates seen so far there are at most k undominated secretaries. This latter condition may make them pass on undominated candidates that arrive early in the sequence, but it will never pass on the last maximal candidate in the permutation. They show that this algorithm succeeds wit probability at least $k^{-k/(k-1)}((1 + \log k^{1/(k-1)})^k - 1)$. This

approaches 1/e as $k \to 1$, and 1 as k grows large. For the special case where the poset consists of k chains, they show that their algorithm succeeds with probability $k^{-1/(k-1)} - O(k/n)$, while no online algorithm can achieve a better success probability than $k^{-1/(k-1)} - o(1)$. This final result is also applicable to our problem, but only in the special case where the best secretary is equally likely to come from any of the groups.

[66] study the problem of selecting candidates in parallel: each candidate is randomly assigned to one of Q queues and candidates can only be compared with other candidates in the same queue. Only the first D candidates in each queue can be hired. The objective is to select k secretaries, and to hire as many of the top k secretaries as possible. For a given parameter $\mathrm{d} \in \mathbb{N}$, they show that if $k = 1$ and $D = n/\mathrm{d}$, then with $Q = 1$ queue the best possible ratio is $(\mathrm{de})^{-1}$, while with $Q = \mathrm{d}$ queues it is possible to achieve a ratio of $\mathrm{d}^{-\mathrm{d}/(\mathrm{d}-1)} - o(1)$. The connection to the poset model and our model is that the uniform random assignment to queues can be interpreted as assigning each secretary one of Q colors uniformly at random, with all secretaries assigned to the same color forming a chain. The best secretary in a chain is the best secretary overall with probability $1/Q$. So this model is again restricted to the case where the best secretary overall is equally like to come from any of the groups.

A final set of related works consider secretary and/or prophet problems with restrictions on which information is available to the decision maker. [15], for example, consider an optimal stopping problem where the decision maker gets imperfect information about random variables presented to her one-by-one, such as the expected reward or some multi-dimensional signal and the expected reward conditional on that signal.

3.4 Preliminaries

In the *multi-color secretary problem* n candidates arrive in uniform random order. Candidates are partitioned into k groups $C = \{C_1, \cdots, C_k\}$. We write $\mathbf{n} = (n_1, \ldots, n_k)$ for the vector of group sizes, i.e., $|C_j| = n_j$, for all $1 \leq j \leq k$. We identify each of the groups with a distinct color and denote by $c(\mathrm{i})$ the color of candidate i. We can compare candidates of the same color, but we cannot compare candidates across groups. We assume comparisons are strict, and use $\mathrm{i} \succ \mathrm{i}'$ to denote that candidate i is better than candidate i′. We write $\max C_j$ for the best candidate of color j, and $\max C$ for the best candidate overall. A natural assumption is that the best candidate from a group is the best candidate overall with equal probability $1/k$, but we can also consider the case where these probabilities are different. We denote the probabilities with which the best candidate of group j is the best candidate overall by p_j, and write $\mathbf{p} = (p_1, \ldots, p_k)$ for the vector of these probabilities. Since candidates are incomparable across groups this can be modeled by tossing a coin *after the fact* to decide whether the best candidate of group j is the best candidate overall. The goal is to design an online algorithm that maximizes the probability of selecting the best candidate overall.

Competitive ratio. We evaluate online algorithms by means of their competitive ratio. Consider some online algorithm ALG. The algorithm selects the best candidate overall, if it selects the best candidate of a given color and this color has the best candidate overall. For an instance of the multi-color secretary problem with group sizes $\mathbf{n}$ and probabilities $\mathbf{p}$, we denote by $\mathrm{ALG}(\mathbf{n},\mathbf{p}) \in \{1,...,n\} \cup \{\phi\}$ the random index at which the algorithm stops, where ϕ denotes the case when the algorithm does not stop. The success probability of ALG is $\mathbf{E}[\mathbf{1}_{\mathrm{ALG}(\mathbf{n},\mathbf{p})=\max C}] = \mathbf{E}[p_{c(\mathrm{ALG}(\mathbf{n},\mathbf{p}))} \cdot \mathbf{1}_{\mathrm{ALG}(\mathbf{n},\mathbf{p})=\max C_{c(\mathrm{ALG}(\mathbf{n},\mathbf{p}))}}]$. We compare this to the optimal offline algorithm OPT, i.e., the best algorithm that can select a candidate after all candidates have arrived. An optimal strategy is to choose a color j with maximum p_j, and then, choose the best candidate of that color. We denote by $\mathrm{OPT}(\mathbf{n},\mathbf{p}) \in \{1,...,n\}$ the random index that OPT selects. The success probability of OPT is $\mathbf{E}[\mathbf{1}_{\mathrm{OPT}(\mathbf{n},\mathbf{p})=\max C}] = \max\{p_1,\ldots,p_k\}$.

Definition 3.1 (competitive ratio)**.** Fix k and $\mathbf{p} = (p_1,\ldots,p_k)$. An online algorithm ALG is $\beta(k,\mathbf{p})$-*competitive* if for all input lengths n and partition sizes $\mathbf{n} = (n_1,\ldots,n_k)$,

$$\frac{\mathbf{E}[\mathbf{1}_{\mathrm{OPT}(\mathbf{n},\mathbf{p})=\max C}]}{\mathbf{E}[\mathbf{1}_{\mathrm{ALG}(\mathbf{n},\mathbf{p})=\max C}]} \leq \beta(k,\mathbf{p}).$$

Note that $\beta(k,\mathbf{p}) \geq 1$, and the smaller $\beta(k,\mathbf{p})$ the better the approximation guarantee.

Unbiased selection. We also examine the extent to which online or offline algorithms are biased, where ideally selection should be unbiased. One way to measure this is by quantifying how much the probability of selecting from any given color class j can differ from the corresponding probability p_j.

Definition 3.2 (fairness)**.** Fix k and $\mathbf{p} = (p_1,\ldots,p_k)$. An offline or online algorithm ALG is $\alpha(\mathbf{n},\mathbf{p})$-*fair*, where $\alpha(\mathbf{n},\mathbf{p}) \geq 1$, if for all colors $j \in [k]$,

$$\begin{aligned}\frac{p_j}{\alpha(\mathbf{n},\mathbf{p})} &\leq \mathbf{P}(c(\mathrm{ALG}(\mathbf{n},\mathbf{p})) = j \mid \mathrm{ALG}(\mathbf{n},\mathbf{p}) \neq \phi) \\ &\leq \alpha(\mathbf{n},\mathbf{p}) \cdot p_j.\end{aligned}$$

Uniform arrival times. We model uniform random arrival *order* through uniform random arrival *times*. For this, we sample n independent realizations of the Uniform$[0,1]$ distribution, and denote them by $\tau_1 < \tau_2 < \ldots < \tau_n$ indexed in increasing order.

3.5 Optimal Online Algorithm

We derive the optimal online algorithm (without fairness considerations), and observe that—in sharp contrast to the optimal offline algorithm—it is robustly fair and provides an "equal treatment of equals" guarantee.

3.5.1 The Algorithm

We show that the optimal online algorithm is from the class of algorithms given by Algorithm 2. Algorithms from this class receive as input a vector of thresholds $\mathbf{t} = (t_1,\ldots,t_k)$, one for each color $j \in [k]$. When a candidate i arrives the algorithm first checks if the candidate arrived after the time threshold for its color $t_{c(\mathrm{i})}$, and if it did, then it accepts the candidate if it is the best candidate of that color so far.

Algorithm 2 GroupThresholds($\mathbf{t}$)

Input: $\mathbf{t} \in [0,1]^k$, a threshold in time for each group
Output: $\mathrm{i} \in [n]$, index of chosen candidate
assuming arrival times $\tau_1 < \ldots < \tau_n$

```
for i ← 1 ... n do
    if τ_i > t_{c(i)} then
        if i ≻ max{i' | τ_{i'} ≤ τ_i, c(i') = c(i)} then
            return i
        end if
    end if
end for
```

Notice that the time based arrival model considered in this section is equivalent to the random order arrival model and is used for the sake of simplicity of presentation and proofs. If we are given an algorithm in the time-based model (such as Algorithm 2), then we can translate it into the random arrival model by having the algorithm draw n arrival times from Uniform$[0,1]$, and assign the i-th smallest arrival time to the i-th candidate in the input stream. If, on the other hand we are given an algorithm in the uniform arrival model, then we can translate it into the time-based model by just ignoring the time component and just using that the candidate that arrived at τ_i was the i-th candidate to arrive.

Therefore any algorithm in one model can be easily used in the other model with identical properties.

3.5.2 Competitive Ratio

Surprisingly, we can show that for any probabilities $\mathbf{p} = (p_1, \ldots, p_k)$ there exist optimal thresholds $\mathbf{t}^* = (t_1^*, \ldots, t_k^*)$ that achieve the best competitive ratio. Later on, we show how these thresholds can be computed explicitly (see Lemma 3.8). Using these thresholds in Algorithm 2 results in the promised optimal online algorithm. Let us start by presenting the success probability of our algorithm for general probabilities and then for the special case that $\mathbf{p} = (1/k, \ldots, 1/k)$. Afterwards we provide an overview of the proof of these results.

Theorem 3.3 (competitive ratio, general probabilities). *Fix k and $\mathbf{p} = (p_1, \ldots, p_k)$. Assume wlog that $p_j \geq p_{j+1}$ for all $j < k$. Then there exist thresholds $\mathbf{t}^* = (t_1^*, \ldots, t_k^*)$ such that $t_j^* \leq t_{j+1}^*$ for all $j < k$ that depend only on the number of colors k and the probabilities $\mathbf{p}$ but not on the number of candidates n or the partition sizes $\mathbf{n} = (n_1, \ldots, n_k)$ such that Algorithm 2 with thresholds $\mathbf{t}^*$ succeeds with probability at least*

$$\sum_{j=1}^{k} \int_{t_j^*}^{t_{j+1}^*} \left(\sum_{j'=1}^{j} p_{j'} \right) \frac{T_j^*}{\tau^j} \, \mathrm{d}\tau \,,$$

where $T_j^* = \prod_{j'=1}^{j} t_{j'}$. For all k and $\mathbf{p} = (p_1, \ldots, p_k)$, no online algorithm can achieve a better competitive ratio in the worst-case over all number of candidates n and partition sizes $\mathbf{n} = (n_1, \ldots, n_k)$.

For the special case where $\mathbf{p} = (1/k, \ldots, 1/k)$ we obtain the following corollary. It shows that in this case we can set a single threshold, and it also provides a simpler-to-parse formula for the competitive ratio.

Corollary 3.4 (competitive ratio, equal probabilities)**.** *Fix k and $\mathbf{p} = (1/k, \ldots, 1/k)$. Then there exists a single threshold t^* such that Algorithm 2 with thresholds $\mathbf{t}^* = (t^*, \ldots, t^*)$ achieves a competitive ratio of*

$$k^{\frac{1}{k-1}}.$$

This is 2 for $k = 2$, $\sqrt{3}$ for $k = 3$, and $1 + O(\frac{\log k}{k})$ as $k \to \infty$. For all k and $\mathbf{p} = (1/k, \ldots, 1/k)$, no online algorithm can achieve a better competitive ratio in the worst-case over all number of candidates n and partition sizes $\mathbf{n} = (n_1, \ldots, n_k)$.

The main difficulty in proving Theorem 3.3 and Corollary 3.4 is that in the point-wise optimal online algorithm, which can be obtained by backward induction, thresholds depend on the number of candidates of each color that have already arrived. This dependency leads to a blow-up in algorithm complexity, and complicates the analysis of the success probability. Our high-level approach is to argue that in the worst-case all n_j's are large, and that in this case the point-wise optimal online algorithm is well approximated by the optimal algorithm from the class of algorithms desbribed in Algorithm 2, which simply sets time-dependent thresholds. So we can optimize over these. We decompose the proof into several lemmas, and summarize at the end of the section how they imply Theorem 3.3.

A first ingredient in our proof is Lemma 3.5, which shows that for the class of algorithms in Algorithm 2, for any vector of thresholds $\mathbf{t}$ the worst-case arises when all n_j's are large .

Lemma 3.5. *Fix the probabilities $\mathbf{p} = (p_1, \ldots, p_k)$ and a vector of thresholds $\mathbf{t} \in [0, 1]^k$. For all $j = 1, \ldots k$, the success probability of* GroupThresholds$(\mathbf{t})$ *is decreasing in n_j.*

Proof. By symmetry it is enough to prove the lemma for $j = 1$. Fix values for $n_1, n_2, \ldots, n_k$ and consider an instance with these sizes. We prove that the success probability of GroupThresholds(t) in this instance is lower than the success probability in the instance with $n_1 - 1$ candidates of group 1, and n_j candidates of groups $j > 1$ that results from removing the worst candidate of group 1. In fact, we can couple the realizations of the arrival times of the two instances by taking the arrival times of the smaller instance and sampling the arrival time of the worst candidate of group 1. Consider a realization for the smaller instance where GroupThresholds$(\mathbf{t})$ fails. Then, either it never accepts a candidate or it accepts a candidate that is not the overall maximum. In any of the two cases, adding the worst candidate of group 1 does not alter the relative ranks of other candidates, so the only possible difference is that the algorithm now selects him. Since he cannot be the best candidate, the algorithm also fails. $\square$

Our next pair of lemmas, Lemma 3.6 and Lemma 3.7, allow us to bound the success probability of the point-wise optimal online algorithm by the limit success probability of the best algorithm which sets time-dependent thresholds (Algorithm 2).

Lemma 3.6. *Denote by* $\mathrm{GT}(\mathbf{p},\mathbf{t})$ *the limit when* $\min_j n_j$ *tends to infinity of the success probability of* GroupThresholds($\mathbf{t}$) *for a given vector of probabilities* $\mathbf{p}$*. If* $\min_j t_j \geq c > 0$*, then the success probability of* GroupThresholds($\mathbf{t}$) *is at most* $\mathrm{GT}(\mathbf{p},\mathbf{t}) + k \cdot (1-c)^z$*, where* $z = \min_j n_j$*.*

Proof. Fix a vector of probabilities $\mathbf{p}$. Consider a vector $\mathbf{t}$ such that $\min_j t_j \geq c$ and two vectors of sizes $\mathbf{n}, \mathbf{n}'$, such that $n_j < n'_j$ for all $j \in [k]$. Denote by $GT(\mathbf{p},\mathbf{t},\mathbf{n})$ and $GT(\mathbf{p},\mathbf{t},\mathbf{n}')$ the success probabilities of GroupThresholds($\mathbf{t}$) when the instance is of size $\mathbf{n}$ and $\mathbf{n}'$, respectively.

Couple the arrival times of the two instances in the following way. For each color j, identify the n_j candidates of color j of the smaller instance with the best n_j candidates of color j in the larger instance. Now, we run in parallel GroupThresholds($\mathbf{t}$) in the two coupled instances. This means that in the smaller instance we will be ignoring the smallest $n'_j - n_j$ candidates of the larger instance. Note that the smallest elements do not alter the relative rank of the largest elements. Thus, if the algorithm in the smaller instance stops, then the algorithm running in the larger instance either stops with the same candidate, or stops earlier with one of the smallest $n'_j - n_j$ candidates of some color j. Now, for a given color j, if one of the best n_j candidates arrives before t_j, then the algorithm in the larger instance will never select any of the smallest $n'_j - n_j$ candidates. Therefore, if for all colors j, at least one of the largest n_j candidates of color j arrives before t_j, the two algorithms stop with the same candidate. In other words, if the two algorithms stop with different candidates, necessarily for some color j the best n_j candidates arrived all after t_j. Using a union bound, the latter event happens with probability at most $\sum_{j=1}^{k}(1-t_j)^{n_j}$, which in turn is at most $k \cdot (1-c)^z$, where $z = \min_j n_j$. Therefore,

$$GT(\mathbf{p},\mathbf{t},\mathbf{n}) - k \cdot (1-c)^z \leq GT(\mathbf{p},\mathbf{t},\mathbf{n}').$$

Note that only the right-hand side depends on $\mathbf{n}'$, so we can take limit when $\min_j n'_j$ tends to infinity, which by Lemma 3.5 exists, and conclude the result. $\square$

Lemma 3.7. *For any vector of probabilities* $\mathbf{p}$ *and sizes* $\mathbf{n}$*, denote by* $\mathrm{ON}(\mathbf{n},\mathbf{p})$ *the optimal success guarantee of an online algorithm. Then for every* $\mathbf{p},\mathbf{n}$ *there exists a vector* $\mathbf{t}$ *such that* $\mathrm{ON}(\mathbf{n},\mathbf{p}) \leq \mathrm{GT}(\mathbf{p},\mathbf{t}) + o(1)$*, where* $\min_j t_j \geq 1/(2\mathrm{e})$*.*

Proof. Consider the optimal online algorithm obtained doing backward induction. This is, when facing a candidate i that is the best seen so far of color $c(\mathrm{i})$, and having seen r_j candidates of color j, for $j \in [k]$, accept candidate i if the probability that i is the overall best is larger or equal to the probability that we select the best overall if we do not stop with i and continue using the optimal policy. Denote by $b(c(\mathrm{i}),\mathbf{r})$ the former probability, and by $B(\mathbf{r})$ the latter probability. Thus, the optimal algorithm stops if $b(c(\mathrm{i}),\mathbf{r}) \geq B(\mathbf{r})$.

On the one hand we can calculate exactly $b(c(\mathrm{i}),\mathbf{r})$. Let $j = c(\mathrm{i})$. It is the probability that the best of the first r_j candidates of color j is the best candidate overall. Now, the best of the first r_j candidates of color j is the best of color j with probability r_j/n_j, and the best of color j is the best overall with probability p_j. Therefore, $b(c(\mathrm{i}),\mathbf{r}) = b(j,\mathbf{r}) = p_j \cdot r_j/n_j$, which is an increasing function of r_j. On the other hand, $B(\mathbf{r})$ is not as easy to calculate, but it is easy to see that it is a decreasing function of r_j, for all $j \in [k]$.

