Online Stochastic Max-Weight Bipartite Matching: Beyond Prophet Inequalities*

Christos Papadimitriou[1], Tristan Pollner[2], Amin Saberi[2], and David Wajc[2]

[1]Columbia University
[2]Stanford University

Abstract

The rich literature on online Bayesian selection problems has long focused on so-called prophet inequalities, which compare the gain of an online algorithm to that of a "prophet" who knows the future. An equally-natural, though significantly less well-studied benchmark is the optimum *online* algorithm, which may be omnipotent (i.e., computationally-unbounded), but not omniscient. What is the computational complexity of the optimum online? How well can a polynomial-time algorithm approximate it?

Motivated by applications in ride hailing, we study the above questions for the online stochastic maximum-weight matching problem under vertex arrivals. This problem was recently introduced by Ezra, Feldman, Gravin and Tang (EC'20), who gave a 1/2-competitive algorithm for it. This is the best possible ratio, as this problem is a generalization of the original single-item prophet inequality.

We present a polynomial-time algorithm which approximates optimal online within a factor of 0.51—beating the best-possible prophet inequality. At the core of our result are a new linear program formulation, an algorithm that tries to match the arriving vertices in two attempts, and an analysis that bounds the correlation resulting from the second attempts. In contrast, we show that it is PSPACE-hard to approximate this problem within some constant $\alpha < 1$.

*Research supported in part by NSF Awards CCF1763970, CCF191070, CCF1812919, ONR award N000141912550, and a gift from Cisco Research.

1 Introduction

Decision-making in an uncertain, dynamic environment influenced by one's decisions has arguably always been the essence of life, and yet it appears to have been first confronted mathematically by Herbert Robbins and Richard Bellman, from different perspectives, in the late 1940s and early 1950s. Decision theory initially focused on instantaneous decisions, but later gave us stopping rules and the gem of prophet inequalities [34]. Later, the Internet age brought us new business models relying exclusively on stochastic decision making — online advertising, ride hailing, kidney exchanges — in which the changing environment affected by the agents' decisions can often be abstracted as an evolving weighted bipartite graph.

Here we study one such problem, the online Bayesian bipartite matching, or RideHail, problem. The input to this problem is a random bipartite graph, revealed over time. Initially, the m nodes on one side of the graph, termed taxis or bins, are present. The n nodes on the other side, termed passengers or balls, are revealed over time. Initially, we know for each ball t the probability p_t of it actually arriving, as well as the weight $w_{i,t}$ of the edge connecting it to any bin i — if it arrives. If ball t does not arrive, we do nothing at time t; if it does arrive, we can choose to match it, irrevocably, to some unmatched neighbor i before time $t+1$, yielding a profit $w_{i,t}$. Our goal is to maximize the overall expected profit.

RideHail essentially generalizes the classic single-item online Bayesian stopping rule problem — the so-called *prophet inequality* problem. In particular, our problem with a single offline node already captures the worst-case examples of the prophet inequality, for which no online algorithm is better than $1/2$-competitive. On the other hand, RideHail is a special bipartite case of the online stochastic max-weight matching problem in general graphs studied by Gravin and Wang [23] and Ezra et al. [16]. In the latter work, Ezra et al. present a $1/2$-competitive algorithm for this problem, which is worst-case optimal for our problem too.

There is an extensive literature on numerous variations of online Bayesian selection problems, that relate the performance of online algorithms with the omniscient prophet of inequality fame — that is to say, with the offline optimum (see Section 1.2). In particular, these works study achievable competitive ratios: the worst-case ratio over all inputs between the online algorithm and the best offline algorithm. While this may be the right thing to do when the input is adversarial, when the input is generated *stochastically* one perhaps could do better. In particular, in the stochastic case, the optimum online algorithm *for the given input* is a well-defined benchmark that can be computed in exponential time. Suddenly we are in the realm of *approximation algorithms,* rather than of competitive analysis.

In approximation algorithms, typically one explores two interesting questions: First, is approximation hard? And second, what is the best approximation ratio achievable in polynomial time?

In this paper we address both questions. First, we show that for some $\alpha < 1$ it is PSPACE-hard to approximate the RideHail problem within a factor of α.

Theorem 1.1. *It is* PSPACE*-hard to approximate the optimal online* RideHail *algorithm within a factor of α, for some absolute constant $\alpha < 1$.*

Here, $1-\alpha$ is small, limited by the current status of expander constructions and approximation hardness of MAX-SSAT (see Section 2). To our knowledge, no past work on variants of online matching had demonstrated such level of hardness.

We then develop an approximation algorithm, as well as a technique to bound the (online) optimum. The upper bounding technique is our main innovation. To our knowledge, all past work on approximating this large family of problems, with the exception of [5], has used the prophet inequality bound on the offline optimum, which necessarily limits the approximation ratio for many variations to be below $1/2$.

We go for bounding the online optimum. We achieve this by identifying a new constraint which separates online from offline algorithms. In particular, we note that online algorithms cannot match an edge (i, t) with probability greater than the probability of ball t arriving, times the probability of bin i not being matched by the online algorithm beforehand, due to independence of these events. This constraint, which is not true of offline algorithms, poses restrictions on the marginal probabilities of edges to be matched by the optimal online algorithm. Combining this constraint with the natural matching constraints we obtain a new LP which bounds the optimal online algorithm's gain. Using this new LP bound (and a number of further ideas, see Section 1.1), we design a new algorithm which recovers at least 51% of the online optimum, i.e., a ratio strictly better than the optimal competitive ratio of $1/2$.

Theorem 1.2. *There exists a polynomial-time online algorithm which is a* 0.51*-approximation of the optimal online algorithm for the* RideHail *problem.*

We further generalize our algorithm and achieve the same approximation bound for the more general problem in which weights of any given ball's edges can follow any joint distribution, but weights of different ball's edges are independent. That is, we extend our positive results to the more general problem studied by Ezra et al. [16], in bipartite graphs. (See Section 5.)

1.1 Techniques

Here we give a very brief overview of the key ideas used to obtain our main results.

1.1.1 Hardness

For our PSPACE-hardness result, we refine the result of Condon et al. [11] for maximum satisfiability of stochastic SAT instances. In the stochastic SAT (SSAT) problem, introduced by Papadimitriou [39], a 3CNF formula is given, and variables $x_1, x_2, \ldots, x_n$ are alternatingly set by an (online) algorithm and randomly set by nature. Condon et al. [11] proved that approximating the maximum expected number of satisfiable clauses of an SSAT instance is PSPACE-hard. Using an *expander graph* construction, we extend this result to SSAT instances in which each variable appears in at most a constant number of clauses. We then give a polynomial-time reduction from approximating maximum satisfiability of a bounded-occurrence SSAT instance to approximating the optimal online algorithm for the RideHail problem, implying our claimed PSPACE-hardness.

1.1.2 Algorithm

Our algorithmic results involve a number of ideas. We outline the key ones here.

A New LP. We recall that we want to approximate the optimal online algorithm within a factor strictly greater than the $1/2$ which is best-possible against the optimal offline algorithm. Hence, our first objective is to identify a property which separates online from offline algorithms. To this end, we note that for any online algorithm $\mathcal{A}$, the probability of arrival of ball t is independent of the event that bin i is not matched by Algorithm $\mathcal{A}$ prior to time t.[1] Consequently, the probability that edge (i, t) is matched by Algorithm $\mathcal{A}$ is at most the product of these two events' probabilities. Combining this constraint with natural matching constraints, we obtain an LP which bounds the expected gain of the optimal online algorithm (but not its offline counterpart) .

[1]Note that this constraint does not necessarily hold for the prophetic optimum offline algorithm $\mathcal{A}^*$, which makes its matching choices based on both past and future balls' arrivals.

A Second Chance Algorithm. We present an efficient online algorithm for approximately rounding a solution to the above LP. Let $x_{i,t}$ be the decision variables of this LP. Intuitively, these $x_{i,t}$ serve as proxies for the probability of (i,t) to be matched by the optimal online algorithm. Our online algorithm matches each edge (i,t) with probability at least $x_{i,t} \cdot (1/2 + c)$ for $c = 1/100$. Our algorithm can be seen as a generalization and extension of the $1/2$-competitive algorithm of Ezra et al. [16] for our problem. Their algorithm can be thought of as approximately rounding the above LP (without the new constraint) as follows. After each arrival of ball t, pick a bin i with probability proportional to $x_{i,t}$, and then, if bin i is unmatched, match edge (i,t) with some probability $q_{i,t}$. These $q_{i,t}$ are set to guarantee that each edge (i,t) is matched with marginal probability $x_{i,t} \cdot 1/2$. To improve on this, we first note that modifying these $q_{i,t}$ appropriately results in each edge (i,t) being matched with probability precisely $x_{i,t} \cdot (1/2 + c)$ if $\sum_{t'<t} x_{i,t'}$ is small, and at least $x_{i,t} \cdot (1/2 - O(c))$ otherwise. To increase these marginal probabilities to $x_{i,t} \cdot (1/2 + c)$ for each edge (i,t), we repeat the above process if t is unmatched, letting t pick a second bin i' and possibly matching edge (i',t). For this second pick to achieve its desired effect, bin i should not be matched too often when picked by ball t in its second pick. That is, conditioning on t not being matched after its first pick should not decrease the probability of i being free by too much. This is the core of our analysis.

Analysis. To prove that conditioning on ball t not being matched after its first pick indeed does not decrease the probability of bin i being free by much, we show that (i) the bins' matched statuses by time t have low correlation, and (ii) bin i is unlikely to be picked twice by ball t. To prove Property (i), we show that most of the probability of a bin to be matched by this algorithm is accounted for by variables which are negatively correlated, and even *negatively associated* (see Section 2). For our proof of Property (ii), we finally reap the rewards from our new LP constraint. In particular, this constraint implies that for bins i with $\sum_{t'<t} x_{i,t'}$ large, as above, $x_{i,t}$ must be low, implying that bin i is unlikely to be picked by ball t as its first pick. Properties (i) and (ii) together imply that conditioning on ball t not being matched after its first pick does not decrease the probability of bin i to be unmatched much. This then implies that the second pick is not too unlikely to result in a match of edge (i,t). We thus find that each edge (i,t) is matched by our algorithm with probability $(1/2 + c) \cdot x_{i,t}$, from which our $(1/2 + c)$-approximation follows.

1.2 Related Work

The literature on online Bayesian selection problems is a long and illustrious one. We briefly outline some of the most relevant work here. (See also surveys on the topic [13, 26, 27, 36].)

A seminal result in the stopping theory literature, the first prophet inequality, a $1/2$-competitive algorithm for the single-item online Bayesian selection problem, was first given in the late 70s [34]. Multiple algorithms achieving this bound are known [3, 4, 16, 33, 40]. On the other hand, better bounds are known for various special cases, most prominently for i.i.d. distributions [1, 12, 28].

