Attenuate Locally, Win Globally: An Attenuation-based Framework for Online Stochastic Matching with Timeouts∗

Brian Brubach
University of Maryland, College Park
bbrubach@cs.umd.edu

Karthik Abinav Sankararaman
University of Maryland, College Park
kabinav@cs.umd.edu

Aravind Srinivasan
University of Maryland, College Park
srin@cs.umd.edu

Pan Xu
University of Maryland, College Park
panxu@cs.umd.edu

ABSTRACT
Online matching problems have garnered significant attention in recent years due to numerous applications. Many of them capture the uncertainty in the real world by including stochasticity in both the arrival process and the matching process. The Online Stochastic Matching with Timeouts problem introduced by Bansal, Gupta, Li, Mestre, Nagarajan, and Rudra (Algorithmica, 2012) models matching markets (e.g. E-Bay, Amazon). Buyers arrive from an independent and identically distributed (i.i.d.) known distribution on buyer profiles and can be shown a list of items one at a time. Each buyer has some probability of purchasing each item and a limit (timeout) on the number of items they can be shown.

Bansal et al. (Algorithmica, 2012) gave a 0.12-competitive algorithm which was improved by Adamczyk, Grandoni, and Mukherjee (ESA, 2015) to 0.24. We present an online attenuation framework that uses an algorithm for offline stochastic matching as a black box. Our main contributions are as follows. On the upper bound side, we show that this framework combined with a black box adapted from Bansal et al. (Algorithmica, 2012) yields an online algorithm which nearly doubles the ratio to 0.46. On the lower bound side, we show that no algorithm can achieve a ratio better than 0.632 using the common LP for this problem. This framework has a high potential for further improvements since new algorithms for offline stochastic matching can lead directly to improvements for the online problem.

Our online framework also has the potential for a variety of extensions. For example, we introduce a natural generalization: Online Stochastic Matching with Two-sided Timeouts in which both online and offline vertices have timeouts. Our framework provides the first algorithm for this problem achieving a ratio of 0.31. We accomplish this by proposing a new black box algorithm for offline stochastic matching on star graphs, which may be of independent interest. This new black box improves the approximation ratio for the offline stochastic matching problem on star graphs from 0.56 by Adamczyk et al. (ESA 2015).

Keywords
Randomized Algorithms; Online Algorithm; Online Stochastic Matching

1. INTRODUCTION
Consider a typical problem in matching markets (e.g. E-Bay, Amazon). We have a certain number of buyer profiles and items. Let us denote the set of buyer profiles by $V$ and the set of items by $U$. In our initial problem, we assume that an item is present in the market until bought. We have an $n$ round market process. In each round, a buyer is sampled uniformly at random (with replacement) from the buyer profiles. We assume that we have $n$ buyer profiles (this assumption is called integral arrival rates in the literature [26]). For every item $u \in U$ and buyer $v \in V$, let $e = (u, v)$ and $p_{e}$ denote the probability that buyer $v$ will buy the item $u$. If $v$ buys $u$, then we obtain a reward of $w_{e}$. When a buyer arrives, she will be shown items one-by-one until she chooses to buy one. Since every buyer $v$ has a limited attention-span, she can be shown at most $t_{v}$ items where $t_{v}$ is typically called a timeout or patience constraint for $v$. The goal is to design an algorithm such that the expected reward at the end of the $n$ round process is maximized. Usually, the buyer profiles are gathered based on past information. Hence, we will assume that the system knows all the buyer profiles as well as the buying probabilities for each item profile pair.

We model this as the Online Stochastic Matching with Timeouts problem introduced by Bansal, Gupta, Li, Mestre, Nagarajan, and Rudra [7]. Formally, this is a probe-commit model for online bipartite stochastic matching. Let us represent an item-buyer pair by an edge $e = (u, v)$. The algorithm can “probe” $e$ to see if the buyer $v$ “buys” the item $u$. If she does, the decision of selling $u$ to $v$ is made irrevocably (commit). We now define the model in an abstract setting. We are given a bipartite graph $G = (E, U, V)$ as input. Each edge $e$ has a probability $p_{e}$ (independent of other edges) of existing (modeling a buyer’s interest in an item) and a weight $w_{e}$. Each vertex $v \in V$ has a timeout $t_{v}$; however, the vertices in $U$ have no timeout restrictions (equivalently, we can say they have timeouts of infinity). These values are all known a priori. The algorithm proceeds in $n$ rounds. In each round, a vertex $v$ arrives and we can probe at most $t_{v}$ neighbors in an attempt to match $v$. Arrivals are drawn with replacement from a known i.i.d. distribution on $V$. For"
simplicity, we will consider the uniform distribution. If a probed edge \((u,v)\) is found to exist, we must match \(v\) to \(u\) and no more probing is allowed for that round. The vertices in \(U\) can be matched at most once. The vertices in \(V\) are called types (a buyer profile) and two or more arrivals of the same type \(v \in V\) are considered distinct vertices (two different buyers of a particular profile) which can each be probed up to \(t_v\) times and matched to separate neighbors in \(U\). The objective is to maximize the expected weight (or profit) of the final matching obtained. Unsurprisingly, this model captures problems beyond the buyer/seller scenario described above. Various online stochastic matching problem have been considered for online advertising and many other applications [26].

We give new algorithms for the Online Stochastic Matching with Timeouts problem that improve the competitive ratio over the previous work. We then introduce a new model wherein the seller selling item \(u\) has a limited patience. In this generalization, called Online Stochastic Matching with Two-sided Timeouts, we have an additional constraint that every vertex \(u \in U\) has a timeout \(t_u\) and the algorithm can probe at most \(t_u\) neighbors of \(u\) across the \(n\) rounds. We give the first constant factor approximation algorithm for this generalized setting.

**Related Work:** Online bipartite matching and its variants have been an active area of study beginning with the seminal work of Karp, Vazirani and Vazirani [23]. They studied weighted bipartite matching in the adversarial arrival model and gave an optimal \((1 - 1/e)\) competitive algorithm. The advent of e-commerce and ad-allocation brought more variants of this problem. For an exhaustive literature survey, refer to the book by Aranyak Mehta [26]. Many variants study arrivals in a random or adversarial order on an unknown set of online vertices. In the i.i.d. arrival model, Feldman et al. [17], Bahmani and Kapralov [6], Manshadi et al. [25], Haeupler et al. [20], Jaillet and Lu [21], and Brubach et al. [9] gave improved algorithms to the Online Stochastic Matching problem. The term stochastic here refers to the known i.i.d. arrival model, although some of those papers also address stochastic edge models.

Beyond online matching, other related problems have been studied. The adwords problem was introduced by Mehta et al. [27] and subsequently studied by Buchbinder et al. [10] and Devanur and Hayes [12]. More variants have been considered by Devanur et al. [14], Devanur et al. [15], and Devanur and Jain [13]. Other generalizations that capture the online matching problem are Online Packing Linear Programs by Feldman et al. [16] and Agrawal et al. [3] and the study of Online Convex Programs by Agrawal and Devanur [2].

Bansal et al. [7] introduced the problem of Online Stochastic Matching with Timeouts and gave the first constant factor competitive ratio of 0.12. This was later improved to 0.24 by Adamczyk et al. [1]. In both works, they considered the notion of timeouts only on the online vertices. The original motivation for timeouts came from the patience constraints in the Offline Stochastic Matching problem. That offline problem was first introduced by Chen et al. [11] and later studied by Bansal et al. [7], Adamczyk et al. [1], and Baveja et al. [8]. A generalization to packing problems was studied by Gupta and Nagarajan [19].