Suppose we are considering a candidate arriving at time t. If $\min_{j'} n_{j'}$ is large, we have that $r_j/n_j \approx t$. Taking advantage of this fact, we approximate the optimal online algorithm with the following that we denote by $\mathrm{ON}'(\mathbf{n}, \mathbf{p})$. We accept a candidate of color j that arrives at time t if he is the best of color j seen so far, and $b(j, \lfloor t\mathbf{n} \rfloor) \geq B(\lfloor t\mathbf{n} \rfloor)$, where $\lfloor t\mathbf{n} \rfloor := (\lfloor tr_1 \rfloor, \ldots, \lfloor tr_k \rfloor)$. By the monotonicity of b and B, for each color j there is a value t'_j such that $b(j, \lfloor t\mathbf{n} \rfloor) \geq B(\lfloor t\mathbf{n} \rfloor)$ if and only if $t \geq t'_j$. Thus, it turns out that for $\mathbf{t}'$ defined this way, $\mathrm{ON}'(\mathbf{n}, \mathbf{p})$ is actually $\textsc{GroupThresholds}(\mathbf{t}')$.

We prove first that $\min_j t'_j \geq 1/(2\mathrm{e})$. In fact, note that the success probability of the optimal algorithm must be at least $\frac{1}{\mathrm{e}} \max_j p_j$, because we can always apply the regular secretary algorithm in the color with highest p_j. Also, the probability that the optimal online algorithm selects the overall best candidate before time $1/(2\mathrm{e})$ is at most $\frac{1}{2\mathrm{e}} \max_j p_j$, since the best of a color arrives before time $1/(2\mathrm{e})$ with probability $1/(2\mathrm{e})$. Therefore, the probability that the optimal algorithm finds the overall best candidate after time $1/(2\mathrm{e})$ is at least $\frac{1}{2\mathrm{e}} \max_j p_j$, and therefore, $B(\lfloor \mathbf{n}/(2\mathrm{e}) \rfloor) \geq \frac{1}{2\mathrm{e}} \max_j p_j$. Recall that $b(j, \mathbf{r}) = p_j r_j / n_j$, so $b(j, \lfloor \mathbf{n}/(2\mathrm{e}) \rfloor) \leq p_j/(2\mathrm{e})$. Thus, we conclude that $t'_j \geq 1/(2\mathrm{e})$ for every $j \in [k]$. Using Lemma 3.6 we have that $\mathrm{ON}'(\mathbf{n}, \mathbf{p}) \leq \mathrm{GT}(\mathbf{p}, \mathbf{t}') + k \cdot (1 - \frac{1}{2\mathrm{e}})^{\min_j n_j}$.

Now we prove that $\mathrm{ON}'(\mathbf{n}, \mathbf{p})$ approximates well $\mathrm{ON}(\mathbf{p}, \mathbf{n})$. Consider some small $\varepsilon > 0$. We show that (1) for large $\min_j n_j$, the probability that either of the two algorithms stops with a color j in $[t'_j - \varepsilon, t'_j + \varepsilon]$ is small, and (2) that outside that interval the two algorithms make the same decisions with high probability.

For the first fact, note that both algorithms stop only with a candidate that is the best seen so far of the same color. Now, if the best candidate of color j in the interval $[0, t'_j + \varepsilon]$ arrives in $[0, t'_j - \varepsilon]$, the algorithm will not stop with a candidate of color j in $[t'_j - \varepsilon, t'_j + \varepsilon]$. Thus, either of the two algorithms stops in $[t'_j - \varepsilon, t'_j + \varepsilon]$ with a candidate of color j for some $j \in [k]$, with probability at most $\sum_{j=1}^{k} \frac{2\varepsilon}{t'_j} \leq 4\mathrm{e}\varepsilon k$.

For the second fact, we prove that with high probability, for each $j \in [k]$, at time $t = t'_j - \varepsilon$, $\mathbf{r} \leq t'_j \mathbf{n}$ and at time $t = t'_j + \varepsilon$, $\mathbf{r} \geq t'_j \mathbf{n}$, where the comparisons are element-wise. We use the following standard Chernoff bounds. If X is a binomial random variable and $0 < \delta < 1$, then

$$\mathbf{P}(X \geq (1+\delta)\mathbf{E}(X)) \leq \mathrm{e}^{-\frac{1}{3}\delta^2 \mathbf{E}(X)}, \text{ and}$$
$$\mathbf{P}(X \leq (1-\delta)\mathbf{E}(X)) \leq \mathrm{e}^{-\frac{1}{3}\delta^2 \mathbf{E}(X)}.$$

Since r_j at time t distributes as a $\mathrm{Binomial}(n_j, t)$, we can use these bounds to get that at time $t = t'_j - \varepsilon$,

$$\mathbf{P}(r_j \geq t'_j n_j) \leq \exp\left(-\frac{1}{3}\varepsilon^2 n_j (t'_j - \varepsilon)\right),$$

and at time $t = t'_j + \varepsilon$,

$$\mathbf{P}(r_j \leq t'_j n_j) \leq \exp\left(-\frac{1}{2}\varepsilon^2 n_j (t'_j + \varepsilon)\right).$$

Since $t'_j \geq 1/(2\mathrm{e})$ for all j, we can take $\varepsilon = 20\frac{\log n_j}{\sqrt{n_j}}$ and use a union bound to obtain that with probability $1 - O(\frac{k}{\min_{j'} n_{j'}})$, for each $j \in [k]$, at time $t = t'_j - \varepsilon$, $\mathbf{r} \leq t'_j\mathbf{n}$ and at time $t = t'_j + \varepsilon$, $\mathbf{r} \geq t'_j\mathbf{n}$. From the monotonicity of b and B, we have that with probability $1 - O(\frac{k}{\min_{j'} n_{j'}})$, for all $j \in [k]$, before time $t'_j - \varepsilon$ it holds that $b(j, \mathbf{r}) < B(\mathbf{r})$, and after time $t'_j + \varepsilon$ it holds that $b(j, \mathbf{r}) > B(\mathbf{r})$.

Putting together facts (1) and (2), we get that with probability $1 - O\left(k\frac{\log(\min_j n_j)}{\sqrt{\min_j n_j}}\right)$, $\mathrm{ON}'(\mathbf{n}, \mathbf{p})$ and $\mathrm{ON}(\mathbf{n}, \mathbf{p})$ select the same candidate. Thus, when $\min_j n_j$ tends to infinity,

$$\mathrm{ON}(\mathbf{n}, \mathbf{p}) \leq \mathrm{ON}'(\mathbf{n}, \mathbf{p}) + o(1) \leq \mathrm{GT}(\mathbf{p}, t') + o(1).$$

This concludes the proof of the lemma.

□

The final ingredient is the following pair of lemmas, Lemma 3.8 and Lemma 3.9, which solve for the optimal time-dependent thresholds and give a formula for evaluating the limit success probability in terms of these thresholds.

Lemma 3.8. *Consider a vector* $\mathbf{p}$ *such that* $p_j \geq p_{j+1}$ *for all* $j < k$*. The optimal thresholds* $\mathbf{t}^*$ *are given by*

$$\begin{aligned} t^*_k &= (1 - (k-1)p_k)^{\frac{1}{k-1}} , \\ t^*_j &= t^*_{j+1} \cdot \left(\frac{\sum_{r=1}^{j} \frac{p_r}{j-1} - p_j}{\sum_{r=1}^{j} \frac{p_r}{j-1} - p_{j+1}} \right)^{\frac{1}{j-1}} , \ \textit{for } 2 \leq j \leq k-1 \\ t^*_1 &= t^*_2 \cdot \mathrm{e}^{\frac{p_2}{p_1} - 1} . \end{aligned}$$

Proof. From the formula in Lemma 3.9 we can check that in the optimal vector $\mathbf{t}^*$, it holds that $t^*_j \leq t^*_{j'}$ if $p_j \geq p_{j'}$, by simply interchanging consecutive t_j's. With this in hand, we can assume w.l.o.g. that $p_1 \geq p_2 \geq \cdots \geq p_k$ and $t^*_1 \leq t^*_2 \leq \cdots \leq t^*_k$. We impose first order conditions to obtain the recursive formula.

Consider $1 \leq j \leq k$. We have that

$$\frac{\partial}{\partial t_j} \mathrm{GT}(\mathbf{p}, \mathbf{t}) = \frac{1}{t_j} \sum_{\ell=j}^{k} \int_{t_\ell}^{t_{\ell+1}} \left(\sum_{r=1}^{\ell} p_r \right) \frac{T_\ell}{\tau^\ell} \,\mathrm{d}\tau - p_j \frac{T_j}{t_j^j}.$$

Setting this derivative equal to zero we obtain the equation

$$p_j \frac{T^*_j}{t_j^{*j-1}} = \sum_{\ell=j}^{k} \int_{t^*_\ell}^{t^*_{\ell+1}} \left(\sum_{r=1}^{\ell} p_r \right) \frac{T^*_\ell}{\tau^\ell} \,\mathrm{d}\tau, \tag{3.1}$$

for all $1 \leq j \leq k$.

For the case of $j = k$, note that $\sum_{r=1}^{k} p_r = 1$, so we obtain the following formula for t_k^*.

$$\begin{aligned} \frac{p_k}{t_k^{*k-1}} &= \frac{1}{k-1}\left(\frac{1}{t_k^{*k-1}} - 1\right) \\ \Leftrightarrow \qquad t_k^* &= (1-(k-1)p_k)^{\frac{1}{k-1}}. \end{aligned}$$

For $1 \leq j \leq k-1$ we can substract Equation (3.1) for two consecutive indices, obtaining

$$\begin{aligned} p_j \frac{T_j^*}{t_j^{*j-1}} - p_{j+1}\frac{T_{j+1}^*}{t_{j+1}^{*j}} &= \int_{t_j^*}^{t_{j+1}^*} \left(\sum_{r=1}^{j} p_r\right) \frac{T_j^*}{\tau^j}\,\mathrm{d}\tau \\ \Leftrightarrow \frac{p_j}{t_j^{*j-1}} - \frac{p_{j+1}}{t_{j+1}^{*j-1}} &= \int_{t_j^*}^{t_{j+1}^*} \left(\sum_{r=1}^{j} p_r\right) \frac{1}{\tau^j}\,\mathrm{d}\tau. \end{aligned} \tag{3.2}$$

For i ≥ 2, this is equivalent to

$$\begin{aligned} \frac{p_j}{t_j^{*j-1}} - \frac{p_{j+1}}{t_{j+1}^{*j-1}} &= \frac{1}{j-1}\left(\sum_{r=1}^{j} p_r\right)\left(\frac{1}{t_j^{*j-1}} - \frac{1}{t_{j+1}^{*j-1}}\right) \\ \Leftrightarrow t_j^* &= t_{j+1}^* \left(\frac{\sum_{r=1}^{j} \frac{p_r}{j-1} - p_j}{\sum_{r=1}^{j} \frac{p_r}{j-1} - p_{j+1}}\right)^{\frac{1}{j-1}}. \end{aligned}$$

For $j = 1$ Equation (3.2) becomes

$$\begin{aligned} p_1 - p_2 &= p_1 \int_{t_1^*}^{t_2^*} \frac{1}{\tau}\,\mathrm{d}\tau \\ \Leftrightarrow p_1 - p_2 &= p_1 \log(t_2^*/t_1^*) \\ \Leftrightarrow t_1^* &= t_2^* \exp\left(\frac{p_2}{p_1} - 1\right). \end{aligned}$$

This concludes the proof of the lemma.

□

Lemma 3.9. *Consider vectors of probabilities* $\mathbf{p}$ *and thresholds* $\mathbf{t}$*, and assume* $t_{\mathrm{i}} \leq t_{\mathrm{i}+1}$ *for all* $\mathrm{i} < k$*. The limit success probability of* GroupThresholds($\mathbf{t}$) *is given by*

$$\mathrm{GT}(\mathbf{p}, \mathbf{t}) = \sum_{j=1}^{k} \int_{t_j}^{t_{j+1}} \left(\sum_{j'=1}^{j} p_{j'}\right) \frac{T_j}{\tau^j}\,\mathrm{d}\tau\,,$$

where $T_j = \prod_{j'=1}^{j} t_{j'}$.

Proof. Since we are interested in the limit probability, we can assume that in every interval at least one candidate of each color arrives. First, note that the algorithm does not stop before time τ if and only if for every color j, the best candidate that arrives in $[0, \tau]$ arrives before t_j. This happens with probability

$$\prod_{j=1}^{k} \frac{\min\{\tau, t_j\}}{\tau}.$$

Now, the algorithm stops with the best candidate of color j' if this candidate arrives at a time $\tau \geq t_{j'}$ and the algorithm does not stop before time τ. Therefore, conditioning on τ, the probability that the algorithm selects the best candidate of color j' is

$$\int_{t_{j'}}^{1} \prod_{j=1}^{k} \frac{\min\{\tau, t_j\}}{\tau} \, \mathrm{d}\tau.$$

If the algorithm stops with the best candidate of color j', the algorithm succeeds with probability $p_{j'}$. Therefore, in total the algorithm succeeds with probability

$$\begin{aligned}
\sum_{j'=1}^{k} p_{j'} \int_{t_{j'}}^{1} \prod_{j=1}^{k} \frac{\min\{\tau, t_j\}}{\tau} \, \mathrm{d}\tau
&= \sum_{j'=1}^{k} p_{j'} \sum_{j=j'}^{k} \int_{t_j}^{t_{j+1}} \frac{T_j}{\tau^j} \, \mathrm{d}\tau \\
&= \sum_{j=1}^{k} \int_{t_j}^{t_{j+1}} \left(\sum_{j'=1}^{j} p_{j'} \right) \frac{T_j}{\tau^j} \, \mathrm{d}\tau,
\end{aligned}$$

where $t_{j+1} := 1$ and $T_j = \prod_{r=1}^{j} t_r$.

□

Putting together these lemmas yields Theorem 3.3.

Proof of Theorem 3.3. From Lemma 3.5 we have that the success probability of the algorithm is at least its limit. Lemma 3.8 characterizes the optimal thresholds, and from Lemma 3.7 we have that in the limit the optimal online algorithm has the same success guarantee as GroupThresholds($\mathbf{t}^*$), so no other algorithm can have a better worst-case guarantee.

□

3.6 Fairness

The optimal offline algorithm is 1-fair for $\mathbf{p} = (1/k, \ldots, 1/k)$, but as soon as probabilities are unbalanced it will choose only from the colors which have maximum p_j. In the worst case, $|p_j - p_{j'}| < \varepsilon$ for all j, j', but the optimal offline algorithm is forced to choose from the unique color j which has maximum p_j. We show that in the case where $p_j = 1/k$ for all j, the optimal online algorithm is not exactly 1-fair, but approaches 1-fairness exponentially fast in the minimum group size $\min_j n_j$.

Theorem 3.10 (fairness result, equal probabilities)**.** *For any k and $\mathbf{p} = (1/k, \ldots, 1/k)$, Algorithm 2 with the optimal single threshold t^* is $1 + O(k^2(1 - \frac{1}{\mathrm{e}})^{\min_j n_j})$-fair.*

Proof. From the first order conditions, we have that

$$\begin{aligned} &\frac{\partial}{\partial t_1}\mathrm{GT}(\mathbf{p}, \mathbf{t}^*) = 0 \\ \Leftrightarrow\ &\frac{1}{t_1^*}\mathrm{GT}(\mathbf{p}, \mathbf{t}^*) - p_1 = 0 \\ &\Leftrightarrow \mathrm{GT}(\mathbf{p}, \mathbf{t}^*) = p_1 \cdot t_1^*. \end{aligned}$$

Since the optimal success probability is at least p_1/e, we have that $t_1^* \geq 1/\mathrm{e}$.

To see that the algorithm is $1 + O(k^2(1 - 1/\mathrm{e})^{\min_j n_j})$-fair, consider an instance of sizes $n_1' = n_2' = \cdots = n_k' = \max_j n_j$. Certainly in this other instance the algorithm is 1-fair, because all the thresholds are equal. We couple the two instances by identifying the best n_j candidates of color j of the larger instance with the candidates of color j of the smaller instance, and run the algorithm in both, in parallel. In color j, the worst $n_j' - n_j$ candidates do not alter the relative rank of the best n_j candidates, and if one of the best n_j candidates arrives before time t_j^*, then the algorithm in the larger instance will not select one of the worst $n_j' - n_j$ candidates. Therefore, in order for the two algorithms to select a different candidate we need that for some color j all the best n_j candidates arrive after t_j^*. This happens with probability at most $\sum_{j=1}^{k}(1-t_j^*)^{n_j} \leq k(1-\frac{1}{\mathrm{e}})^{\min_j n_j}$. In the larger instance the algorithm stops with each color with probability $\Theta(1/k)$, so $k(1 - \frac{1}{\mathrm{e}})^{\min_j n_j}$ is an $O(k^2(1 - \frac{1}{\mathrm{e}})^{\min_j n_j})$ fraction of it, and therefore the algorithm in the smaller instance is $1 + O(k^2(1 - \frac{1}{\mathrm{e}})^{\min_j n_j})$-fair.

□

Moreover, we show that the optimal online algorithm is robust and degrades gracefully as we move away from perfectly balanced probabilities.

Theorem 3.11 (fairness result, general probabilities)**.** *Fix k and $\mathbf{p} = (p_1, \ldots, p_k)$. Algorithm 2 with the optimal choice of thresholds $\mathbf{t}^* = (t_1^*, \ldots, t_k^*)$ ensures that if $p_j = p_{j'}$ then $t_j^* = t_{j'}^*$. Moreover, $\mathbf{t}^*$ is a continuous function of $\mathbf{p}$. So if p_j and $p_{j'}$ are close so are t_j^* and $t_{j'}^*$ and so is the probability of selection. More precisely, if $p_j > p_{j'} > (1 - \varepsilon)p_j$, then $t_{j'}^* > t_j^* > (1 - \varepsilon)t_{j'}^*$, and furthermore,*

$$\begin{aligned} 0 < \mathbf{P}(\textsc{GroupThresholds}(\mathbf{t}^*) \text{ selects color } j) \\ - \mathbf{P}(\textsc{GroupThresholds}(\mathbf{t}^*) \text{ selects color } j') < \varepsilon. \end{aligned}$$

Proof. The facts that $p_j = p_{j'}$ implies $t_j^* = t_{j'}^*$ and that $\mathbf{t}^*$ is continuous in $\mathbf{p}$ follow directly from the formulas in Lemma 3.8.

