MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Prophet Inequality for Bipartite Matching: Merits of Being Simple and Non-adaptive *

Nick Gravin
Shanghai University of Finance and Economics, School of Information Management and Engineering, Institute for Theoretical Computer Science, 100 Wudong Road, Shanghai 200433, China nikolai@mail.shufe.edu.cn

Hongao Wang
Purdue University, Department of Computer Science, 305 N University St, West Lafayette, IN, 47907, United States wang5270@purdue.edu

We consider the Bayesian online selection problem of a matching in bipartite graphs, i.e., the weighted online matching problem where the edges arrive online and edge weights are generated from a known distribution. This setting corresponds to the intersection of two matroids in the work of Kleinberg and Weinberg (2012) and Feldman et al. (2016). We study a simple class of non-adaptive policies which we call vertex-additive policies. A vertex-additive policy assigns static prices to every vertex in the graph and accepts only those edges whose weight exceeds the sum of the prices on the edge endpoints. We show that there exists a vertex-additive policy with the expected payoff of at least one third of the prophet's payoff and present a gradient descent algorithm that quickly converges to the desired vector of vertex prices. Our results improve on the adaptive online policy of Kleinberg and Weinberg and Feldman et al. for the intersection of two matroids in two ways: our policy is non-adaptive and has better approximation guarantee of 3 instead of the previous guarantees of 5.82 in Kleinberg and Weinberg and 5.43 in Feldman et al. We give a complementary lower bound of 2.25 for any online algorithm in the bipartite matching setting.

Key words: Prophet Inequality, Online Matching, Edge Arrivals, Non-Adaptive Thresholds
MSC2000 subject classification: Primary: 68W27; secondary: 90C27, 68Q87
OR/MS subject classification: Primary: Networks/graphs matchings; secondary: stochastic model applications, approximation algorithms

* This work is supported by Science and Technology Innovation 2030 –"New Generation of Artificial Intelligence" Major Project No.(2018AAA0100903), Innovation Program of Shanghai Municipal Education Commission, Program for Innovative Research Team of Shanghai University of Finance and Economics (IRTSHUFE) and the Fundamental Research Funds for the Central Universities.

1. Introduction The topic of prophet inequality broadly refers to the study of trade offs between online and offline selection algorithms in Bayesian settings. The first fundamental result is due to Krengel and Sucheston [41], who considered the following problem in the optimal stopping theory. Let $X_1, \ldots, X_n$ be a sequence of independent, non-negative random variables with $\mathbf{E}[X_i] < +\infty$, then there is an online stopping policy τ such that

$$2 \cdot \mathbf{E}\left[X_\tau\right] \geq \mathbf{E}\left[\max_i X_i\right]. \tag{1}$$

I.e., if a gambler plays a sequence of games with rewards $X_1, ..., X_n$ and can stop at any time and take the most recent reward, then a prophet who can foretell which reward to take in the whole sequence cannot gain more than two times the reward of the gambler. In fact, it was later shown by Samuel-Cahn [50] that a simple stopping policy with a uniform threshold $c: \ \tau = \arg\min_i (X_i \geq c)$ gives the same approximation guarantee of 2.

In computer science terms, inequality (1) compares the performance of an online algorithm (the gambler) and the offline optimal solution (the prophet) in the competitive analysis framework. Results in the form of inequality (1) have been extended to Bayesian settings where the algorithm has to select a feasible subset of elements from a random sequence of samples with the knowledge about the distribution from which this sequence is generated and where the value of the selected set is compared to the offline optimum solution with the complete information about the sequence of samples. For example, a general result of this type is due to Kleinberg and Weinberg [40], who gave a $4k-2$ competitive algorithm for the Bayesian selection problem where the system of feasible sets can be represented as the intersection of k matroids.

The interest in prophet inequalities as a tool for the design and analysis of algorithms has been largely driven by important applications in algorithmic mechanism design. In particular, the work in Bayesian mechanism design showed a remarkable power of simple sequential posted price mechanisms in a variety of settings such as auctions for unit-demand bidders [10],[2] and combinatorial auctions [27],[19],[21]. The underlying environment for all these applications is matching in bipartite graphs.

