

# Prophet Secretary and Matching: the Significance of the Largest Item

Ziyun Chen<sup>\*</sup>      Zhiyi Huang<sup>†</sup>      Dongchen Li<sup>‡</sup>      Zhihao Gavin Tang<sup>§</sup>

## Abstract

The prophet secretary problem is a combination of the prophet inequality and the secretary problem, where elements are drawn from known independent distributions and arrive in uniformly random order. In this work, we design 1) a 0.688-competitive algorithm, that breaks the 0.675 barrier of blind strategies (Correa, Saona, Ziliotto, 2021), and 2) a 0.641-competitive algorithm for the prophet secretary matching problem, that breaks the  $1 - 1/e \approx 0.632$  barrier for the first time. Our second result also applies to the query-commit model of weighted stochastic matching and improves the state-of-the-art ratio (Derakhshan and Farhadi, 2023).

## 1 Introduction

The study of prophet inequality dates back to the 1970s [31, 32] from optimal stopping theory. Consider  $n$  items with independent random values arriving one by one in an adversarial order. The value distribution  $F_i$  of each item  $i$  is known upfront to the algorithm, but the realization of value  $v_i \sim F_i$  is only revealed on the item's arrival. After seeing the item's identity  $i$  and value  $v_i$ , the algorithm decides immediately whether to accept the item and collect its value; the algorithm can accept at most one item in this problem. The goal is to maximize the expected value of the accepted item and compete against the prophet, i.e., the expected maximum value  $\mathbf{E}[\max_i v_i]$ . It is known that the optimal competitive ratio is  $\frac{1}{2}$  for this problem.

A fundamental extension of the prophet inequality is prophet matching. Consider an underlying bipartite graph with edge weights drawn from known distributions. The vertices on one side are known upfront and those on the other side arrive online. On the arrival of an online vertex  $v_i$ , the weights of its incident edges are revealed and the algorithm decides whether to match  $v_i$  and to which offline vertex. The classic prophet inequality is captured by this model with one offline vertex. Feldman, Gravin, and Lucier [19] gave a tight  $\frac{1}{2}$  competitive algorithm for the matching setting, and their result was further generalized to settings when all vertices arrive online [17].

In this work, we consider the secretary variants (a.k.a. the random order variants) of prophet inequality and prophet matching, i.e., the setting where the arrival order of items (resp. vertices) is uniformly at random.

The study of prophet secretary was initiated by Esfandiari et al. [16], who designed a  $1 - 1/e \approx 0.632$  competitive algorithm and provided an upper bound of 0.75.<sup>1</sup> Since then, a sequence of follow-up works [3, 7, 10, 23, 25] have focused on closing the gap. The state-of-the-art lower and upper bounds are 0.672 by Harb [25] and 0.723 by Giambartolomei et al. [23] respectively.

Less progress has been made on the prophet secretary matching problem. Ehsani et al. [15] gave a  $1 - 1/e$  competitive algorithm. Very recently, the  $1 - 1/e$  barrier was surpassed in two special cases: 1) the i.i.d. setting studied by Yan [40] and Qiu et al. [37]; and 2) the query-commit setting studied by Derakhshan and Farhadi [12]. Beating  $1 - 1/e$  for the general case of prophet secretary matching remains one of the most intriguing open questions to the online algorithms community.

<sup>\*</sup>Paul G. Allen School of Computer Science & Engineering, University of Washington. [ziyuncc@cs.washington.edu](mailto:ziyuncc@cs.washington.edu)

<sup>†</sup>School of Computing and Data Science, The University of Hong Kong. [zhiyi@cs.hku.hk](mailto:zhiyi@cs.hku.hk). This work was supported by an NSFC grant (No. 6212290003).

<sup>‡</sup>School of Computing and Data Science, The University of Hong Kong. [dongchen.li@connect.hku.hk](mailto:dongchen.li@connect.hku.hk)

<sup>§</sup>ITCS, Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai Institute of International Finance and Economics, Shanghai University of Finance and Economics. [tang.zhihao@mail.shufe.edu.cn](mailto:tang.zhihao@mail.shufe.edu.cn)

<sup>1</sup>The competitive ratio of an algorithm is a number between  $[0, 1]$ . A lower bound corresponds to an algorithm and an upper bound corresponds to an impossibility result.

## 1.1 Our Contributions

**Result for Prophet Secretary.** We design a 0.688 competitive algorithm for the prophet secretary problem. Besides the improvement over the state-of-the-art 0.672 ratio, our result further surpasses the 0.675 barrier of *blind strategies* [10], the family of algorithms that Correa et al. [10] and Harb [25] focused on. Blind strategies rely on only the distribution of the maximum value, but not the fine-grained distributional information of individual items' values. Intuitively, such fine-grained information must be crucial because the items are heterogeneous in the prophet secretary problem. However, it is technically challenging to incorporate such information to design and analyze item-dependent strategies: changing the strategy for one item would unavoidably affect the probability of accepting other items since we can accept only one of them.

**Technique: Activation-Based Algorithms.** We introduce two ideas to address this difficulty. First, we change our point of view from designing acceptance probabilities to choosing *activation rates*. In general, an online algorithm is defined by the probability of accepting an item based on its identity  $i$ , value  $v$ , and the set of future items that will arrive later. Following the conventional wisdom, the current item's arrival time  $t$  is a good surrogate for aggregating information about exponentially many possible sets of future items over their random arrivals. Here, we interpret the random order as having each item arrive within a time horizon from 0 to 1 uniformly at random. In short, algorithms are represented by the acceptance probabilities for item  $i$  when it has value  $v$  and arrives at time  $t$ . However, it is difficult to analyze the algorithms based on this representation.

By contrast, we will consider the activation rates  $a_i^v(t)$  for item  $i$  when it has value  $v$  and arrives at time  $t$ , and the overall activation rates  $A_i(t) = \mathbf{E}_{v \sim F_i}[a_i^v(t)]$  of the item at time  $t$ . We activate this item (and accept it if no item has been accepted yet) with probability:

$$g_i^v(t) = a_i^v(t) \cdot e^{-\int_0^t A_i(x) dx}.$$

This new viewpoint offers two useful invariants. By definition, the probability that we activate item  $i$  before time  $t$  equals:

$$\int_0^t \mathbf{E}_{v \sim F_i}[g_i^v(x)] dx = 1 - e^{-\int_0^t A_i(x) dx}.$$

Hence, the activation events effectively follow a Poisson process with rates  $A_i(t)$ . Accordingly, the probability that we activate item  $i$  with value  $v$  and arrival time  $t$  is:

$$\Pr[v_i = v] \cdot a_i^v(t) \cdot \underbrace{e^{-\int_0^t \sum_{j=1}^n A_j(x) dx}}_{(\star)} dt.$$

Note that the second part  $(\star)$  is independent of the item's identity  $i$  and value  $v$ . Therefore, we can simplify the dependence of different items' strategies by introducing an upper bound on  $\sum_{j=1}^n A_j(t)$  for any time  $t$ . Subject to this invariant, we can freely design the activation rates  $a_i^v(t)$  for each item  $i$  and value  $v$  to approximately match its contribution to the prophet benchmark.

To further simplify the analysis, we focus on activation rates  $a_i^v(t)$  that are step functions that change their values at a common threshold time. We demonstrate the effectiveness of this viewpoint and such simple step activation rates in Section 2.4.2 by proving a 0.694 competitive ratio when all items are small in the sense that each contributes only  $o(1)$  to the prophet benchmark.

**Technique: Significance of the Largest Item.** To further handle the general case of prophet secretary, we will focus on the item  $i_0$  with the largest probability of being selected by the prophet. This is partly inspired by the existing hard instances (e.g., [7, 10, 23]), all of which involve one large item and many small items. The significance of the largest item is twofold: 1) we need to design a special strategy for it beyond the step-function activation rates, and 2) its characteristics provide sufficient information for selecting the invariants for the other items' activation rates.

Why do we need a special strategy for this largest item? Consider the extreme case when it is the only item that matters. Intuitively, we would like to select it with certainty on its arrival. However, we cannot do that using the step-function activation rates. Within a time interval where the activation rates  $a_i^v(t)$  remain a constant, the  $e^{-\int_0^t A_i(x) dx}$  term decreases the activation probability over time. This decrease is mild for smaller items, but could be substantial for the largest item. Remarkably, this seemingly trivial instance plays an important role of establishing the 0.675 barrier for blind strategies [10]. Motivated by this extreme case, we let the largest

item's activation probability rather than its activation rate be piece-wise constant. While it is difficult to analyze algorithms based on the representation by activation/acceptance probabilities in general, we show that it is manageable to do that for just one largest item. This special treatment of just one largest item is sufficient for breaking the 0.675 barrier.

We will consider two characteristics of this largest item: its probability of being selected by the prophet, i.e.,  $x_0 = \Pr[v_{i_0} = \max_j v_j]$ , and a quantity  $h_0$  that measures the extent to which item  $i_0$  would be selected by the prophet with probability more than half, over the randomness of the other items' values. Based on just  $x_0$  and  $h_0$ , we will choose the (1) invariants  $\sum_{j \neq i_0} A_j(t)$  for the activation rates of other items, (2) the shared threshold time for the step-function activation rates of other items, and (3) the three-stage step-function activation probabilities of the largest item  $i_0$ . It is surprising that these two characteristics of the largest item alone are sufficient for choosing all important invariants for our algorithm and analysis.

**Result for Prophet Secretary Matching.** We design a 0.641-competitive algorithm for the prophet secretary matching problem, breaking the  $1 - 1/e$  barrier for the first time. As a corollary of this result, we also improve the state-of-the-art ratio of the query-commit setting from 0.633 [12] to 0.641 through a reduction [11, 22] from the query-commit model to the secretary model.

**Summary of Techniques for Prophet Secretary Matching.** We start by extending the activation-based framework to matching. First, let us consider a simple strategy: upon the arrival of an online vertex, assign it to an offline vertex with probability proportional to how likely the prophet would match them. We remark that the assignment is *independent of the arrival of earlier vertices and their matching results*. Then, from each offline vertex's viewpoint, it may treat the online vertices (more precisely, the corresponding edges) as online items in the prophet secretary problem, treating those not assigned to it as having zero values.

However, the online stochastic matching literature suggests that we should not naïvely follow this approach and apply the two-stage step-function activation rates from prophet secretary, or we would miss the opportunity of exploiting second-chance (re-)assignments (a.k.a. the power of two choices). If an offline vertex is already matched, we should no longer assign online vertices to it, but instead redirect the opportunities to other unmatched offline vertices. Hence, we introduce a third stage into the activation-based algorithm, which has the same activation rates as the second stage, but takes into account the assignments redirected from the other offline vertices. This may be viewed as reinterpreting the three-stage algorithm by Yan [40] for the i.i.d. special case within the activation-based framework. By doing so, we achieve the same 0.645 competitive ratio but more generally for all non-i.i.d. instances in which all edges are small, i.e., when each edge contributes only  $o(1)$  to the prophet benchmark.

On the other hand, the worst-case competitive ratio of this approach degenerates to  $1 - 1/e$  if there is a large edge adjacent to every offline vertex. To complement this scenario, we introduce a variant of the random order contention resolution scheme (RCRS) algorithm for matching [21, 33]. This may be viewed under the activation-based framework as follows. For each offline vertex  $u$  and its largest edge  $(u, v)$ , let its adjacent edges other than  $(u, v)$  have constant activation rates; let edge  $(u, v)$ 's activation probability be a 0-1 step-function. This is consistent with our approach for the prophet secretary problem, but the design of activation rates and probabilities is simpler due to the complications in the analysis of the more general matching problem, and the fact that we only need to beat the  $1 - \frac{1}{e}$  barrier in this case.

We show that a hybrid algorithm that randomizes over the above two approaches achieves the stated 0.641 competitive ratio.

**1.2 Related Works** Dütting et al. [14] studied the computational complexity of the optimal online algorithm for prophet secretary and gave a PTAS, though it does not imply any competitive ratio of the optimal online algorithm. Abolhassani et al. [1] and Liu et al. [34] studied the prophet secretary problem under small-item assumptions. They proved that if either 1) every distribution appears sufficiently many times [1, 34] or 2) every distribution has only a negligible probability of being non-zero [34], there exists a 0.745-competitive algorithm, matching the optimal competitive ratio as in the i.i.d. setting.

The order-selection prophet inequality lies between the i.i.d. setting and the secretary setting. In this variant, the algorithm is given the extra power of selecting the arrival order of the items. This is motivated by the application of prophet inequalities to sequential posted pricing mechanisms, and has been studied by [5, 7, 8, 36]. The current state-of-the-art competitive ratio is 0.725 by Bubna and Chiplunkar [7].

Besides matching, the prophet inequality has also been generalized to other combinatorial settings, including

matroids [30], combinatorial auctions [13, 19], and general downward-closed constraints [38]. Ehsani et al. [15] also studied the prophet secretary problem under matroid constraints and achieved a competitive ratio of  $1 - 1/e$ . Beating this ratio for general matroid constraints remains an important open question.

Finally, the unweighted and vertex-weighted online stochastic matching problems have attracted a lot of attention in the online algorithms community [4, 6, 18, 24, 27, 28, 35, 39]. Most of these works assumed i.i.d. arrivals of online vertices, in order to surpass the optimal  $1 - 1/e$  competitive ratio of online (vertex-weighted) bipartite matching [2, 29].

## 2 Prophet Secretary

**2.1 Model** Throughout the paper, we use the continuous arrival time formulation of the prophet secretary problem. There are  $n$  items with non-negative values  $v_1, \dots, v_n$  drawn independently from distributions  $F_1, \dots, F_n$ . For the ease of exposition, we make two assumptions without loss of generality. First, we consider *discrete distributions*, since continuous distributions can be approximated up to an arbitrary accuracy by discrete distributions. Second, we assume that the distributions have *disjoint supports* so that the maximum value is always unique. This can be achieved without loss of generality by adding an infinitesimal  $\varepsilon_i$  to the values from  $F_i$  for tie-breaking. We shall use  $p_i^v$  to denote the probability that  $v_i = v$ .

The items arrive within a time horizon between 0 and 1, where the arrival time  $t_i$  of item  $i$  is drawn i.i.d. from the uniform distribution over  $[0, 1]$ . The algorithm, only knowing  $F_1, \dots, F_n$ , is presented with item  $i$  at its arrival time  $t_i$ . Upon item  $i$ 's arrival, the algorithm sees the its identity  $i$ , realized value  $v_i$ , and arrival time  $t_i$ , and must decide whether to accept or reject item  $i$  immediately. If the algorithm accepts item  $i$ , it collects the value  $v_i$ ; otherwise, the algorithm continues to the next item. The algorithm can accept at most one item and the goal is to maximize the expected value of the accepted item.

Following standard competitive analysis, we compare an online algorithm's expected objective to the expected offline optimum (a.k.a., the prophet), i.e.  $\mathbf{E}[\max_i v_i]$ . An algorithm is  $\Gamma$ -competitive if its expected accepted value is at least  $\Gamma$  times the expected offline optimum.