We prove now the more precise bound. Assume $p_1 \geq p_2 \geq \cdots \geq p_k$, and that $p_{j+1} \geq (1 - \varepsilon)p_j$. We will prove first that $t_j^* \geq \mathrm{e}^{-\varepsilon}t_{j+1}^*$. From Lemma 3.8, it holds trivially for $j = 1$.

For $j \geq 2$ we have that t_j^*/t_{j+1}^* equals

$$\begin{aligned}
\left(\frac{\sum_{r=1}^{j} \frac{p_r}{j-1} - p_j}{\sum_{r=1}^{j} \frac{p_r}{j-1} - p_{j+1}}\right)^{\frac{1}{j-1}} &\geq \left(\frac{\sum_{r=1}^{j} \frac{p_j}{j-1} - p_j}{\sum_{r=1}^{j} \frac{p_j}{j-1} - p_{j+1}}\right)^{\frac{1}{j-1}} \\
&\geq \left(\frac{\frac{j}{j-1} - 1}{\frac{j}{j-1} - (1-\varepsilon)}\right)^{\frac{1}{j-1}} \\
&= \left(\frac{1}{1 + (j-1)\varepsilon}\right)^{\frac{1}{j-1}} \\
&= \exp\left(-\frac{\log(1+(j-1)\varepsilon)}{j-1}\right) \\
&\geq \mathrm{e}^{-\varepsilon}.
\end{aligned}$$

Consider now non-consecutive $j < j'$. Assume $p_{j'} = (1-\varepsilon)p_j$, and for $j \leq r \leq j'-1$ define ε_r such that $(1-\varepsilon_r) = \frac{p_{r+1}}{p_r}$. This means that $(1-\varepsilon) = \prod_{r=j}^{j'-1}(1-\varepsilon_r)$. Now, we have that

$$\frac{t_j^*}{t_{j'}^*} = \prod_{r=j}^{j'-1} \frac{t_r^*}{t_{r+1}^*} \geq \exp\left(-\sum_{r=j}^{j'-1} \varepsilon_r\right) \geq \prod_{r=j}^{j'-1}(1-\varepsilon_r) = (1-\varepsilon).$$

To bound the difference between the probabilities of selecting colors j and j' when $p_j \geq p_{j'} \geq (1-\varepsilon)p_j$, note first that conditional on that the algorithm does not stop before $t_{j'}^*$, it stops with either of the two colors with equal probability. Thus, the difference is the probability that the algorithm stops with color j in the interval $[t_j^*, t_{j'}^*]$. Now, this is upper bounded by the probability that the best candidate of color j from those that arrive in $[0, t_{j'}^*]$ arrives in $[t_j^*, t_{j'}^*]$, which is at most

$$\frac{t_{j'}^* - t_j^*}{t_{j'}^*} = 1 - \frac{t_j^*}{t_{j'}^*} \leq 1 - (1-\varepsilon) = \varepsilon,$$

Concluding the proof of the theorem.

□

To exemplify the conclusion of the last theorem consider that we have two colors, say men and women, and that the prior is such that the top candidate is a woman with probability 60% and a man with probability 40%. This translates into having $\varepsilon = 1/3$ in the statement of the theorem, which implies that the algorithm will pick a woman at most 33% more often than a man. See Section 3.7 for more examples and empirical validations of these results.

To wrap up the section, observe that for the case of equal probabilities, i.e. $\mathbf{p} = (1/k, \ldots, 1/k)$, Corollary 3.4 and Theorem 3.10 imply that Algorithm 2 is $1+o(1)$-competitive and $1+o(1)$-fair. Unfortunately, these two properties cannot be simultaneously achieved for a general $\mathbf{p}$. Indeed, consider an instance where $p_1 = 1/\sqrt{k}$ and $p_\mathrm{i} = (1-p_1)/(k-1)$ for all $2 \leq \mathrm{i} \leq k$. Let ALG be an α-fair algorithm. Its success probability is

$$\sum_{\mathrm{i}=1}^{k} p_\mathrm{i} \cdot \mathbf{P}(ALG \text{ selects the best of color i}) \leq \sum_{\mathrm{i}=1}^{k} \alpha p_\mathrm{i}^2 = \frac{\alpha}{k} + \alpha\frac{(1-p_1)^2}{k-1} \leq \frac{2\alpha}{k}\,.$$

On the other hand, the optimal offline algorithm always selects from color 1, and therefore gets the best candidate with probability $p_1 = 1/\sqrt{k}$. So the competitive ratio of any α-fair algorithm is not better than $\sqrt{k}/(2\alpha)$. In particular, if we want an algorithm that is β-competitive and γ-fair, then $\max\{\beta, \gamma\} \geq k^{1/4}/2$.

3.7 Empirical Evaluation

In this section we empirically validate our results on synthetical and real-world experiments[1]. We compare our algorithm (Algorithm 2) with the following two baselines, which are based on the optimal solution to the classic secretary problem [100, 55, 68]:

1. Secretary algorithm (SA): This algorithm first computes the maximum value in the first 1/e-fraction of the stream, and then picks any element with higher value afterwards. This algorithm does not consider the colors of elements.
2. Single-color secretary algorithm (SCSA): This algorithm first picks a color proportional to the $\mathbf{p}$ values, and then runs the secretary algorithm on the elements of that color. This algorithm does not consider the elements whose color is different from the chosen one.

For all the experiments in this section, we run all the algorithms $20,000$ times. We report the number of times that i) the algorithm selects an element from each of the colors, ii) the number of times the selected element has the highest value in its color.

Synthetic dataset, equal p values. In this experiment, we create a synthetic dataset as follows. There are four colors with $10, 100, 1000$, and 10000 occurrences. The value of each element is chosen independently and uniformly at random from $[0, 1]$, so the p values are the same for all the colors, i.e., $\mathbf{p} = (1/4, 1/4, 1/4, 1/4)$. In Figure 3.1 ($a$), we present the result for this dataset. We observe that our algorithm and SCSA pick almost equal number of times from each color,[2] while SA picks almost only from the forth color. Therefore both our algorithm and SCSA are fair while SA does not satisfy the fairness expectations. We also observe that the number of elements picked by our algorithm is 1.305 times higher than in SCSA (+30.5%), and it picks the maximum element of the color and hence the best element overall 1.721 times more often than SCSA (+73.1%). Therefore the quality of the solution of our algorithm is significantly higher than that of SCSA.

Synthetic dataset, general p values. In this experiment, we create a synthetic dataset with four colors of sizes $10, 100, 1000, 10000$ and $\mathbf{p} = (0.3, 0.25, 0.25, 0.2)$. The results are presented in Figure 3.1 (b). We observe that both the distributions of the picked element for our algorithm and SCSA is close to the p distribution while for SA it is clearly different. Moreover, our algorithm performs significantly better than SCSA since it picks 1.309 times more elements (+30.9%) and 1.630 times more maximum element of the picked color (+63.0%).

[1] An implementation of these experiments is available at `https://github.com/google-research/google-research/tree/master/fairness_and_bias_in_online_selection`.

[2] The slight difference is due to the random nature of the algorithm.

Feedback maximization. We consider a dataset containing one record for each phone call by a Portuguese banking institution [103]. The goal of this experiment is to select a client and contact them and ask for their feedback. In order to achieve high quality feedback, we want to maximize the length of the call while being fair with respect to the age of the interviewee. We divide the clients into 5 colors: under 30, 31-40, 41-50, 51-60, and more that 61 years old. For the sake of being fair, we let $\mathbf{p} = (1/5, 1/5, 1/5, 1/5, 1/5)$. In Figure 3.1 (c), we present the obtained results (along with the number of the records in the input for each color). Similar to the previous experiments, we observe that our algorithm and SCSA pick almost equal number of the times from each color while SA picks mostly (80% of the runs) from the forth color. Morevoer, we observe that our algorithm picks 1.347 times more elements than SCSA (+34.7%), and that it picks the maximum element of the color 1.760 times more often (+76.0%).

Influence maximization. We consider a dataset containing the influence of the users of the Pokec social network [116]. The influence is computed as the number of the followers for each user. Selecting influencers has numerous applications, e.g., in advertising. In this experiment we want to be fair with respect to the body mass index (BMI) of the selected influencers. Therefore we divide the users into 5 colors according to their BMI: under weighted, normal, over weighted, obese type 1, and obese type 2. We let $\mathbf{p} = (1/5, 1/5, 1/5, 1/5, 1/5)$. The results are presented in Figure 3.1 (d).[3] Similar to the previous experiments our algorithm and SCSA picks almost equal number of each color while the Secretary algorithm picks only from two colors. Moreover, we observe that our algorithm picks 1.373 times more elements than SCSA (+37.3%), and picks the maximum element of the color 1.756 time more often than SCSA (+75.6%).

3.8 Sample-Driven Multi-Color Secretary Problem

In this section we formulate a sample-driven version of our multi-color secretary problem, along the lines of the ROSp model of Chapter 2. In this problem, again we have n candidates, partitioned into k groups $C = \{C_1, \ldots, C_k\}$, of sizes $\mathbf{n} = (n_1, \ldots, n_k)$. Each of the n candidates is placed in the set S independently with a given probability p, and otherwise she is placed in the set V. We get to observe all candidates in S, this is, their colors, and their relative ranks within their colors. Then, we observe one by one in uniformly random order the candidates in V. For each new candidate, we observe her color, and her relative rank within the candidates of the same color we have already observed (including those in S). We are given prior probabilities $(q_1, \ldots, q_k)$, where q_j is the probability that the best candidate of V is of color j. We want to maximize the probability of selecting the best candidate of V. In other words, if we denote by V_j the set of candidates of color j in V, then we want to choose an algorithm ALG that maximizes

$$\sum_{j=1}^{k} q_j \cdot \mathbb{P}(ALG \text{ selects best of } V_j),$$

which we call the success probability of ALG. To avoid some technical issues, when $V_j = \emptyset$ we interpret "ALG selects the best of V_j" as true. Notice that for fixed p, as n_j grows, the

[3] For ease of representation, this experiment is ran 10^6 times.

probability that $V_j = \emptyset$ decays exponentially fast to zero.

We extend the definition of GroupThresholds, combining it with the algorithm in Chapter 2. We take the perspective of uniform arrival times in $[0,1]$, where candidates that arrive before p are in S. For thresholds $\mathbf{t} = (t_{j,\ell})_{j=1,\ell=1}^{k,\infty}$ in $[0,1]$, we define SD-GroupThresholds($\mathbf{t}$) as the algorithm that, upon observing a candidate of color j at time τ that is best so far in V and has relative rank ℓ, accepts her if $t_{j,\ell} \leq \tau$.

As for the regular version of the algorithm, we can show that its success probability decreases with n_j, for all j.

Lemma 3.12. *For fixed p, $(q_1, \ldots, q_k)$, and $\mathbf{t}$, the success probability of the algorithm* SD-GroupThresholds($\mathbf{t}$) *is decreasing with n_j, for all $j = 1 \ldots k$.*

Proof. We take an instance with sizes $\mathbf{n} = (n_1, \ldots, n_k)$ and increase it in one candidate. W.l.o.g. assume the new candidate is of color 1, so we get an instance $\mathbf{n}' = (n_1+1, n_2, \ldots, n_k)$. We couple the decisions of the algorithm in both instances by first drawing the arrival times of the best n_j candidates of each color j, and separately the arrival time of the (n_1+1)-th candidate of color 1. We argue that in a realization of the arrival times where the algorithm fails with $\mathbf{n}$, it also fails with $\mathbf{n}'$. In fact, for such a realization, in the smaller instance $V_j \neq \emptyset$ for all j, and for some j, the algorithm selects a candidate that is not the best of V_j, or never stops. Adding the (n_1+1)-th candidate of color 1 does not affect the relative ranks of the rest, so the only different action the algorithm could do in the larger instance would be to select the (n_1+1)-th candidate of color 1. But V_1 was nonempty before adding this candidate, so she cannot be the best of V_1. Therefore, the algorithm also fails in the larger instance. □

Now, similar to Lemmas 2.5 and 3.9, we can calculate the limit success probability of SD-GroupThresholds($\mathbf{t}$).

Theorem 3.13. *The limit success probability of* SD-GroupThresholds($\mathbf{t}$) *is*

$$\sum_{j=1}^{k} q_j \int_p^1 \sum_{\ell: t_{j,\ell} \leq \tau} p^{\ell-1} \cdot \left(1 - \sum_{s \geq \ell : t_{j,s} \leq \tau} \left(\frac{p}{\tau}\right)^{s-\ell} \left(\frac{\tau - \max\{p, t_{j,s}\}}{\tau}\right)\right)$$
$$\cdot \prod_{j' \neq j} \left(1 - \sum_{s: t_{j',s} \leq \tau} \left(\frac{p}{\tau}\right)^{s-1} \left(\frac{\tau - \max\{p, t_{j',s}\}}{\tau}\right)\right) \mathrm{d}\tau. \tag{3.3}$$

Proof. Within this proof we refer to SD-GroupThresholds($\mathbf{t}$) be simply ALG. For a given color j, we want to calculate $\mathbb{P}(ALG \text{ selects best of } V_j)$. We say a candidate is acceptable if the algorithm would accept her if it had not stopped when she arrived. For a given time $\tau \in [p,1]$, we define $N_{j,\tau}$ as the event that no candidate of color j that arrives in $[p,\tau)$ is acceptable. Note that $N_{j,\tau}$ only depends on the arrival times of candidates of color j, so these events are independent across colors. Now, in the limit, V_j is nonempty, so we can condition on the arrival time of the best in V_j. Thus,

$$
\begin{aligned}
&\mathbb{P}(ALG \text{ selects best of } V_j) \\
&= \frac{1}{1-p}\int_p^1 \mathbb{P}(ALG \text{ selects best of } V_j \,|\, \text{best of } V_j \text{ arrives at } \tau)\,\mathrm{d}\tau \\
&= \frac{1}{1-p}\int_p^1 \mathbb{P}\left(\cap_{j'=1}^k N_{j',\tau} \text{ and best of } V_j \text{ is acceptable} \,\middle|\, \text{best of } V_j \text{ arrives at } \tau\right)\,\mathrm{d}\tau \\
&= \frac{1}{1-p}\int_p^1 \mathbb{P}\left(N_{j,\tau} \text{ and best of } V_j \text{ is acceptable} \,|\, \text{best of } V_j \text{ arrives at } \tau\right)\cdot \prod_{j'\neq j}\mathbb{P}(N_{j',\tau})\,\mathrm{d}\tau \\
&= \int_p^1 \sum_{\ell=1}^{\infty} p^{\ell-1}\cdot \mathbb{P}\left(N_{j,\tau} \text{ and best of } V_j \text{ is acceptable} \,|\, \text{best of } V_j \text{ arrives at } \tau \text{ and has rank } \ell\right) \\
&\quad \cdot \prod_{j'\neq j}\mathbb{P}(N_{j',\tau})\,\mathrm{d}\tau
\end{aligned}
$$

$$
\begin{aligned}
&= \int_p^1 \sum_{\ell=1}^{\infty} p^{\ell-1}\cdot \mathbb{1}_{t_{j,\ell}\le\tau}\cdot \mathbb{P}\left(N_{j,\tau} \,|\, \text{best of } V_j \text{ arrives at } \tau \text{ and has rank } \ell\right)\cdot \prod_{j'\neq j}\mathbb{P}(N_{j',\tau})\,\mathrm{d}\tau \\
&= \int_p^1 \sum_{\ell=1}^{\infty} p^{\ell-1}\cdot \mathbb{1}_{t_{j,\ell}\le\tau}\cdot \left(1-\sum_{s=\ell}^{\infty}\mathbb{1}_{t_{j,s}\le\tau}\left(\frac{p}{\tau}\right)^{s-\ell}\left(\frac{\tau-\max\{p,t_{j,s}\}}{\tau}\right)\right) \\
&\quad \cdot \prod_{j'\neq j}\left(1-\sum_{s=1}^{\infty}\mathbb{1}_{t_{j',s}\le\tau}\left(\frac{p}{\tau}\right)^{s-1}\left(\frac{\tau-\max\{p,t_{j',s}\}}{\tau}\right)\right)\mathrm{d}\tau.
\end{aligned}
$$

Moving the indicator functions to the conditions of the sums we obtain the formula of the lemma. □

Computation of the optimal thresholds. Unfortunately, Equation (3.3) is not a separable function, like the success probability of the analogous algorithm in Chapter 2. However, for fixed $p < 1$ and given $\varepsilon > 0$ it is possible to approximate the optimal solution numerically, by setting $t_{j,\ell} = 1$ for $\ell > \log_p(\varepsilon) = O(1/\varepsilon(1-p))$, and optimizing over the finitely-many remaining thresholds. Doing this results in a reduction in the success probability that is not larger than $\sum_{\ell>\log_p(\varepsilon)}(1-p)p^{\ell-1} = O(\varepsilon)$.

3.9 Conclusion and Open Problems

In this work, we explored questions of fairness and bias in a natural multi-color variant of a canonical problem of online selection. We designed the optimal fair online algorithm for this problem, and validated its efficacy and fairness on synthetic and real-world data.

As in many real-world settings the online decisions go beyond the single selection model studied here, there is ample opportunity for extending this line of work to combinatorial settings. We expect that building on the respective lines of work in the secretary, prophet and optimal stopping literature in general, could prove very fruitful.

Particularly exciting directions include an extension to matching problems [87, 58, 72], allocation problems with matroid structure [9, 65, 90, 49], or even general combinatorial allocation problems [61, 51].

Another interesting direction is to better understand the sample-driven variant presented here. We believe that the class SD-GroupThresholds should contain the optimal algorithm, and proving this is an open question. It would also be interesting to investigate whether the optimal algorithm for this version also satisfies the good fairness properties shown for GroupThresholds.