Numerous *multiple-item* online Bayesian selection problems were studied over the years. Generalizations of the classic $1/2$-competitive prophet inequality of [34] for single-items were given for matroid constraints [33], for multiple items [3], for bipartite matching under one-sided vertex arrivals [4], and for general matching under vertex arrivals [16]. For matching under *edge* arrivals, a number of positive results are known [18, 23, 33], and a competitive ratio of $1/2$ is impossible for this stochastic problem [16, 23]. This mirrors a similar separation between vertex arrivals and edge arrivals for this problem's (unweighted) deterministic counterpart [21]. Much of this work on approximating the optimal offline algorithm (prophet inequalities) for online Bayesian selection problems was motivated by connections discovered between prophet inequalities and algorithmic mechanism design ([9, 14, 17, 25]). The computational complexity of approximating the *online* optimal algorithm, however, was significantly less well studied.

The only previous positive result for approximating the online optimum algorithm (better than offline optimum) for an online Bayesian selection problem is due to Anari et al. [5], who gave a PTAS for a special class of matroid constraints. On the computational complexity front, the only hardness for such problems we are aware of is the recent result of Agrawal et al. [2], who show that computing the optimal *ordering* of the random variables for a single-item problem is NP-hard. The problem of approximating optimum online (and its hardness) was studied for other stochastic online optimization problems recently, including probing problems [10, 22, 41], stochastic matching problems in infinite-horizon settings under Poisson arrivals and departures [6], and two-stage stochastic matching problems [19]. Computational complexity of approximating optimum online for these and other problems remains an intriguing open problem. We are hopeful that the tools we develop here will prove useful in extending the literature on computational complexity and approximability of such problems of decision-making under uncertainty.

2 Preliminaries

For any algorithm $\mathcal{A}$ and instance $\mathcal{I}$ of a problem Π, we let $\mathcal{A}(\mathcal{I})$ denote the value of the output of algorithm $\mathcal{A}$ on instance I. We use $OPT_{on}^{\Pi}(\mathcal{I})$ to denote an optimal online algorithm for Π on $\mathcal{I}$. Since the problem Π will be clear from context, we will usually just write $OPT_{on}(\mathcal{I})$. Our interest is in understanding how well this value can be approximated by efficient online algorithms. Throughout, we say an algorithm gives an α-approximation to a quantity Q, for $\alpha \in (0,1)$, if it outputs a number in the range $[\alpha Q, Q]$. The following simple fact, whose proof is deferred to Appendix A, is useful for reductions involving hardness of approximation.

Fact 2.1. *Let $Q, Q' \geq 0$ be positive quantities, such that $Q'/Q \leq \beta$, and let $\alpha \in (0,1)$. Then, an $\left(\frac{\alpha+\beta}{1+\beta}\right)$-approximation to $Q + Q'$ yields an α-approximation to Q.*

We now turn to providing background on problems and tools used in this work.

Stochastic SAT. The stochastic SAT (SSAT) problem was first defined by Papadimitriou [39]. In this work, we will consider the maximization variant of this problem, defined below.

Definition 2.2. The input to the MAX-SSAT problem is a 3CNF formula ϕ over an ordered list of variables $(x_1, x_2, \ldots, x_n)$. We choose a value of either `True` or `False` for x_1, nature chooses a value of either `True` or `False` for x_2 uniformly at random, we choose a value of either `True` or `False` for x_3, and so on. Our goal is to maximize the expected number of satisfied clauses in ϕ after all the variables have been assigned a value. For convenience, we will refer to $\{x_1, x_3, \ldots\}$ as the "deterministic variables" and $\{x_2, x_4, \ldots\}$ as the "random variables."

In his work introducing SSAT, Papadimitriou [39] proved PSPACE-hardness of determining the probability of satisfiability of an SSAT instance. Over a decade later, this was improved to a *hardness of approximation* result by Condon et al. [11], via extensions of the PCP theorem [7]. In particular, they prove the following hardness of approximation result.

Lemma 2.3. *([11, Theorem 3.3]) There exist constants $k \in \mathbb{N}$ and $\alpha \in (0,1)$ so that it is PSPACE-hard to compute an α-approximation to $OPT_{on}(\phi)$ for a MAX-SSAT instance ϕ satisfying:*

1. *no random variable appears negated in any clause of ϕ, and*
2. *each random variables appears in at most k clauses of ϕ.*

It is worth noting that Theorem 3.3 in [11] only includes the statement about random variables being non-negated. The second property is a direct consequence of the proof of the theorem. In Appendix A we explain the necessary modifications to the proof to add this guarantee.

Expander Graphs. Define the expansion of a graph G as

$$h(G) := \min_{S \subseteq V, |S| \leq |V|/2} \frac{|E(S, V \setminus S)|}{|S|},$$

where $E(X,Y) := \{(x,y) \in E \mid e \in X, y \in Y\}$ denotes the edges with one endpoint in X and the other in Y. We will utilize results providing explicit, deterministic constructions of graphs with constant degree and constant expansion (e.g. [20, 35]).

Lemma 2.4. *There exists a deterministic, polynomial-time construction of a graph on n vertices with expansion at least 1 and maximum degree at most some constant d.*

Negative Association. We briefly review some notions of negative dependence we need in this work, in particular, the notion of *Negatively Associated* random variables.

Definition 2.5 ([30, 32])**.** Random variables $X_1, \ldots, X_n$ are *negatively associated (NA)*, if every two monotone non-decreasing functions f and g defined on disjoint subsets of the variables in $\vec{X}$ are negatively correlated. That is,

$$\mathbb{E}[f \cdot g] \leq \mathbb{E}[f] \cdot \mathbb{E}[g]. \tag{1}$$

A family of independent random variables are trivially negatively associated. A more interesting example of negatively associated random variables is the following.

Proposition 2.6 (0-1 Principle [15])**.** *Let $X_1, \ldots, X_n \in \{0,1\}$ be binary random variables such that $\sum_i X_i \leq 1$ always. Then, the joint distribution $(X_1, \ldots, X_n)$ is negatively associated.*

More elaborate NA distributions can be obtained via the following closure properties.

Proposition 2.7 (NA Closure Properties [15, 30, 32])**.**

1. *Independent union. Let $(X_1, \ldots, X_n)$ and $(Y_1, \ldots, Y_m)$ be two mutually independent negatively associated joint distributions. Then, the joint distribution $(X_1, \ldots, X_n, Y_1, \ldots, Y_m)$ is also NA.*
2. *Function composition. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be NA, and let $f_1, \ldots, f_k$ be monotone non-decreasing functions defined on disjoint subsets of $\mathbf{X}$. Then the joint distribution $(f_1, \ldots, f_k)$ is also NA.*

Negative association implies many powerful concentration inequalities and other useful properties (see e.g., [8, 15, 30, 32]). For our purposes we will use the pairwise negative correlation of NA variables, implied by Equation (1) with the disjoint functions $f(\vec{X}) = X_i$ and $g(\vec{X}) = X_j$ for $i \neq j$.

Proposition 2.8. *Let $X_1, \ldots, X_n$ be NA random variables. Then, for all $i \neq j$,* $\mathrm{Cov}(X_i, X_j) \leq 0$.

3 PSPACE-Hardness

In this section, we prove our PSPACE-hardness result.

Theorem 1.1. *It is* PSPACE*-hard to approximate the optimal online* RideHail *algorithm within a factor of α, for some absolute constant $\alpha < 1$.*

3.1 Extending Stochastic SAT Hardness

We first extend hardness of approximation for MAX-SSAT instances as in Lemma 2.3 to instances which in addition satisfy that *deterministic* variables appear in at most k clauses.

Lemma 3.1. *There exist constants $k \in \mathbb{N}$ and $\alpha \in (0,1)$ so that it is PSPACE-hard to compute an α-approximation to $OPT_{on}(\phi)$ for a MAX-SSAT instance ϕ satisfying*

(1) no random variable appears negated in any clause of ϕ, and

(2) each *variable (both random and deterministic) appears in at most k clauses of ϕ.*

We give a polynomial-time reduction from α-approximating $OPT_{on}(\phi)$ for a MAX-SSAT instance ϕ as in Lemma 2.3 to α'-approximating $OPT_{on}(\phi')$ on a MAX-SSAT instance ϕ' satisfying both properties (1) and (2) for some $k' = O(1)$ and constant $\alpha' \in (0,1)$.

The reduction. For odd (deterministic) i, if the variable x_i appears in $a(i)$ clauses in ϕ, we replace the j^{th} occurrence of x_i with a new variable $x_{i,j}$ for $1 \leq j \leq a(i)$. Let ϕ' denote the new 3CNF formula after these replacements. We also add clauses to force the optimal online algorithm to set all of $(x_{i,1}, x_{i,2}, \ldots x_{i,a(i)})$ equal to each other, without increasing their number of occurrences by more than a constant. Specifically, for each odd i, we construct via Lemma 2.4 an expander graph G_i on $a(i)$ vertices with maximum degree at most $d = O(1)$ and expansion at least 1. Associate the vertices of G_i with the literals $(x_{i,1}, x_{i,2}, \ldots, x_{i,a(i)})$ arbitrarily. For any edge in G_i between $x_{i,j}$ and $x_{i,j'}$, add the following two clauses to ϕ':

$$(x_{i,j} \vee \overline{x_{i,j'}}) \wedge (\overline{x_{i,j}} \vee x_{i,j'}). \tag{2}$$

Note that if $x_{i,j} \neq x_{i,j'}$, we satisfy exactly one of these two clauses, while if $x_{i,j} = x_{i,j'}$ we satisfy both. The order of variables $x_{i,j}$ and x_i in ϕ is some arbitrary order such that variables in ϕ' corresponding to (copies of) variables x_i and x_j in ϕ appear in an order consistent with the variables x_i and x_j in ϕ. By adding dummy random variables, we further guarantee that copies of deterministic/random variables in ϕ are likewise deterministic/random in ϕ'.

The following lemma relates the maximum expected number of satisfiable clauses in ϕ and ϕ', needed to complete our reduction's analysis.

Lemma 3.2. *Let $E_n := \sum_{odd\ i \leq n} 2|E(G_i)|$. Then, the MAX-SSAT instances ϕ and ϕ' satisfy*

$$OPT_{on}(\phi') = OPT_{on}(\phi) + E_n.$$

Proof. We first prove $OPT_{on}(\phi') \geq OPT_{on}(\phi) + E_n$. Consider an online algorithm $\mathcal{A}$ which for odd i sets $x_{i,1} = x_{i,2} = \ldots = x_{i,a(i)} = b_i$, where b_i is the assignment for x_i of OPT_{on} on ϕ given the induced history. This algorithm for ϕ' is clearly implementable. Moreover, this algorithm satisfies each of the E_n clauses of form (2), and satisfies $OPT_{on}(\phi)$ of the original clauses in expectation. Hence $OPT_{on}(\phi') \geq \mathcal{A}(\phi') = OPT_{on}(\phi) + E_n$.

