Improved Guarantees for Offline Stochastic Matching via New Ordered Contention Resolution Schemes

Brian Brubach
Computer Science Department, Wellesley College, bb100@wellesley.edu,

Nathaniel Grammel
Department of Computer Science, University of Maryland, College Park, ngrammel@umd.edu,

Will Ma
Graduate School of Business, Columbia University, New York, wm2428@gsb.columbia.edu,

Calum MacRury
School of Industrial and Systems Engineering (ISyE), Georgia Tech, Atlanta, calum.macrury@isye.gatech.edu

Aravind Srinivasan
Department of Computer Science, University of Maryland, College Park, asriniv1@umd.edu,

Matching is one of the most fundamental and broadly applicable problems across many domains. In these diverse real-world applications, there is often a degree of uncertainty in the input which has led to the study of stochastic matching models. Here, each edge in the graph has a known, independent probability of existing derived from some prediction. Algorithms must probe edges to determine existence and match them irrevocably if they exist. Further, each vertex may have a patience constraint denoting how many of its neighboring edges can be probed. We present new ordered contention resolution schemes yielding improved approximation guarantees for some of the foundational problems studied in this area. For stochastic matching with patience constraints in general graphs, we provide a 0.382-approximate algorithm assuming each vertex has patience at least 2. Under this assumption, we improve upon the previous best 0.31-approximation of Baveja et al. (2018). When the vertices do not have patience constraints, we describe a 0.432-approximate random order probing algorithm with several corollaries such as an improved guarantee for the Prophet Secretary problem under Edge Arrivals. Finally, for the special case of bipartite graphs with unit patience constraints on one of the partitions, we show a 0.632-approximate algorithm that improves on the recent 1/3-guarantee of Hikima et al. (2021).

Key words: stochastic matching, contention resolution schemes, approximation algorithms

---

1. Introduction The offline stochastic matching problem is about finding a maximum matching on a weighted graph. However, each edge e is *active* independently according to a known probability p_e, and only active edges can be matched. The set of active edges is initially unknown. An edge whose endpoints are unmatched can be *probed* to determine whether it is active, and if so, it is irrevocably inserted into the matching. The objective is to sequentially probe the edges in a way to maximize the expected weighted matching at the end.

Matching problems arise in numerous deployed systems, especially those dealing with allocation and scheduling. See, for example, the works of Ahmadi et al. [3], Ahmed et al. [4], Brubach et al. [12, 10], Baveja et al. [8] for applications to advertising, e-commerce, organ exchange, online dating, peer review, school matching, and hiring; and Hikima et al. [28], Nanda et al. [34], Xu et al. [37] for applications to ride sharing, crowdsourcing (worker-task assignment), and recommendation systems. The work of Esfandiari et al. [20], Antoniadis et al. [6] gives further discussion on applications to e-commerce and internet advertising, and illustrates the importance of designing algorithms with good theoretical guarantees.

Stochastic edges play a major role in properly capturing many of these applications when we account for hidden information about whether a proposed match will be accepted. Is a kidney donor a good match? Will a worker accept an offered task? Does a user want to click on this ad? The uncertainty inherent in the presence or absence of various edges naturally leads to the need for stochastic optimization. Achieving good results when the edge probabilities are known demonstrates the value of learning distributions in AI.

We contribute improved algorithms for a range of fundamental problems, as well as useful-special-case problems, in the realm of stochastic matching. Although our results are phrased for the offline setting, some of our algorithms also translate to the *random-order* online setting in which the edges arrive in random order. This online arrival model is well-motivated in these settings due to the inability to control the order in which the agents arrive.

Below, we further describe three typical features that may be present in stochastic matching problems.

Patience constraints. The offline stochastic matching problem is typically considered with the input including an integral *patience parameter* t_v for each vertex v known as a *patience constraint* or timeout (see, e.g., Chen et al. [15], who first introduced this formulation of the problem). This adds the constraint that at most t_v of the edges incident to v can be probed, which is motivated by applications in which the vertices represent users who are only willing to view a finite number of potential matches. A vertex v which has been probed t_v times is said to have *timed out*. To capture the absence of patience constraints in some models, we allow patience parameters to be ∞.

Probing orders allowed. Most generally, we allow algorithms to probe the edges in any order based on the realizations of past probes. Sometimes we impose that the algorithm must make a single pass through the edges, according to a uniform random permutation. This can also represent an "online" setting in which the edges arrive in a uniformly random order, and upon arrival, an edge must either be probed (only possible if both endpoints are unmatched and have remaining patience) or irrevocably discarded. We say that such algorithms are *random-order*.

Special case of graphs considered. We introduce a new class of graphs, *bipartite graphs with a unit-patience* side, for which we derive further improved guarantees. Such a graph can be divided into two sides such that: (1) all edges are between vertices on different sides; and (2) for one of the sides, all patiences are 1. This subclass of graphs captures a problem of interest in AI with applications such as crowdsourcing and ride-hailing, described in Hikima et al. [28].

1.1. Summary of Results To derive computationally-efficient probing algorithms, we address the standard approach of first solving a fractional relaxation, which prescribes a probability $y_e \in [0,1]$ with which each edge e should be probed. These relaxed values y_e only have to satisfy the matching and patience constraints in expectation, while a probing algorithm must satisfy these constraints with probability (w.p.) 1. However, we can still hope to probe every edge e with probability at least $c \cdot y_e$, for some $c \leq 1$. It is well-known that such an algorithm would then be *c-approximate*, i.e. its expected weight matched would be at least c times that of an optimal probing algorithm.

RESULT 1. A **0.382**-approximate *random-order* probing algorithm for general graphs with patience at least 2.

We note that the previous version of the paper incorrectly claimed this result held for all patience values. However, on graphs with patience values of 1, there exists an input where our algorithm fails to exceed 1/3 against the fractional relaxation we consider (see Remark 1 from Subsection 3.2 for details). Restricted to inputs with patience at least 2, Result 1 improves on the **0.25**-approximate algorithm of Bansal et al. [7], the **0.269**-approximate algorithm of Adamczyk et al. [1], and the **0.31**-approximate algorithm of Baveja et al. [8]. Since our paper first appeared, the state-of-the-art for general graphs is now due to Pollner et al. [35], who attained a **0.395**-approximate algorithm.

For the special case of bipartite graphs, Derakhshan et al. [16] recently attained a **0.58**-approximate algorithm, though their algorithm does not probe the edges in random-order, and attains its guarantee against a different fractional relaxation of the optimal probing algorithm.

Our general guarantee can be parametrized by the patiences in the graph and improves in certain special cases.

COROLLARY 1. *A $(1-\mathsf{e}^{-2})/2 \approx$ **0.432**-approximate random-order probing algorithm for general graphs without patience constraints.*

In the special case where every patience is ∞, the guarantee of our random-order probing algorithm improves to 0.432. We note that without patience constraints, the Greedy algorithm which probes edges in decreasing order of weights (ignoring the probabilities p_e) guarantees at least $1/2$ the offline maximum weighted matching knowing the set of active edges in advance. Even better guarantees are known if the edges can be probed in any order: Gamlath et al. [24] derive a $(1-1/\mathsf{e}) \approx 0.632$-approximate algorithm for bipartite graphs by adding tightening constraints to the standard fractional relaxation; Fu et al. [23] derive a $\frac{8}{15}$-approximate algorithm for general graphs by considering random-order contention resolution with vertex arrivals, which was later improved slightly by MacRury and Ma [32]. These papers do not consider the finite patience case, and even in the infinite patience case, our algorithm differs because it probes the *edges* in a uniformly random order. This means that the special case of our algorithm where patiences are ∞ implies results for *Random-order Contention Resolution Schemes* and *Prophet Secretary under Edge Arrivals*, as we discuss below.

COROLLARY 2. *A $(1-\mathsf{e}^{-2})/2$-balanced Random-order Contention Resolution Scheme for the matching polytope of general graphs.*

For $c \leq 1$, the definition of a c-balanced Contention Resolution Scheme can be reduced to our goal of probing every edge e with probability at least $c \cdot y_e$, as we explain in Section 2. Previous work in this area has derived a $(1-\mathsf{e}^{-2})/2$-balanced "offline" Contention Resolution Scheme (CRS), which needs to know the set of active edges in advance, for the matching polytope of general graphs [27]. The balancedness has since been shown to be strictly greater than $(1-\mathsf{e}^{-2})/2$, while also satisfying a monotonicity property [13]. However, both of these results require knowing the set of active edges in advance, while our algorithm observes the activeness of edges in an ordered fashion and must immediately decide whether to insert any active edges into the matching. In fact, our algorithm satisfies the definition of a *Random-order* Contention Resolution Scheme (RCRS) introduced in the papers by Adamczyk and Włodarczyk [2], Lee and Singla [30]. We note that the bound of $(1-\mathsf{e}^{-2})/2$ has since been beaten by Pollner et al. [35], and the state of the art is due to [33], the latter of which also shows that $1/2$ is unbeatable for an RCRS.

We note that the works of Fu et al. [23] and MacRury and Ma [32] also consider random-order contention resolution schemes for stochastic matching, but under *vertex arrivals*, whereas our setting considers *edge arrivals*. To this end, our specific setting of the matching polytope is not captured[1] by these papers.

COROLLARY 3. *A **0.432**-guarantee for the Prophet Secretary problem under Edge Arrivals, in general graphs.*

Contention Resolution Schemes for the matching polytope also imply Prophet Inequalities under the Edge Arrival model introduced in Gravin and Wang [26]. Ezra et al. [21] attained a **0.337**-balanced *adversarial-order* Contention Resolution Scheme for the matching polytope of general

[1] If the graph is bipartite, then its matching polytope can be captured by the intersection of two matroids; however, in this case the balancedness guaranteed by Adamczyk and Włodarczyk [2] is $1/3$, which is worse than our balancedness of 0.432.

graphs which implies a 0.337-guarantee for Prophet Inequalities under Edge Arrivals in any order. Recently, Aminian et al. [5] achieved a $\frac{3-\sqrt{5}}{2} \approx 0.3819$ guarantee on bipartite graphs, which is tight under certain concentration assumptions (see [31] for details). When the edges arrive in a uniformly random order, this can be called the *Prophet Secretary* problem [19] under Edge Arrivals, for which our Random-order Contention Resolution Scheme implies an improved 0.432-guarantee.

COROLLARY 4. *A new class of non-adaptive* $(1-1/\mathsf{e})$*-balanced Random-order Contention Resolution Schemes for rank-1 matroids.*

In the further special case of a star graph with infinite patiences, the balancedness of our probing algorithm improves to $1-1/\mathsf{e}$. The constraint that at most one edge in a star graph can be matched corresponds to a rank-1 matroid, for which a $(1-1/\mathsf{e})$-balanced Random-order Contention Resolution Scheme is already known [30]. However, our analysis yields a wide range of new such schemes, which are simpler than existing ones in that they satisfy a *non-adaptiveness* property, which we we elaborate on in Subsection 1.2.

RESULT 2. A $(1-1/\mathsf{e}) \approx$ **0.632**-approximate probing algorithm for bipartite graphs with a unit-patience side.