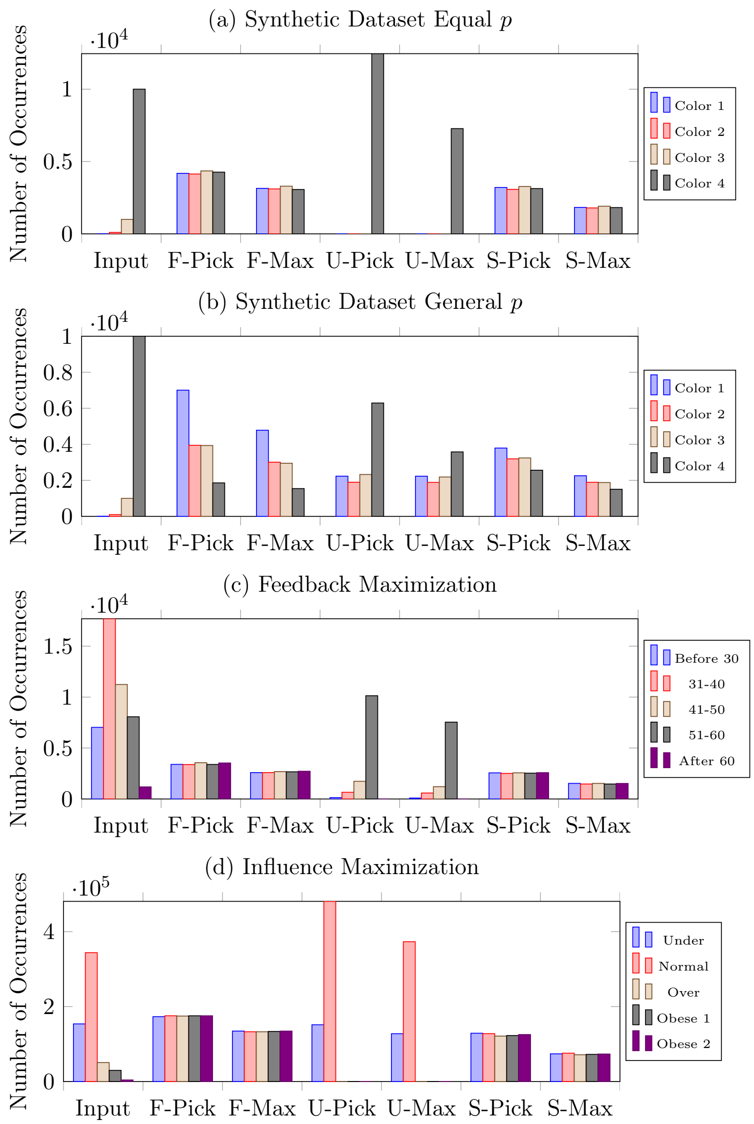


Figure 3.1: In this plot, we compare our fair secretary algorithm with the secretary algorithm (SA) and the single-color secretary algorithm (SCSA) on (a) synthetic dataset, equal **p** values, (b) synthetic dataset, general **p** values, (c) feedback maximization dataset, and (d) influence maximization dataset. Here Input is the number of elements from each color in the input, F-Pick and F-Max are the number of elements picked by our fair secretary algorithm and the number of them that are the maximum among the elements of that color. Similarly, U-Pick (S-Pick) and U-Max (S-Max) are the number of elements picked by SA and SCSA and the number of them that are the maximum among the elements of that color.

Chapter 4

Optimal Item Pricing in Online Combinatorial Auctions

In a combinatorial auction, a set of valuable items is to be allocated among a set of interested agents. Who should get which items in order to maximize the social welfare? This is a fundamental economic question, and a ubiquitous allocation mechanism is to simply set a price for each item and let the agents buy their preferred subset of items under those prices. The study of these mechanisms dates back to the investigations of Leon Walras over a century ago, and is closely related to the notion of Walrasrian equilibrium. Understanding the existence and approximation of Walrasrian equilibrium and related notions under pricing mechanisms has been an active area of research in recent years [11, 60, 62, 12, 109].

In this chapter, we consider a generalization of the prophet inequality in the direction of combinatorial auctions: instead of selling a single item to a stream of buyers, we have several heterogeneous items on sale. We follow the approach of online combinatorial auctions and study the welfare achieved by posted-price mechanisms in a very general setup. Specifically, our mechanisms post a price p_{i} on each item i. Then, buyers with randomly-drawn arbitrary monotone valuations over the subsets of items arrive in arbitrary order, and upon arrival pick their preferred subset among those items that are left (at the posted prices). Of course, in this generality little can be said about the social welfare induced by posted-price mechanisms, so it is common to parametrize the instances by d, the largest size of a set a buyer might be interested in. This parametrization is interesting from a combinatorial perspective: finding a socially optimal allocation is NP-hard already when $\mathrm{d} \geq 3$, and even hard to approximate [117]. Moreover, if we restrict the buyers' valuations to be deterministic and single-minded,[1] we recover the classic hypergraph matching problem.

Our main result in this chapter is to determine the tight approximation guarantee of item pricing as a function of d. Specifically, we prove that there always exists a posted-price mechanism such that the expected welfare of the resulting allocation when adversarial-order buyers iteratively purchase their preferred set (at the posted prices) is at least a $1/(\mathrm{d}+1)$ fraction of the expected welfare of an optimal allocation (Theorem 4.1). Furthermore, we

[1]That is, each buyer has a fixed set T, and values a set S at a certain positive amount if $T \subseteq S$, and at 0 otherwise.

prove this bound is tight (Proposition 4.5).

Interestingly, our result generalizes and/or improves upon several results in the literature, which we now provide context for.

As in the classic prophet inequality, a major drawback of this model in a real-world setup is that we usually don't have enough data to learn the distributions, and decision-makers are reluctant to settle on specific prior distributions. We analyze the case where we only have sample access to the distributions, and show that it is possible to compute prices that achieve a $1/(\mathrm{d}+1+\varepsilon)$-approximation in polynomial time, using a polynomial number of samples.

4.1 Context and Related Work

Posted-price Mechanisms. Posted-price mechanisms are ubiquitous within economics and computation owing to their simplicity. They are commonly used as subroutines in truthful mechanisms that approximately maximize welfare [47, 95, 46, 4, 3]. They are also used as subroutines in simple mechanisms to approximately maximize revenue in Bayesian settings [28, 90, 29, 20]. Our work considers the same model initiated by [60] (welfare maximization in Bayesian settings). Other works consider restrictions on the valuations, such as subadditive [51], while others consider the unrestricted case [50]. Our results contribute to this line of work by providing the tight approximation guarantee of posted-price mechanisms in this model for unrestricted valuations over sets of size at most d. In particular, our results improve the bound of $1/(4\mathrm{d}-2)$ given in [50] to $1/(\mathrm{d}+1)$, which is tight.

Prophet Inequalities. When there is a single item (and thus $\mathrm{d} = 1$) our problem is equivalent to the single-item prophet inequality and thus our result takes the same form as the classic result of Samuel-Cahn [111], who proved that the optimal prophet inequality (whose factor is $1/2$) can be achieved with a single threshold. A special case of our problem when buyers are single-minded corresponds to various multiple-choice prophet inequality settings, and our results improve upon the state-of-the-art. In particular, all prophet inequalities deduced from our main result are *non-adaptive*: for each element e, a threshold T_{e} is set at the beginning of the algorithm. Element e is accepted if and only if $w_{\mathrm{e}} \geq T_{\mathrm{e}}$ (and it is feasible to accept e).

When $\mathrm{d} = 2$ and buyers are single-minded, our problem translates into the matching prophet inequality problem. Our results when $\mathrm{d} = 2$ therefore extend the $1/3$-approximation of Gravin and Wang [73] from bipartite to general graphs. Note that recent work of [59] provides a .337-approximation in this case, although it sets thresholds adaptively.

For arbitrary d when buyers are single-minded, our problem translates into the d-dimensional hypergraph prophet inequality, which generalizes the prophet inequality problem over the intersection of d partition matroids. Here, a $1/(4\mathrm{d}-2)$-approximation was first given in [90], and improved to $1/(e(\mathrm{d}+1))$ in [63]. A corollary of our main result improves this to $1/(\mathrm{d}+1)$, and with non-adaptive thresholds. A lower bound of [90] proves that it is not possible to achieve an $\omega(1/\sqrt{\mathrm{d}})$ approximation even for this special case, but it remains an open problem to determine the tight ratio for prophet inequalities for the intersection of d partition matroids (and for the d-dimensional hypergraph prophet inequality).

4.2 A Technical Highlight and Additional Results

The proof of our main result breaks down the expected welfare into the "revenue" and "utility" achieved by setting prices, and searches for properly "balanced thresholds" as in [90, 60, 73, 50]. In particular, we target prices that are "low enough" so that a buyer with high value for some set will choose to purchase it, yet also "high enough" so that the revenue gained when a bidder purchases items they should not receive in the optimal allocation compensates for the lost welfare. In comparison to prior work using a similar approach, the conditions that guarantee such prices are more involved, and we prove their existence using Brouwer's fixed point theorem.

As our proof makes use of Brouwer's fixed point theorem, it is inherently non-constructive. We however show in Section 4.5 how to compute our prices in polynomial time. Moreover, we show how to compute the prices when we only have access to a polynomial number of samples of the distributions, instead of the distributions themselves. Our approach makes use of a configuration LP relaxation to cope with the APX-hardness of optimizing welfare, and a convex optimization formulation to find our fixed point.

In Section 4.6, we consider the special case that arises when valuations are deterministic and buyers are single-minded. In this situation the welfare optimization problem corresponds to matching in a hypergraph with edges of size at most d. So the problem of finding item prices boils down to finding a set of thresholds, one for each vertex, such that the value of the solution in which hyperedges arrive sequentially (in any order) and are greedily included in the solution when their weight is higher that the sum of the corresponding vertex thresholds, is as close as possible to the optimal solution. For the case of standard matching ($\mathrm{d} = 2$) we prove that there exist prices guaranteeing a factor of $1/2$ of the optimal solution and that there do not exist prices guaranteeing a factor better than $2/3$. The tight factor is left as an open problem. More generally, we prove that there are prices obtaining a fraction $1/\mathrm{d}$ of the optimal solution (thus slightly improving our general $1/(\mathrm{d}+1)$), and that it is not possible to do better than $\sim 1/\sqrt{\mathrm{d}}$.

Summary and Roadmap. We precisely define our model in Section 4.3. Section 4.4 presents our main result: a posted-price mechanism that achieves a $1/(\mathrm{d}+1)$-approximation to the optimal expected welfare, when buyers have arbitrary monotone valuations and are interested in sets of size at most d. Recall that this approximation guarantee is tight (we provide a simple example witnessing this in Proposition 4.5). In Section 4.5, we show how to compute our desired prices in polynomial time and using sample-access to the distributions. In Section 4.6, we consider the special case where the distributions are point-masses.

4.3 Model

In our basic model, we have a (multi)set of items M in which there are $k_j \geq 1$ copies of each item $j \in M$.[2] The set of buyers, denoted by N, arrive sequentially (in arbitrary order) and buy some of those items. Each buyer $\mathrm{i} \in N$ has a valuation function $v_\mathrm{i} : 2^M \to \mathbb{R}_{\geq 0}$, which is randomly and independently chosen according to a given distribution $\mathcal{F}_\mathrm{i}$ (defined over a set of possible valuation functions). As it is standard, we assume that each possible realization of each v_i is monotone (i.e., $A \subseteq B \Rightarrow v_\mathrm{i}(A) \leq v_\mathrm{i}(B)$). We parametrize an instance of the problem by d, the size of the largest set a buyer might be interested in. Thus, if $A \subseteq M$ is such that $|A| > \mathrm{d}$, then

$$v(A) = \max_{B \subseteq A, |B| = \mathrm{d}} v(B) \tag{4.1}$$

Note that while there are $k_j \geq 1$ copies of each item $j \in M$, no bidder achieves value from additional copies (and therefore, without loss of generality, bidders cannot buy more than one copy).

In this chapter, we are interested in exploring the limits of using item prices as a mechanism to assign items to buyers. In a pricing mechanism, we set item prices $p \in \mathbb{R}^M_{\geq 0}$ and then consider an arbitrary arrival order of the buyers (note different copies of the same item must have the same price). Thus, buyer i buys the set of remaining items according to

$$\max_{A \subseteq R_\mathrm{i}} \left(v_\mathrm{i}(A) - \sum_{j \in A} p_j \right), \tag{4.2}$$

where R_i denotes the items for which there remains an unsold copy when i arrives. Note that (4.2) might be solved by $A = \emptyset$, i.e., buyer i might opt not to buy anything. When there is a tie between different sets, the buyer can choose arbitrarily, implying that our results need to be valid even for the worst-case tiebreaking.[3]

More precisely, if σ is the arrival order of the buyers, so that buyer i comes at time $\sigma(\mathrm{i})$, then buyer i gets the set $B_\mathrm{i}(\sigma) = \arg\max_{A \subseteq R_\mathrm{i}(\sigma)} \left(v_\mathrm{i}(A) - \sum_{j \in A} p_j \right)$, where $R_\mathrm{i}(\sigma) = \{ j \in M : k_j > |\{ \ell \in N : \sigma(\ell) < \sigma(\mathrm{i}) \text{ and } j \in B_\ell(\sigma) \}| \}$. With this, given an instance of the problem (determined by M, k_j for all $j \in M$, N, and $\mathcal{F}_\mathrm{i}$ for all $\mathrm{i} \in N$), the quality measure of a price vector $p \in \mathbb{R}^M_{\geq 0}$ is the worst case (over the arrival orders) expected (over the valuations) welfare of the allocation it induces. Denoting this quantity by $ALG(p)$ we have that:

$$ALG(p) := \min_\sigma \mathbf{E} \left(\sum_{\mathrm{i} \in N} v_\mathrm{i}(B_\mathrm{i}(\sigma)) \right).$$

On the other hand, the benchmark we compare to throughout the chapter is the expected value of the optimal welfare-maximizing allocation, OPT, formally defined as

$$OPT := \mathbf{E} \left(\max_{\{A_\mathrm{i}\}_{\mathrm{i} \in N}} \left\{ \sum_{\mathrm{i} \in N} v_\mathrm{i}(A_\mathrm{i}) : \text{ s.t. } |\{ \mathrm{i} \in N : j \in A_\mathrm{i} \}| \leq k_j, \text{ for all } j \in M \right\} \right).$$

[2] Throughout the chapter M is actually a set and refers to the set of different items.

[3] In some of the constructions in Section 4.6 we break ties conveniently but all the results hold by slightly tweaking the instances.

We denote by OPT_{i} the random set that buyer i gets in an optimal allocation.

In Section 4.6 we consider the special case of our problem in which

(i) valuations are deterministic,
(ii) there is a single copy of each item (i.e., $k_j = 1$ for all $j \in M$), and
(iii) buyers are *single-minded*, i.e., each buyer i has a set A_{i}, with $|A_{\mathrm{i}}| \leq \mathrm{d}$, such that $A_{\mathrm{i}} \nsubseteq B \Rightarrow v_{\mathrm{i}}(B) = 0, A_{\mathrm{i}} \subseteq B \Rightarrow v_{\mathrm{i}}(B) = v_{\mathrm{i}}(A_{\mathrm{i}})$.

Interestingly, already in this particular setup, the problem of maximizing the welfare of an allocation corresponds to the classic NP-hard combinatorial optimization problem of hypergraph matching with hyperedges of size at most d. Indeed, in an optimal allocation buyer i either gets A_{i} or $\emptyset$, implying that maximizing the (now deterministic) welfare of the allocation is equivalent to finding a subset of pairwise disjoint A_{i}'s of maximum total valuation.

4.4 Main Result: A $1/(\mathrm{d}+1)$-approximation for Random Valuations

In this section we prove there exists a vector of item prices such that the resulting allocation yields in expectation at least a $1/(\mathrm{d}+1)$ fraction of the optimal social welfare. Additionally, we show that this bound is tight.

Theorem 4.1. *There exists a vector of prices $p \in \mathbb{R}^M_{\geq 0}$ such that*

$$(\mathrm{d}+1) \cdot ALG(p) \geq OPT.$$

To prove the theorem we will make use of the following function. For each $A \subseteq M$ and $\mathrm{i} \in N$, we define $z_{\mathrm{i},A} : \mathbb{R}^M_{\geq 0} \to \mathbb{R}$ as

$$z_{\mathrm{i},A}(p) := \mathbf{E}\left(\mathbb{1}_{OPT_{\mathrm{i}}=A} \cdot \left[v_{\mathrm{i}}(A) - \sum_{j\in A} p_j\right]_+\right),$$

where $[x]_+$ denotes $\max\{x, 0\}$. The function $z_{\mathrm{i},A}(p)$ can be interpreted as follows: imagine we calculate the optimal allocation and offer buyer i the set OPT_{i} at the prices given by p. Then, $z_{\mathrm{i},A}$ would be the contribution of the set A to the non-negative part of the expected utility of buyer i.

We assume without loss of generality that $|OPT_{\mathrm{i}}| \leq \mathrm{d}$ for all $\mathrm{i} \in N$, so $z_{\mathrm{i},A}(p) = 0$ if $|A| > \mathrm{d}$. We start by showing a lower bound for $ALG(p)$ in terms of the values $z_{\mathrm{i},A}(p)$.