We now prove that $OPT_{on}(\phi') \leq OPT_{on}(\phi) + E_n$. Assume that for some odd i, and some fixed history for all variables before $(x_{i,1}, \ldots, x_{i,a(i)})$, an SSAT algorithm $\mathcal{A}$ sets $(x_{i,1}, x_{i,2}, \ldots, x_{i,a(i)})$ such that they do not all take the same value (with some positive probability). Consider the minimum size subset $S \subseteq \{1, 2, \ldots, a(i)\}$ such that flipping all $\{x_{i,j}\}_{j \in S}$ would result in all variables being set to the same value (so, $1 \leq |S| \leq a(i)/2$). Since the expansion of G_i is at least 1, we know that $|E(S, V \setminus S)| \geq |S|$; flipping all the $\{x_{i,j}\}_{j \in S}$ would hence let us satisfy at least $|S|$ additional clauses of the form (2), and possibly satisfy $|S|$ fewer clauses corresponding to clauses in ϕ containing x_i. Thus, $\mathcal{A}$ would satisfy at least as many clauses in expectation by flipping the sign of $\{x_{i,j}\}_{j \in S}$. Repeatedly applying this transformation results in an improved online algorithm $\mathcal{A}'$ as stated in the previous paragraph, from which we find that OPT_{on} satisfies at most $OPT_{on}(\phi) \leq \mathcal{A}'(\phi') \leq OPT_{on}(\phi') + E_n$ clauses in expectation. The lemma follows. $\square$

We now that E_n is bounded from above by a constant times $OPT_{on}(\phi)$.

Observation 3.3. $E_n \leq 12d \cdot OPT_{on}(\phi)$.

Proof. Since for each odd i, the expander graph G_i contains at most d edges per each of the $a(i)$ occurrences of i in ϕ, we have that $E_n = \sum_{\text{odd } i \leq n} 2|E(G_i)| \leq \sum_{\text{odd } i \leq n} 2d \cdot a(i)$. Next, for m the number of clauses in ϕ, since ϕ is a 3-CNF formula, $\sum_{\text{odd } i \leq n} a(i) \leq 3m$. Finally, we note that, since setting each variable randomly satisfies at least half of the clauses in expectation, $m/2 \leq OPT_{on}(\phi)$. Combining these observations, we find that

$$E_n = \sum_{\text{odd } i \leq n} 2|E(G_i)| \leq \sum_{\text{odd } i \leq n} d \cdot a(i) \leq 6dm \leq 12d \cdot OPT_{on}(\phi). \qquad \square$$

Given the above, we are now ready to prove Lemma 3.1.

Proof of Lemma 3.1. Let $\alpha \in (0,1)$ and k be the constants in the statement of Lemma 2.3. Let ϕ be a MAX-SSAT instance as in the statement of that lemma and ϕ' be the obtained instance from the reduction of this section, which is polynomial-time, by Lemma 2.4. By construction, no random variable appears negated in any clause, and each variable appears in at most $k' = \max(d+2, k) = O(1)$ clauses. By Lemma 3.2, $OPT_{on}(\phi') = OPT_{on}(\phi) + E_n$. Next, we let $Q = OPT_{on}(\phi)$, $Q' = E_n$, and $\beta = 12d$, and note that $Q'/Q \leq \beta$, by Observation 3.3. Thus, by Fact 2.1, for the constant $\alpha' := \left(\frac{\alpha+12d}{1+12d}\right) \in (0,1)$, an α'-approximation to $OPT_{on}(\phi') = OPT_{on}(\phi) + E_n = Q + Q'$ yields an α-approximation of $Q = OPT_{on}(\phi)$, which is PSPACE-hard, by Lemma 2.3. $\square$

3.2 Hardness of Algorithms for RideHail

We are now ready to prove our main theorem about the hardness of RideHail. Throughout this proof, we will let $k = O(1)$ be the constant in the statement of Lemma 3.1. Denote the variables in an SSAT instance ϕ as in Lemma 3.1 by $(x_1, x_2, \ldots, x_n)$ and the number of clauses of ϕ by m. Without loss of generality, suppose n is even. From ϕ, we construct a RideHail instance $\mathcal{I}_\phi$, with weights $w_{i,t} = w_t$ for each pair $(i,t) \in E$, where we refer to w_t as the weight of ball t. The instance has $2n$ bins, corresponding to the literals $\{x_i, \overline{x_i} \mid i \in [n]\}$. The instance $\mathcal{I}_\phi$ has $n+m$ balls; we will refer to the first n balls as "literal balls" and the final m balls as "clause balls" (for reasons that will become clear shortly). For odd $t \leq n$, ball t arrives with probability 1, has weight 1, and has an edge only to bins x_t and $\overline{x_t}$. For even $t \leq n$, ball t arrives with probability $1/2$, has weight 1, and has an edge only to bin x_t. The last m clause balls $t = n+1, \ldots, n+m$ each have weight $\frac{m^4}{2k}$ and arrive with probability m^{-4}. The clause ball $t = n+r$ corresponding to clause C_r neighbors only the bins corresponding to literals in C_r. (See Figure 1.)

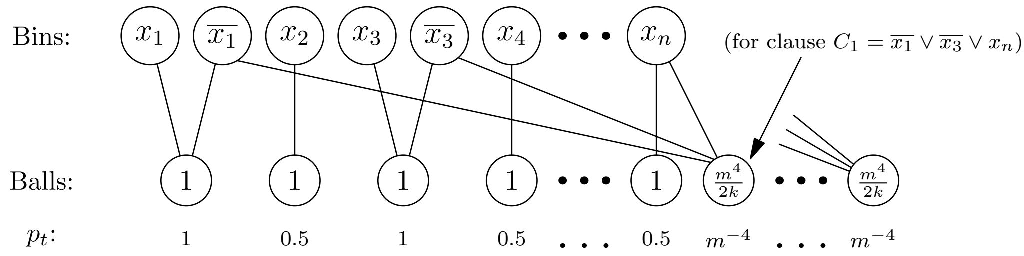

Figure 1: The RideHail instance $\mathcal{I}_\phi$

Bins are labeled by their corresponding literal, while balls are labeled by their weight.

We shall see that $OPT_{on}(\mathcal{I}_\phi)$ and $OPT_{on}(\phi)$ are, up to a negligible error term, related by a simple linear relation. In particular, we will show that

$$OPT_{on}(\mathcal{I}_\phi) = 0.75n + \frac{(1-m^{-4})^{m-1}}{2k} \cdot OPT_{on}(\phi) + o(1). \tag{3}$$

We prove Equation (3) in the following two lemmas. The first proves that OPT_{on} run on $\mathcal{I}_\phi$ matches all arriving literal balls.

Lemma 3.4. *Algorithm OPT_{on} matches all arriving literal balls of $\mathcal{I}_\phi$.*

Proof. Suppose that there is some history h (occurring with probability $q > 0$) after which OPT_{on} does not match a literal ball t which arrives; let $\mathcal{A}'$ be the algorithm that follows exactly what OPT_{on} does, with the exception that it will match t if t arrives after the history h. Then,

$$\mathcal{A}'(\mathcal{I}_\phi) - OPT_{on}(\mathcal{I}_\phi) \geq q \cdot \left(1 - k \cdot \frac{m^4}{2k} \cdot m^{-4}\right) = q/2 > 0.$$

Indeed, if the history h occurs, $\mathcal{A}'$ gets a guaranteed profit of 1 from matching t that OPT_{on} does not receive. The expected profit OPT_{on} gets from having the additional bin available to be potentially matched to clause balls is at most $k \cdot \frac{m^4}{2k} \cdot m^{-4}$, since each literal bin has at most k clause balls adjacent to it, each of which has value $\frac{m^4}{2k}$ and arrives with probability m^{-4}. As the above would imply $\mathcal{A}(\mathcal{I}_\phi) > OPT_{on}(\mathcal{I}_\phi)$, we conclude that OPT_{on} must match each literal ball that arrives. □

A simple corollary of the above is that OPT_{on} gets value of $0.75n$ in expectation from the literal balls it matches. Moreover, this lemma gives a natural correspondence between OPT_{on} on $\mathcal{I}_\phi$ and algorithms for ϕ. The following lemma relies on Lemma 3.4 to bound the value OPT_{on} obtains from the clause balls in terms of the expected number of clauses of ϕ satisfied by OPT_{on}.

Lemma 3.5. *Let B be the gain of OPT_{on} from clause balls of $\mathcal{I}_\phi$. Then, for some $\delta \in [0, 2m^{-1}]$,*

$$\mathbb{E}[B] = \frac{(1 - m^{-4})^{m-1}}{2k} \cdot OPT_{on}(\phi) + \delta.$$

Proof. By Lemma 3.4, OPT_{on} matches each arriving literal ball. We consider the following natural mapping between MAX-SSAT algorithms $\mathcal{A}$ on ϕ and families of algorithms $\mathcal{F}_\mathcal{A}$ which match each literal ball in $\mathcal{I}_\phi$. For odd $t \leq n$, an algorithm $\mathcal{A}' \in \mathcal{F}_\mathcal{A}$ matches ball t to bin $\overline{x_t}$ (x_t) iff algorithm $\mathcal{A}$ sets x_t to `True` (`False`). For even $t \leq n$, if ball t arrives, an algorithm $\mathcal{A}' \in \mathcal{F}_\mathcal{A}$ matches ball t to bin x_t; this corresponds to nature setting $x_t =$ `False`. Otherwise, bin x_t is unmatched up to time $m+1$, and we will think of this as nature setting $x_t =$ `True`. (Note that ball t arrives with probability 50%, so the variables are set to `True`/`False` with the correct probability.) Finally, algorithms $\mathcal{A}' \in \mathcal{F}_\mathcal{A}$ match each arriving clause ball to some available neighboring bin when possible. A simple exchange argument shows that $OPT_{on}(\mathcal{I}_\phi) \in \mathcal{F}_\mathcal{A}$ for some algorithm $\mathcal{A}$.