The Online Stochastic Matching with Two-sided Timeouts problem which we introduce has no direct previous work. One related problem is Online \(b\)-matching wherein the offline vertices can each be matched at most \(b\) times. This is somewhat similar to having timeouts on the offline vertices in online stochastic matching. In the adversarial setting, \(b\)-matching was first studied by Kalyanasundaram and Pruhs [22] and they gave an optimal algorithm. Alaei et al. [4] studied the prophet-inequality problem and considered the stochastic i.i.d. setting. They gave an algorithm whose competitive ratio is \(1 - \frac{1}{b\cdot e}\). Alaei et al. [5] and Brubach et al. [9] studied the Online \(b\)-matching Stochastic Matching problem in the i.i.d. setting and gave a ratio of \(1 - O(1/\sqrt{b})\).

### 1.1 Preliminaries

Let us start by giving the theoretical preliminaries to our problem. Henceforth, we refer to the items \(U\) as the offline vertices and the buyers \(V\) as the online vertices. We refer to the reward \(w_{e, u}\) as the weight of the edge while we refer to the buying probability \(p_{e, v}\) as the edge probability. We use the words time and round of the process interchangeably. When we say at time \(t \in [n]\), it refers to the beginning of step \(t\) in the process. Also, following many of the related works, we assume that the arrival rate of each online vertex type is integral and hence, WLOG assume it to be 1. Additionally, the competitive ratio for this class of problems is defined slightly differently from usual online algorithms (see [26]). We say that a vertex in the offline set \(U\) is safe at some time \(t\) if it has not been matched in a previous round. Similarly we say an edge \(e = (u, v)\) is safe if \(u\) is safe during the arrival of \(v\). We say a safe edge is safely probed if it is probed before \(v\) is matched or the timeout \(t_v\) is reached. Finally, we denote \(\delta(u), \delta(e)\) to denote the edges incident to vertex \(u\) and edge \(e\) respectively.

**Benchmark Linear Program (LP):** As in related works, we use the following Linear Program as a benchmark for our competitive ratios.

\[
\text{maximize } \sum_{e \in E} w_e f_e p_e \quad (1.1)
\]

\[
\text{subject to } \sum_{e \in \delta(u)} f_e p_e \leq 1 \quad \forall u \in U \quad (1.2)
\]

\[
\sum_{e \in \delta(v)} f_e p_e \leq 1 \quad \forall v \in V \quad (1.3)
\]

\[
\sum_{e \in \delta(u)} f_e \leq t_u \quad \forall u \in U \quad (1.4)
\]

\[
\sum_{e \in \delta(v)} f_e \leq t_v \quad \forall v \in V \quad (1.5)
\]

\[
0 \leq f_e \leq 1 \quad \forall e \in E \quad (1.6)
\]

The variable \(f_e\) in the above LP refers to the expected number of probes on \(e\) in the offline optimal. For the problem with timeouts on only the online vertices, WLOG we can assume \(t_u = n, \forall u \in U\).

**Overview of Attenuation Framework:** We present a general online attenuation framework for the design and analysis of algorithms for the Online Stochastic Matching with Timeouts problem. In simple terms, an attenuation framework is a method for balancing the performance of all edges over all rounds to improve worst case analysis. We analyze the performance ratio of every edge across the \(n\) rounds and the competitive ratio for the algorithm is determined by the edge \(e = (u, v)\) with the lowest ratio. In all algorithms, it is common that the neighboring edges \(e' \in \delta(e)\) of \(e\) have much higher ratios. Moreover, performance of \(e\) is negatively affected by that of \(\delta(e)\). Thus we can improve the performance of \(e\) by attenuating those \(e' \in \delta(e)\) and in turn improve the overall competitive ratio.

In each online round when a vertex \(v\) arrives, we need to probe at most \(t_v\) neighbors sequentially until \(v\) is matched. This is essentially an offline stochastic matching problem studied in [7] on a star graph \(G(v)\), which consists of \(v\) and its safe neighbors. In our case, assume we have a black box, which is an LP-based algorithm solving an
offline stochastic matching problem with timeouts on a general star graph. Each online framework takes as an input a black box with a designated property and outputs an online algorithm. The final competitive ratio is jointly determined by the online framework itself and the input black box (see Theorems 3.2, 3.3, 3.4, 3.5).

The idea of edge-attenuation, first proposed in [1], aims to balance the performance of all edges. Suppose our black box guarantees each edge \( e \) will be probed with probability at least \( \alpha f_e \), where \( \alpha \) is some constant and \( f_e \) is the value assigned by an LP. Edge-attenuation will guarantee each edge is probed with probability equal to \( \alpha f_e \). If we know the black box probes \( e \) with probability \( \alpha' f_e > \alpha f_e \), we can achieve our goal by setting a probability \( 1 - \alpha/\alpha' \) with which we “pretend” to probe \( e \), but don’t actually probe it. Unfortunately, the exact value of \( \alpha' \) is hard to find since it is jointly determined by lots of inherent randomness from the algorithm and input itself. Therefore, we resort to simulation for a good estimation, e.g., simulating the algorithm on the input and input itself. Therefore, we resort to simulation to be probed. Since the probability of being matched in a previous round may be smaller in later rounds as the offline neighbors of \( v \) have been matched in earlier rounds. Intuitively, each edge in this smaller graph has less competition and therefore a greater chance to be probed. Since the probability of being matched in a previous round can differ greatly among offline vertices, we apply attenuation to the offline vertices to bound these probabilities in our analysis.

Our Contributions: One of our main contributions is a general framework for solving the Online Stochastic Matching with Timeouts problem which gives rise to a class of algorithms. Notably, we decouple the offline subproblem of which edges to probe when a vertex arrives from the online problem of handling a series of arrivals in a balanced way.

The offline subproblem addressed in Section 2 takes as input a stochastic star graph and an LP solution on that graph. The output is a probing strategy which preserves the LP values up to some factor in expectation. Our framework allows the offline subproblem to be solved by any black box algorithm provided it satisfies one of three basic properties described in Section 3. Thus, any new algorithm for this subproblem can easily be plugged into our overall analysis.

In Section 2, we suggest one such new offline algorithm which shows a novel way to perform the rounding procedure of Gandhi et al. [18] on stochastic star graphs. For clarity, we will denote the rounding procedure of Gandhi et al. with the acronym GKPS. We believe this technique is of independent interest as it appears to be the best known algorithm for solving the offline stochastic matching with timeouts problem on a star graph. We achieve an approximation ratio of 0.56 for this problem while the previous best achieved by [1] is 0.5 and no algorithm using LP (1.1) can perform better than 0.632 (See Section 4). This work may also give insight into better rounding schemes for more general classes of stochastic graphs.

For the online framework (Section 3), we bring tighter, cleaner analysis to the edge-attenuation approach in [1] and generalize it to work with a broad class of algorithms for the offline subproblem. For example, [1] uses only the first arrival of each online vertex type, discarding subsequent arrivals of the same type while we use every arrival. We then present a new vertex-attenuation approach which can be combined with edge-attenuation to achieve further improvement in the competitive ratio. The previous algorithm of [1] gave a ratio of 0.24, while we almost double this to get 0.46.