Lemma 4.2. *For any vector of prices $p \in \mathbb{R}^M_{\geq 0}$,*

$$ALG(p) \geq \min_{C\subseteq M}\left\{\sum_{j\notin C} k_j \cdot p_j + \sum_{\mathrm{i}\in N}\sum_{A\subseteq C} z_{\mathrm{i},A}(p)\right\}.$$

Proof. In this proof we assume the arrival order σ is arbitrary, and for simplicity we denote $B_{\text{i}}(\sigma)$ and $R_{\text{i}}(\sigma)$ simply by B_{i} and R_{i}. We separate the welfare of the resulting allocation into *revenue* and *utility*, i.e., we separate $\sum_{\text{i}\in N} v_{\text{i}}(B_{\text{i}})$ into

$$\text{Revenue} = \sum_{\text{i}\in N}\sum_{j\in B_{\text{i}}} p_j \quad \text{and} \quad \text{Utility} = \sum_{\text{i}\in N}\left(v_{\text{i}}(B_{\text{i}}) - \sum_{j\in B_{\text{i}}} p_j\right).$$

Recall that R_{i} is the set of items with remaining copies when i arrives. Similarly, denote by R the set of items that have remaining copies by the end of the process. Note first that

$$\mathbf{E}(\text{Revenue}) \geq \mathbf{E}\left(\sum_{j\notin R} k_j \cdot p_j\right).$$

This is simply because each item $j \notin R$ has had all k_j copies purchased. As for the utility, for any $\text{i} \in N$, by the definition of B_{i} it holds that

$$v_{\text{i}}(B_{\text{i}}) - \sum_{j\in B_{\text{i}}} p_j = \max_{A\subseteq R_{\text{i}}} v_{\text{i}}(A) - \sum_{j\in A} p_j$$

Note now that v_{i} and R_{i} are independent. Let $(\tilde{v}_{\text{i}})_{\text{i}\in N}$ be independent realizations of the valuations, and $\widetilde{OPT}_{\text{i}}$ the corresponding optimal solution. With this, and noting that $R \subseteq R_{\text{i}}$, we can rewrite the expected utility of agent i as

$$\mathbf{E}\left(\max_{A\subseteq R_{\text{i}}} v_{\text{i}}(A) - \sum_{j\in A} p_j\right) = \mathbf{E}\left(\max_{A\subseteq R_{\text{i}}} \tilde{v}_{\text{i}}(A) - \sum_{j\in A} p_j\right) \geq \mathbf{E}\left(\max_{A\subseteq R} \tilde{v}_{\text{i}}(A) - \sum_{j\in A} p_j\right). \tag{4.3}$$

We replace the maximization over subsets of R with a particular choice, $\widetilde{OPT}_{\text{i}}$, whenever it is contained in R and gives positive utility (otherwise we take $\emptyset$). This obtains the following lower bound on the expected utility of agent i:

$$\begin{aligned}
\mathbf{E}\left(\mathbb{1}_{\{\widetilde{OPT}_{\text{i}}\subseteq R\}} \cdot \left[\tilde{v}_{\text{i}}(\widetilde{OPT}_{\text{i}}) - \sum_{j\in\widetilde{OPT}_{\text{i}}} p_j\right]_+\right) &= \mathbf{E}\left(\sum_{A\subseteq R} \mathbb{1}_{\{\widetilde{OPT}_{\text{i}}=A\}} \cdot \left[\tilde{v}_{\text{i}}(A) - \sum_{j\in A} p_j\right]_+\right) \\
&= \mathbf{E}\left(\sum_{A\subseteq R} \mathbf{E}\left(\mathbb{1}_{\{\widetilde{OPT}_{\text{i}}=A\}} \cdot \left[\tilde{v}_{\text{i}}(A) - \sum_{j\in A} p_j\right]_+\right)\right) \\
&= \mathbf{E}\left(\sum_{A\subseteq R} z_{\text{i},A}(p)\right). \qquad (4.4)
\end{aligned}$$

Summing over all agents, we get that

$$\mathbf{E}(\text{Utility}) \geq \mathbf{E}\left(\sum_{\text{i}\in N}\sum_{A\subseteq R} z_{\text{i},A}(p)\right).$$

Therefore, adding the revenue and the utility we get that

$$ALG(p) \geq \mathbf{E}\left(\sum_{j \notin R} k_j \cdot p_j + \sum_{\mathrm{i} \in N} \sum_{A \subseteq R} z_{\mathrm{i},A}(p)\right).$$

Replacing the expectation over R with a minimization over subsets of M yields the bound of the lemma. □

Lemma 4.3. *For any vector of prices $p \in \mathbb{R}^M_{\geq 0}$,*

$$OPT \leq \sum_{j \in M} k_j \cdot p_j + \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} z_{\mathrm{i},A}(p).$$

Proof. We have that OPT equals

$$\sum_{\mathrm{i} \in N} \mathbf{E}(v_{\mathrm{i}}(OPT_{\mathrm{i}})) = \mathbf{E}\left(\sum_{\mathrm{i} \in N} \sum_{j \in OPT_{\mathrm{i}}} p_j\right) + \sum_{\mathrm{i} \in N} \mathbf{E}\left(v_{\mathrm{i}}(OPT_{\mathrm{i}}) - \sum_{j \in OPT_{\mathrm{i}}} p_j\right).$$

Now we upper bound these two terms separately. Note that in the first term each item $j \in M$ appears at most k_j times, so

$$\mathbf{E}\left(\sum_{\mathrm{i} \in N} \sum_{j \in OPT_{\mathrm{i}}} p_j\right) \leq \sum_{j \in M} k_j \cdot p_j.$$

For the second part, we upper bound with the positive part of the difference, and sum over all possible realizations of OPT_{i}:

$$\begin{aligned}\sum_{\mathrm{i} \in N} \mathbf{E}\left(v_{\mathrm{i}}(OPT_{\mathrm{i}}) - \sum_{j \in OPT_{\mathrm{i}}} p_j\right) &= \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} \mathbf{E}\left(\mathbb{1}_{\{OPT_{\mathrm{i}} = A\}}\left(v_{\mathrm{i}}(A) - \sum_{j \in A} p_j\right)\right) \\ &\leq \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} z_{\mathrm{i},A}(p).\end{aligned}$$

Putting together the two upper bounds we obtain the bound on OPT. □

Lemmas 4.2 and 4.3 provide a similar form to lower bound $ALG(p)$ and upper bound OPT as a function of p. Now, we will prove the existence of a good choice of p where these bounds differ by at most a factor of $\mathrm{d} + 1$.

Lemma 4.4. *There exists a vector of prices $p \in \mathbb{R}^M_{\geq 0}$ such that for every $j \in M$ we have*

$$p_j = \frac{1}{k_j} \sum_{\mathrm{i} \in N} \sum_{A \subseteq M : j \in A} z_{\mathrm{i},A}(p).$$

Proof. The proof will be an application of Brouwer's fixed point theorem. Let K denote the compact set $K := [0, OPT]^M \subseteq \mathbb{R}^M_{\geq 0}$. We define a function $\psi : K \to K$ as follows: for a vector $p \in K$ and item $j \in M$, the j^{th} coordinate of ψ is

$$\psi_j(p) = \frac{1}{k_j} \sum_{\mathrm{i} \in N} \sum_{A \subseteq M : j \in A} z_{\mathrm{i},A}(p). \tag{4.5}$$

We prove now that ψ is a well-defined continuous function, from the compact set K into itself, and therefore it has a fixed point by Brouwer's fixed point theorem. Note that a fixed point of ψ is exactly the vector of prices we are looking for.

In fact, recall that we defined $z_{\mathrm{i},A}(p) = \mathbf{E}(\mathbb{1}_{OPT_{\mathrm{i}}=A} \cdot [v_{\mathrm{i}}(A) - \sum_{j \in A} p_j]_+)$, which is a nonincreasing function of p_j, for all $j \in M$. Moreover, note that since $[\cdot]_+$ is a convex function, $z_{\mathrm{i},A}$ is also a convex function of p_j for all $j \in M$. The monotonicity of $z_{\mathrm{i},A}$ implies that for all $p \in K$ and $j \in M$, $\psi_j(p) \leq \psi_j(0) \leq \frac{1}{k_j} OPT$, and therefore $\psi(p) \in K$ for all $p \in K$. The convexity of $z_{\mathrm{i},A}$ implies it is also continuous, so ψ is a continuous function.

We've now argued that ψ is a continuous function from K to itself, and therefore a fixed point exists, which proves the lemma. □

Proof of Theorem 4.1. Using the vector of prices from Lemma 4.4, we apply the bound of Lemma 4.2 and conclude

$$\begin{aligned}
ALG(p) &\geq \min_{C \subseteq M} \left\{ \sum_{j \notin C} k_j \cdot p_j + \sum_{\mathrm{i} \in N} \sum_{A \subseteq C} z_{\mathrm{i},A}(p) \right\}. \\
&\geq \min_{C \subseteq M} \left\{ \sum_{j \notin C} k_j \cdot \frac{1}{k_j} \sum_{\mathrm{i} \in N} \sum_{A \subseteq M : j \in A} z_{\mathrm{i},A}(p) + \sum_{\mathrm{i} \in N} \sum_{A \subseteq C} z_{\mathrm{i},A}(p) \right\} \\
&= \min_{C \subseteq M} \left\{ \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} z_{\mathrm{i},A}(p) \cdot (|A \setminus C| + \mathbb{1}_{A \subseteq C}) \right\} \\
&\geq \min_{C \subseteq M} \left\{ \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} z_{\mathrm{i},A}(p) \cdot (\mathbb{1}_{|A \setminus C| \geq 1} + \mathbb{1}_{A \subseteq C}) \right\} \\
&= \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} z_{\mathrm{i},A}(p).
\end{aligned}$$

For OPT, substituting our fixed point in the upper bound of Lemma 4.3 gives

$$\begin{aligned}
OPT &\leq \sum_{j \in M} \sum_{\mathrm{i} \in N} \sum_{A \subseteq M : j \in A} z_{\mathrm{i},A}(p) + \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} z_{\mathrm{i},A}(p) \\
&= \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} (|A| + 1) \cdot z_{\mathrm{i},A}(p) \\
&\leq (\mathrm{d} + 1) \sum_{\mathrm{i} \in N} \sum_{A \subseteq M} z_{\mathrm{i},A}(p).
\end{aligned}$$

Comparing the two bounds we see that $(\mathrm{d} + 1) \cdot ALG(p) \geq OPT$. □

To wrap up the section, we establish that the bound of Theorem 4.1 is best possible, by modifying a simple example of Feldman et al. [60]. The example provided by Feldman et al. establishes a lower bound of d if we restrict to deterministic valuations. Here we add stochasticity to match the bound of Theorem 4.1.

Proposition 4.5. *For all* d*, and all* $\delta > 0$*, there exists an instance on* $|N| = 2$ *buyers and* $|M| = \mathrm{d}$ *items such that for all* p*,* $ALG(p) \leq 1$*, yet* $OPT = \mathrm{d} + 1 - \delta$*.*

Proof. Consider a set M of exactly d items with a single copy of each, and a very small $\varepsilon > 0$. There are two buyers. The first buyer values any nonempty subset of the items at 1. The second buyer only assigns value to getting all d items, and this value is $\mathrm{d} - \varepsilon$ with probability $1 - \varepsilon$ and it is $1/\varepsilon$ with probability ε. In any instance where the first buyer purchases a non-empty subset, the resulting social welfare is 1. Note that this is certain to happen if we set the prices so that $\sum_{j \in M} p_j < \mathrm{d}$ and the first buyer arrives before the second. If, on the contrary, $\sum_{j \in M} p_j \geq \mathrm{d}$ and the first buyer does not purchase anything, the second buyer will only purchase items with probability ε. In this case, the expected total welfare is also 1. This establishes that $ALG(p) = 1$ for all p. Finally, it is clear that in this instance the optimal welfare is achieved by always assigning all items to the second buyer, which results in an expected welfare of $(\mathrm{d} - \varepsilon) \cdot (1 - \varepsilon) + \varepsilon \cdot (1/\varepsilon) \geq \mathrm{d} + 1 - (\mathrm{d} + 1)\varepsilon$. Setting $\varepsilon = \delta/(\mathrm{d} + 1)$ completes the proof. □

4.5 Efficient and Sample-Based Computation

So far, our main result is nonconstructive for several reasons. First, it requires a fixed-point computation (which is PPAD-hard in general). Second, evaluating the function for which we hope to find a fixed point requires computing the expected value of a random variable with exponential support (which is #P-hard in general). Finally, even sampling the random variable whose expected value defines our function requires computing the optimal allocation (which is NP-hard in general, even to approximate).

In this section, we show how to overcome all three barriers, and efficiently (in time polynomial in $|M|$ and $|N|$) compute the prices, even when d is not a constant. Notice that when d is not a constant, a complete description of the distributions, or even of a single deterministic valuation function, might be exponentially large. Thus, we assume instead that we can draw samples from the distributions of valuation functions, which we access in a black-box manner via *demand queries*.

Definition 4.6. A demand query of a valuation function $v : 2^M \to \mathbb{R}_{\geq 0}$ accepts a price vector $p \in \mathbb{R}^M_{\geq 0}$ and returns a subset of items A such that $v(A) = \max_{B \subseteq M} \left(v(B) - \sum_{j \in B} p_j \right)$.

We note that while a demand query only returns the subset A and not the associated valuation $v(A)$, we can compute the valuation of any subset using polynomially-many demand queries [106, Lemma 11.22].

Theorem 4.7. *If there is a number $v_{\max}$ such that $v_{\mathrm{i}}(A) \leq v_{\max}$ for all $\mathrm{i} \in N, A \subseteq M$ with probability 1, and such that $v_{\max}/OPT \leq \mathrm{poly}(|M|, |N|)$, then for every $\varepsilon > 0$ we can calculate prices $\hat{p}$ such that*

$$(\mathrm{d} + 1 + \varepsilon) \cdot ALG(\hat{p}) \geq OPT,$$

with probability $1 - \varepsilon$, in time $\mathrm{poly}(|M|, |N|, 1/\varepsilon)$, using $\mathrm{poly}(|M|, |N|, 1/\varepsilon)$ samples of the valuations and $\mathrm{poly}(|M|, |N|, 1/\varepsilon)$ demand queries in total.

To prove this theorem we first show that under the condition $v_{\max}/OPT \leq \mathrm{poly}(|M|, |N|)$ we can approximate the function ψ (as defined in Section 4.4) using polynomially-many samples. This approximation in principle requires computing the optimal allocation for each set of sampled valuations, which is in general intractable. However, as a second step, we show it is enough to solve a linear relaxation of OPT, which can be done in polynomial time. Finally, we show that the structure of ψ allows us to efficiently compute a fixed point through a convex quadratic program.

Even though $v_{\max}/OPT \leq \mathrm{poly}(|M|, |N|)$ is a seemingly strong condition, the following example illustrates its necessity in our approach. Consider an instance with one item and two buyers. For a small $\delta > 0$, the first buyer has a valuation of δ for the item, while the second has a valuation of $1/\delta$ with probability δ, and 0 otherwise. In this instance $OPT = 1 + \delta - \delta^2$. Most of the time the optimal allocation gives the item to the first buyer; however, most of the value in OPT comes from the second buyer. Thus, in order to obtain a good approximation of OPT using samples, we need to sample the valuation functions enough times to see at least once the $1/\delta$ valuation of the second buyer, i.e., we require $\Omega(1/\delta)$ samples. Otherwise, we could not distinguish the instance from one where the second buyer has valuation identically 0 (in which case we should allocate the item to the first buyer, obtaining a welfare of only δ).

4.5.1 Proof of Theorem 4.7

Our strategy to find the prices has several steps. First, we use an estimate $\hat{\psi}$ of ψ (recall the definition of ψ from Section 4.4). The function $\hat{\psi}$ differs from ψ in two ways: first, it replaces the optimal integral allocation with the optimal fractional allocation (according to the configuration LP specified below) in the definition of $z_{\mathrm{i},A}(p)$. Second, it takes polynomially-many samples and computes the empirical average, rather than an exact expected value. This allows us to compute $\hat{\psi}$ in poly-time. Finally, we write a convex quadratic minimization program whose solution is a fixed-point of $\hat{\psi}$. Because we can minimize convex quadratic functions in poly-time, we can then find a fixed point of $\hat{\psi}$.

More precisely we proceed as follows:

1. For $s \in \{1, \ldots, S\}$, with $S = \mathrm{poly}(|M|, |N|, 1/\varepsilon)$, draw independent sets of samples of the valuations $(v_{\mathrm{i}}^{(s)})_{\mathrm{i} \in N}$.
2. For each set of samples $(v_{\mathrm{i}}^{(s)})_{\mathrm{i} \in N}$ find an optimal fractional allocation $x^{(s)} = (x_{\mathrm{i},A}^{(s)})_{\mathrm{i} \in N, A \subseteq M}$,

i.e., one that solves

$$\text{(LP)} \qquad \max_{x\geq 0} \sum_{\mathrm{i}\in N}\sum_{A\subseteq M} x_{\mathrm{i},A}\cdot v_{\mathrm{i}}^{(s)}(A)$$
$$\text{s.t.} \quad \sum_{A\subseteq M} x_{\mathrm{i},A} \leq 1, \text{ for all } \mathrm{i}\in N,$$
$$\sum_{\mathrm{i}\in N}\sum_{A:j\in A} x_{\mathrm{i},A} \leq k_j, \text{ for all } j\in M.$$

3. For each $s = 1,\ldots,S$ define the functions $\hat{\psi}^{(s)} : \mathbb{R}^M_{\geq 0} \to \mathbb{R}^M_{\geq 0}$ as

$$\hat{\psi}^{(s)}_j(p) = \frac{1}{k_j}\sum_{\mathrm{i}\in N}\sum_{A:j\in A} x^{(s)}_{\mathrm{i},A}\cdot\left[v_{\mathrm{i}}^{(s)}(A) - \sum_{j'\in A} p_{j'}\right]_+, \text{ for each } j\in M, \tag{4.6}$$

and denote their average as $\hat{\psi} := \frac{1}{S}\sum_{s=1}^{S}\hat{\psi}^{(s)}$.

4. Find a fixed point of $\hat{\psi}$, i.e., a vector $\hat{p}$ such that $\hat{\psi}(\hat{p}) = \hat{p}$.

As said before, $\hat{\psi}$ does not exactly approximate ψ, but another function $\tilde{\psi} := \mathbf{E}(\hat{\psi})$. Notice that if we define

$$\tilde{z}_{\mathrm{i},A}(p) = \mathbf{E}\left(x_{\mathrm{i},A}\cdot\left[v_{\mathrm{i}}(A) - \sum_{j\in A} p_j\right]_+\right),$$

then $\tilde{\psi}$ is analogous to ψ as defined in (4.5), but using $\tilde{z}$ instead of z, i.e.,

$$\tilde{\psi}_j(p) = \frac{1}{k_j}\sum_{\mathrm{i}\in N}\sum_{A\subset M:j\in A}\tilde{z}_{\mathrm{i},A}(p).$$

To prove Theorem 4.7, we show that (i) given a set of valuation functions, we can efficiently compute an optimal solution of the linear program (LP) using demand queries; (ii) with polynomially many samples, the function $\hat{\psi}$ approximates $\tilde{\psi}$ sufficiently well; (iii) we can efficiently compute a fixed point of $\hat{\psi}$; and (iv) a fixed point of $\hat{\psi}$ (and thus an approximate fixed point of $\tilde{\psi}$) gives a $(\mathrm{d}+1+\varepsilon)$-approximation of OPT.