This result improves and generalizes the **1/3**-guarantee of Hikima et al. [28] which holds for a special case of bipartite graphs with a unit-patience side. Despite the ubiquity of $(1-1/\mathsf{e})$-guarantees in online matching, to the best of our understanding, our guarantee requires the new technical ingredient of a $(1-1/\mathsf{e})$-balanced Ordered Contention Resolution Scheme for rank-1 matroids under *negative correlation*. The aforementioned $(1-1/\mathsf{e})$-balancedness results for rank-1 matroids assume elements to be active independently and do not establish this guarantee, as we discuss in Subsection 1.2. We note that our probing algorithm here must be able to choose the order, though. We also note that Borodin and MacRury [9] study an online problem which implies the same $(1-1/\mathsf{e})$-approximation for the offline problem in the particular case where one side has arbitrary patience and the other side has unlimited patience.

1.2. Description of Techniques

Random-order probing: finding an attenuation function which improves the worst case. As described before, our probing algorithm for general graphs considers the edges in a uniformly random order. This can be implemented by each edge e drawing an "arrival time" x_e independently and uniformly from $[0,1]$. Baveja et al. [8] have previously analyzed a similar algorithm, which probes each incoming edge e according to the fractionally-feasible probability y_e as long as e is safe. They show that every edge e ends up being probed with probability at least $0.31 \cdot y_e$, yielding a 0.31-approximate algorithm. The worst case occurs for an edge e' whose values of $y_{e'}, p_{e'}$ are close to 0, with both endpoints of e' being incident to other edges e'' whose values of $y_{e''}, p_{e''}$ are 1.

To improve this worst case, we attach an "attenuation factor" $a(e) \in [0,1]$ to each edge e such that the probability of an incoming safe edge e being probed is scaled down by a factor of $a(e)$. We make $a(e)$ decreasing in y_e and p_e, to dissuade the aforementioned edges e'' with large values of $y_{e''}, p_{e''}$ from being probed and blocking edge e'. However, given an arbitrary function a defining the attenuation factors, computing the new worst case could be difficult. Therefore, our approach is instead to derive properties on a which cause the *worst case to only involve edges* e *with* $a(e) = 1$. More specifically, we show that for functions a defined by $a(e) = \tilde{a}(y_e p_e)$ for some univariate function $\tilde{a}$ with $\tilde{a}(0) = 1$, it is possible to design the derivatives of $\tilde{a}$ so that in the worst case, edge e' is only incident to edges e'' with $y_{e''} p_{e''} \approx 0$ (which implies that $a(e'') \approx \tilde{a}(0) = 1$). Our final bound is then derived via the same Poisson lower tail bound proven Baveja et al. [8] (namely Lemma 4), except that the worst case ratio has improved to 0.382. We note however that unlike the analysis of Baveja et al. [8], the use of an attenuation function requires us to prove a non-trivial "exchange argument" in Lemma 6 of Subsection 3.2.2 when an endpoint of e has patience 2. The necessity of this computation was overlooked in the previous version of this paper, and so the analysis in this updated version is longer than before. We now make use of a related lemma and construction from Derakhshan et al. [16].

We note that the specific attenuation function we compute is inconsequential to this improved ratio—the key is showing the *existence* of an attenuation function which *eliminates* the previous worst case. However, by deriving such necessary properties, we are able to see not only which attenuation functions work, but also which *don't work* (suggesting alternative attenuation functions can't do better than ours). Further, having a *family* of valid attenuation functions allows for a choice of different attenuation functions for different application domains while keeping the same approximation guarantee.

Using our attenuation for Random-order Contention Resolution Schemes. As stated in Corollaries 2 and 3, our attenuation analysis in the special case of infinite patiences implies a previously-unknown $(1-\mathsf{e}^{-2})/2$-balanced Random-order Contention Resolution Scheme for the matching polytope of general graphs. We now discuss the further special case in Corollary 4 of star graphs, for which we can contrast our technique with that used in the known Random-order Contention Resolution Schemes for rank-1 matroids. Here, Lee and Singla [30] show that $(1-1/\mathsf{e})$-balancedness can be achieved using what we would call the attenuation function $a(e)=\exp(-x_e p_e)$, which depends on the arrival time x_e of each edge e. This function is designed in [18] to yield a closed-form expression for the probability of availability at any particular time $x \in [0,1]$, which allows them to elegantly compute that the balancedness is $1-1/\mathsf{e}$.

In this special case, our analysis also yields a $(1-1/\mathsf{e})$-balanced Random-order Contention Resolution Scheme. However, instead of designing a specific function, our analysis implies a class of functions which sufficiently[2] attenuate large values of p_e to prevent them from blocking smaller values of p_e. To elaborate, we show that the function $a(e)=\frac{1-p_e y_e}{1-\mathsf{e}^{-(1-p_e y_e)}}(1-1/\mathsf{e})$ ensures that in the worst case, all edges e have $p_e \approx 0$. And in this worst case, $a(e) \approx 1$ for all e, from which we can conclude that any edge has an $\approx 1-1/e$ chance of being selected. We note that our function does not depend on the arrival time x_e and can be seen as *non-adaptive* $(1-1/e)$-balanced Random-order Contention Resolution Schemes for a rank-1 matroid, which we believe could be applied elsewhere.

Bipartite graphs with a unit-patience side. For offline stochastic matching on bipartite graphs, a standard technique [7] is to randomly round the fractionally-feasible values y_e to binary values Y_e using the dependent rounding procedure of Gandhi et al. [25], which ensures the patience constraints on both sides to be satisfied w.p. 1. Under our additional assumption that one of the sides V_1 has unit-patience, the vertices in V_1 must have rounded degree at most 1, resulting in the rounded graph being a disjoint collection of stars. One could then separately handle the edges in each star using a Random-order Contention Resolution Scheme for rank-1 matroids, since its edges e will be disjoint from other stars and active independently w.p. p_e.

However, this does not lead to $(1-1/\mathsf{e})$-balancedness. To elaborate, for any vertex $v \notin V_1$, let $\delta(v)$ denote the set of edges incident to v. Fractional feasibility of the y_e values ensures

$$\sum_{e \in \delta(v)} p_e y_e \leq 1, \tag{1}$$

[2] This intuition can be illustrated as follows. If there is no attenuation, then the worst case involves two edges e', e'' with probabilities $p_{e'}=0, p_{e''}=1$, which when shown in a random order implies that e'' will block e' w.p. 1/2, whereas e' will not block e''. The goal of "attenuation" is to scale down the probability of selecting e'', to increase the probability of selecting e'.

but the rounded star graph formed by edges $\{e \in \delta(v) : Y_e = 1\}$ could have $\sum_{e\in\delta(v)} p_e Y_e > 1$ whenever[3] some particular edge e' has $Y_{e'} = 1$, making it difficult to ensure that edge e' gets selected with sufficient probability when $Y_{e'}$ is rounded up.

Reinterpretation as contention resolution under negative correlation. To resolve this issue, we instead imagine each edge $e \in \delta(v)$ as being active with probability $z_e := p_e y_e$, which satisfies $\sum_{e\in\delta(v)} z_e \le 1$, due to (1). The active edges in $\delta(v)$ are correlated in a way such that they cannot conflict with active edges in other stars; however, due to this correlation, any of the aforementioned Random-order Contention Resolution Schemes which are agnostic to the correlation will only be 1/2-balanced, as we will show in Section 4.

Despite this apparent lack of a correlation-agnostic $(1-1/\mathsf{e})$-balanced Contention Resolution Scheme, what we do show is that the optimal online algorithm, which trivially sorts the edges $e \in \delta(v)$ in decreasing order of weights w_e, obtains in expectation at least $1-1/\mathsf{e}$ times the fractional value $\sum_{e\in\delta(v)} w_e z_e$. This is only possible due to the following *negative correlation* property enjoyed by the rounding procedure of Gandhi et al. [25], with $Z_e \in \{0,1\}$ denoting the activeness of an edge e:

$$\Pr\left[\bigcap_{e\in S}(Z_e = b)\right] \le \prod_{e\in S} \Pr[Z_e = b] \qquad\qquad \forall S \subseteq \delta(v), b \in \{0,1\}. \tag{2}$$

Our analysis applies this negative correlation property with $b = 0$ to show that the expected *overall* weight obtained by the online algorithm is *minimized when the Z_e's are independent*, despite the fact that for a *particular* edge e', negative correlation among other edges in $\delta(v)$ *could make it more likely that e' is blocked* than in the independent case. Through the equivalence derived in Lee and Singla [30], our analysis also implies the existence of a $(1-1/\mathsf{e})$-balanced Ordered Contention Resolution Scheme for rank-1 matroids under negative correlation, assuming the order can be chosen. Some other recent papers have considered contention resolution schemes under various notions of negative dependence, such as Dughmi [17], Chekuri and Livanos [14], and Qiu and Singla [36]. To our knowledge, it remains open whether a $(1-1/\mathsf{e})$-balanced *Random-order*[4] Contention Resolution Scheme is possible under property (2).

1.3. Roadmap We begin in Section 2 with some background. Then, in Section 3, we present a random-order algorithm for general graphs, achieving our 0.382-approximation for stochastic matching. We then analyze the same algorithm in the case of infinite patience, where the algorithm yields a $(1-\mathsf{e}^{-2})/2$-approximation (and thus a $(1-\mathsf{e}^{-2})/2$-balanced Random-order Contention Resolution Scheme for the matching polytope); and the case of a star graph, where the algorithm achieves an approximation guarantee of $1-1/\mathsf{e}$ (and thus gives a $(1-1/\mathsf{e})$-balanced Random-order Contention Resolution scheme for rank-1 matroids). Finally, in Section 4, we present an algorithm for stochastic matching on bipartite graphs with a unit patience side, which achieves an approximation guarantee of $1-1/\mathsf{e}$.

2. Notation and Preliminaries The weighted stochastic graph is denoted by $G = (V,E)$, with the weight and probability of being active being denoted by w_e and p_e, respectively, for each edge $e \in E$. The patience parameter is denoted by t_v for each vertex $v \in V$. Given any

[3] As a concrete example, let v have patience $t_v = 2$, and be incident to three edges with $p_1 = 1, y_1 = \varepsilon$; $p_2 = 1-\varepsilon, y_2 = 1$; and $p_3 = 0, y_3 = 1-\varepsilon$, which are fractionally feasible in that $\sum_{e\in\delta(v)} p_e y_e \le 1$ and $\sum_{e\in\delta(v)} y_e \le 2$. The rounding must be such that whenever $Y_1 = 1$, we also have $Y_2 = 1$, which results in $\sum_{e\in\delta(v)} p_e Y_e = 2-\varepsilon$.