Looking forward, we introduce the more general Online Stochastic Matching with Two-sided Timeouts problem in Section 3.2. This new problem is well-motivated from applications and theoretically interesting. Using our framework, we give a constant factor 0.31-competitive algorithm for this problem.

Finally, we show in Section 4 that no algorithm using the LP in this paper and prior work can achieve a ratio better than \( 1 - 1/e \approx 0.632 \).

2. OFFLINE BLACK BOX

The online process consists of \( n \) offline rounds. In each round, we have an offline stochastic matching instance as studied in [7, 1] on a star graph. Consider a single round at time \( t \) : Let \( e \) be the arriving vertex and \( G(v) \) be the star graph of \( v \) and its safe neighbors. For ease of notation, we overload \( G(v) \) to denote both the set of edges and the set of neighbors of \( v \). Consider the following polytope:

\[
\left\{ \sum_{e \in G(v)} f_e p_e \leq 1, \sum_{e \in G(v)} f_e \leq t_v, 0 \leq f_e \leq 1, \forall e \in G(v) \right\}
\]  
(2.1)

An offline black box is any algorithm that transfers any feasible solution to the polytope (2.1) to a feasible probing strategy on \( G(v) \) with a guaranteed performance for each edge. By a feasible probing strategy, we mean one that does not violate the matching and patience constraints on \( v \).

Now, we present two concrete examples of black boxes. Let \( g = \{ f_e | e \in G(v) \} \) be any given feasible solution to the polytope (2.1). We will use \( g_e \) to refer to the value of \( g \) for edge \( e \).

Uniform Random Black Box: The Uniform Random Black Box, denoted by \( \text{BB}_{UR} \), is a direct application of the algorithm in [7] to the star graph \( G(v) \). To be consistent with the notation in [7], we use GKPS to denote the dependent rounding techniques developed in Gandhi et al. [18].

Algorithm 1: \( \text{BB}_{UR} \)

1. Apply GKPS [18] to \( G(v) \) where edge \( e \) is associated with a value \( g_e \). Let \( \hat{G}(v) \) be the set of edges that gets rounded;
2. Choose a random permutation \( \pi \) over \( \hat{G}(v) \). Probe each edge \( e \in \hat{G}(v) \) in the order \( \pi \) until \( v \) is matched.

The performance of \( \text{BB}_{UR} \) is presented in lemma 2.1. For each given \( e \in G(v) \), let \( R_{v,e} = \sum_{e \neq e'} g_{e'} p_{e'} \).

Lemma 2.1. In \( \text{BB}_{UR} \), each edge \( e \) will be safely probed with probability at most \( g_e \) and at least \( \left( 1 - \frac{R_{v,e}}{2} \right) g_e \).

Proof. Each edge \( e \) can be probed only when \( e \) appears in \( \hat{G}(v) \), which occurs with probability \( g_e \) according to GKPS [18]. Therefore, we have that each edge \( e \) will be probed with probability at most \( g_e \).

Consider an edge \( e \). The random permutation order \( \pi \) can be viewed as: each \( e' \) uniformly draws a real number \( a_{e'} \) from \([0, 1]\) and we sort all \( \{a_{e'}\} \) in increasing order. Condition on \( a_x = x \in [0, 1] \) and for each \( e' \neq e \), let \( X_{e'} \) be the indicator random variable that is 1 if \( a_{e'} < x \) (i.e., \( e' \) falls before \( e \) in \( \pi \)). For each \( e' \), let \( Y_{e'} \) and \( Z_{e'} \) be the respective indicator random variables corresponding to the event \( e' \) being rounded and if \( e' \) is present when probed. Let \( 1_e \) be
the indicator for the event \( e \) is probed. Therefore we have
\[
\Pr[e \text{ is probed} | a_e = x, Y_e = 1] = \mathbb{E}[e | a_e = x, Y_e = 1] \geq \mathbb{E}[1 - \sum_{e \in \mathcal{E}} x_e Y_e | a_e = x, Y_e = 1] \\
\geq 1 - x \sum_{e \in \mathcal{E}} g_e p_e = 1 - x R_{e,g}
\]
This we have
\[
\Pr[e \text{ is probed}] = g_e \int_0^1 \Pr[e \text{ is probed} | a_e = x, Y_e = 1] dx \\
\geq g_e (1 - R_{e,g}/2)
\]

**Sorted Dependent Rounding Block Box:** This black box, denoted by \( \text{BB}_{SDR} \), introduces a new way to perform GKPS \([18]\) rounding on stochastic star graphs. This rounding is combined with the probing strategy from \([1]\) to achieve improved results for the offline stochastic matching problem on star graphs.

Algorithms for matching problems on both stochastic and non-stochastic graphs have relied heavily on GKPS as a subroutine \([7, 1]\). In GKPS, the edges rounded in each iteration are chosen arbitrarily. Indeed, part of the beauty is that we are allowed to choose edges arbitrarily. However, for matching problems on stochastic graphs, we show that taking edge probabilities into account improves yields improved results. While this work is limited to the star graph structure, we believe the ideas can be extended to other graph structures in the future. The key insight is that each iteration of GKPS \([18]\) rounding should choose two fractional edges with similar \( p_e \) values since one edge will be rounded up and the other rounded down. Intuitively, this helps approximately preserve the constraint \( \sum_{e \in \mathcal{E}} \delta_e |E_e| \leq 1 \). We achieve this by rounding the two fractional edges with the largest \( p_e \) values in each iteration and modifying the input \( g_e \) values to balance the performance of edges with differing \( p_e \) values.

To optimize our analysis, we partition the edges into three disjoint sets based on their \( p_e \) values: large, medium, and small. We denote these sets \( E_L, E_M, E_S \), respectively. The sets are separated by two thresholds \( T_L \) and \( T_S \) such that large edges have \( p_e > T_L \), medium edges have \( T_L \leq p_e < T_S \), and small edges have \( p_e < T_S \). When we optimize these thresholds for our analysis of all cases, we set \( T_L = 2/3 \) and \( T_S = 1/4 \). Our approach is summarized in Algorithm 2.

**Algorithm 2: \( \text{BB}_{SDR} \)**

1. Sort the edges of \( G(v) \) in descending order of their \( p_e \) values.
2. Compute \( \Gamma \), the expected \( p_e \) value of the first edge rounded to 1 if GKPS were applied to the edges in sorted order (i.e., always rounding the two remaining edges with the largest \( p_e \) values).
3. Modification step: If \( T_S \leq \Gamma \leq T_L \), apply no modification. Otherwise, if \( \Gamma < T_S \), set \( \psi = 1.15 \), let edges in \( E_L \) get \( g_e = \psi g_e \), and edges in \( E_S \) get \( g_e = \left( \frac{\psi - 1}{\psi - 1} \right) g_e \). Finally, if \( \Gamma > T_L \), let \( Q = \sum_{e \in E_L} g_e \) before modification and let edges in \( E_L \) get \( g_e = g_e / Q \).
4. Apply GKPS \([18]\) to the edges of \( G(v) \) in sorted order, where edge \( e \) is associated with a value \( g_e \). Let \( \hat{G}(v) \) be the set of edges that gets rounded.
5. Choose a random permutation \( \pi \) over \( \hat{G}(v) \) according to the distribution in \([1]\). Probe each edge \( e \in \hat{G}(v) \) in the order \( \pi \) until \( v \) is matched.

The performance of \( \text{BB}_{SDR} \) is presented in Lemma 2.2.