Matching. In this paper, we adopt the general online Bayesian model of Kleinberg and Weinberg [40] for selecting a matching in bipartite graphs. Specifically, we assume that *edges* of a bipartite graph are the elements of our online selection problem which may arrive in an arbitrary order. The edge values are independently generated from the prior distributions that are known to the online algorithm. Upon arrival of a new edge the algorithm observes its value and must immediately and irrevocably decide whether to include this edge in the matching or not. The algorithm's objective is to maximize the total value of the selected matching.

This model is more general than the matching (unit-demand) models or sequential posted prices from the mechanism design literature (e.g., [27],[19]). Indeed, the previous work mostly focuses on *one-sided vertex arrival models*, i.e., where vertices on one side of the graph arrive online and the values of all edges incident to a new vertex are revealed to the online algorithm[1]. On the practical side, the edge arrival model (as opposed to vertex arrival) gives a finer control over agent preferences which may not be available upon arrival of a new agent and also may dynamically evolve over time.

Non-adaptive pricing. The online policy for more general selection problems proposed by Kleinberg and Weinberg [40] is highly adaptive, i.e., the threshold rule for accepting or rejecting a particular element changes over time depending on the values of the previous observed elements. A more desirable approach is to use *static* thresholds that are computed beforehand and do not change with the arrival of new elements. Such non-adaptive policies translate to simpler and more robust mechanisms that enjoy better incentive properties and are more fair to the agents. Indeed, such mechanisms are group strategy proof, credible (i.e., agents do not need to trust the mechanism designer), and timing of the agent arrivals has a limited influence on the outcome of the mechanism.

Given this it is not surprising that most applications in the mechanism design literature starting with the work of Hajiaghayi et al. [31] and Chawla et al. [10] rely on non-adaptive versions of prophet inequality. For example, Chawla et al. [10] refers to the result of Samuel-Cahn [50] that uses simple and robust constant threshold policy c but not the earlier version of Krengel and Sucheston [41] which compares the optimal online and offline policies. Similarly, Feldman et al. [27] use static item prices that never change throughout the execution of their algorithm.

Our results. In this paper, we prove a *non-adaptive* prophet inequality for selecting a matching in bipartite graphs. Our online algorithm belongs to a natural class of simple *vertex-additive* threshold policies: (i) vertices on either side of the graph receive threshold values $\mathbf{l} = (l_1, \dots, l_n)$ for the left hand side and $\mathbf{r} = (r_1, \dots, r_m)$ for the right hand side vertices (ii) the online algorithm accepts every edge $e = (i, j)$ that can be added to the current matching if $v(e) \geq l_i + r_j$. We find a vertex-additive policy with the expected payoff of at least one third of the prophet's payoff (the expected value of the optimal matching), i.e., we give a 3 approximation guarantee. We give a complementary lower bound of 2.25 that shows clear separation from the setting where vertices from one side arrive online and where the upper bound is 2 (see, e.g., [27]).

The closest to our results are the papers by Kleinberg and Weinberg [40] and Feldman et al. [28]. The former derives a $4p-2$-approximation prophet inequality for the intersection of p matroids (a more general feasibility constraint than our bipartite matching setting). The later derives prophet

[1] To be fair, the independence assumption (for all edges) is a bit stronger than in some mechanism design models with vertex arrivals. However, without this assumption, it is impossible to recover constant fraction of the prophet.

inequalities via online contention resolution schemes (OCRS) in a variety of settings including intersection of p matroids, matching in general graphs, and knapsack feasibility constraint. There are a few aspects that differentiate our work from these two papers.

- *Assumptions on the arrival order.* The framework of [40] assumes that the elements arrive in an unknown order chosen by the *oblivious adversary*, i.e., the adversary picks the order before seeing the realization of the elements' values. The results in [40] also hold against a stronger *adaptive adversary* who makes decisions about arrival order concurrently with the online algorithm, i.e., the adversary can choose which element will arrive next based on the values of the previous elements and previous choices of the online algorithm. The OCRS approach in [28] allows to obtain results against the strongest *almighty adversary*, who decides on the arrival order after seeing all realized values. Similar to [40] we assume that the adversary is *oblivious*. Our analysis also works against the *adaptive adversary* but not the *almighty adversary*.
- *Approximation guarantee.* Matching in bipartite graphs can be represented as an intersection of $p = 2$ matroids. For this special case [40] slightly improves the approximation guarantee of 6 to 5.82 and [28] has a $2 \cdot e = 5.43$ approximation. In both cases the best known lower bound was 2. We obtain a much tighter approximation of 3 complemented by an improved lower bound of 2.25.
- *Adaptivity and pricing mechanisms.* The policies for accepting new elements in [40] and [28] are more complex than our scheme: they cannot be described by a simple rule with precomputed thresholds on every element subject to the feasibility constraint. The mechanism in [40] is not an *oblivious posted pricing*, i.e., the mechanism keeps changing the item prices over time. The respective mechanisms in [28] are *constraint oblivious posted pricing*, i.e., the item prices are calculated in advance but the feasibility constraint is modified to a subfamily of the original feasible sets.