Even though the linear program (LP) has exponentially many variables, its dual has only $|M|+|N|$ variables. It turns out the demand queries provide a separation oracle for it, and therefore, it can be solved using the Ellipsoid method in polynomial time. For more details we refer to [106, Chapter 11.5.2]. This completes step (i). For each of the other three steps we prove a separate lemma.

Lemma 4.8. *Using $S = \mathrm{poly}(|M|,|N|,1/\varepsilon)$ samples we can guarantee that with probability $1-\varepsilon$ we have that $\sum_{j\in M}|\hat{\psi}_j(p) - \tilde{\psi}_j(p)| \leq \varepsilon\cdot OPT/(|M|\cdot|N|)$, for all $p\in[0,v_{\max}]^M$.*

Proof. Consider the following discretization of $[0,v_{\max}]^M$. In each coordinate we take multiples of $\delta = \varepsilon\cdot OPT/(4\cdot|M|^3\cdot|N|)$, i.e., we consider vectors in $\mathcal{P} = \{\mathrm{i}\cdot\delta : \mathrm{i}\in\mathbb{N}\}^M\cap[0,v_{\max}]^M$.

Recall that $\tilde{\psi} = \mathbf{E}(\hat{\psi})$. For any given $p \in \mathcal{P}$, $\lambda > 0$, $j \in M$, and number of samples S, an additive Chernoff bound indicates that

$$\mathbb{P}(|\hat{\psi}_j(p) - \mathbf{E}(\hat{\psi}_j(p))| > \lambda) \leq 2\exp\left(-\frac{2 \cdot S \cdot \lambda^2}{v_{\max}^2}\right).$$

Taking a union bound over all $j \in M$ and all $p \in \mathcal{P}$, we have that $\sum_{j\in M} |\hat{\psi}_j(p) - \mathbf{E}(\hat{\psi}_j(p))| \leq |M| \cdot \lambda$ for all $p \in \mathcal{P}$ with probability at least

$$\begin{aligned}
&1 - |M| \cdot |\mathcal{P}| \cdot 2\exp\left(-\frac{2 \cdot S \cdot \lambda^2}{v_{\max}^2}\right) \\
&= 1 - |M| \cdot \left(\frac{v_{\max}}{\delta}\right)^{|M|} \cdot 2\exp\left(-\frac{2 \cdot S \cdot \lambda^2}{v_{\max}^2}\right) \\
&= 1 - \varepsilon \cdot \exp\left(\log\frac{|M|}{\varepsilon} + |M| \cdot \log\frac{v_{\max}}{\delta} - S \cdot 2 \cdot \frac{\lambda^2}{v_{\max}^2}\right) \qquad (4.7)
\end{aligned}$$

Now take any vector $p \in [0, v_{\max}]^M$. By the definition of $\mathcal{P}$, there is a vector $\hat{p} \in \mathcal{P}$ such that $||p-\hat{p}||_1 \leq |M|\cdot\delta$. It is easy to check from the definition of $\hat{\psi}$ in (4.6) and the constraints that $x^{(s)}$ satisfies in LP that for all $j \in M$, $|\hat{\psi}_j(p) - \hat{\psi}_j(\hat{p})| \leq ||p - \hat{p}||_1$. Therefore, $\hat{\psi}$ and $\mathbf{E}(\hat{\psi})$ are $|M|$-lipschitz functions. By the triangle inequality, we have that

$$\begin{aligned}
||\hat{\psi}(p) - \mathbf{E}(\hat{\psi}(p))||_1 &\leq ||\hat{\psi}(p) - \hat{\psi}(\hat{p})||_1 + ||\hat{\psi}(\hat{p}) - \mathbf{E}(\hat{\psi}(\hat{p}))||_1 + ||\mathbf{E}(\hat{\psi}(\hat{p})) - \mathbf{E}(\hat{\psi}(p))||_1 \\
&\leq 2 \cdot |M|^2 \cdot \delta + |M| \cdot \lambda. \qquad (4.8)
\end{aligned}$$

Now, taking $\lambda = \varepsilon \cdot OPT/(2 \cdot |M|^2 \cdot |N|)$ and replacing in (4.8), we obtain that $||\hat{\psi}(p) - \mathbf{E}(\hat{\psi}(p))||_1$ is at most $\varepsilon \cdot OPT/(|M| \cdot |N|)$ for all $p \in [0, v_{\max}]^M$; with probability at least the expression in (4.7). Assuming $v_{\max}/OPT \leq \text{poly}(|M|, |N|)$, we can make make the probability in (4.7) larger than $1 - \varepsilon$ by taking $S = \text{poly}(|M|, |N|, |1/\varepsilon)$. $\square$

Lemma 4.9. *If* $S = \text{poly}(|M|, |N|, 1/\varepsilon)$ *we can compute a fixed point of* $\hat{\psi}$ *in time* $\text{poly}(|M|, |N|, 1/\varepsilon)$.

Proof. Recall that p is a fixed point of $\hat{\psi}$ if for all $j \in M$,

$$p_j = \hat{\psi}_j(p) = \frac{1}{S}\sum_{s=1}^{S}\sum_{\text{i}\in N}\sum_{A:j\in A}\frac{1}{k_j} \cdot x_{\text{i},A}^{(s)} \cdot \left[v_{\text{i}}^{(s)}(A) - \sum_{j'\in A} p_{j'}\right]_+.$$

Note that in this sum, the only non-zero terms are those such that $x_{\text{i},A}^{(s)} > 0$. A basic solution for the LP has at most $|M| + |N|$ non-zero variables, so there are at most $S \cdot (|M| + |N|) = \text{poly}(|M|, |N|, 1/\varepsilon)$ combinations of indices such that $x_{\text{i},A}^{(s)} > 0$. Denote by E the set of such indices, i.e., $E = \{(\text{i}, A, s) : \text{i} \in N, A \subseteq M, 1 \leq s \leq S, \text{ and } x_{\text{i},A}^{(s)} > 0\}$.

Now, for a vector p, define

$$y_{\text{e}} := \sqrt{x_{\text{i},A}^{(s)}} \cdot \left[v_{\text{i}}^{(s)}(A) - \sum_{j\in A} p_j\right]_+, \text{ for all e} = (\text{i}, A, s) \in E. \qquad (4.9)$$

If p is a fixed point, then it satisfies

$$p_j = \sum_{\substack{(\mathrm{i}',A',s')=\mathrm{e}': \\ \mathrm{e}'\in E, j\in A'}} \frac{\sqrt{x_{\mathrm{i}',A'}^{(s')}}}{S\cdot k_j}\cdot y_{\mathrm{e}'}. \tag{4.10}$$

By replacing p_j back in (4.9), we have that p is a fixed point if and only if $y = (y_{\mathrm{e}})_{\mathrm{e}\in E}$ satisfies

$$y_{\mathrm{e}} = \sqrt{x_{\mathrm{i},A}^{(s)}}\cdot\left[v_{\mathrm{i}}^{(s)}(A) - \sum_{j\in A}\sum_{\substack{(\mathrm{i}',A',s')=\mathrm{e}': \\ \mathrm{e}'\in E, j\in A'}} \frac{\sqrt{x_{\mathrm{i}',A'}^{(s')}}}{S\cdot k_j}\cdot y_{\mathrm{e}'}\right]_+, \quad \text{for all } \mathrm{e} = (\mathrm{i}, A, s)\in E. \tag{4.11}$$

We write a quadratic program with variables $(y_{\mathrm{e}})_{\mathrm{e}\in E}$ whose optimal solutions correspond to solutions of (4.11).

$$\text{(QP)} \qquad \min_{y} \sum_{\mathrm{e}=(\mathrm{i},A,s)\in E} y_{\mathrm{e}}\cdot\left(y_{\mathrm{e}} - \sqrt{x_{\mathrm{i},A}^{(s)}}\cdot\left(v_{\mathrm{i}}^{(s)}(A) - \sum_{j\in A}\sum_{\substack{(\mathrm{i}',A',s')=\mathrm{e}': \\ \mathrm{e}'\in E, j\in A'}} \frac{\sqrt{x_{\mathrm{i}',A'}^{(s')}}}{S\cdot k_j}\cdot y_{\mathrm{e}'}\right)\right)$$

s.t.

$$y_{\mathrm{e}} \geq \sqrt{x_{\mathrm{i},A}^{(s)}}\cdot\left(v_{\mathrm{i}}^{(s)}(A) - \sum_{j\in A}\sum_{\substack{(\mathrm{i}',A',s')=\mathrm{e}': \\ \mathrm{e}'\in E, j\in A'}} \frac{\sqrt{x_{\mathrm{i}',A'}^{(s')}}}{S\cdot k_j}\cdot y_{\mathrm{e}'}\right), \quad \mathrm{e} = (\mathrm{i}, A, s)\in E \tag{4.12}$$

$$y_{\mathrm{e}} \geq 0, \quad \mathrm{e}\in E. \tag{4.13}$$

To see that it suffices to optimize this quadratic program, take first a vector y that satisfies (4.11) (note that such a vector must exist, by Brouwer's fixed-point theorem). It is immediately implied by (4.11) that y satisfies both (4.12) and (4.13). Moreover, it is also evident that for all $\mathrm{e}\in E$ one of the two constraints must be tight, implying that the objective function must take a value of 0. Notice that the objective function is necessarily non-negative for feasible solutions, so y is an optimal solution. Observe also that for any optimal solution y' to the quadratic program, because the objective function takes a value of zero it must be the case that for every $\mathrm{e}\in E$, at least one of (4.12) and (4.13) is tight. This directly shows that y' satisfies (4.11).

We finally argue that the quadratic program is convex and hence can be solved in polynomial time [92]. To do so, it suffices to argue that the objective function can be written in the form $b^T y + y^T(B^T B + I)y$ for some vector b and matrix B. We define $B\in\mathbb{R}^{M\times E}$ by

$$B_{j,\mathrm{e}=(\mathrm{i},A,s)} := \sqrt{\frac{x_{\mathrm{i},A}^{(s)}}{S\cdot k_j}}\cdot\mathbb{1}_{j\in A}.$$

Now, for $\mathrm{e} = (\mathrm{i}, A, s)$ and $\mathrm{e}' = (\mathrm{i}', A', s')$ observe that

$$(B^TB)_{\mathrm{e},\mathrm{e}'} = \sum_{j\in M} \frac{\sqrt{x^{(s)}_{\mathrm{i},A}\cdot x^{(s')}_{\mathrm{i}',A'}}}{S\cdot k_j}\cdot \mathbb{1}_{j\in A, j\in A'} = \sqrt{x^{(s)}_{\mathrm{i},A}}\cdot\left(\sum_{j\in A\cap A'}\frac{\sqrt{x^{(s')}_{\mathrm{i}',A'}}}{S\cdot k_j}\right).$$

From this is is straightforward to see that all the nonlinear terms in the objective of QP can be written as $y^T(B^TB+I)y$ as we wanted to show. □

Lemma 4.10. *If $\hat{p}\in[0,v_{\max}]^M$ is such that $\sum_{j\in M}|\hat{p}_j-\tilde{\psi}_j(p)|\le\varepsilon\cdot OPT/(|M|\cdot|N|)$, then*

$$(\mathrm{d}+1+O(\varepsilon))\cdot ALG(\hat{p})\ge OPT.$$

Proof. We simply re-do the proof of Theorem 4.1, but replacing with the approximate fixed point. Thus, we take $\hat{p}\in[0,v_{\max}]^M$ such that $\sum_{j\in M}|\hat{p}_j-\tilde{\psi}_j(p)|\le\varepsilon\cdot OPT/(|M|\cdot|N|)$, and replace in the bound of Lemma 4.2.[4] We obtain that

$$\begin{aligned}
ALG(\hat{p}) &\ge \min_{C\subseteq M}\left\{\sum_{j\in C}k_j\cdot\hat{p}_j+\sum_{\mathrm{i}\in N}\sum_{A\subseteq\bar{C}}\tilde{z}_{\mathrm{i},A}(\hat{p})\right\}\\
&\ge \sum_{\mathrm{i}\in N}\sum_{A\subseteq M}\tilde{z}_{\mathrm{i},A}(\hat{p})-\frac{\varepsilon\cdot OPT}{|M|\cdot|N|}\cdot\max_{j\in M}k_j\\
&\ge \sum_{\mathrm{i}\in N}\sum_{A\subseteq M}\tilde{z}_{\mathrm{i},A}(\hat{p})-\frac{\varepsilon\cdot OPT}{|M|}.
\end{aligned}$$

The last inequality comes from the fact that we can assume without loss of generality that $\max_{j\in M}k_j\le|N|$, since a buyer buys at most one copy of each item. Then, we use the upper bound of Lemma 4.3.

$$\begin{aligned}
OPT &\le \sum_{j\in M}k_j\cdot\hat{p}_j+\sum_{\mathrm{i}\in N}\sum_{A\subseteq M}\tilde{z}_{\mathrm{i},A}(\hat{p})\\
&\le (\mathrm{d}+1)\sum_{\mathrm{i}\in N}\sum_{A\subseteq M}\tilde{z}_{\mathrm{i},A}(\hat{p})+\frac{\varepsilon\cdot OPT}{|M|\cdot|N|}\cdot\max_{j\in M}k_j.
\end{aligned}$$

Noting that $\mathrm{d}\le|M|$, this implies that

$$(1-\varepsilon)\cdot OPT\le(\mathrm{d}+1)\cdot\sum_{\mathrm{i}\in N}\sum_{A\subseteq M}\tilde{z}_{\mathrm{i},A}(\hat{p}).$$

Putting together the lower bound on $ALG(\hat{p})$ and the upper bound on OPT we conclude that

$$(\mathrm{d}+1+O(\varepsilon))ALG(\hat{p})\ge OPT.$$

□

[4]Here we use $\tilde{z}_{\mathrm{i},A}(p)$ instead of $z_{\mathrm{i},A}(p)$, as we take an approximate fixed point of $\tilde{\psi}$ instead of ψ. Using almost identical arguments we can show the corresponding versions of Lemmas 4.2 and 4.3. For completeness, we include them in Appendix 4.8.

The proof of the theorem is straightforward from the lemmas: we take $S = \text{poly}(|M|, |N|, \varepsilon)$ samples of the valuations, as required by Lemma 4.8 so that $\hat{\psi}$ is a good approximation of $\tilde{\psi}$. For each sample, we solve (LP) in polynomial time, so we can calculate $\hat{\psi}$ in polynomial time. Then, by Lemma 4.9, we can compute a fixed point of $\hat{\psi}$. Finally, taking the computed fixed point as prices, we get a $(\mathrm{d} + 1 + O(\varepsilon))$-approximation of OPT, by Lemma 4.10.

4.6 Deterministic Single-minded Valuations

In this section, we consider the special case where there is a single copy of each item (i.e., $k_\mathrm{i} = 1$ for all $\mathrm{i} \in M$), buyers' valuations are deterministic, and buyers are *single-minded.* The latter means each buyer i has a set A_i, with $|A_\mathrm{i}| \leq \mathrm{d}$, such that $A_\mathrm{i} \not\subseteq B \Rightarrow v_\mathrm{i}(B) = 0, A_\mathrm{i} \subseteq B \Rightarrow v_\mathrm{i}(B) = v_\mathrm{i}(A_\mathrm{i})$. The problem of maximizing the welfare of an allocation in this context can be seen as the classic combinatorial problem of hypergraph matching with hyperedges of size at most d, where the buyers correspond to the hyperedges and the items are the vertices. Indeed, in an optimal allocation for this setting buyer i either gets A_i or $\emptyset$, implying that maximizing the welfare of the allocation is equivalent to finding a subset of pairwise disjoint A_i's of maximum total valuation. As this is a traditional problem, in the rest of this section we will refer to hypergraphs, hyperedges and vertices, rather than buyers and items, using the usual notation $G = (V, E)$ and denoting by $w(\mathrm{e})$ the valuation (or weight) of the hyperedge e.

4.6.1 Matching in Graphs: $\mathrm{d} = 2$

We first focus on the traditional matching problems, showing that using prices has limits even for this scenario. A similar discussion appears in [33], though not for single-minded valuations. While our results are simple (and similar to those in [33]), we describe them here is full since we believe it helps completing the picture of the single-minded case. In particular, we show in Lemmas 4.11 and 4.13, there are instances in which no pricing scheme can guarantee recovering more than 2/3 of the optimal solution. This is true even if the graph is bipartite or if there is a unique optimal matching; on the other hand, if both conditions are fulfilled — the graph is bipartite and there is a unique optimal matching — using the dual prices leads precisely to such optimal solution.

Lemma 4.11. *Prices cannot guarantee obtaining more than* 2/3 *of the optimal matching, even if the graph is bipartite.*

Proof. Consider the graph depicted in Fig. 4.1, in which all edges have unit weight. There are two optimal solutions, given by the black and the red perfect matchings; no perfect matching can be constructed using arcs of different colors. Assume we have prices that are able to build an optimal solution (i.e., include three edges) regardless of the order in which the edges arrive. This implies that for at least one of the optimal solutions, all the edges will be included if their vertices are available when they arrive. Without loss assume this is the case for the black matching, i.e. for $\mathrm{i} = 1, 2, 3, p_{L_\mathrm{i}} + p_{R_\mathrm{i}} \leq 1$.