**Lemma 2.2.** In \( \text{BB}_{SDR} \), each edge \( e \) will be safely probed with probability at least \( 0.56 g_e \).

Proof. Throughout this section, we will say that an edge \( e \) has been chosen by our rounding scheme if it’s LP value \( g_e \) has been rounded to 1 and not chosen if \( g_e \) has been rounded to 0. Let \( \hat{E} \) be the set of chosen edges after rounding a solution to LP \((1.1)\) on a stochastic star graph with a timeout \( t_v \) on the center vertex \( v \). Similarly, let \( \hat{\delta}(e) \) be the subset of edges in \( \hat{E} \) which share an endpoint with \( e \). Let \( \Delta_e = \mathbb{E}[g_e | \hat{\delta}(e) \cup \{ g_e \}] = 1 \). This is the expected sum of the \( p_f \) values of all the edges in the neighborhood of an edge \( e \) that have been chosen, conditioned on the event that \( e \) has been chosen. We assume the center \( v \) has a timeout \( t_v \geq 2 \) since \( t_v = 1 \) is trivial.

Claim 2.3 restates a claim made as an intermediary step of the proof of Lemma 2 in \([1]\) using our \( \Delta_e \) notation.

**Claim 2.3.** For every edge \( e \) it holds that
\[
\Pr[e \text{ is safe} | e \in \hat{E}] \geq \frac{1}{\Delta_e + p_e} \left( 1 - \exp \left( \left( \frac{-\Delta_e + p_e}{p_e} \right) \ln \frac{1}{1 - p_e} \right) \right)
\]

**Proof.** A detailed proof can be found in \([1]\).

Note that when applying GKPS to a star graph, we can store the edges in a sorted list and use Algorithm 3. This algorithm gives all of the same guarantees as GKPS since we are allowed to select two edges arbitrarily in every iteration of rounding a star graph.

**Algorithm 3: \( \text{GKPS}_{\text{star}} \)**

1. Sort the edges in descending order of \( p_e \) values.
2. Select the first two edges from the list. In other words, select the two remaining fractional edges with the largest \( p_e \) values.
3. Round one upward and the other downward according to the rules of GKPS and remove any edge that has been rounded to 0 or 1 from the list.
4. Repeat until the list is empty and all edges have been rounded.

Let \( e_1 \) be the first fractional edge which is chosen and let \( P_1 \) be a random variable which is equal to \( p_{e_1} \). Let \( \Gamma = \mathbb{E}[P_1] \) and let \( E_1 \) be a prefix of the initial sorted list of fractional edges which satisfies the following property: At the end of the iteration in which \( e_1 \) is chosen, all edges in \( E_1 \) must have been rounded to 0 or 1. Note that \( e_1 \) will be the only edge in \( E_1 \) which is rounded to 1. the rest will be rounded to 0. We can now use \( \Gamma \) to tighten the bounds for edges in both \( E_1 \) and \( E \setminus E_1 \).

**Claim 2.4.** Algorithm 3 followed by the AGM probing order guarantees the following for edge \( e_1 \).

\[
\Pr[e_1 \text{ is safe} | e_1 \in \hat{E} \cup E_1] \geq \frac{1}{2 - \Gamma}
\]

**Proof.** Note that \( e_1 \in \hat{E} \cup E_1 \) by definition and immediately after \( e_1 \) is chosen, we have \( \sum_{e \in E \setminus E_1} g_e p_e \leq 1 - \Gamma \). Therefore, \( \Delta_{e_1} \leq 1 - \Gamma \). Then, we have
\[
\Pr[e_1 \text{ is safe}] \geq \frac{1}{1 + \Gamma} \left( 1 - \exp \left( \left( \frac{1 - \Gamma + p_{e_1}}{p_{e_1}} \right) \ln \frac{1}{1 - p_{e_1}} \right) \right) \\
\geq \frac{1}{1 + \Gamma} \quad \text{\because worst case } p_{e_1} = 1
\]

**Claim 2.5.** Algorithm 3 followed by the AGM probing order guarantees the following for edge \( e \in \hat{E} \setminus E_1 \).

\[
\Pr[e \text{ is safe} | e \in \hat{E} \setminus E_1] \geq \frac{1}{1 + \frac{\Gamma}{1 - \Gamma}} \left( 1 - \exp \left( \left( \frac{1 + \Gamma}{\Gamma} \right) \ln \frac{1}{1 - \Gamma} \right) \right)
\]
Proof. We will upper bound both the $\Delta_e$ and $p_e$ values appearing in the function from Claim 2.3. First, $\Delta_e \leq 1$, due to the LP matching constraint 1.3 and the properties of GKPS. Second, $p_e \leq \Gamma$ for all $e \in E \setminus E_1$ since our list of edges was sorted by $p_e$ values. Then, the result follows by applying Claim 2.3. □

The modification step of Algorithm 2 defines three cases.

Case 1: $T_S \leq \Gamma \leq T_L$

We consider the edge sets $\hat{E} \cap E_1$ (containing the single edge $e_1$) and $\hat{E} \setminus E_1$. In the former, $\Gamma \geq T_S = 1/4$ and Claim 2.4 gives

$$Pr[e_1 \text{ is safe } | e_1 \in \hat{E} \cap E_1] \geq \frac{1}{2 - \Gamma} \geq \frac{1}{2 - T_S} \geq 0.56$$

In the latter, $p_e \leq \Gamma \leq T_L = 2/3$ and Claim 2.5 gives

$$Pr[e \text{ is safe } | e \in \hat{E} \setminus E_1] \geq \frac{1}{1 + T_L} \left(1 - \exp\left(-\frac{(1+T_L)}{T_L} \ln \frac{1}{1-T_L}\right)\right) \geq 0.56$$

Case 2: $\Gamma < T_S$

In this case, we have an modification factor $\psi = 1.15$, which we use to “help” $E_1$ while not “hurting” $E_S$ too much. In the modification step, all edges in $E_1 \setminus E \setminus ge = \psi g_e$ and all edges in $E_S$ get $g_e = \left(\frac{1 - 0.5\psi}{1 - \psi}\right) g_e$. After modification, we must make sure the patience constraint is not violated and $\Delta_e$ remains small. Note that $\Gamma \leq T_S = 1/4$ implies $\sum_{e \in E_1} s_e \psi e \leq 1/4$ and therefore $\sum_{e \in E_1} s_e \leq 3/8$ since $p_e > T_L = 2/3$ for $e \in E_1$. Similarly, $\sum_{e \in E_S} s_e \geq t_v - 1$ (assuming WLOG that the LP patience constraint was tight). It follows that

$$\sum_{e \in E} s_e = \sum_{e \in E_1} s_e + \sum_{e \in E_S} s_e + \sum_{e \in E_M} s_e \leq \frac{3\psi}{8} + \frac{3}{8} + \left(\frac{1 - 0.5\psi}{1 - \psi}\right) (t_v - 1) = t_v$$

Also, note that for all $e \in E$, $\Delta_e \leq 3/4 + \psi/4$ which is again due to the fact that $\sum_{e \in E_1} s_e \psi e < T_S < 1/4$. Finally, it is important that for $e \in E\setminus E \setminus ge \leq 1$ ensuring that $ge$ will only be rounded to 0 or 1. From above, we know $g_e \leq 3/8$ and therefore $\psi g_e \leq 1$.