[4] A 1/2-balanced Random-order Contention Resolution Scheme for star graphs under negative correlation is implied by the "uniform black box" in Brubach et al. [12]. This black box has also been extended in some cases by Fata et al. [22].

problem instance defined by these values, finding the optimal probing algorithm is computationally challenging [7]. For $c \leq 1$, a probing algorithm is said to be *c-approximate* if its expected weight matched is at least c times that of the optimal probing algorithm, for any problem instance. The following LP relaxation is commonly used to derive computationally efficient probing algorithms.

$$\mathsf{LP} := \max \sum_{e \in E} w_e z_e \tag{3}$$

$$\text{subject to} \quad \sum_{e \in \delta_G(v)} z_e \leq 1 \qquad \forall v \in V \tag{3a}$$

$$\sum_{e \in \delta_G(v)} y_e \leq t_v \qquad \forall v \in V \tag{3b}$$

$$0 \leq y_e \leq 1 \qquad \forall e \in E \tag{3c}$$

$$z_e = y_e p_e \qquad \forall e \in E \tag{3d}$$

Here $\delta_G(v)$ denotes the set of edges incident to a vertex v, and $N_G(v)$ will be used to denote the neighbors of v. When clear, we shall drop the dependence on G. The variable $y_e \in [0,1]$ corresponds to the probability of probing edge e. The variable z_e is then the probability that edge e is included in the matching (that is, it is both active and probed). Constraint (3a) for a vertex $v \in V$ is the *matching constraint*: it is satisfied when v is matched to at most one of its neighbors in expectation. Constraint (3b) for a vertex $v \in V$ is the *patience constraint*: it is satisfied when at most t_v edges incident to v are probed in expectation. We note that for a solution $(y_e)_{e \in E}$ to (3), we denote $y(S) := \sum_{e \in S} y_e$ for $S \subseteq E$.

Lemma 1 **(Bansal et al. [7])**. *For any problem instance, the optimal objective value* LP *is an upper bound on the expected weight matched by any optimal probing algorithm.*

Due to Lemma 1, for an algorithm to be c-approximate, it suffices to show that its expected weight matched is at least $c \cdot \mathsf{LP}$. All of our algorithms will be based on taking an optimal LP solution given by $(y_e)_{e \in E}$, and randomizing in a way to probe every edge e with probability at least $c \cdot y_e$, which suffices for matching expected weight at least $c \cdot \mathsf{LP}$. We note that the gap between the LP and the optimal probing algorithm can be large: Brubach et al. [10] show, via a result of Karp and Sipser [29], that for some graphs, the ratio between the maximum-weight matching and the LP objective value can be as large as 0.544.

Definition 1 (Ordered Contention Resolution Scheme Problem). A graph $G = (V, E)$ and a vector $(\tilde{z}_e)_{e \in E}$ lying in its *matching polytope* (i.e. satisfying $\sum_{e \in \delta(v)} \tilde{z}_e \leq 1$ for all v) is given. Each edge $e \in E$ has an "activeness" in $\{0,1\}$ whose state is initially unknown other than that it equals 1 w.p. $\tilde{z}_e$. The activeness of edges is observed sequentially, and if an edge is both active and eligible to be matched (i.e. not incident to any edges already matched), then it can be either immediately matched or irrevocably discarded. A (randomized) algorithm that guarantees every edge $e \in E$ of being matched with ex ante probability at least $c \cdot \tilde{z}_e$ is said to be a *c-balanced Ordered Contention Resolution Scheme* for the matching polytope.

A probing algorithm which guarantees every edge e probability at least $c \cdot y_e$ of being probed implies a c-balanced Ordered Contention Resolution Scheme. To see this, given an instance to the problem in Definition 1, we can construct an instance of offline stochastic matching with $p_e = \tilde{z}_e$ for all e and $t_v = \infty$ for all v, which means that setting $y_e = 1$ for all e is a feasible solution to LP (3). The probing algorithm will indicate whether to probe each edge in a way that guarantees the overall probability of any edge e being probed to be at least $c \cdot y_e = c$. Therefore, if in Definition 1 we "accept an edge when active" whenever the probing algorithm would have probed that edge, this translates to an ex ante guarantee of $c \cdot p_e = c \cdot \tilde{z}_e$ on the probability of any edge e being matched, as desired.

DEFINITION 2 (RANDOM-ORDER). A probing algorithm is said to be *random-order* if it can be applied in the online setting where the edges arrive in a uniformly random order, and upon arrival, each edge needs to be either immediately probed (if safe) or irrevocably discarded. Analogously, a *Random-order Contention Resolution Scheme* must observe the activeness of edges in a uniformly random order.

3. Algorithm and Analysis for Result 1

3.1. Description of Algorithm and Attenuation Functions Our algorithm is based on the algorithm of Baveja et al. [8], but with an added attenuation factor. This algorithm first solves LP (3) to get a fractional solution $(y_e)_{e\in E}$, and independently draws $Y_e \in \{0,1\}$ for each $e \in E$, where $\Pr[Y_e = 1] = y_e$. It generates a uniformly random permutation on E, and processes the edges sequentially in the order of the permutation. Specifically, e is probed if and only if $Y_e = 1$ and both endpoints of e are currently *free* (i.e., when e is processed in the permutation, its endpoints are both unmatched and have remaining patience).

Our algorithm adds additional attenuation as follows. Let $a\colon E \to [0,1]$ be our attenuation function. When we get to an edge $e = \{u,v\}$ in the permutation, we draw a new independent Bernoulli random variable A_e such that $\Pr[A_e = 1] = a(e)$. Then, we probe e if

1. e is free,
2. $Y_e = 1$, and
3. $A_e = 1$

Pseudocode is given in Algorithm 1. In what follows, we generate the uniformly random permutation by first independently drawing a uniformly random "arrival time" $\pi(e) \in [0,1]$ for each $e \in E$, and then ordering the edges in increasing order of arrival times.

Algorithm 1 Attenuation-based algorithm for Stochastic Matching

```
function ATTENUATEMATCH(V, E, p)
    for each edge e in uniformly random order do
        Generate random bit Y_e = 1 with probability y_e
        Generate random bit A_e = 1 with probability a(e)
        if e is free ∧ Y_e = 1 ∧ A_e = 1 then
            Probe e
        end if
    end for
end function
```

Our analysis allows for our 0.382-approximation to be achieved for many choices of attenuation function. Specifically, our analysis will require a few key properties of our attenuation function, outlined in Definition 3 below.

DEFINITION 3. We call an attenuation function $a(e)$ *effective* if all of the following conditions hold:

1. $a(e)$ can be expressed as a function $\tilde{a}(z_e)$ of z_e
2. $\tilde{a}(0) = 1$
3. $\ln(1 - xz\tilde{a}(z))$ is a *convex* function of $z \in [0,1]$ for any $x \in (0,1)$

There are many functions which satisfy the conditions of Definition 3. Notice, for instance, that the first two conditions are straightforward: we require only that the attenuation function be a function of z_e and result in no attenuation when $z_e \approx 0$. The final condition is satisfied by many nice classes of functions, with some examples given in Definition 4.

Definition 4. Define the following attenuation functions:

- The *exponential* attenuation function, defined by $\tilde{a}_{\exp}(z) := \mathsf{e}^{-\alpha z}$ for any $\alpha \geq 1/2$
- The *linear* attenuation function, defined by $\tilde{a}_{\text{lin}}(z) := 1 - \alpha z$ for any $\alpha \geq 1/2$
- The function $\tilde{a}(z) := \frac{1-z}{1-\mathsf{e}^{-(1-z)}}(1 - 1/\mathsf{e})$

It can be easily verified, by taking second derivatives, that these functions indeed satisfy the third property of Definition 3, and hence are effective. Although these examples are effective attenuation functions for all $\alpha \geq 1/2$, in our algorithms we will always set $\alpha = 1/2$ to get the best bound at the end.

3.2. Analysis of the Attenuation Algorithm Suppose we execute Algorithm 1 using an attenuation function $a : E \to [0,1]$. For each $e \in E$, it will be convenient to denote $P_e \in \{0,1\}$ as the indicator for whether e is active, where $\Pr[P_e = 1] = p_e$. Note that Algorithm 1 decides to probe e prior to learning P_e, and the event "e is free" is independent of Y_e and A_e. Thus, the expected weight of the matching produced by Algorithm 1 is

$$\sum_{e \in E} w_e p_e \Pr[e \text{ is free}] \Pr[Y_e = 1 \cap A_e = 1] = \sum_{e \in E} w_e p_e y_e a(e) \Pr[e \text{ is free}]. \tag{4}$$

Let us hereby assume that $a : E \to [0,1]$ is effective, and so $a(e) = \tilde{a}(p_e y_e)$ for each $e \in E$, for some function $\tilde{a} : [0,1] \to [0,1]$ which satisfies Definition 3. Our goal is to lower bound $\tilde{a}(p_e y_e) \cdot \Pr[e \text{ is free}]$ for all $e \in E$. Specifically, define

$$\beta^* := \int_0^1 \left(\mathsf{e}^{-2x} + x\mathsf{e}^{-2x}\right)^2 dx \approx 0.38278. \tag{5}$$

We prove the following theorem which holds for any solution $(y_e)_{e \in E}$ to (3). Our main result then follows by taking $(y_e)_{e \in E}$ to be an optimal solution to (3), and applying (4) and Lemma 1.

Theorem 1 **(corresponds to Result 1 from Subsection 1.1)**. *Suppose the stochastic graph $G = (V, E)$ has minimum patience 2, and $(y_f)_{f \in E}$ is an arbitrary solution to the LP (3). More, assume that $a : E \to [0,1]$ is taken from Definition 3 where $\alpha = 1/2$ if the exponential or linear attenuation function is chosen. Then, when Algorithm 1 executes on G,*

$$a(e) \cdot \Pr[e \textit{ is free}] \geq \beta^*, \tag{6}$$

for all $e \in E$.

Remark 1. If patience values of 1 are allowed, there is a simple input where the left-hand side of (6) is no greater than 1/3 (no matter which attenuation function $\tilde{a}$ with $\tilde{a}(0) = 1$ is used). To see this, fix $\varepsilon > 0$, and consider a single edge $e = \{u, v\}$ with $p_e = 1$ and $y_e = \varepsilon$, where $t_u = t_v = 1$. More, assume that u and v each are incident to one additional edge, say f and f', respectively, where $y_f = y_{f'} = 1$, yet $p_f = p_{f'} = \varepsilon$. Then, as $\varepsilon \to 0$, there will then be no attenuation since $\tilde{a}(p_f y_f) = \tilde{a}(p_{f'} y_{f'}) = \tilde{a}(p_e y_e) = \tilde{a}(\varepsilon) \to 0$. Moreover, the probability that e is matched tends to $y_e/3$.

In order to prove Theorem 1, we provide a sufficient condition for the event "$e = \{u, v\}$ is free" to occur. This sufficient condition depends only on the random variables associated with the edges incident to the endpoints of e – i.e., $\delta(u) \cup \delta(v)$ – which makes computing its probability tractable. Recall that each $f \in E$ has an arrival time $\pi(f) \in [0,1]$ drawn uniformly and independently. We say that $e = \{u, v\}$ is *safe* when executing Algorithm 1 on G with $(y_f)_{f \in E}$, provided that before e is processed in the uniformly random permutation:

1. At most $t_r - 1$ probes were made to edges of $\delta(r)$ for each $r \in \{u, v\}$.
2. For each $f \in \delta(u) \cup \delta(v) \setminus \{e\}$ with $\pi(f) < \pi(e)$, $A_f P_f X_f = 0$.