On the other hand, we need to prevent the red edges to be included if they appear; to see why this is necessary, consider for instance the case in which the edge (L_1, R_2) is not

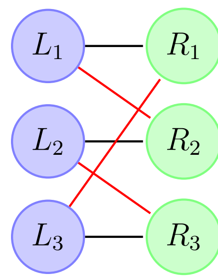


Figure 4.1: Example of a bipartite graph in which, when all edges have the same weight, no pricing scheme can guarantee obtaining more than 2/3 of the optimal solution.

discarded when appearing first. Then, if the edge (L_3, R_3) appears second, no more edges could be added. To preclude this, we need to impose that for $\mathrm{i} = 1, 2, 3$, $p_{L_\mathrm{i}} + p_{R_{(\mathrm{i}+1)\bmod 3}} > 1$. A contradiction follows by adding these and the previous three inequalities.

Finally, if, for instance, all vertex prices are 1/2, two edges will be added regardless of the order in which they appear. $\square$

In the case of bipartite graphs, it is natural to consider the usual linear programming formulation, since it has integer optimal solutions. The following lemma shows that when we require the additional hypothesis that there is a unique optimal matching. the prices given by the optimal solution of the dual problem lead to that optimal assignment.

Lemma 4.12. *If the graph $G = (V, E)$ is bipartite and has a unique optimal matching, then such a matching is obtained using the dual prices.*

Proof. Because the graph is bipartite, the problem reduces to solving the following linear program: $\max\{\sum_{\mathrm{e}\in E} x_\mathrm{e} w(\mathrm{e}) : \sum_{\mathrm{e}\in\delta(v)} x_\mathrm{e} \leq 1 \text{ for all } v \in V,\ x \geq 0\}$; which has an integral optimal solution. Because there is only one optimal matching, the LP has a unique optimal solution $(x^*_\mathrm{e})_{\mathrm{e}\in E}$. Consider the prices $(p^*_u)_{u\in V}$ corresponding to an optimal dual solution, satisfying strict complementary slackness.

Consider an edge $\mathrm{e} = (u, v)$ that is not part of the optimal matching. Hence, the corresponding primal variable takes the value $x^*_\mathrm{e} = 0$. By complementary slackness, the corresponding dual constraint is not tight, i.e. $p^*_u + p^*_v > w(\mathrm{e})$. This last condition implies that buyer e will not buy the edge upon arrival. On the other hand, if e is part of the optimal solution, the corresponding dual constraint must be tight (again due to strict complementary slackness), so that those buyers will choose to buy. $\square$

The assumption of a unique solution is crucial for the dual prices to be useful. Indeed, when there is more than one solution, using the dual prices can be arbitrarily inefficient. Indeed, consider the same example depicted in Figure 4.1, but modify the weight of the edges $f = (L_1, R_1)$ and $g = (L_2, R_3)$ to be ε, so that that the optimal solution has value $2 + \varepsilon$. On the other hand, consider an edge $\mathrm{e} = (u, v)$ and the resulting dual prices p_u, p_v: complementary slackness now states that we have $p_u + p_v = w(\mathrm{e})$ iff e is part of any optimal solution. Edge f is part of the black optimal solution, and edge g is part of the red, hence those edges will be bought if the corresponding vertices are available when they appear. In particular, if they are the first two edges to appear, then they will both be in the final solution, and no other edge can be added, leading to a final weight of 2ε.

However, in general graphs, even the uniqueness assumption is not enough. Indeed we have the following result.

Lemma 4.13. *Prices cannot guarantee obtaining more than $2/3$ of the optimal matching in a general graph, even if there is only one optimal matching.*

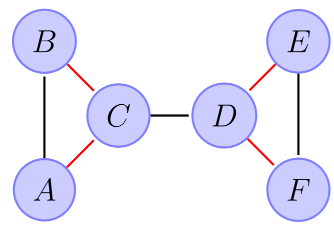


Figure 4.2: Example of a graph in which, when all edges have the same weight, there is a unique optimal matching but no pricing scheme can guarantee obtaining more than 2/3 of its weight.

Proof. Consider the graph depicted in Fig. 4.2, where every edge has unit weight. The optimal matching is given by the three black edges with total value of 3. On the other hand, if any red edge enters the solution, the resulting total weight will be at most 2. We now show that any pricing scheme in which every black edge is willing to buy will also include at least one red edge if it comes first. Let $(p_{\mathrm{i}})_{\mathrm{i}=A,\dots,F}$ prices such that for every black edge, the sum of the involved vertices is lower than 1. In particular, we have that $p_C + p_D \leq 1$, so without loss of generality we assume that $p_C \leq 1/2$. If $p_B \leq 1/2$ as well, then the red edge (B, C) will want to buy and the proof is complete. Otherwise, if $p_B > 1/2$, it implies that $p_A \leq 1/2$ because the black edge (A, B) wants to buy. But this implies that the red edge (A, C) will buy if appearing first.

Finally, if all vertex prices are 1/2, then it is straightforward to see that at least two edges will be added regardless of the order in which they appear. □

In general, there are item prices that guarantee obtaining at least half of the optimal welfare. This is achieved by splitting the weight of the edges of an optimal matching uniformly between the two corresponding vertices. We present this result in Lemma 4.15 for general d.

4.6.2 Hypergraph Matching: $\mathrm{d} > 2$

We begin this section proving two negative results. First we show an upper bound of $\sim \sqrt{\frac{1}{\mathrm{d}}}$ on the fraction of the optimal solution that can be guaranteed with prices. We then show a specific bound for the case $\mathrm{d} = 3$, in which we cannot guarantee obtaining more than $1/2$ of the optimal welfare. Finally, we provide a pricing scheme that always obtains at least $1/\mathrm{d}$ of the optimal welfare.

Lemma 4.14. *Prices cannot guarantee welfare more than an $\sim \sqrt{\frac{1}{\mathrm{d}}}$ fraction of the optimal welfare, even if the arrival order is known.*

Proof. Our example is based on constructions for finite projective planes; namely, we will use the fact that if $q-1$ is a prime power there exists a hypergraph on q^2-q+1 vertices with q^2-q+1 hyperedges that are q-regular, q-uniform and intersecting, i.e. every pair of hyperedges has at least one shared vertex (see, e.g., [79, Chapter 12] for a reference).

To build our example, we will assume that for each hyperedge there exists a corresponding buyer interested in exclusively that subset of items with a total valuation of q. We will also add one buyer whose only subset of interest is the entire set of items, with a valuation of $\mathrm{d} = q^2-q+1$. Note that clearly the optimal welfare attainable is q^2-q+1.

It hence suffices to show that prices cannot achieve welfare greater than q. Assume the buyer interested in the entire set of items arrives last. Note that if there is any edge e such that the sum of the prices of the vertices in e is at most than q, we are guaranteed welfare at most q. However, if every the sum of the prices of the vertices in every hyperedge is more than q, because our graph is q-uniform that means the sum of the prices of *all* vertices is more than q^2-q+1, meaning the final buyer would not select anything and the welfare attained is zero. Hence, the total welfare attainable by prices is at most a

$$\frac{q}{q^2-q+1} \sim \frac{1}{\sqrt{q^2-q+1}}$$

fraction of the optimum.

Finally, if d cannot be written as q^2-q+1, we replicate the same construction for the largest $\mathrm{d}' < \mathrm{d}$ that can, and the result holds. □

When $\mathrm{d} = 3$, we show that no prices exist obtaining a better than 1/2-approximation (tightening the 2/3 bound that can be obtained via Lemma 4.14). Such a bound is obtained using a hypergraph $G = (V, E)$ with $V = \{1, 2, 3, 4, 5, 6\}$ and the following hyperedges with unit weight:

$$\{1, 2, 3\}, \{4, 5, 6\}, \{1, 2, 4\}, \{1, 3, 5\}, \{2, 5, 6\}, \{3, 4, 6\}$$

First, note that there is a perfect matching (of weight 2), given by the two first hyperedges. Also note that each of the remaining hyperedges intersect all other hyperedges, thus only one of them could be included in a feasible solution. Therefore, it suffices to prove that there is no pricing scheme in which the first two edges ($\{1, 2, 3\}, \{4, 5, 6\}$) want to buy but all the others do not.

Let us assume that there are prices $p_1, \ldots, p_6$ that achieve the aforementioned property. For the first two edges to be taken when they appear, we need $p_1+p_2+p_3 \leq 1$ and $p_4+p_5+p_6 \leq 1$. To prevent the other edges to buy when they appear, we need the sum of the corresponding vertices to be strictly greater than 1, hence we obtain four additional inequalities. Adding up all the six inequalities, we obtain $\sum_{\mathrm{i}=1}^{6} p_{\mathrm{i}} > 2$. And the exact opposite result is obtained adding the first two inequalities.

We now provide our positive result. Consider a hypergraph $G = (V, E)$, with weights $(w(\mathrm{e}))_{\mathrm{e}\in E}$. To define the prices, take an optimal matching given by the hyperedges $OPT_1, \dots, OPT_\ell$. For each $a \in OPT_j$, define $p_a = w(OPT_j)/\mathrm{d}$. The prices of the items not covered by the optimal solution are set to ∞. The following simple result shows that these prices obtain at least a fraction $1/\mathrm{d}$ of the optimal welfare.

Lemma 4.15. *Consider prices defined as above, and hyperedges arriving in an arbitrary order. Denote Q the set of edges that are bought. Then*

$$\sum_{\mathrm{e}\in Q} w(\mathrm{e}) \geq \frac{1}{\mathrm{d}} \sum_{j=1}^{\ell} w(OPT_j)$$

Proof. First note that for each $\mathrm{e} \in Q$, it must hold that

$$w(\mathrm{e}) \geq \sum_{\mathrm{i}\in\mathrm{e}} p_{\mathrm{i}} \tag{4.14}$$

As otherwise the buyer associated to e would have decided not to buy. Therefore:

$$\sum_{\mathrm{e}\in Q} w(\mathrm{e}) \geq \sum_{\mathrm{e}\in Q} \sum_{\mathrm{i}\in\mathrm{e}} p_{\mathrm{i}} \tag{4.15}$$

On the other hand, for each OPT_j in the optimal solution, there must be at least one vertex, with its corresponding price $w(OPT_j)/\mathrm{d}$ that is covered by the edges in Q. To see this, note that there are two possible cases: either $OPT_j \in Q$ and all its vertices are covered, or $OPT_j \notin Q$, meaning that when OPT_j arrived, at least one of its vertices was not available, i.e., it was covered by an edge previously bought. The result follows directly, noting that in the right side of (4.15), we are summing at least once $w(OPT_j)/\mathrm{d}$ for each $j = 1 \dots, \ell$. □

4.7 Conclusion and Future Directions

In this chapter, we provided an efficiently computable 1/(d+1)-approximate pricing algorithm for maximizing social welfare when buyers have arbitrary and random monotone valuations on subsets of at most d items. Although this approximation factor is tight in the worst case, numerous interesting directions for future work remain. In the special case where buyers are single-minded and have deterministic valuations, for $\mathrm{d} = 2$ we have bounded the best attainable ratio of pricing algorithms in $[1/2, 2/3]$, so the exact value is yet to be found. It would furthermore be relevant to understand the asympotics of the optimal ratio for general d, which our results place in $[\sim 1/\sqrt{\mathrm{d}}, 1/\mathrm{d}]$. When buyers are single-minded but could have random valuations, our problem is closely related to the design of thresholding prophet inequalities. Many open problems remain in this area; we gave an upper bound for prophet inequalities for bipartite matching and note that there remain large gaps in known bounds for the optimal competitive ratio for matching in bipartite (and general) graphs.

4.8 Bounds Using an Optimal Solution of LP

We prove here that Lemmas 4.2 and 4.3 also hold when in the definition of $z_{\mathrm{i},A}(p)$ we replace $\mathbb{1}_{OPT_\mathrm{i}=A}$ with $x_{\mathrm{i},A}$, an optimal solution of LP, the linear relaxation of the optimal allocation problem. This means we replace $z_{\mathrm{i},A}(p)$ with

$$\tilde{z}_{\mathrm{i},A}(p) = \mathbf{E}\left(x_{\mathrm{i},A} \cdot \left[v_\mathrm{i}(A) - \sum_{j\in A} p_j \right]_+ \right).$$

The proofs are almost identical to the original ones in Section 4.4.

Lemma 4.16. *For any vector of prices $p \in \mathbb{R}^M_{\geq 0}$,*

$$ALG(p) \geq \min_{C\subseteq M} \left\{ \sum_{j\notin C} k_j \cdot p_j + \sum_{\mathrm{i}\in N} \sum_{A\subseteq C} \tilde{z}_{\mathrm{i},A}(p) \right\}.$$

Proof. In this proof we assume the arrival order σ is arbitrary, and for simplicity we denote $B_\mathrm{i}(\sigma)$ and $R_\mathrm{i}(\sigma)$ simply by B_i and R_i. We separate the welfare of the resulting allocation into *revenue* and *utility*, i.e., we separate $\sum_{\mathrm{i}\in N} v_\mathrm{i}(B_\mathrm{i})$ into

$$\text{Revenue} = \sum_{\mathrm{i}\in N} \sum_{j\in B_\mathrm{i}} p_j \quad \text{and} \quad \text{Utility} = \sum_{\mathrm{i}\in N} \left(v_\mathrm{i}(B_\mathrm{i}) - \sum_{j\in B_\mathrm{i}} p_j \right).$$

Recall that R_i is the set of items with remaining copies when i arrives. Similarly, denote by R the set of items that have remaining copies by the end of the process. Note first that

$$\mathbf{E}(\text{Revenue}) \geq \mathbf{E}\left(\sum_{j\notin R} k_j \cdot p_j \right).$$

This is simply because all items $j \notin R$ have sold all k_j copies. For the utility, for any $\mathrm{i} \in N$, by the definition of B_i it holds that

$$v_\mathrm{i}(B_\mathrm{i}) - \sum_{j\in B_\mathrm{i}} p_j = \max_{A\subseteq R_\mathrm{i}} v_\mathrm{i}(A) - \sum_{j\in A} p_j$$

Note now that v_i and R_i are independent. Let $(\tilde{v}_\mathrm{i})_{\mathrm{i}\in N}$ be independent realizations of the valuations. With this and noting that $R \subseteq R_\mathrm{i}$, we can rewrite the expected utility of agent i as

$$\mathbf{E}\left(\max_{A\subseteq R_\mathrm{i}} v_\mathrm{i}(A) - \sum_{j\in A} p_j \right) = \mathbf{E}\left(\max_{A\subseteq R_\mathrm{i}} \tilde{v}_\mathrm{i}(A) - \sum_{j\in A} p_j \right) \geq \mathbf{E}\left(\max_{A\subseteq R} \tilde{v}_\mathrm{i}(A) - \sum_{j\in A} p_j \right). \quad (4.16)$$

Let $\tilde{x}$ denote an optimal solution of LP when the values are $(\tilde{v}_\mathrm{i})_{\mathrm{i}\in N}$. Since $(\tilde{v}_\mathrm{i})_{\mathrm{i}\in N}$ is independent of R, $\tilde{x}$ is also independent of R. Since $\tilde{x}$ is feasible for LP, for any given $\mathrm{i} \in N$, $\sum_{A\subseteq R} \tilde{x}_{\mathrm{i},A} \leq \sum_{A\subseteq M} \tilde{x}_{\mathrm{i},A} \leq 1$. We can replace the maximization over subsets of R in (4.16) with the convex combination of particular choices given by $(\tilde{x}_{\mathrm{i},A})_{A\subseteq R}$. Thus, we obtain the following lower bound.

$$
\begin{aligned}
\mathbf{E}\left(\max_{A\subseteq R} \tilde{v}_{\mathrm{i}}(A) - \sum_{j\in A} p_j\right) &\geq \mathbf{E}\left(\sum_{A\subseteq R} \tilde{x}_{\mathrm{i},A}\cdot\left[\tilde{v}_{\mathrm{i}}(A) - \sum_{j\in A} p_j\right]_+\right) \\
&= \mathbf{E}\left(\sum_{A\subseteq R} \mathbf{E}\left(\tilde{x}_{\mathrm{i},A}\cdot\left[\tilde{v}_{\mathrm{i}}(A) - \sum_{j\in A} p_j\right]_+\right)\right) \\
&= \mathbf{E}\left(\sum_{A\subseteq R} \tilde{z}_{\mathrm{i},A}(p)\right). \qquad (4.17)
\end{aligned}
$$

The positive part $[\cdot]_+$ comes from the fact that we can always choose $\emptyset \subseteq R$ in the maximization in (4.16). Summing over all agents, we get that

$$
\mathbf{E}(\text{Utility}) \geq \mathbf{E}\left(\sum_{\mathrm{i}\in N}\sum_{A\subseteq R} \tilde{z}_{\mathrm{i},A}(p)\right).
$$

Therefore, adding the revenue and the utility we get that

$$
ALG(p) \geq \mathbf{E}\left(\sum_{j\notin R} k_j\cdot p_j + \sum_{\mathrm{i}\in N}\sum_{A\subseteq R} \tilde{z}_{\mathrm{i},A}(p)\right).
$$

Replacing the expectation over R with a minimization over subsets of M we obtain the bound of the lemma. ☐

Lemma 4.17. *For any vector of prices $p \in \mathbb{R}^M_{\geq 0}$,*

$$
OPT \leq \sum_{j\in M} k_j\cdot p_j + \sum_{\mathrm{i}\in N}\sum_{A\subseteq M} \tilde{z}_{\mathrm{i},A}(p).
$$

Proof. Let x be an optimal solution of LP. We have that

$$
\begin{aligned}
OPT &\leq \mathbf{E}\left(\sum_{\mathrm{i}\in N}\sum_{A\subseteq M} x_{\mathrm{i},A}\cdot v_{\mathrm{i}}(A)\right) \\
&= \mathbf{E}\left(\sum_{\mathrm{i}\in N}\sum_{A\subseteq M} x_{\mathrm{i},A}\cdot \sum_{j\in A} p_j\right) + \mathbf{E}\left(\sum_{\mathrm{i}\in N}\sum_{A\subseteq M} x_{\mathrm{i},A}\cdot\left(v_{\mathrm{i}}(A) - \sum_{j\in A} p_j\right)\right).
\end{aligned}
$$

Now we upper bound these two terms separately. Since x is feasible for LP, for all $j \in M$ we have that $\sum_{\mathrm{i}\in N}\sum_{A:j\in A} x_{\mathrm{i},A} \leq k_j$, so the first term satisfies

$$
\mathbf{E}\left(\sum_{\mathrm{i}\in N}\sum_{A\subseteq M} x_{\mathrm{i},A}\cdot\sum_{j\in A} p_j\right) \leq \sum_{j\in M} k_j\cdot p_j.
$$

For the second term we simply upper bound the difference with its positive part.

$$\begin{aligned}
&\mathbf{E}\left(\sum_{\mathrm{i}\in N}\sum_{A\subseteq M} x_{\mathrm{i},A}\cdot\left(v_{\mathrm{i}}(A)-\sum_{j\in A}p_j\right)\right)\\
&\leq \sum_{\mathrm{i}\in N}\sum_{A\subseteq M}\mathbf{E}\left(x_{\mathrm{i},A}\cdot\left[v_{\mathrm{i}}(A)-\sum_{j\in A}p_j\right]_+\right)\\
&\leq \sum_{\mathrm{i}\in N}\sum_{A\subseteq M}\tilde{z}_{\mathrm{i},A}(p).
\end{aligned}$$

Putting together the two upper bounds we obtain the bound on OPT. $\square$

Bibliography

[1] Allaart, P., and Islas, J. A sharp lower bound for choosing the maximum of an independent sequence. *Journal of Applied Probability 53* (2015), 1041–1051.