We can now bound the ratios for edges in $E_1$, $E_S$, and $E_M$.

$$Pr[e \text{ is safe } | e \in \hat{E} \cap E_1] \geq \psi \left(\frac{1}{1 + \frac{3\psi}{8} + p_e} \left(1 - \exp\left(-\frac{(1 + \frac{3\psi}{8} + p_e)}{p_e} \ln \frac{1}{1-p_e}\right)\right)\right) \geq 0.56 \quad \text{worst case } p_e, 1$$

For $E_S$, observe $\left(\frac{1 - 0.5\psi}{1 - \psi}\right)$ is minimized for $t_v \geq 2$ at 0.943.

$$Pr[e \text{ is safe } | e \in \hat{E} \setminus E_1] \geq 0.943 \left(\frac{1}{1 + \frac{3\psi}{8} + p_e} \left(1 - \exp\left(-\frac{(1 + \frac{3\psi}{8} + p_e)}{p_e} \ln \frac{1}{1-p_e}\right)\right)\right) \geq 0.56$$

For an edge $e \in \hat{E} \setminus E_M$, we first show that $e$ must be in $E_1$ and therefore $\Delta_e \leq 1 - \Gamma \leq 1$. To do so, we prove the stronger claim that $E \setminus E_S \subseteq E_1$ by showing that before modification

$$\sum_{e \in E_1} s_e \leq \frac{1 - \sum_{e \in E_M} s_e}{\psi}$$

This ensures that $\sum_{e \in E_1} \psi g_e + \sum_{e \in E_M} s_e < 1$ and thus, $E \setminus E_S \subseteq E_1$. Before modification, $\Gamma < T_S = 1/4$. So we have

$$\sum_{e \in E_1} s_e \psi g_e + \sum_{e \in E_M} s_e < T_S - T_L \sum_{e \in E_M} s_e$$

Then, since $p_e \leq T_L = 2/3$, Claim 2.3 gives

$$Pr[e \text{ is safe } | e \in \hat{E} \cap E_M] \geq \frac{1}{1 + T_L} \left(1 - \exp\left(-\frac{(1 + T_L)}{T_L} \ln \frac{1}{1-T_L}\right)\right) \geq 0.56$$

Case 3: $\Gamma > T_L$

Here, we use a different modification strategy to ensure $E_L = E_1$ and thus only one edge from $E_L$ will be chosen. Let $Q = \sum_{e \in E_L} s_e$ before modification. In the modification step, all edges $e \in E_L$ get $g_e = \psi e / Q$. Then $\Delta_e \leq 1 - T_L Q$ since $\sum_{e \in E_L} s_e p_e$ before modification is at least $T_L Q$. Similar to Claim 2.4, we have

$$Pr[e \text{ is safe } | e \in \hat{E} \cap E_L] \geq \left(\frac{1}{Q} - \frac{1 - \psi}{1 - \psi} \left(1 - \exp\left(-\frac{(1 + T_L)}{T_L} \ln \frac{1}{1-T_L}\right)\right)\right) \geq 0.56$$

This concludes the proof of Lemma 2.2. □

3. ATTENUATION FRAMEWORK FOR ONLINE MATCHING WITH TIMEOUTS

The main idea of our attenuation framework is to decouple the offline and online subproblems. The offline problem is what to do with an arriving vertex once it has arrived and we must choose which edges to probe. This is handled by a black box offline algorithm. The black box is only restricted by one of the three properties listed below. The online problem is how to manage a series of arrivals and that is the primary focus of this section.

We will begin in Section 3 by defining some desirable properties of an offline black box and showing which of those properties are satisfied by the two black boxes proposed in this paper. Section 3.1 describes the main attenuation techniques applied during the online phase. Finally, Section 3.2 will introduce a more general model with timeouts on both online and offline vertices and show how to extend our results to this model.

Throughout this section, we assume that through simulations we can always get an accurate estimation of our target probabilities. As shown in [1, 24], we can manipulate the simulation error properly such that we lose at most an additive factor of $\epsilon$ in the final ratio.

The three properties of a black box: Property A states that the black box BB is guaranteed to probe each edge with probability at least $\alpha g_e$ for some constant $\alpha \in (0, 1)$. It gives a lower bound on the performance of each edge without any further restrictions on the black box. More formally:

Property A: For any feasible solution $\mathbf{g}$ to LP (2.1), BB outputs a feasible probing strategy $BB[\mathbf{g}]$ such that every edge $e$ will be probed with probability at least $\alpha g_e$ for some constant $\alpha \in (0, 1)$. 


For Properties B and C, recall that \( R_{e,G} = \sum_{e \in \mathcal{E}_{G}} g_e \mathbf{e}^{e} \) and this value expresses the amount of competition \( e \) will face from its neighbors. These two properties both add the restriction that the probability of probing a given edge will be a function of both \( g_e \) and \( R_{e,G} \). This allows us to take advantage of the fact that \( R_{e,G} \) may decrease as the number of arrivals increases. The conditions of non-increasing and convexity on the function \( R_{e,G} \) are required to ensure that the offline ratio of BB can be used to bound the overall competitive ratio. The condition of finitely bounded first derivative guarantees that the error accumulated from simulation is bounded.

**Property B:** For any feasible \( g \) to LP (2.1), BB outputs a feasible probing strategy \( \text{BB}[g] \) such that each edge \( e \) is probed with probability at least \( g_e \mathbf{e}^{e} \mathbf{R}_{\mathcal{E},G}[\mathcal{E},G] \), where \( \mathbf{R}_{\mathcal{E},G} \) is a non-increasing and convex function and has finitely bounded first derivative on \([0, 1]\).

Property C adds a further restriction that each edge is probed with probability at most \( g_e \).

**Property C:** For any feasible \( g \) to LP (2.1), BB outputs a feasible probing strategy \( \text{BB}[g] \) such that every edge \( e \) is probed with probability at most \( g_e \mathbf{e}^{e} \mathbf{R}_{\mathcal{E},G}[\mathcal{E},G] \), where \( \mathbf{R}_{\mathcal{E},G} \) is a non-increasing and convex function and has finitely bounded first derivative on \([0, 1]\).

**Observation 3.1.** For any BB satisfying Property B or Property C with \( \mathbf{R}_{\mathcal{E}} \), we have \( \mathbf{R}_{\mathcal{E}}[x] \leq \mathbf{R}_{\mathcal{E}}[0] \leq 1 \) for all \( x \in [0, 1] \).

The fact that \( \mathbf{R}_{\mathcal{E}}[0] \leq 1 \) can be seen from this example: consider the graph \( G(v) \) which has exactly one edge \( e = (u, v) \). Clearly, \( g_e = 1 \) is a feasible solution to LP (2.1). Then, \( \text{BB}[g] \) will probe \( e \) with probability at least \( g_e \mathbf{e}^{e} \mathbf{R}_{\mathcal{E},G}[\mathcal{E},G] \), implying \( \mathbf{R}_{\mathcal{E}}[0] \leq 1 \).

We note that our first black box \( \text{BB}_{U,R} \) satisfies all three properties. However, our second BB satisfies Property A.

### 3.1 Attenuation

Our black box properties give us lower bounds on the probability that an edge or vertex will be matched at any given time during the online phase. Attenuation allows us to make those bounds tight by reducing the performance of any edge or vertex which is exceeding the lower bound. The intuition is that weakening the over-performing edges will increase the performance of the lowest performing edges that provide the worst case competitive ratio.