Observe that if e is safe, then e is free. Thus, using that $a(e) = \tilde{a}(p_e y_e)$, $a(e)\Pr[e \text{ is free}]$ is lower bounded by

$$\tilde{a}(p_e y_e)\Pr[e \text{ is safe}]. \tag{7}$$

We next argue that with respect to lower bounding (7), we can assume our input is of a specific form:

Lemma 2. *Suppose Algorithm 1 is executed using an effective attenuation function. Then, with respect to lower bounding (7), we may assume without loss of generality that:*

1. *The edges incident to u and v form the entire graph; i.e., $E = \delta_G(u) \cup \delta_G(v)$ and $V = N_G(u) \cup N_G(v)$.*
2. *For all $f \in \delta(u) \cup \delta(v)$, $p_f \in \{0,1\}$, and $p_e = 1$.*
3. *For each $r \in \{u,v\}$, $\sum_{f\in\delta(r)} y_f = t_r$.*

Proof. We first observe that since (7) depends only on the edges of $\delta(u)\cup\delta(v)$ for $e = \{u,v\}$, we may assume without loss that $E = \delta_G(u) \cup \delta_G(v)$ and $V = N_G(u) \cup N_G(v)$.

Under this assumption, we next argue that we can assume $p_e = 1$. To see this, construct $\bar{G}$, on the same edges and vertices of G, yet with LP solution $(\bar{y}_f)_{f\in E}$ and edge probabilities $(\bar{p}_f)_{f\in E}$, where $\bar{y}_e = p_e y_e$, $\bar{p}_e = 1$ and $\bar{y}_f = y_f$, $\bar{p}_f = p_f$ for all $f \in E \setminus \{e\}$. It is easy to check that $(\bar{y}_f)_{f\in E}$ is a feasible solution to (3). Moreover, a is effective, and so $a(e) = \tilde{a}(\bar{p}_e\bar{y}_e) = \tilde{a}(p_e y_e)$. Finally, the probability that e is safe is the same in either input, and so (7) is the same in either input.

Let us now assume that $E = \delta_G(u) \cup \delta_G(v)$, $V = N_G(u) \cup N_G(v)$ and $p_e = 1$. We next argue that we can assume without loss that $p_f \in \{0,1\}$ for all $f \in E = \delta(u) \cup \delta(v) \setminus \{e\}$. We consider the "worst-case structure", i.e., the one that leads to the worst bound in (7). Suppose for contradiction that in this worst-case structure, there exists some edge $f \in E(u)$ with $p_f \in (0,1)$. We will "split" this edge into two edges with integer probabilities and then see that this splitting can only *decrease* the probability that e is safe.

We split the edge f into two new edges f_0 and f_1 such that $p_{f_0} = 0$ and $p_{f_1} = 1$. Then, set $y_{f_0} = y_f(1 - p_f)$ and $y_{f_1} = y_f p_f$. It is easy to see that this splitting does not affect any of the constraints of the LP and so is still feasible. This is because $E = \delta(u) \cup \delta(v)$, and $y_{f_0} + y_{f_1} = y_f$ and $y_{f_0}p_{f_0} + y_{f_1}p_{f_1} = y_f p_f$.

Further, we argue that this splitting can only decrease the probability that e is safe. Notice that if e is not safe, then there must be some edge $f \in \delta(u) \cup \delta(v) \setminus \{e\}$ that is probed before e and either: (a) is matched, or (b) causes u to time out (i.e., it exhausts u's patience by being the t_u^{th} failed probe); when this happens, we say that f "blocks" e.

We do this by showing that the probability of e being blocked by the probing of f is at most the probability of e being blocked by the probing of either f_0 or f_1. Edge f can block e either by being successfully probed before e, or by an unsuccessful probe that causes f to time out.

Let us condition on $\pi(e) = x$ for $x \in [0,1]$. First, we consider the (conditional) probability that f blocks e by a successful probe. This occurs with probability $xy_f a(f) p_f$. To see, note that f occurs before e with probability x; then, it is probed with probability $y_f a(f)$; finally, this probe is successful with probability p_f.

Next, consider the probability that either f_0 or f_1 is successfully probed before e. This is $xy_{f_1}a(f_1)p_{f_1} + xy_{f_0}a(f_0)p_{f_0}$. Since $p_{f_0} = 0$, this is simply $xy_{f_1}a(f_1)p_{f_1} = xy_f p_f a(f_1)$. Next, we use the fact that a is effective, so $a(f) = \tilde{a}(z_f)$. Since $z_f = y_f p_f = y_{f_1} = y_{f_1}p_{f_1} = z_{f_1}$, we have that $\tilde{a}(z_f) = \tilde{a}(z_{f_1})$ and so the probability is unchanged.

Now consider the case of blocking e by an unsuccessful probe (causing e to time out). The probability that f is unsuccessfully probed before e is $xy_f a(f)(1 - p_f)$. Then, consider the probability that either f_0 or f_1 is probed unsuccessfully before e. For edge f_1, this occurs with probability 0 since $p_{f_1} = 1$. The probability that edge f_0 is unsuccessfully probed before e is $xy_{f_0}a(f_0)(1 - p_{f_0}) = xy_{f_0}a(f_0) = xy_f a(f_0)(1 - p_f)$. We again utilize the fact that a is effective, noting that $a(f_0) =$

$\tilde{a}(z_{f_0}) = \tilde{a}(0) = 1$. Thus, the probability of blocking e by an unsuccessful probe is only increased by this splitting.

All other edges are unchanged, and so after averaging over $x \in [0,1]$, the overall probability of e being safe can only decrease by this splitting.

Finally, given the other properties, it is clear we may assume $\sum_{f \in \delta(r)} x_f = t_r$ for each $r \in \{u,v\}$. This can be attained by adding additional edges f' incident to each r with $p_{f'} = 0$ and $y_{f'} = \min\{1, t_r - \sum_{f \in \delta(r)} y_f\}$, and then by applying a simple coupling argument. ☐

Let $E(r) := \delta(r) \setminus \{e\}$ for $r \in \{u,v\}$. We also define $E_b(r) = \{f \in E(r) \mid p_f = b\}$ for $b \in \{0,1\}$. Given Lemma 2, we may assume that $p_e = 1$ and $E(r) = E_0(r) \cup E_1(r)$ for each $r \in \{u,v\}$.

For $r \in \{u,v\}$, let us define $Q_0(r)$ as the number of queries Algorithm 1 makes to $E_0(r)$ before e is processed. Observe then that if we condition on $\pi(e) = x$ for $x \in [0,1]$, then

$$\begin{aligned}\Pr[e \text{ is safe} \mid \pi(e) = x] &= \Pr[\cap_{f \in E_1(u) \cup E_1(v)} \{A_f Y_f \mathbf{1}_{\pi(f) < x} = 0\} \mid \pi(e) = x] \prod_{r \in \{u,v\}} \Pr[Q_0(r) < t_r \mid \pi(e) = x] \\ &= \prod_{f \in E_1(u) \cup E_1(v)} (1 - x a(f) y_f) \prod_{r \in \{u,v\}} \Pr[Q_0(r) < t_r \mid \pi(e) = x].\end{aligned}$$

Using the the analytic properties of the attenuation function $\tilde{a} : [0,1] \to [0,1]$ from Definition 3, we prove the following:

Lemma 3. *For each $r \in \{u,v\}$, we have that $\prod_{f \in E_1(r)} (1 - x a(f) y_f) \geq \mathsf{e}^{-x \sum_{f \in E_1(r)} y_f}$.*

Proof. Fix an arbitrary $r \in \{u,v\}$. Let us first define the follow quantities for convenience:

$$\lambda_1 := \prod_{f \in E_1(r)} (1 - x y_f a(f)) \text{ and } Y_1 := \sum_{f \in E_1(u)} y_f.$$

We consider $\ln \lambda_1 = \sum_{f \in E_1(r)} \ln(1 - x y_f a(f))$. Since a is effective, it can be expressed as a function $\tilde{a}$ of $p_f y_f$. Notice that for $f \in E_1(r)$, $p_f = 1$, and so we can write $a(f) = \tilde{a}(p_f y_f) = \tilde{a}(y_f)$. Further, since a is effective, we have that $\ln(1 - x y_f \tilde{a}(y_f))$ is a convex function of $y_f \in [0,1]$ for any $x \in (0,1)$.

Thus, the quantity $\ln \lambda_1$ is minimized by setting all y_f to be equal, i.e. $y_f = Y_1 / |E_1(r)|$ for all $f \in E_1(r)$, and letting $|E_1(r)|$ tend to infinity. That is,

$$\min \lambda_1 = \min \exp\left(\sum_{f \in E_1(r)} \ln(1 - x y_f \tilde{a}(y_f))\right) = \lim_{k \to \infty} \left(1 - \frac{x Y_1}{k} \tilde{a}\left(\frac{x Y_1}{k}\right)\right)^k = \mathsf{e}^{-x Y_1},$$

where the last line uses that $\tilde{a}(0) = 1$. Thus, for each $r \in \{u,v\}$,

$$\prod_{f \in E_1(r)} (1 - x a(y_f) y_f) \geq \mathsf{e}^{-x \sum_{f \in E_1(r)} y_f}.$$

☐

Recall that for any $S \subseteq E$, $y(S) := \sum_{f \in S} y_f$. Using this notation, if we apply Lemma 3 to each $r \in \{u,v\}$, then

$$\prod_{f \in E_1(u) \cup E_1(v)} (1 - x a(y_f) y_f) \geq \mathsf{e}^{-x(y(E_1(u)) + y(E_1(v)))},$$

Thus, $\tilde{a}(y_e) \Pr[e \text{ is safe} \mid \pi(e) = x]$ is lower bounded by

$$\tilde{a}(y_e) \cdot \int_0^1 \mathsf{e}^{-x(y(E_1(u)) + y(E_1(v)))} \prod_{r \in \{u,v\}} \Pr[Q_0(r) < t_r \mid \pi(e) = x] dx. \tag{8}$$

It remains to bound each term $\Pr[Q_0(r) < t_r \mid \pi(e) = x]$ for $r \in \{u,v\}$. In order to do so, we require the following two lemmas. The first is proven in Baveja et al. [8], and the second is a standard Chernoff bound.