Let C be the number of clause balls of $\mathcal{I}_\phi$ that arrive. Then, with probability $\Pr[C = 1] = m \cdot m^{-4} \cdot (1 - m^{-4})^{m-1}$, exactly one such clause ball arrives, equally likely to correspond to any of the m clauses in ϕ. On the other hand, a literal x_t (respectively, $\overline{x_t}$) is unmatched by $\mathcal{A}' \in \mathcal{F}_\mathcal{A}$ immediately prior to time $m+1$ iff $\mathcal{A}(\phi)$ or nature set x_t to `True` (respectively, `False`). We conclude that Algorithm $\mathcal{A}' \in \mathcal{F}_\mathcal{A}$ gains $\frac{\mathcal{A}(\phi)}{m} \cdot \frac{m^4}{2^k}$ expected value from conditioned on a single clause ball arriving. Thus, the expected gain $\mathbb{E}[B]$ of $OPT_{on}(\mathcal{I}_\phi)$ from clause balls is at least

$$\mathbb{E}[B] \geq \mathbb{E}[B \mid C = 1] \cdot \Pr[C = 1] = \frac{(1 - m^{-4})^{m-1}}{2k} \cdot OPT_{on}(\phi). \tag{4}$$

Let $\mathcal{A}$ be the MAX-SSAT algorithm for which $OPT_{on}(\mathcal{I}_\phi) \in \mathcal{F}_\mathcal{A}$. By the above argument yielding Equation (4), the expected gain of $OPT_{on}(\mathcal{I}_\phi)$ from clause balls conditioned on $C = 1$ is precisely

$$\Pr[B \mid C = 1] = \frac{\mathcal{A}(\phi)}{m} \cdot \frac{m^4}{2k} \leq \frac{OPT_{on}(\phi)}{m} \cdot \frac{m^4}{2k}. \tag{5}$$

Next, we note that the probability that multiple clause balls arrive is inverse polynomial in m.

$$\Pr[C \geq 2] = \sum_{t=2}^{m} \binom{m}{t} m^{-4t}(1 - m^{-4})^{m-t} \leq \sum_{t=2}^{m} m^t \cdot m^{-4t} \leq m^{-6} + m \cdot m^{-9} \leq 2m^{-6}. \tag{6}$$

On the other hand, conditioned on at multiple clause balls arriving, the expected profit of OPT_{on} from clause balls is at most

$$\mathbb{E}[B \mid C \geq 2] \leq m \cdot \frac{m^4}{2k} \leq m^5. \tag{7}$$

Combining equations (5), (6) and (7), we find that the expected gain of $OPT_{on}(\mathcal{I}_\phi)$ from matching clause balls is at most

$$\begin{aligned}\mathbb{E}[B] &= \mathbb{E}[B \mid C = 1] \cdot \Pr[C = 1] + \mathbb{E}[B \mid C \geq 2] \cdot \Pr[C \geq 2] \\ &\leq \frac{OPT_{on}(\phi)}{m} \cdot \frac{m^4}{2k} \cdot m \cdot m^{-4}(1 - m^{-4})^{m-1} + m^5 \cdot 2m^{-6} \\ &= \frac{(1 - m^{-4})^{m-1}}{2k} \cdot OPT_{on}(\phi) + 2m^{-1}. \end{aligned}$$ □

We now conclude the reduction, and obtain the proof of our hardness result.

Proof of Theorem 1.1. Let $\alpha \in (0, 1)$ be the constant from the statement of Lemma 3.1 and ϕ be a MAX-SSAT instance as in the statement of that lemma. Without loss of generality, we assume that ϕ has no pairs of consecutive variables x_{2k-1} and x_{2k} which appear in no clauses. (Else, we remove these variable pairs and relabel the remaining variables while preserving parity of indices. This does not change the clauses, nor does it change the expected number of clauses satisfied by OPT_{on}.) Next, let $\mathcal{I}_\phi$ be the obtained RideHail instance from the (clearly polynomial-time) reduction of this section. From Lemma 3.4, the expected gain of $OPT_{on}(\mathcal{I}_\phi)$ from literal balls is $0.75n$. Combining this with Lemma 3.5 we find that for $\gamma := \frac{(1-m^{-4})^{m-1}}{2k}$ and some $\delta \in [0, 2m^{-1}]$,

$$OPT_{on}(\mathcal{I}_\phi) = 0.75n + \gamma \cdot OPT_{on}(\phi) + \delta.$$

Next, since ϕ is a 3-CNF formula with at least half its variables appear in at least one clause, the number of variables is at most $n \leq 6m$. Moreover, since setting all variables randomly satisfies at least half of the clauses in expectation, we have $m/2 \leq OPT_{on}(\phi)$. Combining these two observations, we get

$$0.75n < n \leq 12 \cdot OPT_{on}(\phi), \tag{8}$$

Next, let $Q = \gamma \cdot OPT_{on}(\phi) + \delta$, $Q' = 0.75n$, and $\beta = \frac{12}{\gamma}$. Note that $Q'/Q \leq \beta$ by Equation (8), and that $\beta = O(1)$, since $k = O(1)$. Therefore, by Fact 2.1, for the constant $\alpha' := (\frac{\alpha\cdot(\gamma+2m^{-1})/\gamma+\beta}{1+\beta})$, which is in the range $(0, 1)$ for sufficiently large m, an α'-approximation to $OPT_{on}(\mathcal{I}_\phi) = OPT_{on}(\phi) + 0.75n = Q+Q'$ yields an $\alpha\cdot(\gamma+2m^{-1})/\gamma$-approximation of $Q \in [\gamma\cdot OPT_{on}(\phi), (\gamma+2m^{-1})\cdot OPT_{on}(\phi)]$. By scaling appropriately, this yields an α-approximation to $OPT_{on}(\phi)$, which is PSPACE-hard to obtain, by Lemma 3.1. The theorem follows. □

4 Algorithmic Results

In this section we give an algorithm to approximate the profit of OPT_{on}, for any joint distributions over edge weights of each ball t.

Theorem 1.2. *There exists a polynomial-time online algorithm which is a* 0.51*-approximation of the optimal online algorithm for the* RideHail *problem.*

An LP Relaxation. Our starting point is a linear program (LP) called LP-Match, which we show upper bounds the gain of any online algorithm for RideHail. Below, the variables we optimize over are $\{x_{i,t}\}$, which we think of as "the probability that the online algorithm matches ball t to bin i". Recall that ball t arrives with probability p_t.

$$
\begin{aligned}
\textbf{LP-Match:} \qquad & \max \quad \sum_{i,t} w_{i,t} \cdot x_{i,t} & & \\
& \text{s.t.} \quad \sum_{t} x_{i,t} \leq 1 & \text{for all } i & \qquad (9) \\
& \sum_{i} x_{i,t} \leq p_t & \text{for all } t & \qquad (10) \\
& x_{i,t} \leq p_t \cdot \left(1 - \sum_{t'<t} x_{i,t'}\right) & \text{for all } i,t & \qquad (11) \\
& x_{i,t} \geq 0 & \text{for all } i,t & \qquad (12)
\end{aligned}
$$

Denoting by LP-Match($\mathcal{I}$) the optimal value of LP-Match on Instance $\mathcal{I}$, we have the following.

Lemma 4.1. *For any* RIDEHAIL *instance $\mathcal{I}$, we have that*

$$\text{LP-Match}(\mathcal{I}) \geq OPT_{on}(\mathcal{I}).$$

Proof. Let $x^*_{i,t}$ denote the probability that OPT_{on} matches bin i to ball t. We note that x^* constitutes a feasible solution for LP-Match because (i) the probability OPT_{on} matches a bin i is at most 1, (ii) the probability OPT_{on} matches a ball t is at most p_t (the probability that t arrives), (iii) the probability OPT_{on} matches a bin i to a ball t is at most p_t (the probability t arrives) times $1 - \sum_{t'<t} x_{i,t'}$ (the probability that i is not matched by time t),[2] and (iv) these probabilities are non-negative. On the other hand, for this $x^*_{i,t}$, the objective of LP-Match is precisely the expected profit of OPT_{on} on this instance, and therefore LP-Mathc($\mathcal{I}$) $\geq OPT_{on}(\mathcal{I})$. □

4.1 The Algorithm

Given a solution to LP-Match, whose objective upper bounds OPT_{on} by Lemma 4.1, a natural approach to approximate OPT_{on} is to round this solution online. By simple "integrality gap" examples (see Appendix B), this is impossible to do perfectly. Instead, we show how to do so approximately, by rounding a solution to LP-Match while only incurring a $1/2 + c$ multiplicative loss in the rounding, for the constant $c := 0.01$.

For notational simplicity, assume without loss of generality that an optimal solution to LP-Match to the input instance $\mathcal{I}$ satisfies all Constraints (10) at equality, i.e., $\sum_i x_{i,t} = p_t$ for all balls t. This can be guaranteed by adding a dummy bin i_t for each ball t with $w_{i,t} = 0$, and setting $x_{i_t,t} \leftarrow p_t - \sum_i x_{i,t}$. These dummy edges do not affect the gain of OPT_{on}, nor that of the online algorithm.

After computing a solution to LP-Match as above, our algorithm proceeds iteratively as follows. For each time t, if ball t arrives, we pick a single bin i with probability $x_{i,t}/p_t$, and if this is bin is vacant (unmatched), we match (i,t) with some probability $q_{i,t}$. (We sometimes refer to this as i *accepts* t.) If this did not result in t being matched, we repeat the process a second time, but this time we match t to its picked bin i, provided i is vacant, and the edges until time t have nearly saturated Constraint (9) for i. See Algorithm 1.

[2] Here, we use the fact that arrival of t is independent of the online algorithm's previous choices. Note that this constraint is not valid for the probabilities induced by an *offline* algorithm, so our LP does not upper bound $OPT_{off}(\mathcal{I})$.

Algorithm 1 Rounding LP-Match Online

1: solve LP-Match for $\{x_{i,t}, y_{i,t,r}\}$
2: add dummy neighbor for each t so that $\sum_i y_{i,t,r} = p_{t,r}$ for all r
3: $\mathcal{M} \leftarrow \emptyset$
4: **for** all balls $t = 1, 2, \ldots$ **do**
5: let r be the index of the realization of $\{w_{i,t} = w_{i,t,r} \mid i\}$
6: pick a single bin i with probability $\frac{y_{i,t,r}}{p_{t,r}}$
7: **if** i is unmatched in $\mathcal{M}$ **then**
8: **with probability** $q_{i,t} := \min\left(1, \frac{1/2+c}{1-\sum_{t'<t} x_{i,t'}\cdot(1/2+c)}\right)$ **do**
9: $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,t)\}$
10: **if** t is still unmatched in $\mathcal{M}$ **then**
11: pick a single bin i with probability $\frac{y_{i,t,r}}{p_{t,r}}$
12: **if** i is unmatched in $\mathcal{M}$ **and** $\sum_{t'<t} x_{i,t'} > \frac{1/2-c}{1/2+c}$ **then**
13: $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,t)\}$
14: **Output** $\mathcal{M}$

By Constraint (10), Lines 6 and 11 are well-defined. Also, by Constraint (9), Line 9 is well-defined since $c < 1/2$. We also note that the algorithm clearly outputs a matching.

As we shall show, our Algorithm 1 fares well in comparison to $OPT_{on}(\mathcal{I})$. In particular, we will show the following per-edge guarantees.

Theorem 4.2. *Each edge $(i,t) \in E$ is matched by Algorithm 1 with probability at least*

$$\Pr[(i,t) \in \mathcal{M}] \geq x_{i,t} \cdot (1/2 + c).$$

Theorem 4.2 implies that our algorithm is a polynomial-time 0.51-approximation of the optimal online algorithm, thus proving Theorem 1.2.

Proof of Theorem 1.2. All steps of Algorithm 1, including solving the polynomially-sized LP in Line 1, can be implemented in polynomial time. The approximation ratio follows directly from linearity of expectation, together with Lemma 4.1 and Theorem 4.2. □

The remainder of this section is dedicated to proving Theorem 4.2. To this end, we consider two events for edge (i,t) being matched—depending on whether it was matched as a first pick or second pick, in Line 9 or Line 13, respectively. We bound the probability of an edge being matched either as a first pick or as a second pick in the following sections.

4.2 Analysis: First Pick

In this section we bound the probability of an edge being matched as a first pick. That is, the probability that edge (i,t) is added to $\mathcal{M}$ in Line 9. We start with the following useful definition.