We define three distinct attenuation frameworks: **edge attenuation** which requires an offline black box satisfying Property A, **vertex-attenuation** which requires a black box satisfying Property C, and **edge and vertex-attenuation** which requires an offline black box satisfying Property B. The edge-attenuation framework generalizes and clarifies the edge-attenuation approach of [1]. We also give an improved result due to tighter analysis and a more powerful black box. Vertex-attenuation is a novel approach introduced in this paper that upper bounds the probability that a vertex in \( U \) will be safe at time \( t \). This lets us to exploit the fact that the star graph \( G(v) \) will be smaller in later rounds leading to a higher probability of safely probing each of the remaining edges. It can be combined with edge-attenuation to get the best known result for this problem.

Let \( f = \{ f_e \mid e \in E \} \) be an optimal solution to the LP (1.1). Let \( v \) be the vertex arriving at time \( t \in [n] \) and \( G_t(v) \) be the star graph consisting of \( v \) and its safe neighbors. Throughout this section we assume \( f_{t,v} = \{ f_e \mid e \in G_t(v) \} \), which is a feasible solution to the LP (2.1) on \( G_t(v) \).

**Edge-attenuation:** The most basic form of attenuation we consider is edge attenuation. Suppose we are given a black box BB satisfying Property A and guaranteeing that each edge is probed with probability at least \( \alpha g_e \). This attenuation will guarantee that each offline subproblem, each edge is probed with probability equal to \( \alpha g_e \). From Property A, we know that \( \text{BB}[f_{t,v}] \) will probe each edge \( e \) with probability at least \( \alpha f_e \). In this framework, we maintain that each safe edge \( e \) is probed with probability exactly equal to \( \alpha f_e \) in all rounds via appropriate edge-attenuation. Algorithm 4 gives a formal description of the algorithm.

**Algorithm 4:** \( \text{ATTN}_t[\text{BB}] \)

1. For each \( t \in [n] \), let \( v \) be the vertex arriving at time \( t \) and \( G_t(v) \) be the graph consisting of \( v \) and its safe neighbors.
2. Let \( f_{t,v} = \{ f_e \mid e \in G_t(v) \} \) be the induced feasible solution to LP (2.1).
3. Apply \( \text{BB}[f_{t,v}] \) and simulation-based edge-attenuation (see Sec. 1.1) to \( G_t(v) \), such that each \( e \) is probed with probability exactly equal to \( \alpha f_e \).

**Theorem 3.2.** For any BB satisfying Property A, \( \text{ATTN}_t[\text{BB}] \) has an online competitive ratio of \( 1 - \epsilon^\alpha \).

**Proof.** Consider an edge \( e = (u,v) \) and let \( f_u = \sum_{e \in \mathcal{R}(u)} f_e \mathbf{e}^{e} \). From Algorithm 4, we know that during any round \( t \in [n] \), \( u \) will be matched with probability exactly equal to \( \alpha f_u / n \) (conditioned on \( u \) being safe at the beginning of round \( t \)). Therefore \( u \) will be safe at \( t \) with probability equal to \( 1 - (1 - \alpha f_u / n)^t \). Thus we have

\[
\Pr[e \text{ is probed}] = \sum_{t=1}^{n} \frac{1}{n} \alpha f_e \left( 1 - \frac{\alpha f_u}{n} \right)^{t-1} \\
\geq f_e \left( 1 - (1 - \frac{\alpha}{n})^t \right) > f_e \left( 1 - e^{-\alpha} \right)
\]

We claim that after incorporating the simulation error as shown in Section 1.1, we can get a ratio of \( 1 - (1 - \alpha/n)^t - \epsilon \) for any given \( \epsilon \). Thus by setting \( \epsilon = e^{-\alpha} = (1 - \alpha/n)^t = O(1/n) \), we get the result in theorem 3.2. \( \Box \)

Notice that \( \text{BB}_{U,R} \) and \( \text{BB}_{S,D,R} \) satisfy Property A with \( \alpha = 1/2 \) and \( \alpha = 0.56 \), respectively. Plugging those values into the above theorem, we get ratios of 0.3934 and 0.4287, respectively.

**Corollary 3.1.** When combined with \( \text{BB}_{U,R} \) and \( \text{BB}_{S,D,R} \), the first framework will yield an algorithm which achieves a competitive ratio of 0.3934 and 0.4287, respectively, for the Online Stochastic Matching with Timeouts problem.

Although this approach does not give our best result, we note that it places fewer restrictions on the black box than the other approaches presented in this paper. Thus, developing a stronger black box satisfying only Property A could lead to the edge-attenuation framework yielding the best result for this problem in the future.

**Vertex-attenuation:** Applying vertex-attenuation without any edge-attenuation requires an offline black box BB satisfying our most strict property, **Property C**. Here is the intuition behind vertex-attenuation. Notice that over time, the offline vertices in \( U \) will be matched and therefore removed from the graph. Suppose we apply \( \text{BB}[f_{t,v}] \) to \( G_t(v) \) on each round \( t \) when \( v \) arrives. Thus when \( t \) gets larger and \( G_t(v) \) gets smaller, \( R_{e,f_{t,v}} \) will decrease for each safe edge \( e = (u,v) \). This means the lower bound on probing an edge, \( f_e \mathbf{R}_{\mathcal{E},G_t(v)} \), will increase with time. We can think of previous approaches to this problem as using a weak bound of \( R_{e,f_{t,v}} \). Vertex-attenuation lets us take advantage of \( R_{e,f_{t,v}} \)
considering a specific round $t$ when $v$ comes. Let $S_{u,t}$ be the event that $u$ is safe at $t$ for each $u \in U$. We have the below Lemma 3.1.

**Lemma 3.1.** Suppose we apply BB[$f_{t-1}$] to $G_t(v)$ during each round $t$ when $v$ arrives. Then for each round $t \in [n]$, we have (1) $\Pr[S_{u,t}] \geq \Pr[S_{u,t-1}] \big(1 - \frac{1}{n}\big)$ and (2) $\Pr[S_{u,t} \cap S_{u',t-1}, S_{u,t-1}] \leq \Pr[S_{u,t}] \big(1 - \frac{1}{n}\big) \Pr[S_{u',t}]\big(1 - \frac{1}{n}\big)$.

**Proof.** First, we show the proof of inequality (1). Assume $u$ is safe at (the beginning of) $t-1$. Notice that in the round $t-1$, every edge $e \in \partial(u)$ will be matched with probability at most $f_{e,t-1}$. Therefore, we have that $u$ will be matched in round $t-1$ with probability $1 - F_{u,t} \geq 1 - \sum_{e \in \partial(u)} f_{e,t-1}$. Thus, we have $\Pr[S_{u,t}] \geq \Pr[S_{u,t-1}] \big(1 - \frac{1}{n}\big)$. Now we show the proof of inequality (2). Assume both $u$ and $u'$ are safe at time $t-1$. Consider the round $t-1$ and assume each edge $e$ is probed with probability $\alpha_{e,t}$ with $\alpha_e \in [0,1]$. Notice that $\Pr[S_{u,t} \cap S_{u',t-1}, S_{u',t-1}] = 1 - \sum_{e \in \partial(u'), \partial(e')} f_{e,t-1}$ and $\Pr[S_{u,t}] \big(1 - \frac{1}{n}\big) \Pr[S_{u',t}] \big(1 - \frac{1}{n}\big)$ is safe at $t-1$. Therefore, we get the inequality (2), since $\partial(u)$ and $\partial(u')$ are disjoint. □

Lemma 3.1 implies that adding vertex-attenuation independently to every offline $u$ at the start of each round $t \in [n]$ ensures every $u$ is safe at time $t$ with probability equal to $(1 - \frac{1}{n})^{t-1}$. Additionally, for two vertices $u$ and $u'$, this event is negatively correlated. Algorithm 3 describes this online framework, denoted ATTN.