LEMMA 4 **(Baveja et al. [8]).** *Fix an integer $t \geq 1$. Suppose $B_1, \ldots, B_k$ are independent Bernoulli's, and $\mu := \sum_{i=1}^k \mathbb{E}[B_i] < t-1$. Then, if $\mathrm{Pois}(\mu)$ is a Poisson random variable of parameter μ,*

$$\Pr\left[\sum_{i=1}^k B_i < t\right] \geq \Pr[\mathrm{Pois}(\mu) < t] = \sum_{i=0}^{t-1} \frac{\mathsf{e}^{-x\mu}\mu^i}{i!}.$$

LEMMA 5. *Fix $\varepsilon > 0$. Suppose $B_1, \ldots, B_k$ are independent Bernoulli's, and $\mu := \sum_{i=1}^k \mathbb{E}[B_i]$. Then,*

$$\Pr\left[\sum_{i=1}^k B_i < (1+\epsilon)\mu\right] \geq 1 - \exp\left(-\frac{\epsilon^2}{2+\epsilon}\mu\right).$$

To apply these bounds, note that conditional on $\pi(e) = x$ for $x \in [0,1]$, $Q_0(r)$ is a sum of $|E_0(r)|$ Bernoulli random variables. Moreover,

$$\mathbb{E}[Q_0(r) \mid \pi(e) = x] = \sum_{f \in E_0(r)} a(p_f y_f) y_f x = y(E_0(r))x$$

Thus, if $x \in [0, \frac{t_r - 1}{y(E_0(r))})$, then Lemma 4 applies, and we may conclude that

$$\Pr[Q_0(r) < t_r \mid \pi(e) = x] \geq \Pr[\mathrm{Pois}(y(E_0(r))x) < t_r] = \sum_{i=0}^{t_r-1} \frac{\mathsf{e}^{-xy(E_0(r))}(xy(E_0(r)))^i}{i!}. \tag{9}$$

On the other hand, for any $x \in [0,1]$, setting $\varepsilon = (1-x)/x$ and $\mu = xy(E_0(r))$, note that

$$\frac{\mu\varepsilon^2}{2+\varepsilon} = \frac{y(E_0(r))(1-x)^2}{1+x} \text{ and } (1+\varepsilon)xy(E_0(r)) = y(E_0(r)).$$

Thus, we can apply Lemma 5 to conclude that

$$\Pr[Q_0(r) < y(E_0(r)) \mid \pi(e) = x] \geq 1 - \exp\left(\frac{-y(E_0(r))(1-x)^2}{1+x}\right)$$

However, since $t_r - 1 \leq y(E_0(r)) \leq t_r$ by Lemma 2, we may combine the previous equation with each bound on $y(E_0(r))$ to get that

$$\Pr[Q_0(r) < t_r \mid \pi(e) = x] \geq 1 - \exp\left(\frac{-(t_r-1)(1-x)^2}{1+x}\right). \tag{10}$$

We now apply (9) and (10) to (8) over $x \in [0,1]$, where which bound we use depends on the values of t_u and t_v. Due to the symmetry of the problem, we hereby assume that $t_u \leq t_v$. To eliminate a variable in the subsequent bounds, we shall use that $y(E_0(r)) = t_r - y(E_1(r)) - y_e$ for each $r \in \{u, v\}$, as guaranteed by Lemma 2. More, it will be convenient to define for $r \in \{u, v\}$,

$$x_{r,c} := (t_r - 1)/(t_r - y(E_1(r)) - y_e), \tag{11}$$

as well as $x_c := \min\{x_{u,c}, x_{v,c}\}$. Observe that (9) applies to $\Pr[Q_0(r) < t_r \mid \pi(e) = x]$ for all $x \in [0, x_{r,c})$.

Next, we consider different cases for the patience: first, when $t_u \geq 3$, and then when $t_u = 2$.

3.2.1. The case $t_u \geq 3$ We first consider the case where $t_u \geq 3$. Let us further assume that $\max\{t_u, t_v\} < 48$. Then, by the definition of x_c, we can apply (9) for $x \in [0, x_c)$, followed by the trivial lower bound of 0 for $x \in [x_c, 1]$, and so (8) is lower bounded by

$$\tilde{a}(y_e) \cdot \int_0^{x_c} \mathsf{e}^{-x(y(E_1(u))+y(E_1(v)))} \prod_{r \in \{u,v\}} \sum_{k=0}^{t_r-1} \frac{\mathsf{e}^{-x(t_r - y_e - y(E_1(r)))}(x(t_r - y_e - y(E_1(r)))^k}{k!} dy.$$
$$= \tilde{a}(y_e) \cdot \int_0^{x_c} \prod_{r \in \{u,v\}} \sum_{k=0}^{t_r-1} \frac{\mathsf{e}^{-x(t_r - y_e)}(x(t_r - y_e - y(E_1(r)))^k}{k!} dy. \tag{12}$$

For each choice of $3 \leq t_u \leq t_v \leq 47$, we numerically minimize (12) over variables y_e, $y(E_1(u))$ and $y(E_1(v))$, subject to the constraints $y(E_1(r)) + y_e \leq 1$ for each $r \in \{u, v\}$ (as guaranteed by Lemma 2). The minimum of (12) occurs when $t_u = t_v = 3$, $y_e = 0$, and $y(E_1(u)) = y(E_1(v)) = 1$, in which case $x_{u,c} = x_{v,c} = 1$, and so the value $\int_0^1 (\mathsf{e}^{-3x} + 2x\mathsf{e}^{-3x} + 2x^2\mathsf{e}^{-3x})^2 \, dx \geq 0.385$, is attained, which is strictly greater than β^*.

Next, suppose that $t_u \leq 47$, yet $t_v \geq 48$. In this case, we apply (9) to $\Pr[Q_0(u) < t_u \mid \pi(e) = x]$ for $x \in [0, x_{u,c})$, yet (10) to $\Pr[Q_0(v) < t_v \mid \pi(e) = x]$ for $x \in [0, 1]$. After simplification, this leaves us with the function

$$\tilde{a}(y_e) \cdot \int_0^{x_{u,c}} \mathsf{e}^{-xy(E_1(v))} \left(1 - \mathsf{e}^{\frac{-(t_v-1)(1-x)^2}{1+x}}\right) \sum_{k=0}^{t_u-1} \frac{\mathsf{e}^{-x(t_u-y_e)}(x(t_u - y(E_1(u)) - y_e))^k}{k!} dx, \tag{13}$$

of variables $y_e, y(E_1(v))$, and $y(E_1(u))$ subject to the constraints $y(E_1(r)) + y_e \leq 1$ for $r \in \{u, v\}$. Now, for any $x \in [0, 1]$, $t_v \to \left(1 - \mathsf{e}^{\frac{-(t_v-1)(1-x)^2}{1+x}}\right)$ is an increasing function in t_v, and $y(E_1(v)) \to \mathsf{e}^{-xy(E_1(v))}$ is a decreasing function in $y(E_1(v))$. Thus, the minimum of (13) occurs when $t_v = 48$, and $y(E_1(v)) = 1 - y_e$, in which case we are left with

$$\tilde{a}(y_e) \cdot \int_0^{x_{u,c}} \mathsf{e}^{-x(1-y_e)} \left(1 - \mathsf{e}^{\frac{-47(1-x)^2}{1+x}}\right) \sum_{k=0}^{t_u-1} \frac{\mathsf{e}^{-x(t_u-y_e)}(x(t_u - y(E_1(u)) - y_e))^k}{k!} dx. \tag{14}$$

By numerically minimizing (14) over y_e and $y(E_0(u))$ for each integer $t_u \in [3, 47]$, the minimum occurs when $t_u = 3$, $t_v = 48$, $y_e = 0$ and $y(E_1(u)) = 1$, in which case we get a lower bound of

$$\int_0^1 \mathsf{e}^{-x} \left(1 - \mathsf{e}^{\frac{-47(1-x)^2}{1+x}}\right) \left(\mathsf{e}^{-3x} + 2x\mathsf{e}^{-3x} + 2x^2\mathsf{e}^{-3x}\right) dx \geq 0.384,$$

which is again strictly larger than β^*.

Finally, when $t_u, t_v \geq 48$, we apply (10) to $\Pr[Q_0(r) < t_r \mid \pi(e) = x]$ for each $r \in \{u, v\}$ and all $x \in [0, 1]$. In this case, by again observing that the minimum occurs when $t_u = t_v = 48$ and $y(E_1(u)) = y(E_1(v)) = 1 - y_e$, we are left with

$$\tilde{a}(y_e) \cdot \int_0^1 \mathsf{e}^{-x(2-2y_e)} \left(1 - \mathsf{e}^{\frac{-47(1-x)^2}{1+x}}\right)^2 dx, \tag{15}$$

whose minimum is $\int_0^1 \mathsf{e}^{-2x} \left(1 - \mathsf{e}^{\frac{-47(1-y)^2}{1+y}}\right)^2 dy \geq 0.392$ when $y_e = 0$, which is strictly greater than β^*.

3.2.2. The case $t_u = 2$ When $t_u = t_v = 2$, suppose that $y_e = 0$, and $y(E_1(u)) = 0$, $y(E_1(v)) = 1$, in which case $y(E_0(u)) = 1$, $y(E_0(v)) = 2$ and so $x_{u,c} = 1/2$, $x_{v,c} = 1$ and $x_c = 1/2$. If we apply the bound of (9) for $x \in [0, x_c]$ as done in (12), then the lower bound attained is

$$\int_0^{1/2} \left(\mathsf{e}^{-2x} + \mathsf{e}^{-2x}2x\right)\left(\mathsf{e}^{-2x} + \mathsf{e}^{-2x}x\right) dx \approx 0.347, \tag{16}$$

which is strictly less than β^*. In fact, even if we additionally apply (10) for $x \in [x_c, 1]$, the resulting bound is still less than β^*. Similarly, using this analysis leads to bounds strictly less than β^* when $t_u = 2$ and $t_v > 2$.

To get resolve this issue in the analysis, we revisit (8) for the case when $t_u = 2$, in which we have

$$\tilde{a}(y_e)\int_0^1 \mathsf{e}^{-x(y(E_1(u))+y(E_1(v)))} \Pr[Q_0(u) < 2 \mid \pi(e) = x] \cdot \Pr[Q_0(v) < t_v \mid \pi(e) = x] dx. \tag{17}$$

Instead of directly applying Lemma 4 to (17), we shall first use an *exchange argument*, based on the construction from Derakhshan et al. [16] which was done for the special case when G is a star graph. Due to Lemma 2, we can apply this to our problem, and construct a super-graph of $\widetilde{G}$ from G by modifying *only* the neighborhood of u. In $\widetilde{G}$, the analogously defined term (17) is no larger than that of G. More, the new input $\widetilde{G}$ will be constructed so that we can ultimately apply Lemma 4 over a larger range of $x \in [0, 1]$, allowing us to avoid the complication in (16).

We now provide the exact details, where we recall that $E = \delta(u) \cup \delta(v)$ and $V = N(u) \cup N(v)$ by Lemma 2. By applying the construction from Lemma 3.10 of Derakhshan et al. [16] to G and $(y_f)_{f \in E}$, we can modify the neighborhood of u to get a super-graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ of G, whose patience values, edges probabilities, solution to (3), and arrival times we denote by $(\widetilde{t}_r)_{r \in \widetilde{V}}$, $(\widetilde{p}_f)_{f \in \widetilde{E}}$, $(\widetilde{y}_f)_{f \in \widetilde{E}}$ and $(\widetilde{\pi}(f))_{f \in \widetilde{E}}$, respectively. This input has $t_r = \widetilde{t}_r$ for each $r \in \{u, v\}$, satisfies the properties of Lemma 2, leaves the neighboring edges of v unchanged (including e), and assigns the maximum fractional value to the edges $f \in \delta_{\widetilde{G}}(u)$ with $\widetilde{p}_f = 1$. Formally,

1. $t_r = \widetilde{t}_r$ for each $r \in \{u, v\}$,
2. $\sum_{f \in \delta_{\widetilde{G}}(r)} \widetilde{x}_f = t_r$ for each $r \in \{u, v\}$, $\widetilde{p}_f \in \{0, 1\}$ for each $f \in \delta_{\widetilde{G}}(e)$, and $\widetilde{p}_e = 1$.
3. $\delta_G(v) = \delta_{\widetilde{G}}(v)$, and $y_f = \widetilde{y}_f$, $p_f = \widetilde{p}_f$ for all $f \in \delta_G(v)$,
4. $\sum_{f \in \delta_{\widetilde{G}}(u) : \widetilde{p}_f = 1} \widetilde{y}_f = 1 - \widetilde{y}_e$.