Definition 4.3. Ball t is *early* for bin i if $\sum_{t'<t} x_{i,t'} \leq \frac{1/2-c}{1/2+c}$. Otherwise, it is *late*. Edge (i,t) is early (late) if t is early (late) for i. We use E_i and L_i to denote the early and late balls for i, respectively.

Intuitively, a ball is late for bin i if most balls t' (weighted by $x_{i,t'}$-value) precede t. Note that the early/late distinction determines whether or not the probability $q_{i,t}$ in Line 9 is 1. In particular, this probability is less than 1 only if (i,t) is early, and equal to 1 when (i,t) is late. We will use this observation frequently in the subsequent analysis.

For every (i,t), we let $V_{i,t}$ be an indicator random variable for the event that bin i is *vacant* (i.e., unmatched) at time t. We additionally let $\mathcal{M}_1 \subseteq \mathcal{M}$ denote the edges in $\mathcal{M}$ added as a result of a bin i accepting a ball's first pick (i.e., in Line 9), and $\mathcal{M}_2 \subseteq \mathcal{M}$ denote the edges in $\mathcal{M}$ added as a result of a bin i accepting a ball's second pick (i.e., in Line 13). Note that $\mathcal{M} = \mathcal{M}_1 \sqcup \mathcal{M}_2$.

The next lemma bounds the probability of an edge (i,t) being matched as a first pick (in Line 9).

Lemma 4.4. *If edge* $(i,t) \in E$ *is early, then*

$$\Pr[(i,t) \in \mathcal{M}_1] = x_{i,t} \cdot (1/2 + c).$$

In addition, for any edge $(i,t) \in E$,

$$x_{i,t} \cdot (1/2 - 3c) \le \Pr[(i,t) \in \mathcal{M}_1] \le x_{i,t} \cdot (1/2 + c).$$

Proof. Fix i. We prove by strong induction that these bounds hold for all edges (i,t') with $t' < t$. The base case, for $t = 1$, is vacuously true. Assume the claim holds for all $t' < t$; we will prove it holds for t as well.

The event $(i,t) \in \mathcal{M}_1$ requires that ball t arrives and bin i is picked in Line 6, that bin i is vacant at time t, and that bin i accepts the offer. Note that i being vacant at time t is independent from the arrival of t, and the first pick of t. Therefore,

$$\Pr[(i,t) \in \mathcal{M}_1] = x_{i,t} \cdot \Pr[V_{i,t}] \cdot q_{i,t}. \tag{13}$$

For this reason, we turn our attention to bounding the probability of i being vacant at time t,

$$\Pr[V_{i,t}] = 1 - \sum_{t'<t} \Pr[(i,t') \in \mathcal{M}] = 1 - \sum_{t'<t} \Pr[(i,t') \in \mathcal{M}_1] - \sum_{t'<t} \Pr[(i,t') \in \mathcal{M}_2]. \tag{14}$$

First, the inductive hypothesis and the definition of $x_{i,t}$ imply the following upper bound on $\Pr[V_{i,t}]$.

$$\Pr[V_{i,t}] \le 1 - \sum_{r} \sum_{\substack{t'<t,\\ t' \in E_i}} \Pr[(i,t') \in \mathcal{M}_1] = 1 - \sum_{\substack{t'<t,\\ t' \in E_i}} x_{i,t'} \cdot (1/2 + c). \tag{15}$$

If (i,t) is early, this bound is tight because (i,t') is early for any $t' < t$; hence, for early (i,t) we have that $\Pr[V_{i,t}] = 1 - \sum_{t'<t} x_{i,t'} \cdot (1/2 + c)$. Recalling that $q_{i,t} = \frac{1/2+c}{1-\sum_{t'<t} x_{i,t'} \cdot (1/2+c)}$ for early (i,t), Equation (13) then implies that $\Pr[(i,t) \in \mathcal{M}_1] = x_{i,t} \cdot (1/2 + c)$ for early edges (i,t).

If (i,t) is late, then $\sum_{t'<t, t' \in E_i} x_{i,t'} = \sum_{t' \in E_i} x_{i,t'} \ge \frac{1/2-c}{1/2+c}$. Hence, by Equation (15) we have that

$$\Pr[V_{i,t}] \le 1 - \left(\frac{1/2 - c}{1/2 + c}\right) \cdot (1/2 + c) = 1/2 + c. \tag{16}$$

Again, Equation (13) then implies that $\Pr[(i,t) \in \mathcal{M}_1] \le x_{i,t} \cdot (1/2 + c)$ for late edges (i,t).

Finally, we lower bound $\Pr[(i,t) \in \mathcal{M}_1]$ for late (i,t). To do so, we lower bound $\Pr[V_{i,t}]$; here, our analysis must account for the fact that late edges can be matched in either $\mathcal{M}_1$ or $\mathcal{M}_2$. Hence, we first note that for any $t' < t$ that is late for i, we have, similarly to Equation (15) that the probability of edge (i,t') being matched as a second pick is at most

$$\Pr[(i,t') \in \mathcal{M}_2] \le x_{i,t'} \cdot \Pr[V_{i,t'}] \le x_{i,t'} \cdot (1/2 + c). \tag{17}$$

Now, combining equations (14) and (17), we lower bound $\Pr[V_{i,t}]$ as follows:

$$\Pr[V_{i,t}] \ge 1 - \sum_{t'<t} x_{i,t'} \cdot (1/2 + c) - \sum_{\substack{t'<t,\\ t' \in L_i}} x_{i,t'} \cdot (1/2 + c) \ge 1 - (1/2 + c) - \left(1 - \frac{1/2 - c}{1/2 + c}\right) \cdot (1/2 + c),$$

which simplifies to

$$\Pr[V_{i,t}] \ge 1/2 - 3c. \tag{18}$$

Again, Equation (13) then implies that $\Pr[(i,t) \in \mathcal{M}_1] \ge x_{i,t} \cdot (1/2 - 3c)$. □

The proof of Lemma 4.4 yields upper and lower bounds on $\Pr[V_{i,t}]$ (equations (15), (16) and (18)), which will prove useful later. For convenience, we extract these bounds in the following corollary.

Corollary 4.5. *For any edge* (i,t)*, we have that* $\Pr[V_{i,t}] \ge 1/2 - 3c$*. For any late* (i,t)*, we have that* $\Pr[V_{i,t}] \le 1/2 + c$*. For any early* (i,t)*, we have that* $\Pr[V_{i,t}] = 1 - \sum_{t'<t} x_{i,t'} \cdot (1/2 + c)$.

Given Lemma 4.4, in order to prove Theorem 4.2, we wish to prove that the second attempt of t to match will ensure late edges (i,t) a probability of at least $x_{i,t} \cdot 4c$ of being matched. This is the meat of our analysis, and the next section is dedicated to its proof.

4.3 Analysis: Second Pick

In this section we prove that the second pick of ball t, in Lines 10-13, does indeed increase the probability of late edges (i, t) to be matched. In particular, we prove the following theorem.

Theorem 4.6. *For any late edge $(i, t) \in E$,*

$$\Pr[(i, t) \in \mathcal{M}_2] \geq x_{i,t} \cdot 4c.$$

Before proving the above theorem, we provide some useful intuition and outline the challenges the proof of Theorem 4.6 needs to overcome.

By Lemma 4.4, the probability of a late edge (i, t) being matched as a first pick is at least

$$\Pr[(i, t) \in \mathcal{M}_1] \geq x_{i,t} \cdot (1/2 - 3c). \tag{19}$$

Moreover, by the same lemma, each edge $(i, t) \in E$ (whether early or late) is matched as a first pick with probability at most $\Pr[(i, t) \in \mathcal{M}_1] \leq x_{i,t} \cdot (1/2 + c)$. Denote by A_t the event that t arrives and denote by $U_1(t)$ the event that t is unmatched after its first pick of $i_1 = j$. Then, we have

$$\Pr[U_1(t) \mid A_t, i_1 = j] = 1 - \Pr[V_{j,t}] \cdot q_{j,t}.$$

If (j, t) is late, then because $\Pr[V_{j,t}] \leq 1/2 + c$ by Corollary 4.5, the above quantity is at least $1/2 - c$. If (j, t) is early, then because $\Pr[V_{j,t}] = 1 - \sum_{t'<t} x_{j,t'}(1/2 + c)$, by Corollary 4.5, combined with the definition of $q_{j,t}$, we have that the above quantity is exactly equal to $1/2 - c$. In summary,

$$\Pr[U_1(t) \mid A_t, i_1 = j] \geq 1/2 - c. \tag{20}$$

Now, we recall that for late edges (i, t), we have that $q_{i,t} = 1$. So, a late edge (i, t) is matched iff i is vacant by time t and i is picked in Line 6 or Line 11. One might then be tempted to guess that $\Pr[(i, t) \in \mathcal{M}_2 \mid U_1(t)]$ is equal to $\Pr[(i, t) \in \mathcal{M}_1]$, which by (19) and (20) would imply that $\Pr[(i, t) \in \mathcal{M}_2] \geq x_{i,t} \cdot (1/2 - c) \cdot (1/2 - 3c) \geq x_{i,t} \cdot 4c$ (the last inequality using $c \leq 0.01$), as desired.

4.3.1 The Key Challenges

There are two key issues with the simplifistic argument above.

Challenge 1: Re-drawing i. Unfortunately, conditioning on $U_1(t)$ does not result in the probability of (i, t) being matched in the second pick equalling that of it being matched in the first pick. To see this, suppose a ball t was late for a single bin i, and $x_{i,t}/p_t = 1$. In that case, conditioning on $U_1(t)$ is equivalent to conditioning on i being occupied (matched) before time t. Consequently, for this late edge (i, t), we have that $\Pr[(i, t) \in \mathcal{M}_1] \geq x_{i,t} \cdot (1/2 - 3c)$ by Lemma 4.4, while $\Pr[(i, t) \in \mathcal{M}_2 \mid U_1(t)] = 0$, which implies that the second pick does not increase the probability of (i, t) to be matched *at all*, as $\Pr[(i, t) \in \mathcal{M}_2] = 0$(!).

This is where Constraint (11) of LP-Match comes in: This constraint implies that if t is late for bin i, then the probability that i was picked in Line 6 at time t conditioned on arrival of t is at most

$$\frac{x_{i,t}}{p_t} \leq 1 - \sum_{t'<t} x_{i,t'} \leq 1 - \frac{1/2 - c}{1/2 + c} = \frac{2c}{1/2 + c} \leq 4c.$$

This implies that there is a (high) constant probability of i not being picked in Line 6.

Lemma 4.7. *For any late edge (i, t), for i_1 the bin picked in Line 6 at time t,*

$$\Pr[i_1 \neq i \mid A_t] \geq 1 - 4c.$$

Challenge 2: Positive Correlation Between Bins. Lemma 4.7 alone does not resolve our problems. Suppose that ball t is late for all bins for which $x_{i,t} \neq 0$, and all these bins have perfectly positively correlated matched status, i.e., $V_{i,t} = V_{j,t}$ for all bins i, j always. If this were the case, then we would have that $\Pr[V_{i,t} \mid U_1(t)] = 0$, since if t is not matched to its first i_1, then i_1 and i must both have been matched before. This again would result in $\Pr[(i,t) \in \mathcal{M}_2] = 0$.