**Algorithm 3:** ATTN$_{\alpha}$

1. For each $t \in [n]$, let $v$ be the vertex arriving at time $t$ and $G_t(v)$ be the star graph consisting of $v$ and its safe neighbors.
2. Compute and add attenuation factors to each offline $u$ such that each $u$ is safe at time $t$ with probability equal to $(1 - \frac{1}{n})^{t-1}$.
3. Let $f_{t,v} = \{f_e \in G_t(v)\}$ be the induced feasible solution to LP (2.1) and BB[$f_{t,v}$] be the feasible probing strategy of BB.
4. Apply BB[$f_{t,v}$] to $G_t(v)$.

**Theorem 3.3.** For any BB satisfying Property C with function $\text{BB}_t$, ATTN$_{\alpha}$ has a competitive ratio of $\int_0^1 e^{-x} \text{BB}_t[e^{-x}]dx - e$.

**Proof.** Consider a single edge $e = (u,v)$. Let $A_{e,t}$ be the event that $e$ is effectively probed during round $t$, i.e., $v$ comes at $t$, $u$ is safe, and $e$ is probed. Let $S_{u,t}$ be the event that $u$ is safe at the beginning of $t$, i.e., $G_t(v)$. From ATTN$_{\alpha}$, we have that $\Pr[S_{u,t}] = (1 - \frac{1}{n})^{t-1}$. Then, from Property C, we have $\Pr[A_{e,t}] \geq (f_e/n) \Pr[S_{u,t}] \Pr[R_{BB}([f_e,t,v])] \geq (f_e/n)(1 - \frac{1}{n})^{t-1} \Pr[R_{BB}([f_e,t,v])] \geq (f_e/n)(1 - \frac{1}{n})^{t-1} h(t - 1/n)$ and (The right-most equality below is obtained by letting $n \rightarrow \infty$), $\Pr[e \text{ is probed}] \geq \sum_{t=1}^n \Pr[A_{e,t}] \geq \sum_{t=1}^n \frac{f_e}{n}(1 - \frac{1}{n})^{t-1} \Pr[R_{BB}([f_e,t,v])] \geq \int_0^1 e^{-x} \text{BB}_t[e^{-x}]dx$. Incorporating simulation errors (see Sec. 1.1), we get an online ratio of $\int_0^1 e^{-x} \text{BB}_t[e^{-x}]dx - e$ for any given $e > 0$. □}

Plugging the BB$_t$ function $R_{BB}([x]) = 1 - x/2$ into the above formula, we get a ratio of 0.4159.

**Corollary 3.2.** The second framework combined with BB$_t$ yields an algorithm, which achieves a competitive ratio of 0.4159 for the Online Stochastic Matching with Timeout problem.

**Edge and Vertex-attenuation Combined:** Our final and currently most powerful framework combines both edge and vertex-attenuation. Notice that by design, edge-attenuation upper bounds the probability an edge will be probed in an offline step. Therefore, our black box only needs to satisfy Property B which is slightly less restrictive than Property C.

At the start of each round $t$, let every $u$ be safe with a target probability equal to $\gamma_t \in [0,1]$. From Property B, we have that each safe edge $e = (u,v)$ is probed during round $t$ with probability at least $\alpha_{e,t}$, where $\alpha_t = \mathbb{E}[R_{BB}([f_{e,t,v}])] \geq R_{BB}([\gamma_t])$ (same analysis as Theorem 3.3). Using edge-attenuation, each safe edge is probed with probability equal to $\alpha_{e,t}/\gamma_t$. Consequently, each safe $u$ at time $t$ will remain safe at $t + 1$ with probability at least $1 - R_{BB}([\gamma_t])$. Through vertex-attenuation, each $u$ remains safe at $t + 1$ with probability equal to $\gamma_{t+1} = \gamma_t(1 - R_{BB}([\gamma_t])$. Thus, we ensure each edge is probed with a uniformly increasing ratio and every offline node is safe with a uniformly decreasing probability. Algorithm 6 describes this online framework denoted ATTN$_{\alpha}$.

**Algorithm 6:** ATTN$_{\alpha}$

1. For time steps $1, 2, \ldots, t$ do
2. Let each $u$ be safe with probability equal to $\gamma_t$.
3. Let $v$ arrive at time $t$ and $G_t(v)$ be the graph of $v$ and its safe neighbors. Let $f_{t,v}$ be the feasible solution to LP (2.1).
4. Apply BB[$f_{t,v}$] and edge-attenuation to $G_t(v)$ such that each edge $e$ is probed with probability equal to $\alpha_{e,t}/\gamma_t$, where $\alpha_t = \mathbb{E}[R_{BB}([\gamma_t])$.
5. Apply vertex-attenuation to each $u$ such that each $u$ is safe at time $t + 1$ with probability equal to $\gamma_{t+1} = \gamma_t(1 - \alpha_t/\gamma_t)$.

We can express a recurrence relation for $(\gamma_t, \alpha_t)$ as follows.

$$\gamma_1 = 1, \quad \alpha_t = R_{BB}([\gamma_t]), \quad \gamma_{t+1} = \gamma_t \left(1 - \frac{\alpha_t}{\gamma_t}\right) \tag{3.1}$$

**Theorem 3.4.** For any BB satisfying Property B, ATTN$_{\alpha}$ has an online competitive ratio of $(1 - h(1))/e$, where $h$ is the unique function satisfying $h' = -hR_{BB}([h])$ with boundary condition $h(0) = 1$. Here, $h'$ represents the first-order derivative of function $h$.

**Proof.** Consider an edge $e = (u,v)$. It will be probed with probability equal to $f_e \sum_{t=1}^n \frac{\alpha_t}{\gamma_t}$. From Observation 3.1, $R_{BB}[x] \in [0,1]$ for all $x \in [0,1]$. From Property B and Equation (3.1), we know that $(\alpha_t)$ is an increasing sequence and $(\gamma_t)$ is a decreasing sequence with $\gamma_t \geq e/\alpha_t$ and $\alpha_t \leq R_{BB}([\gamma_t])$ for all $t$.

Define a function $h : [0,1] \rightarrow [0,1]$ such that $h((t-1)/n) = \gamma_t$ for all $t \in [n]$. Then we have $h(0) = 1$. Equation (3.1) implies that

$$\frac{h(t/n) - h((t-1)/n)}{n} = -h((t-1)/n)$$

Letting $x = (t-1)/n$ and the above equation yields $h(x+1/n) - h(x) = -h(x)\frac{n}{h(x)}$. Now letting $n \rightarrow \infty$, we can see that $h$ satisfies the differential equation $h' = -hR_{BB}([h])$ with boundary condition $h(0) = 1$.

Given $h$, we have

$$\sum_{t=1}^n \frac{\alpha_t}{\gamma_t} \frac{n}{h} = \frac{1}{n} \sum_{t=1}^n h((t-1)/n)R_{BB}([h((t-1)/n)])$$

$$= \int_0^1 h(x)R_{BB}([h(x)])dx = h(0) - h(1) = 1 - h(1)$$


Simulation error subtracts at most $O(\varepsilon)$ in the final ratio (see Sec. 1.1). Hence, this completes the proof of the theorem. \hfill \square

$\text{BB}_{UR}$ satisfies Property B with $\text{R}_{\text{BB}_{UR}}[x] = 1 - x/2$. Plugging $\text{R}_{\text{BB}_{UR}}$ into the above theorem, we get $h(x) = 2/(1 + e^x)$, which implies ATTN[\text{BB}_{UR}] has an online ratio of $1 - h(1) - e \geq 0.4621$.