Moreover, suppose that for each $r \in \{u, v\}$, we define $\widetilde{E}_q(r) := \{f \in \delta_{\widetilde{G}}(r) \setminus \{e\} : \widetilde{p}_f = q\}$ for $q \in \{0, 1\}$ and $\widetilde{Q}_0(r)$ as the number of queries Algorithm 1 makes to $\widetilde{E}_0(r)$ before e when executing on $\widetilde{G}$ with $(\widetilde{y}_f)_{f \in \widetilde{E}}$. Then,

$$\int_0^1 \left(\mathsf{e}^{-xy(E_1(u))} \Pr[Q_0(u) < 2 \mid \pi(e) = x] dy - \mathsf{e}^{-x(1-y_e)} \Pr[\widetilde{Q}_0(u) < 2 \mid \widetilde{\pi}(e) = x]\right) dx \geq 0 \tag{18}$$

Let us denote $\widetilde{y}(S) := \sum_{f \in S} \widetilde{y}_f$ for each $S \subseteq \widetilde{E}$. We claim the following relation:

Lemma 6. *If $\widetilde{G}$ and $(\widetilde{y}_f)_{f \in \widetilde{E}}$ are constructed from G and $(y_f)_{f \in E}$ as above, then (17) is lower bounded by*

$$\tilde{a}(\widetilde{y}_e)\int_0^1 \mathsf{e}^{-x(\widetilde{y}(\widetilde{E}_1(u))+\widetilde{y}(\widetilde{E}_1(v)))} \Pr[\widetilde{Q}_0(u) < \widetilde{t}_u \mid \widetilde{\pi}(e) = x] \cdot \Pr[\widetilde{Q}_0(v) < \widetilde{t}_v \mid \widetilde{\pi}(e) = x] dx. \tag{19}$$

In order to prove Lemma 6, it will be convenient to use the following elementary bound involving the integral of a product of functions from [33].

Proposition 1 **(Proposition 11 in [33])**. *Suppose that $\lambda, \phi : [0, 1] \to \mathbb{R}$ are integrable, $\lambda \geq 0$, and λ is non-increasing. Moreover, assume that there exists $0 \leq z^* \leq 1$ such that $\phi(z) \geq 0$ for all $z \in [0, z^*]$, and $\phi(z) \leq 0$ for all $z \in [z^*, 1]$. Then,*

$$\int_0^1 \lambda(z)\phi(z)\, dz \geq \lambda(z^*) \int_0^1 \phi(z)\, dz$$

Proof of Lemma 6. We first simplify (19) using the properties 1. to 4. of $\widetilde{G}$ and $(\widetilde{y}_f)_{f \in \widetilde{E}}$ in relation to G and $(y_f)_{f \in E}$. Specifically, $\widetilde{t}_r = t_r$ for $r \in \{u, v\}$, $\widetilde{y}_e = y_e$, $\widetilde{y}(E_1(v)) = y(E_1(v))$, and $\widetilde{y}(E_1(u)) = 1 - y_e$. Moreover, by coupling the two executions, we may also conclude that $\Pr[Q_0(v) < t_v \mid \pi(e) = x] = \Pr[\widetilde{Q}_0(v) < \widetilde{t}_v \mid \widetilde{\pi}(e) = x]$ for all $x \in [0,1]$. Thus, (19) is equal to

$$\tilde{a}(y_e) \int_0^1 \mathsf{e}^{-x(1-y_e+y(E_1(v)))} \Pr[\widetilde{Q}_0(u) < 2 \mid \widetilde{\pi}(e) = x] \cdot \Pr[Q_0(v) < t_v \mid \pi(e) = x] dx. \tag{20}$$

Now, subtracting (20) from (17), and dividing by $\tilde{a}(y_e)$, we are left with

$$\int_0^1 \left(\mathsf{e}^{-xy(E_1(u))} \Pr[Q_0(u) < 2 \mid \pi(e) = x] - \mathsf{e}^{-x(1-y_e)} \Pr[\widetilde{Q}_0(u) < 2 \mid \widetilde{\pi}(e) = x] \right) \mathsf{e}^{-y(E_1(v))} \Pr[Q_0(v) < t_v \mid \pi(e) = x] dx \tag{21}$$

The function $\lambda(x) := \mathsf{e}^{-y(E_1(v))} \Pr[Q_0(v) < t_v \mid \pi(e) = x]$ is non-negative and non-increasing. Moreover, the function $\phi(x) := \mathsf{e}^{-xy(E_1(u))} \Pr[Q_0(u) < 2 \mid \pi(e) = x] - \mathsf{e}^{-x(1-y_e)} \Pr[\widetilde{Q}_0(u) < 2 \mid \widetilde{\pi}(e) = x]$ is initially non-negative and changes sign at most once on $[0,1]$. By combining Proposition 1 with (18), we conclude that (21) is non-negative, and so the lemma follows. □

Due to Lemma 6, with respect to lower bounding (17), we may assume without loss that the original input G and fractional solution $(y_f)_{f \in E}$ satisfies $y(E_0(u)) = 1 - y_e$. Recalling the definition of $x_{u,c}$ in (11), this implies that $x_{u,c} = 1$. The remainder of the proof depends on the value of t_v and follows a similar structure to the computations done in Subsection 3.2.1.

If $t_v = 2$, then by the symmetry of u and v, we may apply the exchange argument of Lemma 6 to v so as to ensure without loss that $y(E_0(v)) = 1 - y_e$, and so $x_{v,c} = 1$. If we then apply (9) over all $x \in [0,1]$, we get a lower bound of

$$\tilde{a}(y_e) \int_0^1 \left(\mathsf{e}^{-x(2-y_e)} + x\mathsf{e}^{-x(2-y_e)} \right)^2 dx, \tag{22}$$

which is minimized when $y_e = 0$, in which case we get a lower bound of exactly β^*.

If $2 < t_v \leq 47$, then since $x_{u,c} = 1$, we have $x_{v,c} = x_c = (t_v - 1)/(t_v - y(E_1(v)) - y_e)$. Thus, we get a lower bound of

$$\tilde{a}(y_e) \cdot \int_0^{x_{v,c}} (\mathsf{e}^{-x(2-y_e)} + x\mathsf{e}^{-x(2-y_e)}) \sum_{k=0}^{t_v-1} \frac{\mathsf{e}^{-x(t_v-y_e)}(x(t_v - y_e - y(E_1(v)))^k}{k!} dx, \tag{23}$$

which is a function of variables y_e and $y(E_1(v))$, subject to the constraint $y(E_1(v)) + y_e \leq 1$. We numerically verify that (23) is minimized when $t_v = 3$, $y_e = 0$ and $y(E_1(v)) = 1$, where it takes a value of $\int_0^1 (\mathsf{e}^{-2x} + x\mathsf{e}^{-2x})(\mathsf{e}^{-3x} + x\mathsf{e}^{-3x} + 2x^2\mathsf{e}^{-3x}) dx \geq 0.383$.

Finally, for $t_v \geq 48$, we apply Lemma 5 to get a lower bound of

$$\tilde{a}(y_e) \cdot \int_0^1 (\mathsf{e}^{-x(2-y_e)} + x\mathsf{e}^{-x(2-y_e)}) \mathsf{e}^{-xy(E_1(v))} \left(1 - \mathsf{e}^{\frac{-(t_v-1)(1-x)^2}{1+x}} \right) dx, \tag{24}$$

which is a function of y_e and $y(E_1(v))$, subject to the constraint $y(E_1(v)) + y_e \leq 1$. By applying the same observations used to simplify (13), we know that the minimum of (24) occurs when $t_v = 48$ and $y(E_1(v)) = 1 - y_e$. This leads to

$$\tilde{a}(y_e) \cdot \int_0^1 (\mathsf{e}^{-x(2-y_e)} + x\mathsf{e}^{-x(2-y_e)}) \mathsf{e}^{-x(1-y_e)} \left(1 - \mathsf{e}^{\frac{-47(1-x)^2}{1+x}} \right) dx, \tag{25}$$

and we numerically verify that (25) attains its minimum at $y_e = 0$, where it is ≥ 0.3828, a value strictly larger than β^*.

3.3. Infinite Patience When all patience values are ∞, an edge cannot be blocked by any endpoint exhausting its patience, and so the lower bound on $a(e)\Pr[e \text{ is safe}]$ becomes $\tilde{a}(y_e)\int_0^1 \mathsf{e}^{-2x(1-y_e)}dx$. It is easy to verify that for the linear and exponential attenuation functions with $\alpha = 1/2$, the minimum still occurs at $y_e = 0$. This gives us Corollary 5.

COROLLARY 5 **(corresponds to Corollaries 1 to 3 from Subsection 1.1)**. *In the case where all patiences are ∞, for any feasible solution $(y_e)_{e\in E}$ to LP (3): Algorithm 1, using the effective attenuation functions from Definition 4 with $\alpha = 1/2$, considers the edges in a random order and probes every edge $e = \{u,v\}$ with probability at least*

$$\left(\int_{x=0}^{1} \mathsf{e}^{-2x}dx\right) y_e = \frac{1}{2}\left(1-\mathsf{e}^{-2}\right) y_e$$

which is $\approx 0.432 y_e$. In this case, the algorithm is a 0.432-approximation and a 0.432-balanced Random-order Contention Resolution Scheme for the matching polytope.

Similarly, when all patience values are ∞, and we have a star graph, then the lower bound on $a(e)\Pr[e \text{ is safe}]$ simplifies to $\tilde{a}(y_e)\int_0^1 \mathsf{e}^{-x(1-y_e)}dx$. In this case, taking $\tilde{a}(z) = \frac{1-z}{1-\mathsf{e}^{-(1-z)}}(1-1/\mathsf{e}) = (1-1/\mathsf{e})/\int_0^1 \mathsf{e}^{-x(1-z)}dx$ gives an attenuation function, for which $\tilde{a}(y_e)\int_0^1 \mathsf{e}^{-x(1-y_e)}dx = 1-1/\mathsf{e}$ for all $y_e \in [0,1]$.

COROLLARY 6 **(corresponds to Corollary 4 from Section 1)**. *For any star graph with infinite patiences and any feasible solution $(y_e)_{e\in E}$ to the LP (3), Algorithm 1, using the attenuation function $\tilde{a}(z) = \frac{1-z}{1-\mathsf{e}^{-(1-z)}}(1-1/\mathsf{e})$, considers the edges in a random order and probes every edge $e = \{u,v\} \in E$ with probability at least*

$$\left(\int_{x=0}^{1} \mathsf{e}^{-x}dx\right) y_e = \left(1-\frac{1}{\mathsf{e}}\right) y_e.$$

This yields a $(1-1/e)$-balanced Random-order Contention Resolution Scheme for rank-1 matroids which does not adapt to the time of arrival of each element.