To overcome the above, we show that the above scenario does not occur. In particular, we show that while positive correlations between different bins' matched statuses are possible, such correlations cannot be too large. More formally, we show the following.

Lemma 4.8. *For any time t and bins $i \neq j$, we have that*

$$\mathrm{Cov}(V_{i,t}, V_{j,t}) \leq 12c.$$

The crux of our analysis is proving Lemma 4.8. Using it, we will be able to argue that for any late edge (i,t), the probability that i is free at time t, conditioned on $U_1(t)$ and on the first pick satisfying $i_1 \neq i$ (a likely event, by Lemma 4.7), is not changed much compared to the unconditional probability of i being free at time t. In particular, this implies that the probability of (i,t) being matched as a second pick, conditioned on $U_1(t)$, is not too much smaller compared to its probability of being matched as a first pick. In particular, we will show that $\Pr[(i,t) \in \mathcal{M}_2] \geq x_{i,t} \cdot 4c$, for sufficiently small $c > 0$, as stated in Theorem 4.6.

We prove that lemmas 4.7 and 4.8 indeed imply Theorem 4.6, as outlined above, in Section 4.3.3. But first, we turn to proving our key technical lemma, namely Lemma 4.8.

4.3.2 Bounding Correlations of Occupancies

To bound the correlation of vacancy indicators, it is convenient to define the indicator random variable $O_{i,t} := 1 - V_{i,t}$, which indicate whether i is occupied (i.e., matched) at time t. We additionally decompose the variables $O_{i,t}$ into two variables, based on whether i was matched (became occupied) along an early or late edge. In particular, we let $O^E_{i,t} \leq O_{i,t}$ be an indicator for the event that i is matched along an early edge before t arrives. Similarly, we let $O^L_{i,t} := O_{i,t} - O^E_{i,t}$ be an indicator for the event that i is matched along a late edge before t arrives. To bound the pairwise correlations of variables $O_{i,t}$, we will show that $O^E_{i,t}$ contributes most of the probability mass of $O_{i,t}$, and that the variables $O^E_{i,t}$ and $O^E_{j,t}$ are negatively correlated. To prove this negative correlation, we will prove the following, stronger statement.

Lemma 4.9. *For any time t, the variables $\{O^E_{i,t}\}_i$ are negatively associated (NA).*

Proof. For every edge (i,t), let $X_{i,t}$ be the indicator random variable for the event that ball t arrives and picks bin i as its first pick. Let $Y_{i,t} \sim \mathrm{Ber}(q_{i,t})$ be an indicator for the event that bin i accepts, i.e., it will be matched to ball t if it arrives and picks i as its first pick and i is free.

For fixed t, the variables $\{X_{i,t}\}$ are 0/1 random variables whose sum is at most 1 always, so they are NA by the 0-1 Principle (Proposition 2.6). On the other hand, the variables $\{Y_{i,t}\}_i$ are independent, and hence NA. Moreover, $\{X_{i,t}\}_i$, $\{Y_{i,t}\}_i$ are mutually independent distributions, and so by closure of NA under independent union (Proposition 2.7), we also have that $\{X_{i,t}, Y_{i,t}\}_i$ is NA. Likewise, the lists $\{X_{i,t}, Y_{i,t}\}_i$ are mutually independent as we vary t; again using closure of NA under independent union we find that $\{X_{i,t}, Y_{i,t}\}_{i,t}$ are also NA.

Fix t. For each bin i, let t_i denote the largest $t' < t$ so that (i,t') is early. We note that bin i cannot be matched as a second pick to any $t' \leq t_i$. So, it is matched along an early edge before t arrives if and only if there are some $t' \leq t_i$ and r such that ball t' arrives and picks bin i, and bin i accepts the proposal (for the smallest such t', bin i is guaranteed to be free). Therefore, we have that

$$O^E_{i,t} = \bigvee_{t' \leq t_i} (X_{i,t'} \wedge Y_{i,t'}).$$

Note that we have written $\{O^E_{i,t}\}_i$ as the output of monotone non-decreasing functions defined on disjoint subsets of the variables in $\{X_{i,t}, Y_{i,t}\}_{i,t}$. Hence, by closure of NA under monotone function composition (Proposition 2.7), we have that $\{O^E_{i,t}\}_i$ are NA. □

By Proposition 2.8, the above lemma implies that any $O^E_{i,t}$ and $O^E_{j,t}$ are negatively correlated.

Corollary 4.10. *For any time t and bins $i \neq j$, we have that* $\mathrm{Cov}(O^E_{i,t}, O^E_{j,t}) \leq 0$.

We are now ready to prove Lemma 4.8.

Proof of Lemma 4.8. First, we show that the probability of a bin i being matched along a late edge before time t is small, which we later use to bound the covariance of $O^L_{i,t}$ and other binary variables. Indeed, as $\Pr[V_{i,t}] \geq 1/2 - 3c$ (Corollary 4.5), we have that $\Pr[O_{i,t}] \leq 1/2 + 3c$. Additionally, $\Pr[O^E_{i,t}] \geq \frac{1/2-c}{1/2+c} \cdot (1/2 + c) = 1/2 - c$ by Lemma 4.4. Therefore,

$$\Pr[O^L_{i,t}] = \Pr[O_{i,t}] - \Pr[O^E_{i,t}] \leq (1/2 + 3c) - (1/2 - c) = 4c. \tag{21}$$

Thus, using the additive law of covariance for $\mathrm{Cov}(O_{i,t}, O_{j,t}) = \mathrm{Cov}(1-O_{i,t}, 1-O_{j,t}) = \mathrm{Cov}(V_{i,t}, V_{j,t})$, we obtain the desired bound,

$$\begin{aligned}
\mathrm{Cov}(V_{i,t}, V_{j,t}) &= \mathrm{Cov}(O^E_{i,t} + O^L_{i,t}, O^E_{j,t} + O^L_{j,t}) \\
&= \mathrm{Cov}(O^E_{i,t}, O^E_{j,t}) + \mathrm{Cov}(O^E_{i,t}, O^L_{j,t}) + \mathrm{Cov}(O^L_{i,t}, O^E_{j,t}) + \mathrm{Cov}(O^L_{i,t}, O^L_{j,t}) \\
&\leq 0 + \Pr[O^E_{i,t}, O^L_{j,t}] + \Pr[O^L_{i,t}, O^E_{j,t}] + \Pr[O^L_{i,t}, O^L_{j,t}] && \text{(Cor. 4.10)} \\
&\leq 0 + \Pr[O^L_{j,t}] + \Pr[O^L_{i,t}] + \Pr[O^L_{i,t}] \\
&\leq 12c. && \text{(Eq. (21))}
\end{aligned}$$
□

4.3.3 Putting it All Together

We are now ready to use weak positive correlation (if any) between vacancy indicators $V_{i,t}$ and $V_{j,t}$. In particular, we will show that the probability of bin i to be occupied a time t is not changed much when conditioning on A_t (arrival of t), the first picked bin at time t being $i_1 \neq i$, and $U_1(t)$ (ball t bot being matched to its first pick).

Lemma 4.11. *For any late edge (i, t), we have that*

$$\Pr[O_{i,t} \mid A_t, i_1 \neq i, U_1(t)] \leq \Pr[O_{i,t}] \cdot \left(1 + \frac{12c}{(1/2 - c)^2}\right).$$

Proof. To analyze the conditional probability above, we first look at $\Pr[O_{i,t}, A_t, i_1 = j, U_1(t)]$. This is the probability of bin i being occupied at time t, ball t arriving and picking j as its first pick, and not being matched due to this first pick. Note that A_t and the first pick is independent of bins' occupancy statuses at time t. Additionally, we notice that with probability $1 - q_{j,t}$ bin j will deterministically reject. With probability $q_{j,t}$, it rejects if and only if j is occupied. So, for any $j \neq i$,

$$\Pr[O_{i,t}, A_t, i_1 = j, U_1(t)] = \Pr[O_{i,t}] \cdot \Pr[A_t, i_1 = j] \cdot ((1 - q_{j,t}) + q_{j,t} \cdot \Pr[O_{j,t} \mid O_{i,t}]). \tag{22}$$

We now turn to relating the last term in the above product, namely $(1-q_{j,t})+q_{j,t} \cdot \Pr[O_{j,t} \mid O_{i,t}]$, to its "unconditional" counterpart, $\Pr[U_1(t) \mid A_t, i_1 = j] = (1 - q_{j,t}) + q_{j,t} \cdot \Pr[O_{j,t}]$. For notational convenience, we which we abbreviate by

$$z_{i,j,t} := (1 - q_{j,t}) + q_{j,t} \cdot \Pr[O_{j,t} \mid O_{i,t}].$$

Recalling that $\mathrm{Cov}(O_{i,t}, O_{j,t}) = \mathrm{Cov}(V_{i,t}, V_{j,t}) \le 12c$, by Lemma 4.8, we have

$$\Pr[O_{j,t} \mid O_{i,t}] = \frac{\Pr[O_{j,t}, O_{i,t}]}{\Pr[O_{i,t}]} = \frac{\Pr[O_{j,t}] \cdot \Pr[O_{i,t}] + \mathrm{Cov}(O_{j,t}, O_{i,t})}{\Pr[O_{i,t}]} \le \Pr[O_{j,t}] + \frac{12c}{\Pr[O_{i,t}]}. \tag{23}$$

Hence,

$$\begin{aligned}
z_{i,j,t} &\le (1 - q_{j,t}) + q_{j,t} \cdot \left(\Pr[O_{j,t}] + \frac{12c}{\Pr[O_{i,t}]}\right) && \text{(Eq. (23))} \\
&\le (1 - q_{j,t}) + q_{j,t} \cdot \left(\Pr[O_{j,t}] + \frac{12c}{1/2 - c}\right) && \text{(Cor. 4.5, } c < 1/2\text{)} \\
&= \Pr[U_1(t) \mid A_t, i_1 = j] + q_{j,t} \cdot \frac{12c}{1/2 - c} \\
&\le \Pr[U_1(t) \mid A_t, i_1 = j] \cdot \left(1 + \frac{12c}{(1/2 - c)^2}\right) && \text{(Eq. (20), } q_{j,t} \le 1\text{)} \qquad (24)
\end{aligned}$$

Using this bound in Equation (22) and summing over all $j \ne i$, we have

$$\Pr[O_{i,t}, A_t, i_1 \ne i, U_1(t)] \le \Pr[O_{i,t}] \cdot \Pr[A_t, i_1 \ne i, U_1(t)] \cdot \left(1 + \frac{12c}{(1/2 - c)^2}\right).$$

The desired inequality therefore follows by Bayes' theorem. □

With this lemma in place, we are ready to conclude this section by proving Theorem 4.6, i.e. that $\Pr[(i,t) \in \mathcal{M}_2] \ge x_{i,t} \cdot 4c$ for any late edge (i,t).