[2] Alpern, S., and Baston, V. The secretary problem with a selection committee: Do conformist committees hire better secretaries? *Management Science 63*, 4 (2017), 901–1269.

[3] Assadi, S., Kesselheim, T., and Singla, S. Improved truthful mechanisms for subadditive combinatorial auctions: Breaking the logarithmic barrier. In *SODA* (2021).

[4] Assadi, S., and Singla, S. Improved truthful mechanisms for combinatorial auctions with submodular bidders. In *FOCS* (2019).

[5] Azar, P. D., Kleinberg, R., and Weinberg, S. M. Prophet inequalities with limited information. In *SODA* (2014).

[6] Babaioff, M., Dinitz, M., Gupta, A., Immorlica, N., and Talwar, K. Secretary problems: weights and discounts. In *SODA* (2009).

[7] Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. A knapsack secretary problem with applications. In *APPROX-RANDOM* (2007).

[8] Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. Matroid secretary problems. *Journal of the ACM (JACM) 65*, 6 (2018), 1–26.

[9] Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. Matroid secretary problems. *Journal of the ACM 65*, 6 (2018), 35:1–35:26.

[10] Babaioff, M., Immorlica, N., and Kleinberg, R. Matroids, secretary problems, and online mechanisms. In *SODA* (2007).

[11] Babaioff, M., Lucier, B., Nisan, N., and Paes Leme, R. On the efficiency of the walrasian mechanism. In *EC* (2014).

[12] Baldwin, E., and Klemperer, P. Understanding preferences:"demand types", and the existence of equilibrium with indivisibilities. *Econometrica 87*, 3 (2019), 867–932.

[13] Bearden, J. N., Rapoport, A., and Murphy, R. O. Sequential observation

and and selection with rank-dependent payoffs: An experimental study. *Management Science 52*, 9 (2006), 1437–1449.

[14] Beyhaghi, H., and Kleinberg, R. Pandora's problem with nonobligatory inspection. In *EC* (2019).

[15] Bhattacharjya, D., and Deleris, L. A. The value of information in some variations of the stopping problem. *Decision Analysis 11*, 3 (2014), 189–203.

[16] Bradac, D., Gupta, A., Singla, S., and Zuzic, G. Robust algorithms for the secretary problem. In *ITCS* (2020).

[17] Bruss, F. T. A unified approach to a class of best choice problems with an unknown number of options. *Annals of Probability 12*, 3 (1984), 882–889.

[18] Buchbinder, N., Jain, K., and Singh, M. Secretary problems via linear programming. *Mathematics of Operations Research 39*, 1 (2014), 190–206.

[19] Buchbinder, N., Jain, K., and Singh, M. Secretary problems via linear programming. *Mathematics of Operations Research 39*, 1 (2014), 190–206.

[20] Cai, Y., and Zhao, M. Simple mechanisms for subadditive buyers via duality. In *STOC* (2017).

[21] Campbell, G., and Samuels, S. M. Choosing the best of the current crop. *Advances in Applied Probability 13*, 3 (1981), 510–532.

[22] Cayci, S., Gupta, S., and Eryilmaz, A. Group-fair online allocation in continuous time. In *NeurIPS* (2020).

[23] Celis, E., Keswani, V., Straszak, D., Deshpande, A., Kathuria, T., and Vishnoi, N. Fair and diverse DPP-based data summarization. In *ICML* (2018).

[24] Celis, L. E., Huang, L., and Vishnoi, N. K. Multiwinner voting with fairness constraints. In *IJCAI* (2018).

[25] Celis, L. E., Straszak, D., and Vishnoi, N. K. Ranking with fairness constraints. In *ICALP* (2018).

[26] Chan, T. H., Chen, F., and Jiang, S. H.-C. Revealing optimal thresholds for generalized secretary problem via continuous lp: Impacts on online k-item auction and bipartite k-matching with random arrival order. In *SODA* (2015).

[27] Chawla, S., Hartline, J. D., Malec, D. L., and Sivan, B. Multi-parameter mechanism design and sequential posted pricing. In *STOC* (2010).

[28] Chawla, S., Hartline, J. D., Malec, D. L., and Sivan, B. Multi-parameter mechanism design and sequential posted pricing. In *STOC* (2010).

[29] Chawla, S., and Miller, J. B. Mechanism design for subadditive agents via an ex

ante relaxation. In *EC* (2016).

[30] Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. In *NeurIPS* (2017).

[31] Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvtiskii, S. Matroids, matchings, and fairness. In *AISTATS* (2019).

[32] Chow, Y., Moriguti, S., Robbins, H., and Samuels, S. Optimal selection based on relative rank. *Israel Journal of Mathematics 2*, 2 (1964), 81–90.

[33] Cohen-Addad, V., Eden, A., Feldman, M., and Fiat, A. The invisible hand of dynamic market pricing. In *EC* (2016).

[34] Cole, R., and Roughgarden, T. The sample complexity of revenue maximization. In *STOC* (2014).

[35] Correa, J., Cristi, A., Duetting, P., and Norouzi-Fard, A. Fairness and bias in online selection. In *ICML* (2021).

[36] Correa, J., Cristi, A., Epstein, B., and Soto, J. Sample-driven optimal stopping: From the secretary problem to the iid prophet inequality. *arXiv preprint arXiv:2011.06516* (2021).

[37] Correa, J., Cristi, A., Epstein, B., and Soto, J. A. The two-sided game of googol and sample-based prophet inequalities. In *SODA* (2020).

[38] Correa, J., Cristi, A., Feuilloley, L., Oosterwijk, T., and Tsigonias-Dimitriadis, A. The secretary problem with independent sampling. In *SODA* (2021).

[39] Correa, J., Cristi, A., Fielbaum, A., Pollner, T., and Weinberg, S. M. Optimal item pricing in online combinatorial auctions. In *IPCO* (2022).

[40] Correa, J., Dütting, P., Fischer, F., and Schewior, K. Prophet inequalities for i.i.d. random variables from an unknown distribution. In *EC* (2019).

[41] Correa, J., Dutting, P., Fischer, F., and Schewior, K. Prophet inequalities for iid random variables from an unknown distribution. In *EC* (2019).

[42] Correa, J., Dütting, P., Fischer, F., Schewior, K., and Ziliotto, B. Unknown i. i. d. prophets: Better bounds and streaming algorithms and and a new impossibility. In *ITCS* (2021).

[43] Correa, J., Foncea, P., Hoeksma, R., Oosterwijk, T., and Vredeveld, T. Posted price mechanisms for a random stream of customers. In *EC* (2017).

[44] Correa, J., Foncea, P., Hoeksma, R., Oosterwijk, T., and Vredeveld, T. Posted price mechanisms and optimal threshold strategies for random arrivals. *Mathematics of Operations Research 46*, 4 (2021), 1452–1478.

[45] Correa, J., Saona, R., and Ziliotto, B. Prophet secretary through blind strategies. In *SODA* (2019).

[46] Dobzinski, S. Breaking the logarithmic barrier for truthful combinatorial auctions with submodular bidders. In *STOC* (2016).

[47] Dobzinski, S., Nisan, N., and Schapira, M. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC* (2005).

[48] Doval, L. Whether or not to open pandora's box. *Journal of Economic Theory 175* (2018), 127–158.

[49] Dütting, P., Feldman, M., Kesselheim, T., and Lucier, B. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput 49*, 3 (2020), 540–582.

[50] Dutting, P., Feldman, M., Kesselheim, T., and Lucier, B. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM Journal on Computing 49*, 3 (2020), 540–582.

[51] Dütting, P., Kesselheim, T., and Lucier, B. An O(log log m) prophet inequality for subadditive combinatorial auctions. In *FOCS* (2020).

[52] Dütting, P., Lattanzi, S., Paes Leme, R., and Vassilvitskii, S. Secretaries with advice. In *EC* (2021).

[53] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *TCC* (2006).

[54] Dynkin, E. B. The optimum choice of the instant for stopping a Markov process. *Soviet Math. Dokl 4* (1963), 627–629.

[55] Dynkin, E. B. The optimum choice of the instant for stopping a markov process. *Soviet Mathematics Doklady 4* (1963), 627–629.

[56] Ehsani, S., Hajiaghayi, M., Kesselheim, T., and Singla, S. Prophet secretary for combinatorial auctions and matroids. In *SODA* (2018).

[57] Esfandiari, H., HajiAghayi, M., Lucier, B., and Mitzenmacher, M. Prophets, secretaries, and maximizing the probability of choosing the best. In *AISTATS* (2020).

[58] Ezra, T., Feldman, M., Gravin, N., and Tang, Z. G. Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. In *EC* (2020).

[59] Ezra, T., Feldman, M., Gravin, N., and Tang, Z. G. Online stochastic max-weight matching: prophet inequality for vertex and edge arrival models. In *EC* (2020).

[60] Feldman, M., Gravin, N., and Lucier, B. Combinatorial auctions via posted prices. In *SODA* (2014).

[61] Feldman, M., Gravin, N., and Lucier, B. Combinatorial auctions via posted prices. In *SODA* (2015).

[62] Feldman, M., Gravin, N., and Lucier, B. Combinatorial walrasian equilibrium. *SIAM Journal on Computing 45*, 1 (2016), 29–48.

[63] Feldman, M., Svensson, O., and Zenklusen, R. Online contention resolution schemes. In *SODA* (2016).

[64] Feldman, M., Svensson, O., and Zenklusen, R. Simple O(log log(rank))-competitive algorithm for the matroid secretary problem. *Math. Oper. Res 43*, 2 (2018), 638–650.

[65] Feldman, M., Svensson, O., and Zenklusen, R. A. A simple O(log log(rank))-competitive algorithm for the matroid secretary problem. In *SODA* (2015).

[66] Feldman, M., and Tennenholtz, M. Interviewing secretaries in parallel. In *EC* (2012).

[67] Ferguson, T. S. Who solved the secretary problem? *Statistical Science 4*, 3 (1989), 282–296.

[68] Ferguson, T. S. Who solved the secretary problem? *Statistical Science 4* (1989), 282–289.

[69] Gilbert, J., and Mosteller, F. Recognizing the maximum of a sequence. *J. Am. Statist. Assoc 61* (1966), 35–73.

[70] Goldstein, D. G., McAfee, P. R., Suri, S., and Wright, J. R. Learning when to stop searching. *Management Science 66*, 3 (2020), 1375–1394.

[71] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

[72] Gravin, N., and Wang, H. Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In *EC* (2019).

[73] Gravin, N., and Wang, H. Prophet inequality for bipartite matching: merits of being simple and non adaptive. In *EC* (2019).

[74] Guo, C., Huang, Z., and Zhang, X. Settling the sample complexity of single-parameter revenue maximization. In *STOC* (2019).

[75] Gupta, S., and Salem, J. Closing the gap: Mitigating bias in online resume-filtering. In *WINE* (2020).

[76] Gusein-Zade, S. M. The problem of choice and the optimal stopping rule for a sequence of in trials. *Theory of Probability and Applications 11*, 3 (1966), 472–476.

[77] Hajiaghayi, M. T., Kleinberg, R., and Sandholm, T. Automated online mech-

anism design and prophet inequalities. In *AAAI* (2007).

[78] Halabi, M. E., Mitrovic, S., Norouzi-Fard, A., Tardos, J., and Tarnawski, J. Fairness in streaming submodular maximization: Algorithms and hardness. In *NeurIPS* (2020).

[79] Hall, M. *Combinatorial theory*, vol. 71. John Wiley & Sons, 1998.

[80] Hill, T. P., and Kertz, R. P. Comparisons of stop rule and supremum expectations of i.i.d. random variables. *The Annals of Probability 10*, 2 (1982), 336–345.

[81] Immorlica, N., Kleinberg, R. D., and Mahdian, M. Secretary problems with competing employers. In *WINE* (2006).

[82] Kaplan, H., Naori, D., and Raz, D. Competitive analysis with a sample and the secretary problem. In *SODA* (2020).

[83] Kearns, M., and Roth, A. *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, 2019.

[84] Kertz, R. P. Stop rule and supremum expectations of iid random variables: a complete comparison by conjugate duality. *Journal of Multivariate Analysis 19* (1986), 88–112.

[85] Kesselheim, T., and Molinaro, M. Knapsack secretary with bursty adversary. In *ICALP* (2020).

[86] Kesselheim, T., Radke, K., Tönnis, A., and Vöcking, B. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *ESA* (2013).

[87] Kesselheim, T., Radke, K., Tönnis, A., and Vöcking, B. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *ESA* (2013).

[88] Kesselheim, T., Tönnis, A., Radke, K., and Vöcking, B. Primal beats dual on online packing lps in the random-order model. In *STOC* (2014).

[89] Khatibi, A., and Jacobson, S. Generalized sequential stochastic assignment problem. *Stochastic Systems 8*, 4 (2018), 293–306.

[90] Kleinberg, R., and Weinberg, S. M. Matroid prophet inequalities. In *STOC* (2012).

[91] Korula, N., and Pál, M. Algorithms for secretary problems on graphs and hypergraphs. In *ICALP* (2009).

[92] Kozlov, M. K., Tarasov, S. P., and Khachiyan, L. G. Polynomial solvability of convex quadratic programming. In *Doklady Akademii Nauk* (1979), vol. 248, Russian

Academy of Sciences, pp. 1049–1051.

[93] Krengel, U., and Sucheston, L. Semiamarts and finite values. *Bulletin of the American Mathematical Society 83*, 4 (1977), 745–747.

[94] Krengel, U., and Sucheston, L. On semiamarts, amarts, and processes with finite value. *Adv. in Probability 4* (1978), 197–266.

[95] Krysta, P., and Vöcking, B. Online mechanism design (randomized rounding on the fly). In *ICALP* (2012).

[96] Kumar, R., Lattanzi, S., Vassilvitskii, S., and Vattani, A. Hiring a secretary from a poset. In *EC* (2011).

[97] Lachish, O. O(log log rank) competitive ratio for the matroid secretary problem. In *FOCS* (2014).

[98] Lien, R. W., Iravani, S. M. R., and R., K. Smilowitz sequential resource allocation for nonprofit operations. *Operations Research 62*, 2 (2014), 301–317.

[99] Lindley, D. V. and decision theory. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 10* (1961), 39–51.

[100] Lindley, D. V. Dynamic programming andand decision theory. *Applied Statistics 10* (1961), 39–51.

[101] Liu, A., Leme, R. P., Pál, M., Schneider, J., and Sivan, B. Variable decomposition for prophet inequalities and optimal ordering. In *EC* (2021).

[102] Moran, S., Snir, M., and Manber, U. Applications of ramsey's theorem to decision tree complexity. *Journal of the ACM 32*, 4 (1985), 938–949.

[103] Moro, S., Cortez, P., and Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems 62* (2014), 22–31.

[104] Mucci, A. G. Differential equations and optimal choice problems. *The Annals of Statistics 1* (1973), 104–113.

[105] Mucci, A. G. On a class of secretary problems. *The Annals of Probability 1*, 3 (1973), 417–427.

[106] Nisan, N., Roughgarden, T., Tardos, É., and Vazirani, V. V., Eds. *Algorithmic Game Theory.* Cambridge University Press, 2007.

[107] Nuti, P. The secretary problem with distributions. In *IPCO* (2022).

[108] Nuti, P., and Vondrák, J. Secretary problems: The power of a single sample. *Manuscript* (2022).

[109] Paes Leme, R., and Wong, S. C.-w. Computing walrasian equilibria: Fast algo-

rithms and structural properties. *Mathematical Programming 179*, 1 (2020), 343–384.

[110] Rubinstein, A., Wang, J. Z., and Weinberg, S. M. Optimal single-choice prophet inequalities from samples. In *ITCS* (2020).

[111] Samuel-Cahn, E. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *The Annals of Probability 12*, 4 (1984), 1213–1216.

[112] Samuels, S. Exact solutions for the full information best choice problem. *Purdue Univ. Stat. Dept. Mimeo Series* (1982), 82–17.

[113] Samuels, S. S. P. Secretary problems. In *Handbook of Sequential Analysis*. CRC Press, 1991.

[114] Soto, J. A. Matroid secretary problem in the random-assignment model. *SIAM J. Comput 42*, 1 (2013), 178–211.

[115] Soto, J. A., Turkieltaub, A., and Verdugo, V. Strong algorithms for the ordinal matroid secretary problem. In *SODA* (2018).

[116] Takac, L., and Zábovský, M. Data analysis in public social networks. In *Proceedings of the International Scientific Conference and International Workshop Present Day Trends of Innovations* (2012).

[117] Trevisan, L. Non-approximability results for optimization problems on bounded degree instances. In *STOC* (2001).

[118] Weitzman, M. Optimal search for the best alternative. *Econometrica 47*, 3 (1979), 641–654.