**Continuous Arrival Time vs. Discrete Arrival Order.** We remark that the continuous time model is equivalent to the classical discrete random order model of prophet secretary. The discrete random order model samples a permutation  $\pi$  of the items uniformly at random over all  $n!$  possible permutations. Then, the items arrive in a sequence according to  $\pi$ . On one hand, any algorithm for the discrete random order model can be simulated in the continuous time model, by letting  $\pi$  be the permutation of the items in ascending order of their arrival times  $t_i$ . On the other hand, any algorithm for the continuous time model can be simulated in the discrete random order model by first sampling  $t_{(1)} \leq t_{(2)} \leq \dots \leq t_{(n)}$  i.i.d. from the uniform distribution over  $[0, 1]$ , sorted in ascending order, and letting  $t_{(\pi(i))}$  be the arrival time of item  $i$ .

## 2.2 Our Result

**THEOREM 2.1.** *There exists a 0.688-competitive algorithm for prophet secretary.*

Prior to our work, the state-of-the-art is a 0.672-competitive algorithm [9, 26]. This algorithm is a *blind strategy*; Correa, Saona, and Ziliotto [9] showed that no blind strategy can be better than 0.675 competitive. Our result breaks this barrier.

Our result can also be viewed as an exploration on the trade-off between the complexity of the state space and the approximate optimality of the online algorithms. On one hand, the online optimal algorithm based on dynamic programming has an exponential-size state space. When it decides whether to accept an item  $i$ , the state consists of the item's identity  $i$ , value  $v_i$ , arrival time  $t_i$ , and the subset of items that have not yet arrived. In particular, the last component makes the state space exponentially large (see Dütting et al. [14] for a PTAS that discretizes of the state space down to polynomial in  $n$  but doubly exponential in  $\frac{1}{\varepsilon}$ ). On the other hand, all existing algorithms in the literature of competitive analysis for prophet secretary, including the aforementioned blind strategies, decide whether to accept an item based only on this item's value and arrival time, but oblivious to its identity and the subset of remaining items. By contrast, our algorithm will further take into account the current item's identity, on top of its value and arrival time, to decide whether to accept the item.

The rest of the section is organized as follows. Section 2.3 introduces the framework of activation-based online algorithms, and how to analyze their competitive ratios. Then, Section 2.4 demonstrates two simple applications of this framework as warm-ups. Finally, Section 2.5 presents a 0.688-competitive algorithm under this framework,

and proves Theorem 2.1.

## 2.3 Activation-Based Online Algorithms

**2.3.1 General Framework** An *activation-based online algorithm* is as follows. Upon the arrival of an item, we see its identity  $i$ , value  $v \sim F_i$ , and arrival time  $t$ . Note that we suppress the subscript  $i$  in the value and arrival time for notation simplicity as the item's identity  $i$  will always be clear from context. Then, we decide whether to activate the item or not based on an *activation probability*  $0 \leq g_i^v(t) \leq 1$ , depending on  $i$ ,  $v$ , and  $t$ , but importantly, independent of the subset of items that have not yet arrived. Finally, we accept an item if it is the first activated item. By definition, an activation-based online algorithm is characterized by the activation probabilities  $g_i^v(t) \in [0, 1]$ .

### Activation Framework

For time  $t$  from 0 to 1:

- (1) If item  $i$  arrives with value  $v$ , activate it with activation probability  $g_i^v(t)$ .
- (2) Accept item  $i$  if it is the first activated item.

Now we show two basic properties of these algorithms.

LEMMA 2.1. *The probability of activating item  $i$  before time  $\theta$  is  $\int_0^\theta \mathbf{E}_{v \sim F_i} [g_i^v(t)] dt$ .*

*Proof.* We will first consider the probability of activating item  $i$  with value  $v$  before time  $\theta$ . This event can be decomposed into three sub-events: (1) the item arrives at time  $t \leq \theta$ ; (2) the item has value  $v$ , and (3) the algorithm activates the item.

Recall that the arrival time  $t$  distributes uniformly over  $[0, 1]$ , and the probability of drawing value  $v$  from  $F_i$  is  $p_i^v$ . Note that  $t$  and  $v$  are independent. Further, conditioned on the first two sub-events, the third happens with probability  $g_i^v(x)$ . Hence, the probability of activating item  $i$  with value  $v$  before time  $\theta$  is:

$$\int_0^\theta p_i^v g_i^v(t) dt.$$

Summing over all possible values  $v$ , the probability of activating a item  $i$  before time  $\theta$  is:  $\int_0^\theta \sum_v p_i^v g_i^v(t) dt = \int_0^\theta \mathbf{E}_{v \sim F_i} [g_i^v(t)] dt$ .  $\square$

LEMMA 2.2. *The probability of accepting item  $i$  with value  $v$  before time  $\theta$  is:*

$$\int_0^\theta p_i^v g_i^v(t) \prod_{j \neq i} \left( 1 - \int_0^t \mathbf{E}_{v \sim F_j} [g_j^v(x)] dx \right) dt.$$

*Proof.* For any time  $t \leq \theta$ , the event that item  $i$  is accepted at time  $t$  can be decomposed into three independent sub-events: (1) item  $i$  has value  $v$  at time  $t$ ; (2) item  $i$  is activated at time  $t$ , and (3) any item  $j \neq i$  is not activated before time  $t$ . The first sub-event happens with probability  $p_i^v$ , and the second sub-event happens with probability  $g_i^v(t)$ . By Lemma 2.1, the probability of activating item  $j$  before time  $t$  is  $\int_0^t \mathbf{E}_{v \sim F_j} [g_j^v(x)] dx$ , thus the third sub-event happens with probability  $\prod_{j \neq i} (1 - \int_0^t \mathbf{E}_{v \sim F_j} [g_j^v(x)] dx)$ . Together, the probability of accepting item  $i$  with value  $v$  before time  $\theta$  is as stated.  $\square$

**2.3.2 Activation Rates** This subsection presents an alternative representation of the activation probabilities  $g_i^v(t)$  that will simplify parts of the subsequent analysis. Formally, for each item  $i$ , we focus on activation probabilities in the following form:

$$g_i^v(t) = a_i^v(t) e^{-\int_0^t A_i(x) dx}, \quad \forall i, v, t$$

where  $A_i(x) \stackrel{\text{def}}{=} \sum_v p_i^v a_i^v(x) = \mathbf{E}_{v \sim F_i} [a_i^v(x)]$ . We call  $a_i^v(t)$  the *activation rate* of  $v$  at time  $t$ . We say that the activation rate  $a_i^v(t)$  is *well-defined* if the corresponding activation probability  $g_i^v(t)$  is at most 1 for every  $i, t, v$ .

LEMMA 2.3. *The probability of activating item  $i$  before a threshold time  $\theta$  is  $1 - e^{-\int_0^\theta A_i(t) dt}$ .*

This lemma justifies why we call  $a_i^v(t)$  the activation rate, as its conclusion resembles the form of a Poisson activation process with rate  $A_i(t)$ .

*Proof.* Replacing  $g_i^v(t)$  with  $a_i^v(t)e^{-\int_0^t A_i(x) dx}$  in Lemma 2.1, the probability of activating item  $i$  before time  $\theta$  is:

$$\int_0^\theta \mathbf{E}_v [a_i^v(t)] e^{-\int_0^t A_i(x) dx} dt = \int_0^\theta A_i(t) e^{-\int_0^t A_i(x) dx} dt = 1 - e^{-\int_0^\theta A_i(t) dt}.$$

□

LEMMA 2.4. *The probability of accepting item  $i$  with value  $v$  is:*

$$\int_0^1 p_i^v a_i^v(t) e^{-\int_0^t \sum_j A_j(x) dx} dt.$$

*Proof.* We substitute  $g_i^v(t)$  with  $a_i^v(t)e^{-\int_0^t A_i(x) dx}$  in Lemma 2.2 and the probability that item  $j$  is not activated before time  $t$  with  $e^{-\int_0^t A_j(x) dx}$  by Lemma 2.4. Hence, the probability of accepting item  $i$  with value  $v$  is:

$$\int_0^1 p_i^v a_i^v(t) e^{-\int_0^t A_i(x) dx} \prod_{j \neq i} \left( e^{-\int_0^t A_j(x) dx} \right) dt = \int_0^1 p_i^v a_i^v(t) e^{-\int_0^t \sum_j A_j(x) dx} dt.$$

□

**2.3.3 Sufficient Condition for Competitive Analysis** For every item  $i$  and value  $v$ , we denote the (conditional) probability that  $v_i = v$  is the maximum value as the following:

$$x_i^v \stackrel{\text{def}}{=} p_i^v \cdot \prod_{j \neq i} \Pr[v_j < v] \quad \text{and} \quad \rho_i^v \stackrel{\text{def}}{=} \prod_{j \neq i} \Pr[v_j < v].$$

We list the following properties of  $x_i^v$  and  $\rho_i^v$  that are straightforward to verify.

LEMMA 2.5. 1)  $x_i^v = p_i^v \rho_i^v$ ; 2)  $\rho_i^v \leq 1$  is non-decreasing in  $v$ ; and 3)  $\sum_{i,v} x_i^v = 1$ .

Our competitive analysis is induced by the following stochastic dominance.

LEMMA 2.6. *An online algorithm is  $\Gamma$ -competitive if for any item  $i$  and any value  $v$ :*

$$\int_0^1 p_i^v g_i^v(t) \prod_{j \neq i} \left( 1 - \int_0^t \mathbf{E}_{v_j \sim F_j} [g_j^{v_j}(x)] dx \right) dt \geq \Gamma \cdot x_i^v = \Gamma \cdot p_i^v \rho_i^v.$$

*Proof.* Multiplying the inequality by  $v$  and summing over all  $v$  and  $i$ , the left hand side gives the gain from the algorithm, and the right hand side is  $\Gamma$  of the offline optimal gain. □

**2.4 Warm-ups** This subsection presents two warm-up algorithms that use simple activation rates. The first algorithm considers a constant activation rate  $a_i^v(t) = \rho_i^v$  for any item  $i$  and any value  $v$ . The second algorithm considers a threshold time  $\beta$ , and lets  $a_i^v(t)$  be a step function that changes its value at time  $\beta$  for any item  $i$  and value  $v$ .

**2.4.1 Constant Activation Rates** We now present the Constant Activation Rates algorithm, and prove that it is  $1 - \frac{1}{e}$  competitive. We note that it is exactly the  $(1 - \frac{1}{e})$ -selectable RCRS algorithm by Lee and Singla [33], rewritten in our activation-based framework.

#### Constant Activation Rates [33]

For time  $t$  from 0 to 1:

- (a) If item  $i$  arrives with value  $v$ , activate it with activation rate  $a_i^v(t) = \rho_i^v$ .
- (b) Accept item  $i$  if it is the first activated item.

**THEOREM 2.2.** *Constant Activation Rates is  $(1 - \frac{1}{e})$ -competitive.*

*Proof.* Recall that by definition of  $A_i(t)$ , we have  $A_i(t) = \sum_v p_i^v \rho_i^v = \sum_v x_i^v$ . According to Lemma 2.4, the probability of accepting item  $i$  with value  $v$  is:

$$\int_0^1 p_i^v \rho_i^v e^{-\sum_j \int_0^t A_j(x) dx} dt = x_i^v \int_0^1 e^{-t \cdot \sum_j \sum_v x_j^v} dt = x_i^v \int_0^1 e^{-t} dt = \left(1 - \frac{1}{e}\right) x_i^v.$$

Here we use the property that  $\sum_j \sum_v x_j^v = 1$  (Lemma 2.5). Further by Lemma 2.6, we get that the algorithm is  $(1 - \frac{1}{e})$ -competitive.  $\square$

Why are constant activation rates suboptimal? Intuitively, we would like to let the activation rate be larger, e.g., converging to 1 when the time approaches 1, because at that time there would be few items left and thus, we should take any occurring item. Accordingly, the activation rate should be smaller at the beginning, unless the realized value is very high, because we anticipate more items to come in the future.

**2.4.2 Step Activation Rates** The next warm-up algorithm implements the above idea by letting the activation rate  $a_i^v(t)$  for any item  $i$  and value  $v$  be a step function that increases at a threshold time  $\beta$ . We maintain the invariant that the activation rates in the two stages sum to  $2 \cdot \rho_i^v$ , but let the rate in the second stage be as high as possible.

#### Step Activation Rates

Parameter:  $\beta = 0.367$ .

For time  $t$  from 0 to 1:

- (a) If item  $i$  arrives with value  $v$ , activate it with activation rate:

$$a_i^v(t) = \begin{cases} (2\rho_i^v - 1)^+ & \text{if } t \in [0, \beta), \\ 2\rho_i^v - (2\rho_i^v - 1)^+ & \text{if } t \in [\beta, 1]. \end{cases}$$

- (b) Accept item  $i$  if it is the first activated item.

We will analyze this algorithm under the assumption that:

$$(2.1) \quad h \stackrel{\text{def}}{=} \sum_{i,v} (2x_i^v - \rho_i^v)^+ \leq 1 - \ln 2.$$

This is inspired by a property from the stochastic online matching literature proved by Manshadi et al. [35]. It would hold up to an  $o(1)$  additive error if every item  $i$  is small in the sense that it has the highest value with probability only  $o(1)$ , namely,  $\sum_v x_i^v = o(1)$ . A formal proof of this claim is deferred to Appendix B.

**Parameter of the Invariant.** We remark that there is actually a parameter  $s$  for the invariant that we can optimize for the above algorithm: the activation rates in the two stages sum to  $s \cdot \rho_i^v$ . In the main body of the paper, we fix  $s$  to be 2 both in this warm-up algorithm and our main algorithm in the next subsection. This simplification makes it clear the connection between the assumption (2.1) and the constraint [28] from previous works. The algorithm for general  $s$  is provided in Appendix C.