Proof of Theorem 4.6. We start by bounding

$$\Pr[(i,t) \in \mathcal{M}_2] \ge \Pr[(i,t) \in \mathcal{M}_2 \mid A_t, i_1 \ne i, U_1(t)] \cdot \Pr[A_t, i_1 \ne i, U_1(t)]. \tag{25}$$

In words, the probability (i,t) is matched as a second pick is at least the probability of the same event and $i_1 \ne i$. By Lemma 4.7 we know that $\Pr[A_t, i_1 \ne i] \ge p_t \cdot (1 - 4c)$; by Equation (20), we know that $\Pr[U_1(t) \mid A_t, i_1 = j] \ge 1/2 - c$ for any $j \ne i$. As a consequence, by Bayes' theorem and our choice of $c < 1/4$, we have that

$$\Pr[A_t, i_1 \ne i, U_1(t)] = \Pr[A_t, i_1 \ne i] \cdot \Pr[U_1(t) \mid A_t, i_1 \ne i] \ge p_t \cdot (1 - 4c) \cdot (1/2 - c). \tag{26}$$

Next, we note that

$$\Pr[(i,t) \in \mathcal{M}_2 \mid A_t, i_1 \ne i, U_1(t)] = \frac{x_{i,t}}{p_t} \cdot \Pr[V_{i,t} \mid A_t, i_1 \ne i, U_1(t)] \tag{27}$$

because conditioned on A_t, picking someone other than i first, and being rejected, we will match (i,t) exactly when t's second pick is i and i is vacant.

Lemma 4.11 yields the following lower bound on the probability of $[V_{i,t} \mid A_t,\, i_1 \ne i, U_1(t)]$:

$$\begin{aligned}
\Pr[V_{i,t} \mid A_t, i_1 \ne i, U_1(t)] &= 1 - \Pr[O_{i,t} \mid A_t, i_1 \ne i, U_1(t)] \\
&\ge 1 - \Pr[O_{i,t}] \cdot \left(1 + \frac{12c}{(1/2 - c)^2}\right) \\
&= \Pr[V_{i,t}] - \frac{12c}{(1/2 - c)^2} \cdot (1 - \Pr[V_{i,t}]) \\
&\ge 1/2 - 3c - \frac{12c}{(1/2 - c)^2} \cdot (1/2 + 3c) && \text{(Cor. 4.5)} \qquad (28)
\end{aligned}$$

Combining equations 27 and 28 we thus have

$$\Pr[(i,t) \in \mathcal{M}_2 \mid A_t, i_1 \ne i, U_1(t)] \ge \frac{x_{i,t}}{p_t} \cdot \left(1/2 - 3c - \frac{12c}{(1/2 - c)^2} \cdot (1/2 + 3c)\right). \tag{29}$$

Putting it all together, equations (25), (26), and (29) and our choice of (sufficiently small) $c = 0.01$ imply the desired inequality,

$$\Pr[(i,t) \in \mathcal{M}_2] \geq \frac{x_{i,t}}{p_t} \cdot \left(1/2 - 3c - \frac{12c}{(1/2 - c)^2} \cdot (1/2 + 3c) \right) \cdot p_t \cdot (1-4c) \cdot (1/2 - c) \geq x_{i,t} \cdot 4c. \quad \square$$

5 Generalizing the Algorithm

Our algorithm and its analysis of Section 4 generalize seamlessly to a setting in which weights of each online node t are drawn from discrete joint distributions. For brevity, we only outline the small changes in the LP, algorithm and analysis here.

Problem Statement. We are given a complete bipartite graph, with vertices of one side (bins) give up front, and vertices of the other side (balls) arriving sequentially, with ball t arriving at time t (with probability one). The vector of edge weights of any ball t, denoted by $w^t := (w_{1,t}, w_{2,t}, \dots)$, is drawn from some discrete joint distribution, $w^t \sim \mathcal{D}_t$. The vector of all edge weights, $w := (w^1, w^2, \dots)$, is drawn from the product distribution, $w \sim \mathcal{D} := \prod_t \mathcal{D}_t$. That is, the weights of any ball's edges may be arbitrarily correlated, but weights of different balls' edges are independent. We assume that these discrete distributions are given explicitly, e.g., via a list of tuples of the form $(v_{t,j}, p_{t,j})$ with $p_{t,j} := \Pr_{\mathcal{D}_t}[w^t = v_{t,j}]$. We note that the problem considered in previous sections is a special instance of this problem with each $\mathcal{D}_t$ consisting of two-point distributions, with one of the possible realizations of $w^t \sim \mathcal{D}_t$ being the all-zeros vector.

Generalizing LP-Match. The generalization of LP-Match now has decision variables $y_{i,t,j}$, which we think of as proxies for the probability of edge (i,t) being matched by the optimal online algorithm when ball t's edge weights are $w^t = v_{t,j}$. Generalizing the argument behind Constraint (11), we note that w^t is independent of bin i not being matched by the optimal online algorithm by time t. From this we obtain Constraint (30) below. The remaining constraints of the obtained LP (below) are matching constraints.

LP-Match:

$$\begin{aligned} \max \quad & \sum_{i,t,j} w_{i,t,j} \cdot y_{i,t,j} & \\ \text{s.t.} \quad & \sum_t \sum_j y_{i,t,j} \leq 1 & \text{for all } i \\ & \sum_i y_{i,t,j} \leq p_{t,j} & \text{for all } t, j \\ & y_{i,t,j} \leq p_{t,j} \cdot \left(1 - \sum_{t'<t} \sum_{j'} y_{i,t',j'} \right) & \text{for all } i,t,j \qquad (30) \\ & y_{i,t,j} \geq 0 & \text{for all } i,t,j \end{aligned}$$

Generalizing the algorithm. Our general algorithm will match each edge (i,j) when $w^t = v_{t,j}$ with marginal probability at least probability

$$\Pr[(i,t) \in \mathcal{M}, w^t = v_{t,j}] \geq y_{i,t,j} \cdot (1/2 + c).$$

To do so, when ball t arrives, we first observe the realization of the edge weight vector $w^t = v_{t,j}$. Then, When picking a bin i (either as first or second pick) at time t, we now do so with probability

$\frac{y_{i,t,r}}{p_{t,j}}$. Moreover, we take $q_{i,t} := \min\left(1, \frac{1/2+c}{1-\sum_{t'<t}\sum_{j'} y_{i,t',j'}\cdot(1/2+c)}\right)$ to be the probability of a vacant picked bin i to be matched to ball t by the algorithm. The dummy nodes i_t are now assigned values $y_{i_t,t,j} \leftarrow p_{t,j} - \sum_i y_{i,t,j}$ for each j. Apart from this, the algorithm is unchanged. We note that this algorithm can be implemented in polynomial time in the size of the input (the representation of $\mathcal{D}$).

Generalizing the Analysis. Extending the analysis of Algorithm 1 to this more general problem is a rather simple syntactic generalization. We therefore only outline the changes in the analysis. Broadly, all changes needed for the analysis require us to refine our claims as follows. Denote by R_t a random variable denoting the random index of the weight vector of edges of t. That is, $R_t = j \iff w^t = v_{t,j}$. Then, all our bounds for the probability of (i,t) being matched (as a first or second pick, or either) now need to refer to $R_t = j$, and relate to $y_{i,t,j}$. So, for example, Lemma 4.4 will be restated to show that for each early edge (i,t) and index j, we have that $\Pr[(i,t) \in \mathcal{M}_1, R_t = j] = y_{i,t,r} \cdot (1/2 + c)$, and for any edge (i,t), we have that $y_{i,t,r} \cdot (1/2 - 3c) \leq \Pr[(i,t) \in \mathcal{M}_1, R_t = j] \leq y_{i,t,r} \cdot (1/2 + c)$. Lemma 4.9 requires some care in setting up the NA variables to prove that $O^E_{i,t}$ are NA, by also accounting for the realization of R_t, with indicators $[R_t = j]$, which are NA by the 0-1 Principle (Proposition 2.6). Apart from that, the proofs are essentially unchanged, except for replacing occurrences of A_t by $R_t = j$ in every probability conditioned on arrival of t, and appropriately replacing $\frac{x_{i,t}}{p_t}$ by $\frac{y_{i,t,j}}{p_{t,j}}$.

6 Conclusions and Open Questions

We studied the online stochastic max-weight bipartite matching problem through the lens of approximation algorithms, rather than that of competitive analysis. In particular, we study the efficient approximability of the optimal online algorithm on any given input. On the one hand, we show that the optimal online algorithm cannot be approximated beyond some constant (barring shocking developments in complexity theory). On the other hand, we present a polynomial-time online algorithm which yields a 0.51 approximation of the optimal online algorithm's gain—surpassing the approximability threshold of $1/2$ of the optimal offline algorithm. Many intriguing research questions remain.

First, it is natural to further study the efficient approximability of our problem. We suspect that much better approximation guarantees are achievable. One might also ask if our general algorithmic approach can be extended to *implicitly* represented weight distribution $\mathcal{D}$. For example, what can one show if $\mathcal{D}_t$ is itself a product distribution, $\mathcal{D}_t = \prod_i \mathcal{D}_{i,t}$, with $w_{i,t} \sim \mathcal{D}_{i,t}$? A related interesting question is to obtain better approximation for the widely-studied special case of balls drawn from some i.i.d distribution (see, e.g., [24, 29, 31, 37, 38]).

More broadly, one might ask how well one can approximate the optimal online algorithm of online Bayesian selection problems under the numerous constraints studied in the literature, including matroid and matroid intersections, knapsack constraints, etc. For which of these problems is the online optimum easy to compute? Which admit a PTAS? Which admit constant approximations? Which are hard to approximate? We are hopeful that the ideas developed here, both algorithmic, as well as our new hardness gadgets, will prove useful when exploring this promising research direction.

A Omitted Proofs of Section 2

In this section we provide proofs deferred from Section 2, restated below for ease of reference.

Fact 2.1. *Let $Q, Q' \geq 0$ be positive quantities, such that $Q'/Q \leq \beta$, and let $\alpha \in (0,1)$. Then, an $\left(\frac{\alpha+\beta}{1+\beta}\right)$-approximation to $Q + Q'$ yields an α-approximation to Q.*

Proof. As $f(x) = \frac{\alpha+x}{1+x} = 1 - \frac{1-\alpha}{1+x}$ is monotone increasing in $x \geq -1$ for $\alpha \in (0,1)$, we have that

$(\frac{\alpha+\beta}{1+\beta}) \geq \frac{\alpha+Q'/Q}{1+Q'/Q} = \frac{\alpha\cdot Q+Q'}{Q+Q'}$. Thus, An $(\frac{\alpha+\beta}{1+\beta})$-approximation to $Q+Q'$ yields a number in the range

$$T \in \left[\frac{\alpha+\beta}{1+\beta}\cdot\left(Q+Q'\right), Q+Q'\right] \subseteq [\alpha\cdot Q+Q', Q+Q'].$$

Subtracting Q' from T then yields a number $T-Q'$ in the range

$$T-Q' \in [\alpha\cdot Q, Q].$$

□

Next, we provide a proof of the underlying PSPACE-hardness result of Condon et al. [11] used in our reductions.