4. Algorithm and Analysis for Result 2 Let $G = (V,E)$ be a bipartite graph with bipartition $V = V_1 \cup V_2$. Assume $t_u = 1$ for all $u \in V_1$.

Description of algorithm. We first solve the standard LP (3) to obtain an optimal solution $(y_e)_{e\in E}$ satisfying $\sum_{e\in\delta(v)} y_e p_e \le 1$ and $\sum_{e\in\delta(v)} y_e \le t_v$ for all $v \in V$. Then, we run the rounding procedure of Gandhi et al. [25] on y_e to get an integral solution $Y_e \in \{0,1\}$. This guarantees that for each vertex $u \in V_1$ that $\sum_{e\in\delta(u)} Y_e \le 1$. Thus, at most one vertex $e \in \delta(u)$ will be rounded to $Y_e = 1$ for every $u \in V_1$. Thus, in the rounded graph $\hat{G} := (V, \hat{E})$, where $\hat{E} := \{e \in E : Y_e = 1\}$, each vertex $v \in V_2$ is the center of a star graph. For each vertex $v \in V_2$, we probe the edges $e \in \delta(v)$ in decreasing order of weight, for each e with $Y_e = 1$.

Analysis of algorithm. The expected value achieved by this strategy is

$$\mathbb{E}[\mathsf{ALG}] := \mathbb{E}\left[\sum_{v\in V_2} W(v)\right] = \sum_{v\in V_2} \mathbb{E}[W(v)]$$

where $W(v)$ denotes the weight of the edge matched by the algorithm (if any) for a vertex v. In our analysis, we consider each vertex $v \in V_2$ separately, since in the rounded graph, it is the center of a star graph that is disconnected from any other vertices of V_2. We first utilize the negative correlation property of our dependent rounding technique [25] to establish the following lemma.

LEMMA 7. *Consider a fixed vertex $v \in V_2$. Label the edges of $\delta(v)$ from 1 to $k := |\delta(v)|$ such that $w_1 \ge w_2 \ge \cdots \ge w_k$. Then:*

$$\mathbb{E}[W(v)] \ge \sum_{i=1}^{k} w_i z_i \prod_{j=1}^{i-1}(1-z_j)$$

Proof. Recall from Section 4 that $Y_i = 1$ if edge i was included in the rounding produced by the algorithm. Let P_i be a Bernoulli random variable indicating whether edge i is active. Then, we can write the expected value of $W(v)$ as

$$\mathbb{E}[W(v)] = \mathbb{E}\left[\sum_{i=1}^{k} w_i Y_i P_i \prod_{i=1}^{i-1}(1 - Y_j P_j)\right]$$

Next, let $\hat{F}(i) = 1 - \prod_{j=1}^{i}(1 - Y_j P_j)$. Observe that

$$\hat{F}(i) = 1 - (1 - P_i Y_i)\prod_{j=1}^{i-1}(1 - P_j Y_j) = 1 - \prod_{j=1}^{i-1}(1 - P_j Y_j) + P_i Y_i \prod_{j=1}^{i-1}(1 - P_j Y_j)$$

and thus we get

$$\hat{F}(i) - \hat{F}(i-1) = P_i Y_i \prod_{j=1}^{i-1}(1 - P_j Y_j).$$

This gives the following expression for $W(v)$, letting $w_{k+1} := 0$ for convenience:

$$\mathbb{E}\left[\sum_{i=1}^{k} w_i(\hat{F}(i) - \hat{F}(i-1))\right] = \mathbb{E}\left[\sum_{i=1}^{k} \hat{F}(i)(w_i - w_{i+1})\right]$$

By the negative correlation property (2) of the dependent rounding [25], and the independence of each P_j from both $P_{j'}$ (for all $j' \neq j$) and from $Y_{j'}$ for all j', we can see that

$$\mathbb{E}\left[\prod_{j=1}^{i}(1 - Y_j P_j)\right] \leq \prod_{j=1}^{i}(1 - y_j p_j)$$

and thus $\mathbb{E}[\hat{F}(i)] \geq 1 - \prod_{j=1}^{i}(1 - y_j p_j)$. As usual, we let $z_i := p_i y_i$. We denote $F(i) := 1 - \prod_{j=1}^{i}(1 - z_j)$, and from the negative correlation (2), we have that $\mathbb{E}[\hat{F}(i)] \geq F(i)$, giving our desired result:

$$\mathbb{E}[W(v)] = \mathbb{E}\left[\sum_{i=1}^{k} \hat{F}(i)(w_i - w_{i+1})\right] \geq \sum_{i=1}^{k} F(i)(w_i - w_{i+1}) = \sum_{i=1}^{k} w_i z_i \prod_{j=1}^{i-1}(1 - z_j)$$

□

Next, we work with the right-hand side expression, giving Lemma 8.

LEMMA 8. *For $i \in [k]$, let $z_i \in [0,1]$ be such that $\sum_{i=1}^{k} z_i \leq 1$. Let $w_1, w_2, \ldots, w_k \in \mathbb{R}^+$ be such that $w_1 \geq w_2 \geq \cdots \geq w_k$. Then,*

$$\sum_{i=1}^{k} w_i z_i \prod_{j=1}^{i-1}(1 - z_j) \geq \left(1 - \frac{1}{\mathsf{e}}\right)\sum_{i=1}^{k} w_i z_i \tag{26}$$

Proof. As in the previous proof, let $w_{k+1} = 0$. Then, let $F(i) = 1 - \prod_{j=1}^{i}(1 - z_j)$ for $i \in \{0, 1, \ldots, k\}$. Note that $F(0) = 0$.

We begin similarly to the proof of Lemma 7: Observe that

$$F(i) = 1 - (1 - z_i)\prod_{j=1}^{i-1}(1 - z_j) = 1 - \prod_{j=1}^{i-1}(1 - z_j) + z_i \prod_{j=1}^{i-1}(1 - z_j) = F(i-1) + z_i \prod_{j=1}^{i-1}(1 - z_j)$$

and thus we get

$$F(i) - F(i-1) = z_i \prod_{j=1}^{i-1}(1-z_j).$$

For convenience, define $w_{k+1} = 0$. Then, we proceed as follows:

$$\sum_{i=1}^{k} w_i z_i \prod_{j=1}^{i-1}(1-z_j) = \sum_{i=1}^{k} w_i (F(i) - F(i-1)) = \sum_{i=1}^{k} F(i)(w_i - w_{i+1}) \tag{27}$$

Next, let $F^*(i) = \sum_{j=1}^{i} z_j$. It is easy to see then that $F^*(i) - F^*(i-1) = z_i$. Then we can observe that

$$\sum_{i=1}^{k} w_i z_i = \sum_{i=1}^{k} w_i (F^*(i) - F^*(i-1)) = \sum_{i=1}^{k} F^*(i)(w_i - w_{i+1}) \tag{28}$$

Using the inequality of arithmetic and geometric means, we see that

$$\prod_{j=1}^{i}(1-z_j) \le \left(\frac{\sum_{j=1}^{i}(1-z_j)}{i}\right)^i = \left(1 - \frac{\sum_{j=1}^{i} z_j}{i}\right)^i = \left(1 - \frac{F^*(i)}{i}\right)^i \le \mathsf{e}^{-F^*(i)}$$

which gives us a bound on $F(i)$, shown in (29).

$$F(i) \ge 1 - \mathsf{e}^{-F^*(i)} \ge (1 - 1/\mathsf{e})\, F^*(i) \tag{29}$$

In (29) above, we use the fact that $0 \le F^*(i) = \sum_{j=1}^{i} z_j \le \sum_{j=1}^{k} z_j \le 1$. Then, combining (27), (29), and (28) gives the desired bound of (26).

$$\sum_{i=1}^{k} w_i z_i \prod_{j=1}^{i}(1-z_j) = \sum_{i=1}^{k} F(i)(w_i - w_{i+1}) \ge \sum_{i=1}^{k}(1-1/\mathsf{e})F^*(i)(w_i - w_{i+1}) = (1-1/\mathsf{e}) \sum_{i=1}^{k} w_i z_i$$

□

For $v \in V_2$, let $\mathsf{OPT}(v) := \sum_{e \in \delta(v)} w_e z_e$. Using Lemma 7, we are then able to derive the following, from which our main result immediately follows.

LEMMA 9. *For any vertex $v \in V_2$, we have*

$$\mathbb{E}[W(v)] \ge \left(1 - \frac{1}{\mathsf{e}}\right) \mathsf{OPT}(v)$$

Proof. We can use Lemmas 7 and 8 together to get:

$$\mathbb{E}[W(v)] \ge \sum_{i=1}^{k} w_i z_i \prod_{j=1}^{i-1}(1-z_j) \ge \left(1 - \frac{1}{\mathsf{e}}\right) \sum_{i=1}^{k} w_i z_i = \left(1 - \frac{1}{\mathsf{e}}\right) \mathsf{OPT}(v)$$

where the first inequality follows from Lemma 7, the second from Lemma 8, and the final equality from the definition of $\mathsf{OPT}(v)$. □

THEOREM 2 **(corresponds to Result 2 from Subsection 1.1)**. *For any bipartite graph with a unit-patience side, let $(y_e)_{e \in E}$ denote an optimal solution to the LP (3). Then our algorithm above matches expected weight at least $(1 - 1/\mathsf{e}) \sum_{v \in V_2} \sum_{e \in \delta(v)} w_e p_e y_e$, yielding a $(1 - 1/\mathsf{e})$-approximation.*

Proof of Theorem 2. Applying Lemma 9, the expected value of our algorithm is

$$\mathbb{E}[\mathsf{ALG}] = \sum_{v \in V_2} \mathbb{E}[W(v)] \geq \sum_{v \in V_2} (1 - 1/\mathsf{e})\mathsf{OPT}(v) = (1 - 1/\mathsf{e})\mathsf{OPT}$$

as desired. □

We note that the problem of stochastic matching on bipartite graphs with a unit patience side is a special case of particular interest, as it captures "Problem A" in Hikima et al. [28]. As shown in Lemma A of Hikima et al. [28], an α-approximation for Problem A implies an α-approximation for the *Integrated Stochastic Problem for Control Variables and Bipartite Matching* (ISPCB). This is discussed further at the end of this section.

Why we cannot use an existing Contention Resolution Scheme. Lemmas 7 and 9 show that the *total* expected weight collected from edges in $\delta(v)$ is at least $(1 - 1/\mathsf{e}) \cdot \mathsf{OPT}(v)$. We now explain why it is not possible to use a correlation-agnostic Contention Resolution Scheme to match *every* edge $e \in \delta(v)$ with probability at least $(1 - 1/\mathsf{e})$. We consider the following example, in which every edge e has the same value of $z_e = p_e y_e$. Therefore, a correlation-agnostic Contention Resolution Scheme would treat the edges symmetrically, doing no better than a strategy which considers the edges in a random order until an edge e is matched (which requires both Y_e to be rounded to 1 and for edge e to exist). However, due to the first-stage dependent rounding for Y_e, the probabilities of edges being matched end up being negatively correlated (as defined in (2) in the introduction). This negative correlation among the neighbors of a particular edge "0" can increase the probability of edge 0 being blocked to 1/2 (something not possible under independence), as we now demonstrate.