**Corollary 3.3.** The third framework combined with $\text{BB}_{UR}$ yields an algorithm, which achieves a competitive ratio of 0.4621 for the Online Stochastic Matching with Timeout problem.

### 3.2 Extensions to a More General Model

The online attenuation framework combined with an offline black box can be extended to more general models. In this section, we give an example by showing how the first attenuation framework together with an offline black box satisfying Property A can be used for the generalization of Stochastic Matching with timeouts on both offline and online vertices. We do believe the other two frameworks can be used to attack the generalized model as well.

In this model, in addition to our previous setting each offline vertex $u$ has a timeout constraint of $t_u$, i.e., each $u$ can be probed at most $t_u$ times over the $n$ rounds. Hence, the constraint 1.4 in LP (1.1) is a valid constraint in the benchmark.

**Theorem 3.5.** For any $\text{BB}$ satisfying Property A with $\alpha$, ATTN$_1[\text{BB}]$ has an online competitive ratio of $ae^{-\alpha}$ for the Online Stochastic Matching with Two-sided Timeouts problem.

Recall that $S_{u,t}$ is the probability that $u$ is safe at time $t$. In this new setting, $u$ is safe if $u$ is not matched and the timeout of $u$ has not been exhausted. The lemma 3.2 gives a lower bound on $\Pr[S_{u,t}]$ when we apply ATTN$_1[\text{BB}]$ using any BB satisfying Property A with $\alpha$.

**Lemma 3.2.**

$$
\Pr[S_{u,t}] \geq \left(1 - \frac{\alpha}{n}\right)^{t-1} \left(1 - \frac{\alpha(t-1)}{n}\right)
$$

**Proof.** Consider a node $u$. For each $e \in \partial(u)$ and each $t' \in [n]$, let $X_{e,t'}$ be the indicator for the event: $e$ comes at $t'$; $Y_{e,t'}$ be the indicator for the event: $e$ is probed at $t$; $Z_{e,t'}$ be the indicator for the event: $e$ is present when probed. Notice that $X_{e,t'} = Y_{e,t'} = Z_{e,t'}$ for all $e \in \partial(u)$, $t' \in [n]$.

Let $S_{u,t}^1$ be the event that $u$ is not matched at time $t$ and $S_{u,t}^2$ be the event that $u$ is probed at most $t_u$ at the beginning of time $t$. Define $A_1$ to be the event that $\sum_{t'=1}^{\alpha} \sum_{e \in \partial(u)} X_{e,t'} Y_{e,t'} Z_{e,t'} = 0$ and $A_2$ to be the event that $\sum_{t'=1}^{\alpha} \sum_{e \in \partial(u)} X_{e,t'} Y_{e,t'} Z_{e,t'} \leq t_u = 1$.

Observe that $\Pr[S_{u,t}^1] = 1 - \Pr[A_1 \wedge A_2]$. Let us now lower bound the value of $\Pr[A_1 \wedge A_2]$.

Recall that $F_u = \sum_{e \in \partial(u)} f_{e} p_e$. For each given $t' < t$, we know that $\Pr[\sum_{e \in \partial(u)} X_{e,t'} Y_{e,t'} Z_{e,t'} = 0] = 1 - \alpha F_u/n \geq 1 - \alpha/n$. Therefore we have $\Pr[A_1] \geq (1 - \alpha/n)^{t-1}$. Notice that for each given $t' < t$,

Thus we get $\exists \sum_{t'=1}^{\alpha} \sum_{e \in \partial(u)} X_{e,t'} Y_{e,t'} |A_1| \leq \frac{\alpha f_u}{n} (t-1)$, which implies $\Pr[A_2|A_1] \geq (1 - \frac{\alpha}{n}) (t-1)$. Therefore $\Pr[S_{u,t}] = \Pr[S_{u,t}^1 \wedge S_{u,t}^2] \geq \Pr[A_1 \wedge A_2] \geq (1 - \frac{\alpha}{n}) (t-1)$. Let us now prove Theorem 3.5.

**Proof.** The proof is very similar to that of Theorem 3.2. Consider a single edge $e = (u, v)$, we have

$$
\Pr[e \text{ is probed}] \geq \sum_{t'=1}^{\alpha} \frac{1}{n} \alpha f_e \left(1 - \frac{\alpha}{n}\right)^{t-1} \left(1 - \frac{\alpha(t-1)}{n}\right) \geq f_e e^{-\alpha}
$$

Plugging $\text{BB}_{SDR}$ with $\alpha = 0.56$, we get a ratio of 0.3198 for this generalized model.

**Corollary 3.4.** The first framework combined with $\text{BB}_{SDR}$ yields an algorithm, which achieves a competitive ratio of 0.3198 for the Online Stochastic Matching with Two-sided Timeouts problem.

### 4. LOWER BOUND TO BENCHMARK LP

Here, we present an unconditional lower bound for this LP due to the stochasticity of the problem. We call this lower bound a stochasticity gap, similar to the concept of an integrality gap.

Consider a complete bipartite graph with $|U| = |V| = n$. Let the edge probabilities $p_e$ for all edges be $1/n$ and the rewards $w_e$ be 1. Let the patience values for all vertices be $n$. Notice that assigning $f_e = 1$ for every edge is a feasible solution to LP 1.1. Hence, the optimal LP value is at least $n$. However, we will show that any online algorithm cannot perform better than $(1 - 1/\epsilon)n$. Therefore, the stochasticity gap for this LP is at least $(1 - 1/\epsilon)n \approx 0.63$.

Consider any vertex $u \in U$. We have the following:

$$
\Pr[u \text{ is matched}] = 1 - \Pr[A_1^n u \text{ is not matched at } t] = 1 - \Pi_{t=1}^{n} \left(1 - \frac{1}{n} \right)^{t-1} \left(1 - \frac{1}{n} \right) \leq 1 - \Pi_{t=1}^{n} \left(1 - \frac{1}{n} \right) \leq 1 - \frac{1}{\epsilon} \approx 0.63
$$

Step 4.2 is due to independence. Step 4.3 uses union bound, $p_e = 1/n$, and probability $1/n$ for each $v$ to be drawn. Using union bound over all $u \in U$, no algorithm can do better than $(1 - 1/\epsilon)n$.

### 5. CONCLUSION/FUTURE DIRECTIONS

We gave a general framework for the Online Stochastic Matching with Timeout problem and its extension. This led to improved competitive ratios for the former and first constant factor ratio for the latter. More importantly, the frameworks are general enough to obtain further improvements by simply finding a better black box for the offline problem on star graphs. One future direction is to increase the competitive ratio by designing better black boxes. In particular, can we obtain a ratio of at least $0.5$? For example, an adaptation of the second black box, $\text{BB}_{SDR}$ from Section 2, to meet Property B from Section 3 would likely accomplish this goal. Another future direction is to design similar framework(s) for the various other online stochastic matching problems, such as $b$-matching. We believe these frameworks have the potential to give a unified framework for many of the stochastic matching problems.
REFERENCES