Lemma 2.3. *([11, Theorem 3.3]) There exist constants $k\in\mathbb{N}$ and $\alpha\in(0,1)$ so that it is PSPACE-hard to compute an α-approximation to $OPT_{on}(\phi)$ for a MAX-SSAT instance ϕ satisfying:*

1. *no random variable appears negated in any clause of ϕ, and*
2. *each random variables appears in at most k clauses of ϕ.*

Proof. This lemma follows from the proof in [11]; here, we briefly explain why.

In that paper, the authors prove their main result that RPCD$(\log n, 1)$ = PSPACE in Theorem 2.4. Using this theorem, they prove that it is PSPACE-hard to approximate MAX-SSAT in Theorem 3.1. In their proof, they start with a language L in PSPACE and an input x, and construct an RPCDS for L flipping $O(\log n)$ coins and reading $O(1)$ bits of the debate. From this, they construct a MAX-SSAT instance ϕ such that if $x\in L$, all clauses of ϕ can be satisfied with probability 1, while if $x\notin L$ there is no way to satisfy more than an $\alpha<1$ fraction of the clauses of ϕ. Their construction of ϕ builds a constant-size 3CNF for each possible realization of the $O(\log n)$ coin flips, and takes the conjunction of these 3CNFs. Each constant-size 3CNF has variables corresponding to the bits of the debate that V queries for a specific realization of the coin-flips. Hence, to show that ϕ only has each random variable appear in $O(1)$ clauses, it suffices to show that each random-bit in the RPCDS constructed is queried for only $O(1)$ realizations of the coin flips.

To show this, we turn to the construction of the RPCDS used to prove Theorem 2.4. Via Lemma 2.1, the authors first show that it is sufficient to consider RPCDSs where the verifier can read a constant number of *rounds* of Player 1 (and not just a constant number of bits).

In Lemma 2.3, the authors describe their protocol for a verifier V which can read $O(1)$ rounds of Player 1. Note that the random coins in this protocol are used to select a "random odd-numbered round $k>1$" and a "random bit of round $k-1$ of Player 0." In fact, this is the *only* time that the verifier reads a random bit of Player 0. So, in this construction, each random bit is only queried in $O(1)$ realizations of the coin flips. With Lemma 2.1, the authors transform this RPCDS to one that only reads a constant number of bits. We note that this transformation only impacts the strings that player 1 writes, and does not affect the coin flips or the bits of player 0 read.

From this, it holds that the MAX-SSAT instance ϕ constructed in Theorem 3.1 has each random variable appear in $O(1)$ clauses. That instance does not yet satisfy the property that random variables only appear non-negated. Condon et al. give a fix for this in the proof of Theorem 3.3; we briefly note that after the modification provided in this proof, it will still hold that random variables appear in $O(1)$ clauses. □

B LP-Match: Additional Observations

Here we make a few additional observations concerning the usefulness of Constraint (11) and LP-Match in general, as well as some natural limits to this LP.

First, we note that LP-Match captures the optimal online algorithm *precisely* for the classic single-item prophet inequality problem. That is, for RideHail instances with a single bin i, solutions to this LP can be rounded online losslessly.

Observation B.1. *LP-Match($\mathcal{I}$) $= OPT_{on}(\mathcal{I})$ for any* RideHail *instance $\mathcal{I}$ with a single bin i.*

Proof. Consider the following online algorithm, which starts by computing a solution $\vec{y}$ to LP-Match. Next, upon arrival of ball t with with $w_{i,t} = w_{i,t,r}$ (i.e., $R_t = r$), match (i,t) with probability

$$\frac{y_{i,t,r}}{p_{t,r} \cdot \left(1 - \sum_{t'<t} \sum_{r'} y_{i,t',r'}\right)}.$$

This last quantity is indeed a probability, by Constraint 11. A simple proof by induction shows that for each t and r, we have that $\Pr[(i,t) \in calM, R_t = r] = y_{i,t,r}$, and consequently $\Pr[F_{i,t}] = 1 - \sum_{t'<t} \sum_{r'} y_{i,t',r'}$, from which we obtain the inductive step, as

$$\Pr[(i,t) \in \mathcal{M}, R_t = r] = p_{t,r} \cdot \frac{y_{i,t,r}}{p_{t,r} \cdot \left(1 - \sum_{t'<t} r' y_{i,t',r'}\right)} \cdot \left(1 - \sum_{t'<t} \sum_{r'} y_{i,t',r'}\right) = y_{i,t,r}.$$

By linearity of expectation, this online algorithm for Instance has expected reward precisely

$$\sum_{i,t,r} w_{i,t,r} \cdot y_{i,t,r} = \text{LP-Match}(\mathcal{I}).$$

Consequently, $OPT_{on}(\mathcal{I}) \geq \text{LP-Match}(\mathcal{I})$. The opposite inequality follows from Lemma 4.1. □

On the other hand, for general RideHail instances, there is a limit to the approximation guarantees obtainable using LP-Match. In particular, simple examples show that there is a gap between the upper bound given by LP-Match and the expected profit of OPT_{on}, appropriately restricting the approximation guarantees provable using this LP. This is to be expected, given our work in Section 3. We present a simple example of such a gap instance below.

Observation B.2. *There exists a* RideHail *instance $\mathcal{I}$ with $w_{i,t} \in \{0,1\}$ for all $(i,t) \in E$ for which LP-Match($\mathcal{I}$) $\geq 8/7 \cdot OPT_{on}(\mathcal{I})$.*

Proof. We consider an instance $\mathcal{I}$ with three balls and two bins. For $k = 1,2$, ball $t = k$ has with probability $p_{k,0} = 1/2$ edge weights $w_{i,t} = 0$ for all i. With the remaining probability $p_{k,1} = 1/2$, its edges have weights $w_{k,k} = 1$ and $w_{k,3-k} = 0$. The last ball has weights $w_{3,k} = 1$ for all bins $k = 1,2$ with probability one. An optimal solution to LP-Match on this Instance $\mathcal{I}$ assigns $y_{k,k,1} = 1/2$ for $k = 1,2$, and $y_{3,k,1} = 1/2$ for $k = 1,2$, achieving an objective value of $\sum_{i,t,r} y_{i,t,r} = 2$. However, with probability $1/4$, both of the first two balls have all their edge weights zero, and so an online algorithm can at most achieve an expected value of $7/4$. That is, $OPT_{on}(\mathcal{I}) \leq 7/8 \cdot \text{LP-Match}(\mathcal{I})$. □

References

[1] Melika Abolhassani, Soheil Ehsani, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Robert Kleinberg, and Brendan Lucier. Beating 1-1/e for ordered prophets. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC)*, pages 61–71, 2017.

[2] Shipra Agrawal, Jay Sethuraman, and Xingyu Zhang. On optimal ordering in the optimal stopping problem. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 187–188, 2020.

[3] Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing (SICOMP)*, 43(2):930–972, 2014.

[4] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pages 18–35, 2012.

[5] Nima Anari, Rad Niazadeh, Amin Saberi, and Ali Shameli. Nearly optimal pricing algorithms for production constrained and laminar bayesian selection. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 91–92, 2019.

[6] Ali Aouad and Ömer Saritaç. Dynamic stochastic matching under limited time. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 789–790, 2020.

[7] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3): 501–555, 1998.

[8] Arash Asadpour, Michel X Goemans, Aleksander Mądry, Shayan Oveis Gharan, and Amin Saberi. An $O(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. *Operations Research*, 65(4):1043–1061, 2017.

[9] Shuchi Chawla, Jason D Hartline, David L Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 311–320, 2010.

[10] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1659–1667, 2016.

[11] Anne Condon, Joan Feigenbaum, Carsten Lund, and Peter Shor. Random debaters and the hardness of approximating stochastic functions. *SIAM Journal on Computing (SICOMP)*, 26 (2):369–400, 1997.

[12] José Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Posted price mechanisms for a random stream of customers. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 169–186, 2017.

[13] Jose Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Recent developments in prophet inequalities. *ACM SIGecom Exchanges*, 17(1):61–70, 2019.

[14] José Correa, Patricio Foncea, Dana Pizarro, and Victor Verdugo. From pricing to prophets, and back! *Operations Research Letters*, 47(1):25–29, 2019.

[15] Devdatt Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996.

[16] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Online stochastic max-weight matching: prophet inequality for vertex and edge arrival models. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 769–787, 2020.

[17] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 123–135, 2014.

[18] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1014–1033, 2016.

[19] Yiding Feng, Rad Niazadeh, and Amin Saberi. Two-stage stochastic matching with application to ride hailing. In *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2862–2877, 2021.

[20] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981.

[21] Buddhima Gamlath, Michael Kapralov, Andreas Maggiori, Ola Svensson, and David Wajc. Online matching with general arrivals. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS)*, pages 26–37, 2019.

[22] Ashish Goel, Sudipto Guha, and Kamesh Munagala. How to probe for an extreme value. *ACM Transactions on Algorithms (TALG)*, 7(1):1–20, 2010.

[23] Nikolai Gravin and Hongao Wang. Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 93–109, 2019.

[24] Bernhard Haeupler, Vahab S Mirrokni, and Morteza Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *Proceedings of the 7th Conference on Web and Internet Economics (WINE)*, pages 170–181. 2011.

[25] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 58–65, 2007.

[26] Jason D Hartline. Approximation in mechanism design. *American Economic Review*, 102(3): 330–36, 2012.

[27] Theodore P Hill and Robert P Kertz. A survey of prophet inequalities in optimal stopping theory. *Contemporary Mathematics*, 125:191–207, 1992.

[28] Theodore P Hill, Robert P Kertz, et al. Comparisons of stop rule and supremum expectations of iid random variables. *The Annals of Probability*, 10(2):336–345, 1982.

[29] Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals. In *Proceedings of the 45th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1070–1081, 2018.

[30] Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295, 1983.

[31] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 587–596, 2011.

[32] Alam Khursheed and KM Lai Saxena. Positive dependence in multivariate distributions. *Communications in Statistics - Theory and Methods*, 10(12):1183–1196, 1981.

[33] Robert Kleinberg and S Matthew Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games and Economic Behavior*, 113:97–115, 2019.

[34] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. *Probability on Banach spaces*, 4:197–266, 1978.

[35] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8 (3):261–277, 1988.

[36] Brendan Lucier. An economic view of prophet inequalities. *ACM SIGecom Exchanges*, 16(1): 24–47, 2017.

[37] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 597–606, 2011.

[38] Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.

[39] Christos H Papadimitriou. Games against nature. *Journal of Computer and System Sciences*, 31(2):288–301, 1985.

[40] Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *the Annals of Probability*, 12(4):1213–1216, 1984.

[41] Danny Segev and Sahil Singla. Efficient approximation schemes for stochastic probing and prophet problems. *arXiv preprint arXiv:2007.13121*, 2020.