Our techniques. The analysis of our algorithm consists of two stages. First, we estimate the expected performance of a vertex-additive policy parametrized by the generic price vectors $\mathbf{l}, \mathbf{r}$. This part of the analysis is similar in many ways to [40, 27, 19]. The performance of the algorithm is decomposed into the revenue and the surplus parts. We count the revenue of the algorithm similarly to [40, 27, 19]: we also put the prices on the vertices rather than the elements of the set system (edges in the bipartite graph). On the other hand, we estimate the surplus term per every edge of the bipartite graph and not per every vertex.

After we take the expectation over all valuation profiles, we get a parametric estimate (11) that depends on two statistics for the distribution of the optimal offline matching: (i) the expected value contributions of every edge (i, j) to the maximum matching; (ii) the probabilities of every edge (i, j) to appear in the maximum matching. We conveniently write down these statistics in the matrix form: $\mathbf{M}$ for (i) and $\mathbf{Q}$ for (ii). The estimate (11) has a new variable - the set of matched vertices.

We treat this variable as if it is chosen by an adversary, similar to the argument in [40, 27, 19]. What separates our analysis from the prior work is that we optimize our prices $\mathbf{l}, \mathbf{r}$ based on the both statistics $\mathbf{M}, \mathbf{Q}$. Note that [40, 27, 19] operate only with the expected contributions to the optimum, while online contention resolution schemes in [28, 24] only look at the probability of the elements to appear in the offline optimum and choose their prices to control the probabilities of accepting elements in the online algorithm.

In the second stage of our analysis, we show the existence of good prices $\mathbf{l}, \mathbf{r}$ in the estimate (11). That stage is highly non trivial, since our maximization problem (11) also contains adversarially chosen variables, i.e., (11) is a max-min optimization problem. We convert (11) into a bilinear algebraic form. Then we relax it to a simpler semi-linear (it is not linear because we have $[\cdot]^+ : x \to \max(x, 0)$ operator besides linear transformations) max-min optimization problem (15) which has a smaller value than (11). The later problem resembles the Lagrangian relaxation of a constrained optimization problem where the constraints are given by a system of equations (16) and the adversarial variables in (15) correspond to the Lagrangian multipliers. The objective in (15) after eliminating the terms with adversarial variables is called *virtual surplus*. We show that the system of constraints (16) has a solution. This solution gives us vertex prices $\mathbf{l}, \mathbf{r}$ and conveniently eliminates all the terms with the adversarial variables from (15). We show in (17) that 3 times the virtual surplus covers the expected optimum $\text{opt} = \mathbb{1}_L^\top \cdot \mathbf{M} \cdot \mathbb{1}_R$ given the constraints (16). Furthermore, the proof that system of equations (16) has a solutions is constructive: we give a gradient-descent algorithm that quickly converges to the solution.

1.1. Related Work Starting with the work of Krengel and Sucheston [41] there has been a lot of work studying different restrictions on stopping rules, distributions, independence assumption etc., which is too broad to discuss in this amount of space. We recommend [32] for a detailed survey. The line of earlier work [37, 38, 39] on multiple-choice prophet inequalities, where the online algorithm and the prophet can choose more than one element, is particularly relevant to our paper.

In computer science literature, the research on prophet inequalities and their applications to posted price mechanisms was initiated by Hajiaghayi et al. [31]. Prophet inequalities were obtained for a variety of multiple choice combinatorial settings: for a matroid and an intersection of matroids [40, 6, 28], for polymatroids [22], for the generalized assignment problem [3, 4], and for the general downward closed feasibility constraints [47]. In algorithmic mechanism design, Chawla et al. [9] have found approximately optimal in terms of revenue posted price mechanisms for the unit-demand buyers. Other results on Bayesian auctions with the objectives of revenue and welfare maximization include [10, 2, 13] for unit-demand buyers, and [2, 27, 19, 48, 21] for combinatorial auctions. A direct connection between pricing mechanisms and prophet inequalities is shown in [16].