LEMMA 2.7. *We have:*

$$\sum_j A_j(t) = \begin{cases} h & \text{if } 0 \leq t < \beta; \\ 2 - h & \text{if } \beta \leq t \leq 1. \end{cases}$$

*Proof.* By the definition of  $A_j(t)$ 's, we have  $\sum_j A_j(t) = \sum_j \sum_v p_j^v a_j^v(t)$ . Further by the definition of  $h$ , we have  $\sum_j \sum_v p_j^v (2\rho_j^v - 1)^+ = \sum_j \sum_v (2x_j^v - p_j^v)^+ = h$ . Then, according to Lemma 2.5, we have  $\sum_j \sum_v p_j^v \rho_j^v = \sum_j \sum_v x_j^v = 1$ , and thus,  $\sum_j \sum_v p_j^v (2\rho_j^v - (2\rho_j^v - 1)^+) = 2 - h$ .  $\square$

THEOREM 2.3. *Assuming (2.1), Step Activation Rates is 0.694-competitive.*

*Proof.* We denote the cumulative total activation rates up to time  $t$  as:

$$G(t) \stackrel{\text{def}}{=} \int_0^t \sum_j A_j(t) \, dy = \begin{cases} h \cdot t & \text{if } 0 \leq t < \beta; \\ h \cdot \beta + (2 - h) \cdot (t - \beta) & \text{if } \beta \leq t \leq 1. \end{cases}$$

By Lemma 2.4, the probability of accepting item  $i$  when its value is  $v$  equals:

$$\begin{aligned} & \int_0^1 p_i^v a_i^v(t) e^{-\int_0^t \sum_j A_j(t) \, dy} \, dt \\ &= p_i^v \cdot \left( (2\rho_i^v - 1)^+ \int_0^\beta e^{-G(t)} \, dt + (2\rho_i^v - (2\rho_i^v - 1)^+) \int_\beta^1 e^{-G(t)} \, dt \right) \\ &= p_i^v \cdot \left( (2\rho_i^v - 1)^+ \cdot \underbrace{\frac{1 - e^{-h\beta}}{h}}_{(a)} + (2\rho_i^v - (2\rho_i^v - 1)^+) \cdot \underbrace{\frac{e^{-h\beta} - e^{-(2-h)\beta}}{2-h}}_{(b)} \right). \end{aligned}$$

Our choice of threshold time  $\beta \approx 0.367$  makes  $(a) \approx (b) > 0.347$ , i.e., half the claimed competitive ratio, when  $h = 1 - \ln 2$ . For smaller  $h$ , we still have  $(a), (b) > 0.347$  because both  $(a)$  and  $(b)$  are decreasing in  $h$ . Hence the above bound is at least:

$$p_i^v \cdot \left( (2\rho_i^v - 1)^+ + (2\rho_i^v - (2\rho_i^v - 1)^+) \right) \cdot 0.347 = 0.694 \cdot p_i^v \rho_i^v.$$

By Lemma 2.6, this algorithm is 0.694-competitive.  $\square$

**2.5 Step Activation Rates Except One Large Item** Finally, we present our algorithm for the general case of prophet secretary. As mentioned in the last subsection, we shall optimize the parameter  $s$  from the invariant to maximize our competitive ratio. Within this subsection, we fix  $s = 2$  and establish a competitive ratio of 0.686. The more general version of our algorithm and its analysis, which prove the 0.688 competitive ratio stated in Theorem 2.1, are provided in Appendix C.

Recall that the analysis in Section 2.4.2 relies on the assumption that  $\sum_{i,v} (2x_i^v - p_i^v)^+ \leq 1 - \ln 2$ . To get rid of the assumption, we consider the importance of the largest item  $i_0 = \arg \max_i \sum_v x_i^v$ :

$$x_0 \stackrel{\text{def}}{=} \sum_v x_{i_0}^v = \Pr \left[ v_{i_0} = \max_j v_j \right].$$



LEMMA 2.8. *For any instance of prophet secretary in which the largest item  $i_0$ 's probability of having the highest value is  $x_0$ , we have:*

$$\sum_{i,v} (2x_i^v - p_i^v)^+ \leq h_2(x_0) \stackrel{\text{def}}{=} \max_{t \in [0, x_0)} t + \sum_{k \geq 0} \left( 2x_0 - \frac{x_0}{1 - t - kx_0} \right)^+.$$

Moreover,  $\lim_{x \rightarrow 0^+} h_2(x) = 1 - \ln 2$ , and  $h_2(1) = 1$ .

Intuitively,  $(2x_i^v - p_i^v)^+$  captures the extent to which an item-value pair  $(i, v)$  is the largest more than half the time when it is present. Lemma 2.8 upper bounds the sum of this quantity over all item-value pairs, depending on the characteristics of the largest item. It generalizes (2.1) by Manshadi et al. [35] in the context of i.i.d. online stochastic matching, which plays an important role in that literature (e.g., [27, 28, 35, 40]).

In Appendix B, we prove the following generalized statement to optimize over  $s$ .

LEMMA 2.9. *For any instance of prophet secretary in which the largest item  $i_0$ 's probability of having the highest value is  $x_0$ , we have:*

$$\frac{1}{s-1} \cdot \sum_{i,v} (sx_i^v - p_i^v)^+ \leq h_s(x_0) \stackrel{\text{def}}{=} \max_{t \in [0, x_0)} t + \frac{1}{s-1} \sum_{k \geq 0} \left( sx_0 - \frac{x_0}{1 - t - kx_0} \right)^+.$$

This immediately implies Lemma 2.8, which is the special case when  $s = 2$ . We will also use this lemma in the prophet secretary matching section.

Figure 1 presents our algorithm for the prophet secretary problem. It handles the largest item and the other items using two different strategies.

For the largest item, we let its activation probabilities, rather than activation rates, be step functions that change values at some thresholds  $\beta_0 < \beta_2$  to be determined. We never activate this item from time 0 to  $\beta_0$ . The activation probabilities from time  $\beta_0$  to  $\beta_2$ , and from time  $\beta_2$  to 1 sum to  $2\rho_{i_0}^v$ , twice the probability that the largest item  $i_0$  has the highest value *conditioned on its value being  $v$* ; we further let the latter (from time  $\beta_2$  to 1) be as large as possible.

For the other items, we let their activation rates be step functions that change values at some threshold  $\beta_1$  to be determined. We will let  $\beta_0 \leq \beta_1 \leq \beta_2$  and hence, the choice of their subscripts. Like the warm-up case in the last subsection, we also let the activate rates before and after the threshold  $\beta_1$  sum to  $2\rho_i^v$ . Unlike the warm-up case, where we let the activation rates before  $\beta_1$  be as small as possible, here we let them be parameters  $z_i^v$  that sum to (weighted by probability  $p_i^v$ ) a fixed invariant depending on the characteristics of the largest item.

We next prove that the algorithm is well-defined.

LEMMA 2.10. *There exist  $z_i^v$ 's satisfying the stated constraints in (2) of Figure 1.*

*Proof.* Note that  $(2\rho_i^v - 1)^+ \leq \rho_i^v$ , it suffices to show that

$$\sum_{i \neq i_0} \sum_v p_i^v (2\rho_i^v - 1)^+ \leq \min\{h_2(x_0) - h_0, 1 - x_0\} \leq \sum_{i \neq i_0} \sum_v p_i^v \rho_i^v.$$

The first inequality follows by:

$$\begin{aligned} \sum_{i \neq i_0} \sum_v p_i^v (2\rho_i^v - 1)^+ &= \sum_{i \neq i_0} \sum_v (2x_i^v - p_i^v)^+ \\ &= \sum_i \sum_v (2x_i^v - p_i^v)^+ - h_0 \leq h_2(x_0) - h_0, \end{aligned}$$

The second inequality follows by  $\sum_{i \neq i_0} \sum_v p_i^v \rho_i^v = \sum_{i \neq i_0} \sum_v x_i^v = 1 - x_0$ .  $\square$

LEMMA 2.11. *The activation rates and probabilities are well-defined, namely they are between 0 and 1.*

### Step Activation Rates Except One Large Item

(1) Let  $i_0 = \arg \max_i \sum_v x_i^v$  be the largest item, and let:

$$x_0 = \sum_v x_{i_0}^v, \quad z_{i_0}^v = (2\rho_{i_0}^v - 1)^+, \quad h_0 = \sum_v p_{i_0}^v z_{i_0}^v.$$

(2) Calculate a set of  $z_i^v$ 's for the remaining items  $i \neq i_0$  such that:

$$h_{ot} \stackrel{\text{def}}{=} \sum_{i \neq i_0} \sum_v p_i^v z_i^v = \min \{h_2(x_0) - h_0, 1 - x_0\}, \quad z_i^v \in [(2\rho_i^v - 1)^+, \rho_i^v],$$

where  $h_2(x_0)$  is the function as defined in Lemma 2.8.

(3) Select threshold times  $\beta_0 \leq \beta_1 \leq \beta_2$  based on  $x_0, h_0$  to maximize Equation (2.2).

(4) For time  $t$  from 0 to 1:

(a) If the largest item  $i_0$  arrives with value  $v$ , activate it with probability:

$$g_{i_0}^v(t) = \begin{cases} 0 & \text{if } t \in [0, \beta_0), \\ z_{i_0}^v & \text{if } t \in [\beta_0, \beta_2), \\ 2\rho_{i_0}^v - z_{i_0}^v & \text{if } t \in [\beta_2, 1]. \end{cases}$$

(b) If an item  $i \neq i_0$  arrives with value  $v$ , activate it with activation rate:

$$a_i^v(t) = \begin{cases} z_i^v & \text{if } t \in [0, \beta_1), \\ 2\rho_i^v - z_i^v & \text{if } t \in [\beta_1, 1]. \end{cases}$$

(c) Accept the item if it is the first activated item.

Figure 1: Algorithm for Prophet Secretary

*Proof.* Recall that  $0 \leq x_i^v \leq \rho_i^v \leq 1$ . For any item  $i$ , we have  $0 \leq z_i^v \leq \rho_i^v \leq 1$ , and  $2\rho_i^v - z_i^v = 2\rho_i^v - (2\rho_i^v - 1)^+ = \min\{2\rho_i^v, 1\} \in [0, 1]$ .

Hence, for the largest item  $i_0$ , the activation probability  $g_{i_0}^v(t)$  is always between 0 and 1. For any other item  $i \neq i_0$ , the activation rate  $a_i^v(t)$  is between 0 and 1 and the resulting activation probability  $g_i^v(t) = a_i^v(t)e^{-\int_0^t A_i(x) dx}$  is also between 0 and 1.  $\square$

We are now ready to prove Theorem 2.1, which says that the Step Activation Rates Except One Large Item is 0.686-competitive.

*Proof.* [Proof of Theorem 2.1] We will leave  $\beta_0 \leq \beta_1 \leq \beta_2$  as parameters to be determined, and optimize them to maximize the competitive ratio via a computer-aided search.

**Largest Item.** By Lemma 2.3, the probability that an item  $j \neq i_0$  was not activated before time  $t$  is  $e^{-\int_0^t A_j(x) dx}$ , where  $A_j(x)$  equals  $\sum_v p_j^v z_j^v$  when  $x \in [0, \beta_1]$  and equals  $2\sum_v x_j^v - \sum_v p_j^v z_j^v$  otherwise.

By the definition of  $h_{ot}$  and  $\sum_j \sum_v x_j^v = 1$  (Lemma 2.5), we have:

$$\sum_{j \neq i_0} A_j(x) = \begin{cases} \sum_{j \neq i_0} \sum_v p_j^v z_j^v = h_{ot}, & \text{for } x \in [0, \beta_1] \\ \sum_{j \neq i_0} (2\sum_v x_j^v - \sum_v p_j^v z_j^v) = 2(1 - x_0) - h_{ot}. & \text{for } x \in [\beta_1, 1] \end{cases}$$

Thus, the probability of not having any item  $j \neq i_0$  activated before time threshold  $t$  is:

$$\prod_{j \neq i_0} e^{-\int_0^t A_j(x) dx} = e^{-\int_0^t \sum_{j \neq i_0} A_j(x) dx} = e^{-h_{ot} \cdot t - 2(1-x_0-h_{ot}) \cdot (t-\beta_1)^+} \stackrel{\text{def}}{=} e^{-K(t)}.$$

Then, according to Lemma 2.2, the probability of accepting item  $i_0$  value  $v_{i_0} = v$  is:

$$\begin{aligned} \int_0^1 p_{i_0}^v g_{i_0}^v(t) e^{-K(t)} dt &= p_{i_0}^v \left( z_{i_0}^v \int_{\beta_0}^1 e^{-K(t)} dt + (\rho_{i_0}^v - z_{i_0}^v) \cdot 2 \int_{\beta_2}^1 e^{-K(t)} dt \right) \\ &\geq p_{i_0}^v \rho_{i_0}^v \cdot \min \left( \int_{\beta_0}^1 e^{-K(t)} dt, 2 \int_{\beta_2}^1 e^{-K(t)} dt \right). \end{aligned}$$

**Other Items.** We define  $L(t)$  as the probability that item  $i_0$  is activated before time  $t$ . Then, according to the definition of  $g_{i_0}^v(t)$  and  $h_0$ ,  $\sum_v p_{i_0}^v g_{i_0}^v(t)$  equals 0 for  $t \in [0, \beta_0]$ , equals  $h_0$  for  $t \in [\beta_0, \beta_2]$ , and equals  $2x_0 - h_0$  for  $t \in [\beta_2, 1]$ . Thus, by Lemma 2.1, we have

$$L(t) = \int_0^t \sum_v p_{i_0}^v g_{i_0}^v(x) dx = h_0 \cdot (t - \beta_0)^+ + 2(x_0 - h_0) \cdot (t - \beta_2)^+.$$

Therefore, for any item  $i \neq i_0$ , according to Lemma 2.2, the probability of accepting item  $i$  with value  $v$  is equal to:

$$\begin{aligned} &\int_0^1 p_i^v a_i^v(t) e^{-\int_0^t \sum_i A_i(x) dx} (1 - L(t)) \prod_{j \neq i, i_0} e^{-\int_0^t A_j(y) dy} dt \\ &= p_i^v \int_0^1 a_i^v(t) \prod_{j \neq i_0} e^{-\int_0^t A_j(y) dy} (1 - L(t)) dt \\ &= p_i^v \int_0^1 a_i^v(t) e^{-K(t)} (1 - L(t)) dt \\ &= p_i^v \left( z_i^v \cdot \int_0^1 e^{-K(t)} (1 - L(t)) dt + (\rho_i^v - z_i^v) \cdot 2 \int_{\beta_1}^1 e^{-K(t)} (1 - L(t)) dt \right) \\ &\geq p_i^v \rho_i^v \cdot \min \left( \int_0^1 e^{-K(t)} (1 - L(t)) dt, 2 \int_{\beta_1}^1 e^{-K(t)} (1 - L(t)) dt \right). \end{aligned}$$

**Choice of Parameters.** Therefore, we define:

$$(2.2) \quad \Gamma(x_0, h_0, \beta_0, \beta_1, \beta_2) = \min \left( \int_{\beta_0}^1 e^{-K(t)} dt, 2 \int_{\beta_2}^1 e^{-K(t)} dt, \int_0^1 e^{-K(t)} (1 - L(t)) dt, 2 \int_{\beta_1}^1 e^{-K(t)} (1 - L(t)) dt \right),$$

which is a function with respect to  $x_0, h_0, \beta_0, \beta_1, \beta_2$ . It remains to show that:

**LEMMA 2.12.** *For every  $x_0 \in [0, 1]$  and  $h_0 \in [0, x_0]$ , there exist  $0 \leq \beta_0 \leq \beta_1 \leq \beta_2 \leq 1$ , such that  $\Gamma(x_0, h_0, \beta_0, \beta_1, \beta_2) > 0.686$ .*

We use a computer-aided search to numerically verify this lemma (see Appendix D). Then, taking such  $\beta_0, \beta_1, \beta_2$  in the algorithm, we get the 0.686 competitive ratio by Lemma 2.6.  $\square$

### 3 Prophet Secretary Matching

#### 3.1 Preliminaries

**Model.** Let there be an edge-weighted bipartite type graph  $G = (U, V, E, w)$ , where  $U$  corresponds to the set of offline vertices that are known in advance to the algorithm and  $V$  corresponds to the set of all possible types of each online vertices.

There are  $n$  online vertices. Each online vertex  $v_i$  has a type  $v \in V$  drawn independently from a known distribution  $F_i$ , and an arrival time of  $t_i$  drawn uniformly from  $[0, 1]$ . We use  $p_i^v$  to denote the probability of  $v_i = v$ . When  $v_i$  arrives at time  $t_i$ , its type is realized and the algorithm decides immediately whether to match  $v_i$  to an unmatched offline vertex  $u$  (if exists). The goal is to maximize the total weight of the selected matching and to compete against the expected maximum matching.