Example 1. Consider a star graph with $T + 1$ edges, whose central vertex has patience 2. Take the fractionally-feasible solution $y_0 = 1$, $p_0 = 1/(T + 1)$, and $y_1 = \cdots = y_T = 1/T$, $p_1 = \cdots = p_T = T/(T + 1)$. Any rounding procedure which satisfies the patience w.p. 1 and preserves the marginal probabilities will set $Y_0 = Y_i = 1$, $Y_{i'} = 0$ for all other i', with i drawn uniformly from $\{1, \ldots, T\}$. This implies that w.p. $1 - 1/(T + 1)$, one of the edges $1, \ldots, T$ will match upon being uncountered. Any correlation-agnostic procedure would treat all edges symmetrically, since they all have the same value of $z_i = p_i y_i = 1/(T + 1)$. Consequently, edge 0 will have a $(1 - 1/(T + 1))/2$ probability of being blocked, whenever it is considered later than the aforementioned edge which matches upon being encountered. Therefore, the correlation-agnostic procedure cannot be better than 1/2-balanced as $T \to \infty$.

4.1. Improvement to Approximation Ratio of Hikima et al. [28]

The work of Hikima et al. [28] introduces and studies a problem they call *Integrated Stochastic Problem for Control variables and Bipartite Matching* (ISPCB). This problem is a two-stage bipartite matching problem where the algorithm is given a bipartite graph $G = (V_1 \cup V_2, E)$ and must set a control variable x_u for each $u \in V_1$. Then, each vertex $u \in V_1$ *leaves* the graph with probability $p_u(x_u)$ (where $p_u(x_u)$ is some known probability which depends on x_u), and then the algorithm computes a maximum-weight matching on the resulting graph.

Hikima et al. [28] prove an approximation guarantee for ISPCB by first studying a problem they denote Problem A. Problem A can be seen as the problem of stochastic bipartite matching with a unit-patience side which we study here: For each edge $(u, v) \in E$, $p_{uv} = p_u(x_u)$, $t_u = 1$ for each $u \in V_1$, and $t_v = \infty$ for each $v \in V_2$. The proof of Theorem 1 in Hikima et al. [28] shows that an α-approximation of Problem A implies an α-approximation for ISPCB. The 1/3-approximation then follows from the 1/3-approximation for stochastic bipartite matching in Bansal et al. [7]. Our Result 2 captures Problem A, which thus implies a $(1 - 1/\mathsf{e})$-approximation for ISPCB, improving on the previous 1/3-approximation of Hikima et al. [28]

Acknowledgments. A preliminary version of this paper [11] appeared at the 35th Conference on Neural Information Processing Systems (NeurIPS), 2021. Nathaniel Grammel and Aravind Srinivasan were supported in part by NSF award CCF-1749864, and by research awards from Amazon and Google.

References

[1] Adamczyk M, Grandoni F, Mukherjee J (2015) Improved approximation algorithms for stochastic matching. *Algorithms-ESA 2015*, 1–12 (Springer).

[2] Adamczyk M, Włodarczyk M (2018) Random order contention resolution schemes. *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 790–801 (IEEE).

[3] Ahmadi S, Ahmed F, Dickerson JP, Fuge M, Khuller S (2020) An algorithm for multi-attribute diverse matching. Bessiere C, ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 3–9 (ijcai.org).

[4] Ahmed F, Dickerson JP, Fuge M (2017) Diverse weighted bipartite b-matching. Sierra C, ed., *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 35–41 (ijcai.org).

[5] Aminian MR, Niazadeh R, Nuti P (2026) Stationary online contention resolution schemes. URL http://dx.doi.org/10.2139/ssrn.6384199.

[6] Antoniadis A, Gouleakis T, Kleer P, Kolev P (2020) Secretary and online matching problems with machine learned advice. Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds., *Advances in Neural Information Processing Systems*, volume 33, 7933–7944 (Curran Associates, Inc.), URL https://proceedings.neurips.cc/paper/2020/file/5a378f8490c8d6af8647a753812f6e31-Paper.pdf.

[7] Bansal N, Gupta A, Li J, Mestre J, Nagarajan V, Rudra A (2012) When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica* 63(4):733–762.

[8] Baveja A, Chavan A, Nikiforov A, Srinivasan A, Xu P (2018) Improved bounds in stochastic matching and optimization. *Algorithmica* 80(11):3225–3252.

[9] Borodin A, MacRury C (2025) Online bipartite matching in the probe-commit model. *Mathematical Programming* 1–54.

[10] Brubach B, Grammel N, Ma W, Srinivasan A (2021) Follow your star: New frameworks for online stochastic matching with known and unknown patience. Banerjee A, Fukumizu K, eds., *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, 2872–2880 (PMLR), URL http://proceedings.mlr.press/v130/brubach21a.html.

[11] Brubach B, Grammel N, Ma W, Srinivasan A (2021) Improved guarantees for offline stochastic matching via new ordered contention resolution schemes. *Advances in Neural Information Processing Systems* 34.

[12] Brubach B, Sankararaman KA, Srinivasan A, Xu P (2020) Attenuate locally, win globally: Attenuation-based frameworks for online stochastic matching with timeouts. *Algorithmica* 82(1):64–87.

[13] Bruggmann S, Zenklusen R (2020) An optimal monotone contention resolution scheme for bipartite matchings via a polyhedral viewpoint. *Mathematical Programming* 1–51.

[14] Chekuri C, Livanos V (2021) On submodular prophet inequalities and correlation gap. *CoRR* abs/2107.03662, URL https://arxiv.org/abs/2107.03662.

[15] Chen N, Immorlica N, Karlin AR, Mahdian M, Rudra A (2009) Approximating matches made in heaven. Albers S, Marchetti-Spaccamela A, Matias Y, Nikoletseas S, Thomas W, eds., *Automata, Languages and Programming*, 266–278 (Berlin, Heidelberg: Springer Berlin Heidelberg), ISBN 978-3-642-02927-1.

[16] Derakhshan M, Ghasemi A, MacRury C (2026) Approximation algorithms for action-reward query-commit matching. *arXiv preprint arXiv:2603.13487* .

[17] Dughmi S (2020) The Outer Limits of Contention Resolution on Matroids and Connections to the Secretary Problem. Czumaj A, Dawar A, Merelli E, eds., *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics*

(LIPIcs), 42:1–42:18 (Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik), ISBN 978-3-95977-138-2, ISSN 1868-8969, URL http://dx.doi.org/10.4230/LIPIcs.ICALP.2020.42.

[18] Ehsani S, Hajiaghayi M, Kesselheim T, Singla S (2018) Prophet secretary for combinatorial auctions and matroids. *Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms*, 700–714 (SIAM).

[19] Esfandiari H, Hajiaghayi M, Liaghat V, Monemizadeh M (2017) Prophet secretary. *SIAM Journal on Discrete Mathematics* 31(3):1685–1701.

[20] Esfandiari H, Korula N, Mirrokni V (2016) Bi-objective online matching and submodular allocations. Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, eds., *Advances in Neural Information Processing Systems*, volume 29 (Curran Associates, Inc.), URL https://proceedings.neurips.cc/paper/2016/file/0966289037ad9846c5e994be2a91bafa-Paper.pdf.

[21] Ezra T, Feldman M, Gravin N, Tang ZG (2020) Online stochastic max-weight matching: prophet inequality for vertex and edge arrival models. *Proceedings of the 21st ACM Conference on Economics and Computation*, 769–787.

[22] Fata E, Ma W, Simchi-Levi D (2019) Multi-stage and multi-customer assortment optimization with inventory constraints. *Available at SSRN 3443109* .

[23] Fu H, Tang ZG, Wu H, Wu J, Zhang Q (2021) Random Order Vertex Arrival Contention Resolution Schemes for Matching, with Applications. Bansal N, Merelli E, Worrell J, eds., *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 68:1–68:20 (Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik), ISBN 978-3-95977-195-5, ISSN 1868-8969, URL http://dx.doi.org/10.4230/LIPIcs.ICALP.2021.68.

[24] Gamlath B, Kale S, Svensso O (2019) Beating greedy for stochastic bipartite matching. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2841–2854, SODA '19 (USA: Society for Industrial and Applied Mathematics).

[25] Gandhi R, Khuller S, Parthasarathy S, Srinivasan A (2006) Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)* 53(3):324–360.

[26] Gravin N, Wang H (2019) Prophet inequality for bipartite matching: merits of being simple and non adaptive. *Proceedings of the 2019 ACM Conference on Economics and Computation*, 93–109.

[27] Guruganesh G, Lee E (2017) Understanding the correlation gap for matchings. *arXiv preprint arXiv:1710.06339* .

[28] Hikima Y, Akagi Y, Kim H, Kohjima M, Kurashima T, Toda H (2021) Integrated optimization of bipartite matching and its stochastic behavior: New formulation and approximation algorithm via min-cost flow optimization. *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*.

[29] Karp RM, Sipser M (1981) Maximum matching in sparse random graphs. *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, 364–375 (IEEE).

[30] Lee E, Singla S (2018) Optimal online contention resolution schemes via ex-ante prophet inequalities. *26th Annual European Symposium on Algorithms (ESA 2018)* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik).

[31] Ma W, MacRury C, Nuti P (2024) Online matching and contention resolution for edge arrivals with vanishing probabilities. *Proceedings of the 25th ACM Conference on Economics and Computation*, 159, EC '24 (New York, NY, USA: Association for Computing Machinery), ISBN 9798400707049, URL http://dx.doi.org/10.1145/3670865.3673588.

[32] MacRury C, Ma W (2024) Random-order contention resolution via continuous induction: Tightness for bipartite matching under vertex arrivals. Mohar B, Shinkar I, O'Donnell R, eds., *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, 1629–1640 (ACM), URL http://dx.doi.org/10.1145/3618260.3649788.

[33] MacRury C, Ma W, Grammel N (2025) On (random-order) online contention resolution schemes for the matching polytope of (bipartite) graphs. *Oper. Res.* 73(2):689–703, ISSN 0030-364X, URL http://dx.doi.org/10.1287/opre.2023.0339.

[34] Nanda V, Xu P, Sankararaman KA, Dickerson JP, Srinivasan A (2020) Balancing the tradeoff between profit and fairness in rideshare platforms during high-demand hours. *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2210–2217 (AAAI Press).

[35] Pollner T, Roghani M, Saberi A, Wajc D (2024) Improved online contention resolution for matchings and applications to the gig economy. *Math. Oper. Res.* 49(3):1582–1606, URL http://dx.doi.org/10.1287/MOOR.2023.1388.

[36] Qiu F, Singla S (2022) Submodular Dominance and Applications. Chakrabarti A, Swamy C, eds., *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, volume 245 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 44:1–44:21 (Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik), ISBN 978-3-95977-249-5, ISSN 1868-8969, URL http://dx.doi.org/10.4230/LIPIcs.APPROX/RANDOM.2022.44.

[37] Xu P, Shi Y, Cheng H, Dickerson JP, Sankararaman KA, Srinivasan A, Tong Y, Tsepenekas L (2019) A unified approach to online matching with conflict-aware constraints. *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 2221–2228 (AAAI Press).