**LP Relaxation.** Consider the following linear program for stochastic matching by Gamlath, Kale, and Svensson [22].

$$\begin{aligned} \text{LP:} \quad & \max_{\{x_i^{(u,v)}\}} : \sum_{i,u,v} w_{uv} \cdot x_i^{(u,v)} \\ & \text{subject to: } \sum_u x_i^{(u,v)} \leq p_i^v \quad \forall i \in [n], v \in V \\ & \sum_i \sum_{v \in S} x_i^{(u,v)} \leq 1 - \prod_i \left(1 - \sum_{v \in S} p_i^v\right) \quad \forall u \in U, S \subseteq V \end{aligned}$$

FACT 3.1. ([22]) *This linear program can be solved efficiently and its value is an upper bound of the expected optimum matching.*

Our algorithm starts by solving the linear program and throughout the section, we use  $\{x_i^{(u,v)}\}$  to denote the optimal solution of the LP. For each online vertex  $u$ , we consider the following quantities that shall be crucial for our algorithm design:

$$\forall i \in [n], u \in U, \quad x_i^u \stackrel{\text{def}}{=} \sum_v x_i^{(u,v)}, \quad x^u \stackrel{\text{def}}{=} \max_i x_i^u, \quad \text{and} \quad h_u \stackrel{\text{def}}{=} \sum_{i,v} \left(2x_i^{(u,v)} - p_i^v\right)^+.$$

According to the constraints of the LP and by Lemma 2.8, we have that  $h_u \leq h_2(x^u)$ .

**Assumption.** Without loss generality, we assume that the graph is 1-regular. I.e., for every offline vertex  $u \in U$ ,  $\sum_{i,v} x_i^{(u,v)} = 1$  and for every  $i \in [n]$ ,  $\sum_{u,v} x_i^{(u,v)} = 1$ . This can be achieved by first adding dummy offline vertices and dummy online types to the type graph, and then introducing dummy online vertices to the instance. A similar pre-processing step is applied by Fu et al. [21] (refer to Lemma 2 of [21] for a detailed construction). Finally, we simulate the random arrival of the dummy online vertices on our own.

**3.2 Overview of Our Result** Our main result is a 0.641-competitive algorithm for the prophet secretary matching problem, beating the  $1 - 1/e \approx 0.632$  barrier for the first time.

We introduce two algorithms that we call 1) Multistage Activation-based Matching (MAM) and 2) Constant Activation Rate except One Large Vertex (CAR).

- Our Multistage Activation-based Matching algorithm is a generalization of the algorithm by Yan [40] that works in the i.i.d. arrival setting. Our algorithm achieves the same competitive ratio of Yan for an arbitrary “infinitesimal instance”, i.e., when  $x^u = o(1)$  for every  $u \in U$ . In other words, an instance is infinitesimal if every online vertex  $v_i$  contributes  $o(1)$  to every offline vertex  $u$  in the offline maximum matching. Crucially, our algorithm avoids the pre-processing step of Yan, which to our knowledge, is hard to generalize beyond the i.i.d. case.

Formally, our algorithm matches each edge  $(u, v_i = v)$  with probability  $\Gamma_{\text{MAM}}(x^u) \cdot x_i^{(u,v)}$ , where  $\Gamma_{\text{MAM}}(\cdot)$  is a decreasing function with  $\Gamma_{\text{MAM}}(0) \approx 0.645$  and  $\Gamma_{\text{MAM}}(1) = 1 - 1/e$ . As an implication, the MAM algorithm achieves the goal of beating  $1 - 1/e$  as long as every  $x^u$  is bounded away from 1. This algorithm and its analysis (including the definition of  $\Gamma_{\text{MAM}}$ ) are provided in Section 3.4.

- The second algorithm is built on the technique of random order contention resolution schemes. It is noticed by Fu et al. [21] that the RCRS by Lee and Singla [33] can be adapted to a  $1 - 1/e$  competitive algorithm

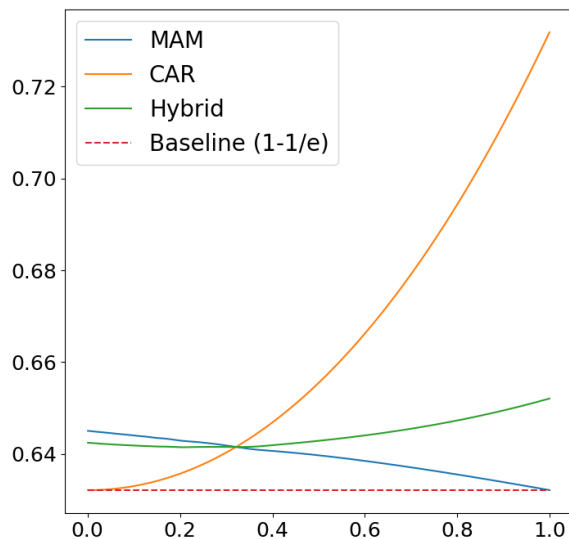


Figure 2: Competitive Ratio Curves For MAM, CAR, Hybrid

for the prophet secretary matching problem and the ratio is tight. Moreover, we observe that the worst-case instance for the RCRS based approach is the infinitesimal case when  $x^u$ 's are small.

To this end, we refine the RCRS algorithm by Lee and Singla so that our algorithm matches each edge  $(u, v_i = v)$  with probability  $\Gamma_{\text{CAR}}(x^u)$ , where  $\Gamma_{\text{CAR}}(\cdot)$  is an increasing function with  $\Gamma_{\text{CAR}}(0) = 1 - 1/e$  and  $\Gamma_{\text{CAR}}(1) = \sqrt{3} - 1 \approx 0.732$ . This algorithm and its analysis (including the definition of  $\Gamma_{\text{CAR}}$ ) are provided in Section 3.5.

Though each of the two algorithms only achieves a competitive ratio of  $1 - 1/e$  in the worst case, they nicely complement each other. Finally, we apply a randomization between the two algorithms.

**THEOREM 3.1.** *A hybrid algorithm by running MAM with probability 0.8 and running CPR with probability 0.2 is at least 0.641-competitive for prophet secretary matching.*

*Proof.* By Theorem 3.3 and Theorem 3.4, the hybrid algorithm matches each edge  $(u, v_i = v)$  with probability

$$(0.8 \cdot \Gamma_{\text{MAM}}(x^u) + 0.2 \cdot \Gamma_{\text{CAR}}(x^u)) \cdot x_i^{(u,v)} \geq 0.641 \cdot x_i^{(u,v)}, \quad \forall x^u \in [0, 1],$$

where the inequality is verified with computer assistance. Refer to Figure 2. Recall that  $\sum_{i,u,v} w_{uv} \cdot x_i^{(u,v)}$  is an upper bound of the expected maximum matching. The property achieved directly implies a competitive ratio of 0.641.  $\square$

**3.3 Query-Commit Stochastic Bipartite Matching** Our algorithm for the prophet secretary matching can be applied to the query-commit stochastic bipartite matching problem. Consider a bipartite graph  $G = (U, V, E)$  where each edge  $e$  is associated with a weight  $w_e$  and a probability  $p_e$ . At each step, the algorithm queries the existence of an edge  $e$ , which happens independently with probability  $p_e$ . If the edge exists, the algorithm must select it. The goal is to maximize the selected matching and competes against the optimum matching.

Gamlath, Kale, and Svensson [22] designed an  $1 - 1/e$  approximation algorithm for this problem, and the ratio is recently improved to  $1 - 1/e + 0.0014$  by Derakhshan and Farhadi [12]. As an implication of Theorem 3.1 through a folklore reduction [20, 21] from the query-commit model to secretary model, we also improve the state-of-the-art approximation ratio of this problem. For completeness, we sketch a proof of the reduction in Appendix A.

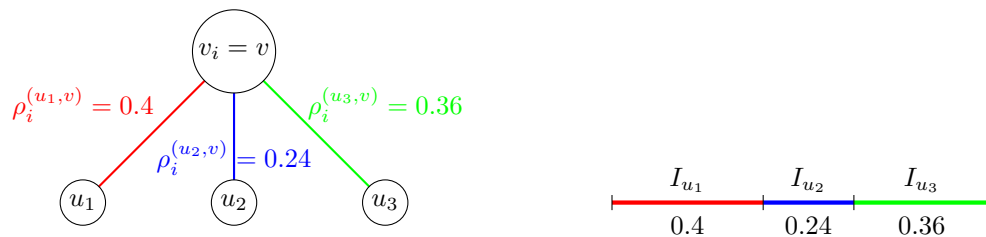


Figure 3: Illustration of intervals  $I_{u_1}$ ,  $I_{u_2}$  and  $I_{u_3}$  for  $v_i = v$ .

**THEOREM 3.2.** *There exists a 0.641 approximation algorithm for the query-commit stochastic bipartite matching problem.*

**3.4 Multistage Activation-based Matching** In this section, we introduce the multistage activation-based matching algorithm. We first recall the correlated sampling method of Jaillet and Lu [28].

**Jaillet and Lu Sampling[28].** For every online vertex  $v_i$  and type  $v$ :

- Consider an interval  $[0, 1]$  and align subintervals  $I_u$  of length  $\rho_i^{(u, v)} \stackrel{\text{def}}{=} \frac{x_i^{(u, v)}}{p_i^v}$  from left to right for each offline vertex  $u \in U$ . Refer to Figure 3.
- Sample  $\eta$  uniformly at random from  $[0, 1]$ . Let  $\eta' = \eta \pm 1/2$  such that  $\eta' \in [0, 1]$ .
- Let  $\mu_i^v(u \rightarrow u') \stackrel{\text{def}}{=} \mathbf{Pr}[\eta \in I_u \text{ and } \eta' \in I_{u'}]$ .

The quantities  $\mu_i^v(u \rightarrow u')$  are important parameters of our algorithm and analysis. We list the following properties of  $\mu_i^v(u \rightarrow u')$  that are straightforward to verify.

**LEMMA 3.1.** *We have that*

- $\mu_i^v(u \rightarrow u') = \mu_i^v(u' \rightarrow u)$ ;
- $\sum_{u' \neq u} \mu_i^v(u \rightarrow u') = \rho_i^{(u, v)} - (2\rho_i^{(u, v)} - 1)^+ \leq \rho_i^{(u, v)}$ .

Now, we present our algorithm, a generalization of the algorithm by Yan [40].

## Multistage Activation-based Matching

Parameters:  $\beta_0 = 0.05, \beta_1 = 0.75$ .

Notations:  $M(t), N(t)$  are the sets of matched and unmatched offline vertices at time  $t$ .

- (1) First Stage: for  $t \in [0, \beta_0)$ , let  $a_i^{(u,v)}(t) = (2\rho_i^{(u,v)} - 1)^+$  for every  $u, v, i$ ;
- (2) Second Stage: for  $t \in [t_0, \beta_1)$ , let  $a_i^{(u,v)}(t) = \rho_i^{(u,v)}$  for every  $u, v, i$ ;
- (3) If vertex  $v_i$  arrives at time  $t \in [0, \beta_1)$  with type  $v \sim F_i$ :  
propose to a random offline vertex  $u \in U$  with probability

$$(3.3) \quad g_i^{(u,v)}(t) = a_i^{(u,v)}(t) \cdot \exp\left(-\int_0^t A_i^u(y) \, dy\right),$$

where  $A_i^u(t) \stackrel{\text{def}}{=} \mathbf{E}_v[a_i^{(u,v)}(t)] = \sum_v p_i^v \cdot a_i^{(u,v)}(t)$ .

Accept the edge  $(u, v_i = v)$  if  $u$  is unmatched.

- (4) Third Stage: at time  $\beta_1$ , let  $a_i^{(u,v)}(t)$  for  $t \in [\beta_1, 1)$  be the following:  
if  $u$  is unmatched at time  $t_1$ , for every  $v, i$ ,

$$\begin{aligned} a_i^{(u,v)}(t) = & \rho_i^{(u,v)} + \sum_{u' \neq u} \mu_i^v(u' \rightarrow u) \cdot \left(1 - \exp\left(-\int_0^{\beta_1} A_i^{u'}(y) \, dy\right)\right) \\ & + \sum_{u' \in M(\beta_1)} \mu_i^v(u' \rightarrow u) \cdot \exp\left(-\int_0^{\beta_1} A_i^{u'}(y) \, dy\right), \end{aligned}$$

else, let  $a_i^{(u,v)}(t) = 0$  for every  $v, i$ .

- (5) If vertex  $v_i$  arrives at time  $t \in [\beta_1, 1]$  with type  $v \sim F_i$ :  
propose to a random offline vertex  $u \in U$  with probability

$$(3.4) \quad g_i^{(u,v)}(t) = a_i^{(u,v)}(t) \cdot \exp\left(-\int_0^t A_i^u(y) \, dy\right),$$

where  $A_i^u(t) \stackrel{\text{def}}{=} \mathbf{E}_v[2\rho_i^{(u,v)} - (2\rho_i^{(u,v)} - 1)^+] = \sum_v p_i^v \cdot (2\rho_i^{(u,v)} - (2\rho_i^{(u,v)} - 1)^+)$ .

Accept the edge  $(u, v_i = v)$  if  $u$  is unmatched.

Our algorithm consists of three stages, in which the first two stages are non-adaptive and are similar to the step activation rates algorithm for the prophet secretary problem; the third stage is adaptive to the randomness of the instance and our algorithm from the first two stages.

The parameters  $a_i^{(u,v)}(t)$  play a similar role as the activation rates for the prophet secretary problem. We first verify that the stated algorithm is well-defined. That is, the vector  $\{g_i^{(u,v)}(t)\}_u$  is a valid probability distribution for every  $t \in [0, 1]$ .

LEMMA 3.2. *For every  $v \in V, i \in [n], t \in [0, 1]$ , we have  $\sum_u g_i^{(u,v)}(t) \leq 1$ .*

*Proof.* For  $t \in [0, \beta_1)$ , we have

$$\sum_u g_i^{(u,v)}(t) \leq \sum_u a_i^{(u,v)}(t) \leq \sum_u \rho_i^{(u,v)} = 1.$$

For  $t \in [\beta_1, 1]$ , notice that  $a_i^{(u,v)}(t) = a_i^{(u,v)}(t_1)$  and  $g_i^{(u,v)}(t)$  is decreasing in  $t$ . It suffices to verify that  $\sum_u g_i^{(u,v)}(\beta_1) \leq 1$ . We redistribute the terms with factor  $\mu_i^v(u' \rightarrow u)$  in  $a_i^{(u,v)}(t)$  to  $u'$  and consider the

(redistributed) contribution of every  $u$  to the sum. If  $u \in M(\beta_1)$ , its contribution is:

$$\begin{aligned} & \rho_i^{(u,v)} \cdot \exp \left( - \int_0^{\beta_1} A_i^u(y) \, dy \right) \\ & \quad + \sum_{u' \neq u} \mu_i^v(u \rightarrow u') \cdot \left( 1 - \exp \left( - \int_0^{\beta_1} A_i^{u'}(y) \, dy \right) \right) \cdot \exp \left( - \int_0^{\beta_1} A_i^{u'}(y) \, dy \right) \\ & \leq \rho_i^{(u,v)} \cdot \exp \left( - \int_0^{\beta_1} A_i^u(y) \, dy \right) + \sum_{u' \neq u} \mu_i^v(u \rightarrow u') \cdot \left( 1 - \exp \left( - \int_0^{\beta_1} A_i^{u'}(y) \, dy \right) \right) \\ & \leq \rho_i^{(u,v)} \cdot \exp \left( - \int_0^{\beta_1} A_i^u(y) \, dy \right) + \rho_i^{(u,v)} \cdot \left( 1 - \exp \left( - \int_0^{\beta_1} A_i^u(y) \, dy \right) \right) = \rho_i^{(u,v)} , \end{aligned}$$

where the first inequality is by dropping exponent term; the second inequality is by Lemma 3.1. Else  $u \in N(\beta_1)$ , its contribution is:

$$\sum_{u' \neq u} \mu_i^v(u \rightarrow u') \cdot \exp \left( - \int_0^{\beta_1} A_i^{u'}(y) \, dy \right) \leq \sum_{u' \neq u} \mu_i^v(u \rightarrow u') \leq \rho_i^{(u,v)} .$$

Therefore, we have  $\sum_u g_i^{(u,v)}(\beta_1) \leq \sum_u \rho_i^{(u,v)} = 1$ .  $\square$

For every vertex  $u \in U$  and  $t \in [0, 1]$ , let  $A^u(t) \stackrel{\text{def}}{=} \sum_i A_i^u(t)$  and  $A_{-i}^u(t) \stackrel{\text{def}}{=} \sum_{j \neq i} A_j^u(t)$ .

LEMMA 3.3. *We have:*

$$A^u(t) = \begin{cases} h_u & t \in [0, \beta_0] ; \\ 1 & t \in (\beta_0, \beta_1] ; \\ 2 - h_u & t \in (\beta_1, 1] . \end{cases}$$

*Proof.* For  $t \in [0, \beta_0]$ , we have  $A^u(t) = \sum_i A_i^u(t) = \sum_i \mathbf{E}_v[(2\rho_i^{(u,v)} - 1)^+] = h_u$ , where the last equality holds by the definition of  $h_u$ . For  $t \in (\beta_0, \beta_1]$ , we have  $A^u(t) = \sum_i \mathbf{E}_v[\rho_i^{(u,v)}] = \sum_i \sum_v p_i^v \cdot \rho_i^{(u,v)} = \sum_{i,v} x_i^{(u,v)} = 1$ . For  $t \in (\beta_1, 1]$ , we have  $A^u(t) = \sum_i \mathbf{E}_v[2\rho_i^{(u,v)} - (2\rho_i^{(u,v)} - 1)^+] = 2 - h_u$ .  $\square$

As a consequence, we define and calculate the following functions:

- $c_1(h_u) \stackrel{\text{def}}{=} \frac{1 - e^{-h_u \beta_0}}{h_u} = \int_0^{\beta_0} \exp \left( - \int_0^t A^u(y) \, dy \right) \, dt ;$
- $c_2(h_u) \stackrel{\text{def}}{=} e^{-h_u \beta_0} (1 - e^{-(\beta_1 - \beta_0)}) = \int_{\beta_0}^{\beta_1} \exp \left( - \int_0^t A^u(y) \, dy \right) \, dt ;$
- $c_3(h_u) \stackrel{\text{def}}{=} e^{-h_u \beta_0 - (\beta_1 - \beta_0)} \cdot \frac{1 - e^{-(2 - h_u)(1 - \beta_1)}}{2 - h_u} = \int_{\beta_1}^1 \exp \left( - \int_0^t A^u(y) \, dy \right) \, dt .$

THEOREM 3.3. *For every  $u \in U, v \in V, i \in [n]$ , Multistage Proposal-based Matching matches edge  $(u, v_i = v)$  with probability  $\Gamma_{MAM}(x^u) \cdot x_i^{(u,v)}$ , where*

$$\Gamma_{MAM}(x) \stackrel{\text{def}}{=} \min \left( c_1(h_2(x)) + c_2(h_2(x)) + c_3(h_2(x)), c_2(h_2(x)) + \left( 2 - e^{-(\beta_1 - \beta_0)} \right) \cdot c_3(h_2(x)) \right) .$$

As an immediate corollary of the theorem, in the infinitesimal case when  $\max_u x_u = o(1)$ , our algorithm achieves a competitive ratio of  $\Gamma_{MAM}(0^+) = c_2(1 - \ln 2) + (2 - e^{-(\beta_1 - \beta_0)})c_3(1 - \ln 2) \approx 0.645$ . Note that we intentionally use the same parameters  $\beta_0, \beta_1$  as Yan [40] so that our competitive ratio is exactly the same as Yan's ratio. Moreover, the competitive ratio of the algorithm is at least  $\Gamma_{MAM}(1) = 1 - 1/e$  for all instances.



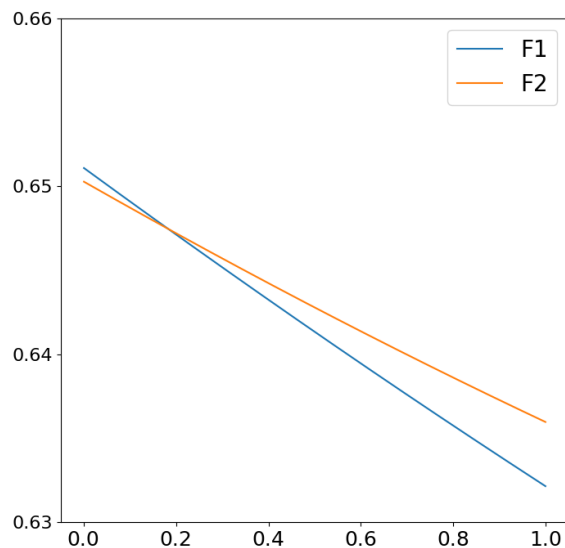


Figure 4: A Plot of  $F_1(h) = c_1(h) + c_2(h) + c_3(h)$  and  $F_2(h) = c_2(h) + (2 - e^{-(\beta_1 - \beta_0)})c_3(h)$ .

**3.4.1 Proof of Theorem 3.3** From now on, we fix arbitrary  $u \in U, v \in V, i \in [n]$  and study the probability that  $(u, v_i = v)$  is matched by our algorithm. We shall prove that

$$(3.5) \quad \Pr[(u, v_i = v) \text{ is matched}] \geq \min\left(c_1(h_u) + c_2(h_u) + c_3(h_u), c_2(h_u) + \left(2 - e^{-(\beta_1 - \beta_0)}\right) \cdot c_3(h_u)\right) \cdot x_i^{(u,v)}.$$

This inequality together with the following lemma and that  $h_u \leq h_2(x^u)$  would conclude the statement of the theorem.

LEMMA 3.4. *The function  $(c_1(h) + c_2(h) + c_3(h), c_2(h) + (2 - e^{-(\beta_1 - \beta_0)}) \cdot c_3(h))$  is decreasing for  $h \in [0, 1]$ .*

We omit the tedious formal proof of this lemma and provide a plot of the function. Refer to Figure 4.

The next lemma is crucial for proving Equation (3.5), that lower bounds the probability of  $u$  remains unmatched on the arrival time  $t_i = t$  of vertex  $v_i$ .

LEMMA 3.5. *For every  $t \in [0, \beta_1)$  and  $u \in U$ ,*

$$\Pr[u \in N(t) \mid t_i = t] = \exp\left(-\int_0^t A_{-i}^u(y) dy\right).$$

*For every  $t \in [\beta_1, 1]$  and  $u \neq u' \in U$ ,*

$$\begin{aligned} \Pr[u \in N(t) \mid t_i = t] &\geq \exp\left(-\int_0^t A_{-i}^u(y) dy\right); \\ \Pr[u \in N(t), u' \in M(\beta_1) \mid t_i = t] &\geq \exp\left(-\int_0^t A_{-i}^u(y) dy\right) \left(1 - \exp\left(-\int_0^{t_1} A_{-i}^{u'}(y) dy\right)\right) \end{aligned}$$

*Proof.* For every  $t \in [0, \beta_1)$  and  $u \in U$ , the statement and its proof is essentially the same of Lemma 2.3 for the prophet secretary problem. Nevertheless, we include a proof for completeness.

$$\begin{aligned}
\Pr[u \in N(t) \mid t_i = t] &= \prod_{j \neq i} \Pr[j \text{ doesn't propose } u \text{ before } t] = \prod_{j \neq i} \left( 1 - \sum_v p_i^v \int_0^t g_i^{(u,v)}(y) dy \right) \\
&= \prod_{j \neq i} \left( 1 - \int_0^t A_j^u(y) \cdot \exp \left( - \int_0^y A_j^u(z) dz \right) dy \right) \\
&= \prod_{j \neq i} \exp \left( - \int_0^t A_{\bullet}^u(y) dy \right) = \exp \left( - \int_0^t A_{-i}^u(y) dy \right).
\end{aligned}$$

For any  $t \in [\beta_1, 1]$  and  $u \neq u' \in U$ , conditioned on box  $i$  arrive exactly at  $t$ , we consider following hypothetical algorithm for other boxes and for time range  $[1, t]$  with activation probability  $\hat{g}_j^{(\cdot, v)}(\bullet)$ . It use the same activation method before  $t_1$ , which means  $\hat{g}_j^{(\cdot, v)}(\bullet) = g_j^{(\cdot, v)}(\bullet)$ , while for time  $\bullet$  after  $t_1$ , the algorithm proposed to  $u$  with probability  $\hat{g}_j^{(u, v)}(\bullet) = (2\rho_j^{(u, v)} - (2\rho_j^{(u, v)} - 1)^+) \cdot \exp \left( - \int_0^{\bullet} A_j^u(y) dy \right)$ , and do not propose to other vertices.

Since the activation probability for vertex  $u$  of original algorithm is at most the probability in this algorithm and the activation probability for vertex  $u'$  is unchanged, the original probability of the event we concern is no smaller than the the probability of the same event while in this hypothetical algorithm. Following we work on the hypothetical algorithm, and lower bound the probability of the event.

An observation is that for any  $j \neq i$ :

$$\sum_v p_j^v \hat{g}_j^{(u, v)}(t) = A_j^u(t) \cdot \exp \left( - \int_0^t A_{\bullet}^u(y) dy \right).$$

Hence, the probability of  $u$  is unmatched before  $t$  is:

$$\begin{aligned}
\prod_{j \neq i} \Pr[j \text{ doesn't propose } u \text{ before } t] &= \prod_{j \neq i} \left( 1 - \sum_{\bullet} p_{\bullet}^v \int_0^t \hat{g}_{\bullet}^{(u, v)}(y) dy \right) \\
&= \prod_{j \neq i} \exp \left( - \int_0^t A_j^u(y) dy \right).
\end{aligned}$$

The probability of  $u$  is unmatched before  $t$  and  $u'$  is unmatched before  $t_1$  is:

$$\begin{aligned}
&\prod_{j \neq i} \Pr[j \text{ doesn't propose } u^{\bullet} \text{ before } \bullet_1 \text{ and } u' \text{ before } t] \\
&= \prod_{j \neq i} \left( 1 - \sum_v p_i^v \int_0^t \hat{g}_{\bullet}^{(u, v)}(y) dy - \sum_v p_{\bullet}^v \int_0^{t_1} g_{\bullet}^{(u', v)}(y) dy \right) \\
&\leq \prod_{j \neq i} \left( 1 - \sum_v p_{\bullet}^v \int_0^t \hat{g}_{\bullet}^{(u, v)}(y) dy \right) \left( 1 - \sum_v p_{\bullet}^v \int_0^{t_1} g_{\bullet}^{(u', v)}(y) dy \right) \\
&= \prod_{j \neq i} \exp \left( - \int_0^t A_{\bullet}^u(y) dy \right) \cdot \prod_{j \neq i} \exp \left( - \int_0^{t_1} A_j^{u'}(y) dy \right).
\end{aligned}$$

The probability of  $u$  is unmatched before  $t$  and  $u'$  is matched before  $t_1$  is at least the difference of above two terms, as we desired.  $\square$

We first study the probability of matching  $(u, v_i = v)$  in the first two stages of our algorithm, i.e., before time  $\beta_1$ .

LEMMA 3.6. *The probability of  $(u, v_i = v)$  being matched before time  $\beta_1$  equals:*

$$(3.6) \quad (2x_i^{(u, v)} - p_i^v)^+ \cdot c_1(h_u) + x_i^{(u, v)} \cdot c_2(h_u)$$

*Proof.* The probability of edge  $(u, v_i = v)$  being matched in time  $[0, \beta_1)$  is that

$$\begin{aligned} p_i^v \cdot \int_0^{\beta_1} g_i^{(u,v)}(t) \cdot \mathbf{Pr}[u \in N(t) \mid t_i = t] \, dt &= p_i^v \cdot \int_0^{\beta_1} g_i^{(u,v)}(t) \cdot \exp\left(-\int_0^t A_i^u(y) \, dy\right) \, dt \\ &= p_i^v \cdot \int_0^{\beta_1} a_i^{(u,v)}(t) \cdot \exp\left(-\int_0^t A_i^u(y) \, dy\right) \, dt = (2x_i^{(u,v)} - p_i^v)^+ \cdot c_1(h_u) + x_i^{(u,v)} \cdot c_2(h_u) . \end{aligned}$$

Here, the first equality is by Lemma 3.5; the second and third equalities come from the construction of  $a_i^{(u,v)}(t)$  and  $c_1(h_u), c_2(h_u)$ .  $\square$

Next, we study the probability of matching  $(u, v_i = v)$  in the third stage of our algorithm, i.e., after time  $\beta_1$ .

LEMMA 3.7. *The probability of  $(u, v_i = v)$  being matched between time  $[\beta_1, 1]$  is at least:*

$$\left(x_i^{(u,v)} + (x_i^{(u,v)} - (2x_i^{(u,v)} - p_i^v)^+ ) \cdot (1 - e^{-(\beta_1 - \beta_0)})\right) \cdot c_3(h_u) .$$

*Proof.* The probability of the edge being matched in the third stage  $[\beta_1, 1]$  equals the following:

$$\begin{aligned} (3.7) \quad \mathbf{E} \left[ \int_{\beta_1}^1 p_i^v \cdot g_i^{(u,v)}(t) \cdot \mathbf{1}[u \in N(t)] \, dt \right] \\ &= p_i^v \cdot \mathbf{E} \left[ \int_{\beta_1}^1 a_i^{(u,v)}(\beta_1) \cdot \mathbf{1}[u \in N(t)] \cdot \exp\left(-\int_0^t A_i^u(y) \, dy\right) \, dt \right] \\ &= p_i^v \cdot \int_{\beta_1}^1 \mathbf{E} \left[ a_i^{(u,v)}(\beta_1) \cdot \mathbf{1}[u \in N(t)] \right] \cdot \exp\left(-\int_0^t A_i^u(y) \, dy\right) \, dt , \end{aligned}$$

where the expectations are taken over both the randomness from the instance (i.e., the realization of the values and the arrival times except for vertex  $v_i$ ) and the randomness from our algorithm; the second equation follows from the fact that  $A_i^u(y)$  are prefixed constants.

Further, we have that

$$\begin{aligned} \mathbf{E} \left[ a_i^{(u,v)}(\beta_1) \cdot \mathbf{1}[u \in N(t)] \right] &= \mathbf{E} \left[ \rho_i^{(u,v)} \cdot \mathbf{1}[u \in N(t)] \right] \\ &+ \sum_{u' \neq u} \mathbf{E} \left[ \mu_i^v(u' \rightarrow u) \cdot \left(1 - \exp\left(-\int_0^{\beta_1} A_i^{u'}(y) \, dy\right)\right) \cdot \mathbf{1}[u \in N(t)] \right] \\ &+ \sum_{u' \neq u} \mathbf{E} \left[ \mu_i^v(u' \rightarrow u) \cdot \exp\left(-\int_0^{\beta_1} A_i^{u'}(y) \, dy\right) \cdot \mathbf{1}[u \in N(t), u' \in M(\beta_1)] \right] \\ &\geq \exp\left(-\int_0^t A_i^u(y) \, dy\right) \cdot \left( \rho_i^{(u,v)} + \sum_{u' \neq u} \mu_i^v(u' \rightarrow u) \cdot \left(1 - \exp\left(-\int_0^{\beta_1} A_i^{u'}(y) \, dy\right)\right) \right) \\ &\geq \exp\left(-\int_0^t A_i^u(y) \, dy\right) \cdot \left( \rho_i^{(u,v)} + \sum_{u' \neq u} \mu_i^v(u' \rightarrow u) \cdot (1 - e^{-(\beta_1 - \beta_0)}) \right) \\ &= \exp\left(-\int_0^t A_i^u(y) \, dy\right) \cdot \left( \rho_i^{(u,v)} + (\rho_i^{(u,v)} - (2\rho_i^{(u,v)} - 1)^+) \cdot (1 - e^{-(\beta_1 - \beta_0)}) \right) , \end{aligned}$$

where the first inequality follows from Lemma 3.5; the second inequality follows from the fact that  $\exp\left(-\int_0^{\beta_1} A_i^{u'}(y) \, dy\right) \leq \exp\left(-\int_{\beta_0}^{\beta_1} 1 \, dy\right) = e^{-(\beta_1 - \beta_0)}$ . Finally, we have that

$$\begin{aligned} (3.7) &\geq p_i^v \cdot \left( \rho_i^{(u,v)} + (\rho_i^{(u,v)} - (2\rho_i^{(u,v)} - 1)^+) \cdot (1 - e^{-(\beta_1 - \beta_0)}) \right) \cdot \int_{\beta_1}^1 \exp\left(-\int_0^t A_i^u(y) \, dy\right) \, dy \\ &= \left( x_i^{(u,v)} + (x_i^{(u,v)} - (2x_i^{(u,v)} - p_i^v)^+) \cdot (1 - e^{-(\beta_1 - \beta_0)}) \right) \cdot c_3(h_u) . \end{aligned}$$

$\square$

Putting the two lemmas together, we have that

$$\begin{aligned} \Pr[(u, v_i = v) \text{ is matched}] &\geq (2x_i^{(u,v)} - p_i^v)^+ \cdot c_1(h_u) + x_i^{(u,v)} \cdot c_2(h_u) \\ &\quad + \left( x_i^{(u,v)} + (x_i^{(u,v)} - (2x_i^{(u,v)} - p_i^v)^+ ) \cdot \left( 1 - e^{-(\beta_1 - \beta_0)} \right) \right) \cdot c_3(h_u) \\ &\geq \min \left( c_1(h_u) + c_2(h_u) + c_3(h_u), c_2(h_u) + \left( 2 - e^{-(\beta_1 - \beta_0)} \right) \cdot c_3(h_u) \right) \cdot x_i^{(u,v)}, \end{aligned}$$

where the last inequality holds by the fact that  $x_i^{(u,v)} \geq (2x_i^{(u,v)} - p_i^v)^+$ .

**3.5 Constant Activation Rate Except One Large Vertex** The multistage activation-based matching algorithm achieves an improved competitive ratio over  $1 - 1/e$  on instances with small  $x^u$ 's, we are left to design an algorithm working against large  $x^u$ 's, which is relatively the easier task. We adapt the constant activation rate algorithm for the prophet secretary problem to each offline vertex  $u$ , and at the same time apply a special treatment to the most important online neighbor of  $u$ . Our algorithm is as the following.

#### Constant Activation Rate Except One Large Vertex

Parameters: for every  $x \in [0, 1]$ , let  $\alpha(x)$  be the unique solution to the following equation:

$$\int_{\alpha(x)}^1 e^{-t(1-x)} dt = \int_0^1 e^{-t(1-x)} \cdot (1 - (t - \alpha(x))^+ \cdot x) dt.$$

(1) For every  $u \in U$ , let  $i_u \stackrel{\text{def}}{=} \operatorname{argmax}_i x_i^u$  be the most important online vertex of  $u$ .

(2) For time  $t$  from 0 to 1:

- (a) If vertex  $v_i$  arrives with type  $v \sim F_i$ :  
propose to a random offline vertex  $u \in U$  with probability  $\rho_i^{(u,v)}$ .
- (b) If vertex  $u$  receives a proposal from vertex  $v_i = v$ :
  - if  $i = i_u$ , activate the edge iff  $t \geq \alpha(x_u)$ ;
  - if  $i \neq i_u$ , activate the edge with probability  $e^{-t \cdot x_i^u}$ .

Accept the first active edge that  $u$  receives.

**THEOREM 3.4.** *For every  $u \in U, v \in V, i \in [n]$ , the Constant Activation Rate Except One Large Vertex algorithm matches edge  $(u, v_i = v)$  with probability  $\Gamma_{CAR}(x^u) \cdot x_i^{(u,v)}$ , where*

$$\Gamma_{CAR}(x) \stackrel{\text{def}}{=} \int_{\alpha(x)}^1 e^{-t(1-x)} dt$$

By straightforward calculation,  $\Gamma_{CAR}(0) = 1 - 1/e$  and  $\Gamma_{CAR}(1) = \sqrt{3} - 1$ . As an implication, this algorithm achieves a competitive ratio better than  $1 - 1/e$  if every  $x^u$  is bounded away from 0.

**3.5.1 Proof of Theorem 3.4** Fix an arbitrary  $u \in U$ . According to the definition of our algorithm, the probability that the edge  $(u, v_j)$  is not active before time  $t$ :

$$(3.8) \quad 1 - \int_0^t \sum_v \left( p_j^v \cdot \rho_j^{(u,v)} \cdot e^{-yx_j^u} \right) dy = 1 - \int_0^t x_j^u \cdot e^{-yx_j^u} dy = e^{-tx_j^u}, \quad \forall j \neq i_u;$$

$$(3.9) \quad 1 - \int_0^t \mathbb{1}[y \geq \alpha(x^u)] \cdot \sum_v p_j^v \cdot \rho_j^{(u,v)} dy = 1 - (t - \alpha(x^u))^+ \cdot x^u, \quad j = i_u;$$

Next, we consider probability that edge  $(u, v_i = v)$  being matched.

**Case 1.** ( $i = i_u$ ) The edge  $(u, v_i = v)$  is matched with probability:

$$\int_{\alpha(x^u)}^1 p_i^v \cdot \rho_i^{(u,v)} \cdot \Pr[u \in N(t) \mid t_i = t] \, dt = x_i^{(u,v)} \cdot \int_{\alpha(x^u)}^1 \prod_{j \neq i} e^{-tx_j^u} \, dt \geq x_i^{(u,v)} \cdot \int_{\alpha(x^u)}^1 e^{-t(1-x^u)} \, dt ,$$

where the equality is by (3.8) and the inequality holds by the fact that  $\sum_{j \neq i} x_j^u \leq 1 - x_i^u = 1 - x^u$ .

**Case 2.** ( $i \neq i_u$ ) The edge  $(u, v_i = v)$  is matched with probability:

$$\begin{aligned} \int_0^1 p_i^v \cdot \rho_i^{(u,v)} \cdot \Pr[u \in N(t) \mid t_i = t] \, dt &= x_i^{(u,v)} \cdot \int_0^1 \prod_{j \neq i, i_u} e^{-tx_j^u} \cdot (1 - (t - \alpha(x^u))^+ \cdot x^u) \, dt \\ &\geq x_i^{(u,v)} \cdot \int_0^1 e^{-t(1-x^u)} \cdot (1 - (t - \alpha(x^u))^+ \cdot x^u) \, dt = x_i^{(u,v)} \cdot \int_{\alpha(x^u)}^1 e^{-t(1-x^u)} \, dt \end{aligned}$$

where the first equality is by (3.8) and (3.9), and the last equality is by the definition of  $\alpha(\cdot)$ .

This concludes the proof of the theorem.

## References

- [1] Melika Abolhassani, Soheil Ehsani, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Robert D. Kleinberg, and Brendan Lucier. Beating 1-1/e for ordered prophets. In *STOC*, pages 61–71. ACM, 2017.
- [2] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *SODA*, pages 1253–1264. SIAM, 2011.
- [3] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Prophet secretary: Surpassing the 1-1/e barrier. In *EC*, pages 303–318. ACM, 2018.
- [4] Bahman Bahmani and Michael Kapralov. Improved bounds for online stochastic matching. In *ESA (1)*, volume 6346 of *Lecture Notes in Computer Science*, pages 170–181. Springer, 2010.
- [5] Hedyeh Beyhaghi, Negin Golrezaei, Renato Paes Leme, Martin Pál, and Balasubramanian Sivan. Improved revenue bounds for posted-price and second-price mechanisms. *Oper. Res.*, 69(6):1805–1822, 2021.
- [6] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Online stochastic matching: New algorithms and bounds. *Algorithmica*, 82(10):2737–2783, 2020.
- [7] Archit Bubna and Ashish Chiplunkar. Prophet inequality: Order selection beats random order. In *EC*, pages 302–336. ACM, 2023.
- [8] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC*, pages 311–320. ACM, 2010.
- [9] José R. Correa, Raimundo Saona, and Bruno Ziliotto. Prophet Secretary Through Blind Strategies. *Mathematical Programming*, 190(1):483–521, 2021. doi: 10.1007/S10107-020-01544-8.
- [10] José R. Correa, Raimundo Saona, and Bruno Ziliotto. Prophet secretary through blind strategies. *Math. Program.*, 190(1):483–521, 2021.
- [11] Kevin P Costello, Prasad Tetali, and Pushkar Tripathi. Stochastic matching with commitment. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming*, pages 822–833. Springer, 2012.
- [12] Mahsa Derakhshan and Alireza Farhadi. Beating  $(1 - 1/e)$ -approximation for weighted stochastic matching. In *SODA*, pages 1931–1961. SIAM, 2023.
- [13] Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput.*, 49(3):540–582, 2020.

- [14] Paul Dütting, Evangelia Gergatsouli, Rojin Rezvan, Yifeng Teng, and Alexandros Tsigonias-Dimitriadis. Prophet Secretary Against the Online Optimal. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC '23, pages 561–581, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 9798400701047. doi: 10.1145/3580507.3597736.
- [15] Soheil Ehsani, MohammadTaghi Hajiaghayi, Thomas Kesselheim, and Sahil Singla. Prophet secretary for combinatorial auctions and matroids. In *SODA*, pages 700–714. SIAM, 2018.
- [16] Hossein Esfandiari, MohammadTaghi Hajiaghayi, Vahid Liaghat, and Morteza Monemizadeh. Prophet secretary. In *ESA*, volume 9294 of *Lecture Notes in Computer Science*, pages 496–508. Springer, 2015.
- [17] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Prophet matching with general arrivals. *Math. Oper. Res.*, 47(2):878–898, 2022.
- [18] Jon Feldman, Aranyak Mehta, Vahab S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *FOCS*, pages 117–126. IEEE Computer Society, 2009.
- [19] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *SODA*, pages 123–135. SIAM, 2015.
- [20] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes with applications to bayesian selection problems. *SIAM J. Comput.*, 50(2):255–300, 2021.
- [21] Hu Fu, Zhihao Gavin Tang, Hongxun Wu, Jinzhao Wu, and Qianfan Zhang. Random order vertex arrival contention resolution schemes for matching, with applications. In *ICALP*, volume 198 of *LIPIcs*, pages 68:1–68:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [22] Buddhima Gamlath, Sagar Kale, and Ola Svensson. Beating greedy for stochastic bipartite matching. In *SODA*, pages 2841–2854. SIAM, 2019.
- [23] Giordano Giambartolomei, Frederik Mallmann-Trenn, and Raimundo Saona. Prophet inequalities: Separating random order from order selection. *CoRR*, abs/2304.04024, 2023.
- [24] Bernhard Haeupler, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*, volume 7090 of *Lecture Notes in Computer Science*, pages 170–181. Springer, 2011.
- [25] Elfarouk Harb. Fishing for better constants: The prophet secretary via poissonization. *CoRR*, abs/2307.00971, 2023.
- [26] Elfarouk Harb. Fishing For Better Constants: The Prophet Secretary Via Poissonization, November 2023.
- [27] Zhiyi Huang and Xinkai Shu. Online Stochastic Matching, Poisson Arrivals, and the Natural Linear Program. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 682–693. ACM, 2021. doi: 10.1145/3406325.3451079.
- [28] Patrick Jaillet and Xin Lu. Online stochastic matching: New algorithms with better bounds. *Math. Oper. Res.*, 39(3):624–646, 2014.
- [29] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358. ACM, 1990.
- [30] Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games Econ. Behav.*, 113:97–115, 2019.
- [31] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bulletin of the American Mathematical Society*, 83(4):745–747, 1977.

- [32] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. *Probability on Banach spaces*, 4:197–266, 1978.
- [33] Euiwoong Lee and Sahil Singla. Optimal online contention resolution schemes via ex-ante prophet inequalities. In *ESA*, volume 112 of *LIPIcs*, pages 57:1–57:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [34] Allen Liu, Renato Paes Leme, Martin Pál, Jon Schneider, and Balasubramanian Sivan. Variable decomposition for prophet inequalities and optimal ordering. In *EC*, page 692. ACM, 2021.
- [35] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Math. Oper. Res.*, 37(4):559–573, 2012.
- [36] Bo Peng and Zhihao Gavin Tang. Order selection prophet inequality: From threshold optimization to arrival time design. In *FOCS*, pages 171–178. IEEE, 2022.
- [37] Guoliang Qiu, Yilong Feng, Shengwei Zhou, and Xiaowei Wu. Improved competitive ratio for edge-weighted online stochastic matching. In *WINE*, volume 14413 of *Lecture Notes in Computer Science*, pages 527–544. Springer, 2023.
- [38] Aviad Rubinstein. Beyond matroids: secretary problem and prophet inequality with general constraints. In *STOC*, pages 324–332. ACM, 2016.
- [39] Zhihao Gavin Tang, Jinzhao Wu, and Hongxun Wu. (fractional) online stochastic matching via fine-grained offline statistics. In *STOC*, pages 77–90. ACM, 2022.
- [40] Shuyi Yan. Edge-weighted online stochastic matching: Beating  $1-1/e$ . In *SODA*. SIAM, 2024.

## A Reduction from Query-Commit to Prophet Secretary Matching

First of all, notice that the stochastic model of the underlying graph of query-commit is a special case of the model of prophet secretary matching. Indeed, the prophet secretary problem only assumes independent realizations of vertices, while query-commit further assumes independent realizations of edges.

Therefore, we can think of an arbitrary instance for query-commit to be an instance for prophet secretary by artificially separating the two sides of the vertices to be online and offline. Say  $V$  is the set of online vertices and  $U$  is the set of offline vertices. In order to apply an online algorithm  $A_{PSM}$  of the prophet secretary matching problem for the instance, we draw a uniform at random arrival time  $t_v$  for each vertex  $v \in V$  on our own to simulate the random arrival of the vertices. Then, the difficulty is that we are not able to see the realization of  $v$ 's type before we query the existence of an edge.

Nevertheless, on the arrival of an online vertex  $v$ , we first compute the expected matching probability  $x_v^u$  of vertex  $v$  to any offline vertex  $u$ , according to algorithm  $A_{PSM}$  in an imaginary run of the prophet secretary problem. The vector  $(x_v^u)_{u \in U}$  then must be in the following polytope:

$$\begin{aligned} \sum_{u \in S} x_v^u &\leq 1 - \prod_{u \in S} (1 - p_{(v,u)}), & \forall S \subseteq U, \\ x_v^u &\geq 0, & \forall u \in U, \end{aligned}$$

where the first inequality states that the probability of  $v$  being matched to some  $u \in S$  is no larger than the probability that at least one of the edges  $\{(v, u)\}_{u \in S}$  exists. We import the following lemma by Gamlath et al. [22] which characterizes the extreme points of this polytope.

LEMMA A.1. (LEMMA 5 OF [22] REPHRASED) *Every vertex of above polytope corresponds to a permutation of a subset of the offline vertices  $u_1, u_2, \dots, u_\ell$  such that:*

$$\begin{aligned} x_v^{u_1} &= p_{(v,u_1)} \\ x_v^{u_2} &= (1 - p_{(v,u_1)}) p_{(v,u_2)} \\ &\dots \\ x_v^{u_\ell} &= (1 - p_{(v,u_1)}) \dots (1 - p_{(v,u_{\ell-1})}) p_{(v,u_\ell)} \\ x_v^u &= 0 \quad \text{if } u \neq u_1, u_2, \dots, u_\ell \end{aligned}$$

Notice that querying the vertices in the order of  $u_1, u_2, \dots, u_\ell$ , the vector of the expected probability exactly equals to the extreme point showed above. Thus, we can match the expected probability  $(x_v^u)_{u \in U}$  in the query-commit model through the following: 1) decompose  $(x_v^u)_{u \in U}$  as a convex combination of the extreme points and sample from the extreme points, 2) then query the offline vertices in the order with respect to the sampled extreme point. Through this simulation, our algorithm matches the same performance of  $A_{PSM}$ . Finally notice that the benchmark (expected offline optimum matching) for the two problems are also the same.

## B Proof of Lemma 2.9

We study the following optimization problem and define  $h_s(x)$  to be the optimal value:

$$\begin{aligned} \max_{\{x_i^v\}, \{p_i^v\}} : & \quad \frac{1}{s-1} \cdot \sum_{i,v} (sx_i^v - p_i^v)^+ \\ \text{subject to :} & \quad \sum_i \sum_{v \in V_i} x_i^v \leq 1 - \prod_i \left(1 - \sum_{v \in V_i} p_i^v\right) \quad \forall V_i \subseteq V \\ & \quad \sum_v x_i^v \leq x \quad \forall i \in [n] \end{aligned}$$

First, we reduce the dimension of the optimization by showing that it is without loss of generality to study when  $|V| = 1$ . Specifically, we show that  $h_s(x)$  equals to the value of the following program:

$$\begin{aligned} \max_{\{x_i\}, \{p_i\}} : & \quad \frac{1}{s-1} \cdot \sum_i (sx_i - p_i) \\ \text{subject to :} & \quad \sum_{i \in S} x_i \leq 1 - \prod_{i \in S} (1 - p_i) \quad \forall S \subseteq [n] \\ & \quad x_i \leq x \quad \forall i \in [n] \end{aligned}$$

It is obvious that this program is a special case of the original one. On the other hand, given an arbitrary feasible solution  $\{x_i^v\}, \{p_i^v\}$  of the original program. Consider the following vectors:

$$x_i = \sum_{v: sx_i^v - p_i^v \geq 0} x_i^v \quad \text{and} \quad p_i = \sum_{v: sx_i^v - p_i^v \geq 0} p_i^v, \quad \forall i \in [n]$$

It is straightforward to verify that the constraints of the new program are satisfied, and the objective value equals to the objective of the original program.

Next, we fix the vector  $\{x_i\}$  and consider the program as an optimization over  $\alpha_i = \ln(1 - p_i)$ . The constraints can be rewritten as the following:

$$(B.1) \quad \sum_{i \in S} \alpha_i \leq \ln \left(1 - \sum_{i \in S} x_i\right), \quad \forall S \subseteq [n]$$

And the goal is now to minimize  $\sum_i 1 - e^{\alpha_i}$ , which is a concave function of  $\{\alpha_i\}$ . Thus, it attains its minimum value at the extreme point of above polytope (B.1).

Let  $\{\alpha_i\}$  be an extreme point. Then, there are  $n$  tight constraints, corresponding to  $n$  sets  $S_1, \dots, S_n$ . We claim that the  $n$  sets must form a chain

$$S_1 \subset S_2 \subset \dots \subset S_n.$$

Otherwise suppose there are two sets  $S, T$  with  $S \not\subset T$  and  $T \not\subset S$  while the two corresponding constraints are tight. Then we have

$$\begin{aligned} & \ln \left(1 - \sum_{i \in S} x_i\right) + \ln \left(1 - \sum_{i \in T} x_i\right) = \sum_{i \in S} \alpha_i + \sum_{i \in T} \alpha_i \\ & = \sum_{i \in S \cap T} \alpha_i + \sum_{i \in S \cup T} \alpha_i \leq \ln \left(1 - \sum_{i \in S \cap T} x_i\right) + \ln \left(1 - \sum_{i \in S \cup T} x_i\right), \end{aligned}$$



a contradiction due to the concavity of  $\ln(1-x)$ . Furthermore, since  $|S_n| \leq n$ , we must have that  $|S_i| = i$  for  $i \in [n]$ . Without loss of generality, we reorder the indices and assume that  $S_i = [i]$ . According to those tight constraints, we have

$$\alpha_i = \ln \left( \frac{1 - \sum_{j \leq i} x_j}{1 - \sum_{j < i} x_j} \right) \quad \text{and} \quad p_i = \frac{x_i}{1 - \sum_{j < i} x_j}.$$

We simplify the program as the following.

$$\begin{aligned} \max_{\{x_i\}} : \quad & \frac{1}{s-1} \cdot \sum_i \left( s x_i - \frac{x_i}{1 - \sum_{j < i} x_j} \right) \\ \text{subject to :} \quad & \sum_i x_i \leq 1 \\ & x_i \leq x \quad \forall i \in [n] \end{aligned}$$

Let  $\{x_i\}$  be the optimal solution. We prove that  $x_i$ 's are non-decreasing by contradiction. Suppose  $x_i > x_{i+1}$ . Consider switching the order of them, the objective function strictly improves since

$$\begin{aligned} & \frac{x_i}{1-y} + \frac{x_{i+1}}{1-y-x_i} > \frac{x_{i+1}}{1-y} + \frac{x_i}{1-y-x_{i+1}} \\ \Leftrightarrow & \frac{x_i - x_{i+1}}{1-y} \geq \frac{(1-y-x_i)x_i - (1-y-x_{i+1})x_{i+1}}{(1-y-x_i)(1-y-x_{i+1})} \\ \Leftrightarrow & \frac{1}{1-y} > \frac{1-y-x_i-x_{i+1}}{(1-y-x_i)(1-y-x_{i+1})}, \end{aligned}$$

which holds for every  $x_i > x_{i+1}$  and  $y = \sum_{j < i} x_j$ . Thus, we have  $x_1 \leq x_2 \leq \dots \leq x_n$ . Finally, we prove that there is at most one  $x_i$  in the open interval  $(0, x)$  by contradiction. Suppose  $0 < x_i \leq x_{i+1} < x$ . Consider changing  $(x_i, x_{i+1})$  to  $(x_i - \epsilon, x_{i+1} + \epsilon)$ , the objective strictly improves since

$$\begin{aligned} & \frac{\partial}{\partial \epsilon} \left( \frac{x_i - \epsilon}{1-y} + \frac{x_{i+1} + \epsilon}{1-y-x_i + \epsilon} \right) < 0 \\ \Leftrightarrow & \frac{1-y-x_i-x_{i+1}}{(1-y-x_i+\epsilon)^2} < \frac{1}{1-y} \\ \Leftrightarrow & \frac{1-y-x_i-x_{i+1}}{(1-y-x_i)^2} < \frac{1}{1-y}, \end{aligned}$$

which holds for every  $x_i \leq x_{i+1}$  and  $y = \sum_{j < i} x_j$ . To sum up, the optimal solution has the form of  $0 \leq x_1 < x_2 = x_3 = \dots = x_n = x$  and the objective is

$$x_1 + \frac{1}{s-1} \cdot \sum_{k=0}^{n-1} \left( s x - \frac{x}{1-x_1-kx} \right) \leq \max_{t \in [0, x)} t + \frac{1}{s-1} \sum_{k \geq 0} \left( s x - \frac{x}{1-t-kx} \right)^+,$$

that concludes the proof of the lemma.

**B.1 Some properties of  $h_s(x)$**  As promised in Lemma 2.8, we verify that  $h_2(0^+) = 1 - \ln 2$ .

LEMMA B.1. For any  $s > 1$ ,  $\lim_{x \rightarrow 0^+} h_s(x) = 1 - \frac{\ln s}{s-1}$ .

*Proof.* Notice that  $h_s(x)$  is a Riemann sum for the integral of function  $f_s(y) = \frac{1}{s-1} \left( s - \frac{1}{1-y} \right)^+$  on  $[0, 1 - \frac{1}{s}]$ , with intervals smaller than  $x$ . Hence

$$\lim_{x \rightarrow 0^+} h_s(x) = \frac{1}{s-1} \cdot \int_0^{1-\frac{1}{s}} \left( s - \frac{1}{1-y} \right) dy = 1 - \frac{\ln s}{s-1}.$$

□

LEMMA B.2. For any  $s > 1$  and any  $x \in [0, 1]$ ,  $h_s(x)$  is right continuous.

Actually,  $h_s(x)$  is also left continuous, but we will conclude a stronger property for the left side in the next lemma.

*Proof.* For any  $x \in (0, 1)$ , and any  $x' > x$ , suppose that

$$h_s(x') = t + \frac{1}{s-1} \sum_{k \in \mathbb{N}, kx' + t \leq 1-1/s} \left( sx' - \frac{x'}{1-t-kx'} \right).$$

Let  $n \in \mathbb{N}$  be that  $(n-1)x + t < 1-1/s \leq nx + t$ , then  $n$  is bounded according to  $x$ ,  $n < (1-1/s)/x + 1$ . The range of the summation can be rewritten as from 0 to  $n-1$ . Then, for any  $x' \in (t, x)$ , we have

$$h_s(x) \geq \min\{t, x\} + \frac{1}{s-1} \sum_{k=0}^{n-1} \left( sx - \frac{x}{1-t-kx} \right).$$

Hence,

$$\begin{aligned} h_s(x') - h_s(x) &\leq x' - x + \frac{1}{s-1} \sum_{k=0}^{n-1} (sx' - sx) \\ &= \left(1 + \frac{sn}{s-1}\right)(x' - x) < \left(1 + \frac{1}{x} + \frac{s}{s-1}\right)(x' - x) \end{aligned}$$

and our lemma follows.  $\square$

LEMMA B.3. For any  $s > 1$  and any  $x \in (0, 1]$ , the left derivative of  $h_s(x)$  is upper bounded by  $\frac{s+1}{2}$ ,

$$\frac{\partial h_s(x)}{\partial x^-} \leq \frac{s+1}{2}.$$

*Proof.* Suppose that

$$h_s(x) = t + \frac{1}{s-1} \sum_{k \in \mathbb{N}, kx + t \leq 1-1/s} \left( sx - \frac{x}{1-t-kx} \right).$$

Let  $n \in \mathbb{N}$  be that  $(n-1)x + t < 1-1/s \leq nx + t$ . The range of the summation can be rewritten as from 0 to  $n-1$ . Then, for any  $x' \in (t, x)$ , we have

$$h_s(x') \geq t + \frac{1}{s-1} \sum_{k=0}^{n-1} \left( sx' - \frac{x'}{1-t-kx'} \right).$$

Therefore the left derivative of  $h_s(x)$  is at most:

$$\frac{1}{s-1} \sum_{k=0}^{n-1} \left( s - \frac{1-t}{(1-t-kx)^2} \right).$$

This formula is non-increasing with respect to  $x$ . Without loss of generality, we assume  $nx + t = 1-1/s$ . Consider the following equation:

$$\begin{aligned} \sum_{k=0}^{n-1} \frac{1-t}{(1-t-kx)(1-t-(k+1)x)} &= \frac{1-t}{x} \sum_{k=0}^{n-1} \frac{1}{1-t-(k+1)x} - \frac{1}{1-t-kx} \\ &= \frac{1-t}{x} \cdot \left( s - \frac{1}{1-t} \right) = sn. \end{aligned}$$

Finally, we have

$$\begin{aligned}
(s-1) \cdot \frac{\partial h_s(x)}{\partial x^-} &\leq (1-t) \cdot \sum_{k=0}^{n-1} \frac{1}{(1-t-kx)(1-t-(k+1)x)} - \frac{1}{(1-t-kx)^2} \\
&\leq (1-t) \cdot \sum_{k=0}^{n-1} \frac{1}{2} \cdot \frac{1}{(1-t-(k+1)x)^2} - \frac{1}{2} \cdot \frac{1}{(1-t-kx)^2} \\
&= (1-t) \cdot \left( \frac{s^2}{2} - \frac{1}{2(1-t)^2} \right) \leq \frac{s^2-1}{2}.
\end{aligned}$$

where the second inequality is by AM-GM inequality.  $\square$

Now, we are able to globally bound the difference of  $h_s(x)$  and  $h_s(x')$  by their difference  $x - x'$  within a constant factor, that shall be used in Appendix D.

LEMMA B.4. *For any  $s > 1$  and any  $x' \leq x \in [0, 1]$  we have  $h_s(x) - h_s(x') \leq \frac{s+1}{2}(x - x')$ .*

*Proof.* For any  $x \in [0, 1]$ , consider set  $A = \{a : \forall b \in [a, x], h_s(x) - h_s(b) \leq \frac{s+1}{2}(x - b), 0 \leq a \leq x\}$ . This set is non-empty since  $x \in A$ . We denote  $\inf A$  by  $a$ .

First we prove  $a \in \inf A$ . Since  $a = \inf A$ , we know that  $h_s(x) - h_s(b) \leq \frac{s+1}{2}(x - b)$  for any  $a < b \leq x$ . Take limit  $b \rightarrow a^+$  on both sides gives us  $h_s(x) - h_s(a) \leq \frac{s+1}{2}(x - a)$ , so  $a \in A$ .

Next we prove  $a = 0$ . Otherwise suppose that  $a > 0$ , since the left derivative of  $a$  is upper bounded by  $\frac{s+1}{2}$ , there exist  $0 < t < a$  such that for any  $b \in (t, a)$ ,  $h_s(a) - h_s(b) \leq \frac{s+1}{2}(a - b)$ , which implies that  $h_s(x) - h_s(b) \leq \frac{s+1}{2}(x - b)$ . As a result, for any  $b \in (t, a)$ ,  $b \in A$ , contradict to  $a = \inf A$ . Therefore we must have  $a = 0$ .

Hence,  $A = [0, x]$  and our lemma follows.  $\square$

## C Algorithm with general $s$

### Step Activation Rates Except One Large Item

(1) Let  $i_0 = \arg \max_i \sum_v x_i^v$  be the largest item, and let:

$$x_0 = \sum_v x_{i_0}^v, \quad z_{i_0}^v = \frac{(s\rho_{i_0}^v - 1)^+}{s-1}, \quad h_0 = \sum_v p_{i_0}^v z_{i_0}^v.$$

(2) For calculate a set of  $z_i^v$ 's for the remaining items  $i \neq i_0$  such that:

$$h_{ot} \stackrel{\text{def}}{=} \sum_{i \neq i_0} \sum_v p_i^v z_i^v = \min\{h_s(x_0) - h_0, 1 - x_0\}, \quad z_i^v \in \left[ \frac{1}{s-1}(s\rho_i^v - 1)^+, \rho_i^v \right],$$

where  $h_s(x_0)$  is the function as defined in Lemma 2.9.

(3) Select threshold times  $\beta_0 \leq \beta_1 \leq \beta_2$  based on the instance.

(4) For time  $t$  from 0 to 1:

(a) If the largest item  $i_0$  arrives with value  $v$ , activate it with probability:

$$g_{i_0}^v(t) = \begin{cases} 0 & \text{if } t \in [0, \beta_0), \\ z_{i_0}^v & \text{if } t \in [\beta_0, \beta_2), \\ s\rho_{i_0}^v - (s-1)z_{i_0}^v & \text{if } t \in [\beta_2, 1]. \end{cases}$$

(b) If an item  $i \neq i_0$  arrives with value  $v$ , activate it with activation rate:

$$a_i^v(t) = \begin{cases} z_i^v & \text{if } t \in [0, \beta_1), \\ s\rho_i^v - (s-1)z_i^v & \text{if } t \in [\beta_1, 1]. \end{cases}$$

(c) Accept the item if it is the first activated item.

LEMMA C.1. *There exist  $z_i^v$ 's satisfying the stated constraints.*

*Proof.* Note that  $\frac{1}{s-1}(s\rho_i^v - 1)^+ \leq \rho_i^v$ , it suffices to show that

$$\sum_{i \neq i_0} \sum_v p_i^v \cdot \frac{1}{s-1}(s\rho_i^v - 1)^+ \leq \min\{h_s(x_0) - h_0, 1 - x_0\} \leq \sum_{i \neq i_0} \sum_v p_i^v \rho_i^v.$$

Recall that  $h_s(x_0)$  is an upper bound of  $\sum_{i,v} \frac{(sx_i^v - p_i^v)^+}{s-1}$  (see Appendix B). The first inequality follows by:

$$\begin{aligned} \sum_{i \neq i_0} \sum_v p_i^v \cdot \frac{1}{s-1}(s\rho_i^v - 1)^+ &= \sum_{i \neq i_0} \sum_v \frac{1}{s-1}(sx_i^v - p_i^v)^+ \\ &= \sum_i \sum_v \frac{1}{s-1}(sx_i^v - p_i^v)^+ - h_0 \leq h_s(x_0) - h_0. \end{aligned}$$

The second inequality follows by  $\sum_{i \neq i_0} \sum_v p_i^v \rho_i^v = \sum_{i \neq i_0} \sum_v x_i^v = 1 - x_0$ .  $\square$

LEMMA C.2. *The activation rates and probabilities are well-defined.*

*Proof.* Recall that  $0 \leq x_i^v \leq \rho_i^v \leq 1$ . For any item  $i$ , we have  $0 \leq z_i^v \leq \rho_i^v \leq 1$ , and  $s\rho_{i_0}^v - (s-1)z_{i_0}^v = s\rho_{i_0}^v - (s\rho_{i_0}^v - 1)^+ = \min\{s\rho_{i_0}^v, 1\} \in [0, 1]$

Hence, for the largest item  $i_0$ , the activation probability  $g_{i_0}^v(t)$  is always between 0 and 1. For any other item  $i \neq i_0$ , the activation rate  $a_i^v(t)$  is between 0 and 1 and the resulting activation probability  $g_i^v(t) = a_i^v(t)e^{-\int_0^t A_i(x) dx}$  is also between 0 and 1.  $\square$

Then, we are ready to prove the main theorem of this section Theorem 2.1, which says that the Step Activation Rates Except One Large Item is 0.688-competitive.

*Proof.* [Proof of Theorem 2.1] Actually, we do not give the explicit form of  $\beta_0, \beta_1, \beta_2$ . We only show that there always *exists* proper choices of them based on any  $x_0, h_0$  such that the competitive ratio is 0.688. In the following, we let  $\beta_0 \leq \beta_1 \leq \beta_2$  to be determined.

**Largest Item.** By Lemma 2.3, the probability that the item  $i (\neq i_0)$  was not activated before time  $t$  is  $e^{-\int_0^t A_i(x) dx}$ , where  $A_i(x)$  equals  $\sum_v p_i^v z_i^v$  when  $x \in [0, \beta_1]$  and equals  $s \sum_v x_i^v - \sum_v p_i^v z_i^v$  otherwise.

By the definition of  $h_{ot}$  and  $\sum_j \sum_v x_j^v = 1$  (Lemma 2.5), we have:

$$\sum_{j \neq i_0} A_j(x) = \begin{cases} \sum_{j \neq i_0} \sum_v p_j^v z_j^v = h_{ot}, & \text{for } x \in [0, \beta_1] \\ \sum_{j \neq i_0} (s \sum_v x_j^v - \sum_v p_j^v z_j^v) = s(1 - x_0) - h_{ot}. & \text{for } x \in [\beta_1, 1] \end{cases}$$

Thus, the probability of not existing any item  $j \neq i_0$  was activated before time threshold  $t$  is:

$$\prod_{j \neq i_0} e^{-\int_0^t A_j(x) dx} = e^{-\int_0^t \sum_{j \neq i_0} A_j(x) dx} = e^{-h_{ot} \cdot t - s(1-x_0-h_{ot}) \cdot (t-\beta_1)^+} \stackrel{\text{def}}{=} e^{-K(t)},$$

Then, according to Lemma 2.2, the probability of item  $i_0$  is accepted with value  $v_{i_0} = v$  is:

$$\begin{aligned} & \int_0^1 p_{i_0}^v g_{i_0}^v(t) e^{-K(t)} dt \\ &= p_{i_0}^v \left( z_{i_0}^v \int_{\beta_0}^1 e^{-K(t)} dt + (\rho_{i_0}^v - z_{i_0}^v) \cdot s \int_{\beta_2}^1 e^{-K(t)} dt \right) \\ &\geq p_{i_0}^v \rho_{i_0}^v \cdot \min \left( \int_{\beta_0}^1 e^{-K(t)} dt, s \int_{\beta_2}^1 e^{-K(t)} dt \right) \end{aligned}$$

**Other Items.** We define  $L(t)$  as the probability that item  $i_0$  is activated before time  $t$ . Then, according to the definition of  $g_{i_0}^v(t)$  and  $h_0$ ,  $\sum_v p_{i_0}^v g_{i_0}^v(t)$  equals 0 for  $t \in [0, \beta_0]$ , equals  $h_0$  for  $t \in [\beta_0, \beta_2]$ , and equals  $s x_0 - h_0$  for  $t \in [\beta_2, 1]$ . Thus, by Lemma 2.1, we have

$$L(t) = \int_0^t \sum_v p_{i_0}^v g_{i_0}^v(x) dx = h_0 \cdot (t - \beta_0)^+ + s(x_0 - h_0) \cdot (t - \beta_2)^+.$$

Thus, for any item  $i \neq i_0$ , according to Lemma 2.2, the probability of item  $i$  to be accepted with value  $v$  is:

$$\begin{aligned} & \int_0^1 p_i^v a_i^v(t) e^{-\int_0^t \sum_i A_i(x) dx} \cdot (1 - L(t)) \prod_{j \neq i, i_0} e^{-\int_0^t A_j(y) dy} dt \\ &= p_i^v \int_0^1 a_i^v(t) \prod_{j \neq i_0} e^{-\int_0^t A_j(y) dy} (1 - L(t)) dt \\ &= p_i^v \int_0^1 a_i^v(t) e^{-K(t)} (1 - L(t)) dt \\ &= p_i^v \left( z_i^v \cdot \int_0^1 e^{-K(t)} (1 - L(t)) dt + (\rho_i^v - z_i^v) \cdot s \int_{\beta_1}^1 e^{-K(t)} (1 - L(t)) dt \right) \\ &\geq p_i^v \rho_i^v \min \left( \int_0^1 e^{-K(t)} (1 - L(t)) dt, s \int_{\beta_1}^1 e^{-K(t)} (1 - L(t)) dt \right) \end{aligned}$$

**Choice of Parameters.** Therefore, we define:

$$(C.2) \quad \Gamma(x_0, h_0, s, \beta_0, \beta_1, \beta_2) = \min \left( \int_{\beta_0}^1 e^{-K(t)} dt, s \int_{\beta_2}^1 e^{-K(t)} dt, \right.$$

$$(C.3) \quad \left. \int_0^1 e^{-K(t)} (1 - L(t)) dt, s \int_{\beta_1}^1 e^{-K(t)} (1 - L(t)) dt \right),$$

We use computer to numerically verify the following, that we discuss in detail in Appendix D.

**LEMMA C.3.** *For every  $x_0 \in [0, 1]$  and  $h_0 \in [0, x_0]$ , there exists  $s \geq 1, 0 \leq \beta_0 \leq \beta_1 \leq \beta_2 \leq 1$ , such that  $\Gamma(x_0, h_0, s, \beta_0, \beta_1, \beta_2) > 0.688$ .*

This concludes a competitive ratio of 0.688.  $\square$

## D Proof of Lemma 2.12 and Lemma C.3 with Computer Assistance

Below we prove Lemma C.3. Lemma 2.12 can be proved in a similar way except that the function  $\Gamma$  has one less input  $s$  which is fixed to be 2.

First, we set the parameter  $s$  as the following:

$$s = \begin{cases} 3 & x_0 \in [0, 0.35] \\ 2.5 & x_0 \in (0.35, 0.6] \\ 2 & x_0 \in (0.6, 1] \end{cases}$$

Next, let  $\epsilon = 1/10000$ . For every  $x_0 \in [0, 1]$ , and  $h_0 \in [0, x_0]$ , we round them down to multiples of  $\epsilon$ . I.e., let  $x'_0 = \epsilon \cdot \lfloor x_0/\epsilon \rfloor$  and  $h'_0 = \epsilon \cdot \lfloor h_0/\epsilon \rfloor$ .

Next, we prove that for any  $0 \leq \beta_0 \leq \beta_1 \leq \beta_2 \leq 1$ , we have:

$$(D.4) \quad \Gamma(x_0, h_0, s, \beta_0, \beta_1, \beta_2) \geq \Gamma(x'_0, h'_0, s, \beta_0, \beta_1, \beta_2) - \left( \frac{3}{2}s^2 + \frac{1}{2}s \right) \epsilon.$$

We naturally consider variables according to the rounding:  $h'_{ot} = \min\{1 - x'_0, h_s(x'_0) - h'_0\}$  and  $K(t)' = h_{ot}' \cdot t + s(1 - x'_0 - h'_{ot})(t - \beta_1)^+$ , and  $L(t)' = h'_0 \cdot (t - \beta_0)^+ + s(x'_0 - h'_0)(t - \beta_2)^+$ .

We first bound  $K(t)' - K(t)$  for any  $t$ :

$$\begin{aligned} K(t)' - K(t) &= s(t - \beta_1)^+ \cdot (x_0 - x'_0) + (s(t - \beta_1)^+ - t) \cdot (h_{ot} - h'_{ot}) \\ &\geq (s(t - \beta_1)^+ - t) \cdot (h_{ot} - h'_{ot}) \geq -\frac{s+1}{2}\epsilon, \end{aligned}$$

where the last inequality holds by the following case analysis.

- If  $s(t - \beta_1)^+ - t \geq 0$ , then  $s(t - \beta_1)^+ - t \leq s - 1$ . Further by  $h_{ot} \geq h'_{ot} - \epsilon$  due to  $1 - x_0 \geq 1 - x'_0 - \epsilon$  and  $h_s(x_0) - h_0 \geq h_s(x'_0) - h'_0 - \epsilon$ , we have  $K(t)' - K(t) \geq -(s - 1)\epsilon \geq -\frac{s+1}{2}\epsilon$ .
- If  $s(t - \beta_1)^+ - t \leq 0$ , then  $t - s(t - \beta_1)^+ \leq 1$ . Further by  $h_{ot}' \geq h'_{ot} - \frac{s+1}{2}\epsilon$  due to  $1 - x'_0 \geq 1 - x_0$  and  $h_s(x'_0) - h'_0 \geq h_s(x_0) - h_0 \geq h_s(x_0) - h_0 - \frac{s+1}{2}\epsilon$  by Lemma B.4, we have  $K(t)' - K(t) \geq -\frac{s+1}{2}\epsilon$ .

Second we lower bound  $L(t)' - L(t)$  for any  $t$ :

$$L(t)' - L(t) = -s(t - \beta_0)^+ \cdot (x_0 - x'_0) + ((t - \beta_0)^+ + s(t - \beta_2)^+) \cdot (h_0 - h'_0) \geq -s \cdot \epsilon.$$

Putting these together, we have for any  $t$ :

$$\begin{aligned} e^{-K(t)} - e^{-K'(t)} &\geq -\frac{s+1}{2} \cdot \epsilon, \\ e^{-K(t)}(1 - L(t)) - e^{-K'(t)}(1 - L(t)') &\geq -\left( \frac{s+1}{2} + s \right) \cdot \epsilon. \end{aligned}$$

Recall the definition of  $\Gamma$  function, we conclude the proof of (D.4).

Finally, we are left to verify that for every  $(x_0, h_0) = (i\epsilon, j\epsilon)$ ,  $0 \leq i, j \leq \frac{1}{\epsilon}$ , there exist  $0 \leq \beta_0 \leq \beta_1 \leq \beta_2 \leq 1$ ,

$$\Gamma(x_0, h_0, s, \beta_0, \beta_1, \beta_2) - s \left( \frac{3}{2}s + \frac{1}{2} \right) \cdot \epsilon > 0.688 .$$

This final step is done by brute force search with computer assistance. For verification, the source code can be found at a github repository<sup>2</sup>.

---

<sup>2</sup>[https://github.com/billyldc/prophet\\_secretary\\_and\\_matching\\_code](https://github.com/billyldc/prophet_secretary_and_matching_code)