Mechanism design applications of the prophet inequality have instigated further studies of dynamic posted prices in other online optimization settings such as a k-server problem [12], and the makespan minimization for the scheduling problems [26].

A recent literature on prophet inequalities mostly focuses on the issue of limited knowledge of the value distributions, i.e., the online algorithm instead of the perfect information about the prior distributions is given only certain statistics or has an access to a limited number of samples from the priors [6, 15, 14, 49].

Online matching. Our matching setting is closely related to the online bipartite matching problem introduced by Karp et al. [36] which is a central topic in the area of online algorithms with a wide range of applications. In this model one side of the bipartite graph is given in advance, while the vertices of the other side arrive online in an arbitrary order. The online algorithm observes all the edges incident to a new vertex, however the algorithm does not have any prior information about distribution of the edges (typically in this line of work edges have only $0-1$ weights). Karp et al. gave the tight 1.58 approximation guarantee for the adversarial order of vertex arrivals. The result was extended to the settings with weighted vertices [1] and Adword problem [45]. The latter work considers generalized matching where each edge has a weight and each vertex in the given side of the graph has a budget (capacity constraint). Earlier work [45, 7] on the Adword problem focused on the worst-case performance guarantees for the adversarial or random arrival order [29, 17]. More recent papers adopt the Bayesian framework in which distribution of weights (bids) is usually known in advance, e.g., [25, 42, 35, 43, 30, 30, 46, 18]. A few papers consider matching models closer to our edge arrival setting. E.g., McGregor [44] gave a 5.82-competitive online algorithm for the (weighted) edge arrival model with preemption.[2] Better approximation guarantees were obtained for randomized algorithms [23] and for the special cases of growing tree and forests [11, 8]. A very recent line of work [33, 5, 34] considers the fully online matching model in (unweighted) bipartite graphs where vertices of both sides arrive online and reveal edges to all previous vertices. In this model [34] gave a tight 1.76-competitive algorithm.

A follow up paper (to the conference version of our work) by Ezra et al. [24] has slightly improved the approximation guarantee to 2.96. They used a different OCRS-type algorithm that works for general (non bipartite) graphs. However, their algorithm is highly adaptive (it updates the thresholds on the edges at every round) and unlike other OCRS in [28] works only against the adaptive adversary.

[2] I.e., the online algorithm can discard any previously matched edges. Note that $5.82 = 3 + 2\sqrt{2}$ is exactly the same approximation ratio as in Kleinberg and Weinberg.

2. Preliminaries Let $G(L,R)$ be a bipartite multi-graph between the left and the right parts L,R with $|L|=n$ and $|R|=m$ vertices. The graph has a set of multi-edges E between L and R with different values $v_e \in \mathbb{R}_+$ for each edge $e \in E$, where $\mathbb{R}_+$ denotes the set of non-negative real numbers. The edges arrive online in an unknown order $\sigma = (e(t))_{t=1}^{t=T}$, where edges are enumerated by their arrival time from 1 to $T=|E|$. For a given profile of values $\mathbf{v}=\{v_e\}_{e\in E}$, we let $\texttt{opt}(\mathbf{v})$ to denote the value of the maximum (offline) matching in the graph G with edge values $\mathbf{v}$.

Bayesian online selection problem. We consider a Bayesian setting, where edge values are drawn independently from the distributions $\{\mathcal{F}_e\}_{e\in E}$. Write $\mathcal{F}=\prod_{e\in E}\mathcal{F}_e$, so that the joint valuation profile $\mathbf{v}$ of all edges is drawn from the joint distribution $\mathbf{v}\sim\mathcal{F}$. Both the graph G and the distribution $\mathcal{F}$ are known in advance. An input to the Bayesian online selection problem (BOSP) is a sequence σ of pairs $(e(t), v_{e(t)})_{t=1}^{t=T}$ revealed one by one from time $t=1$ to $t=T$. An *online selection algorithm* (also called *online policy*) $\mathcal{A}$[3] upon receiving a new piece of input $(e(t), v_{e(t)})$ at time t must irrevocably decide whether to take the edge $e(t)$ subject to the feasibility constraint that the selected set of edges is a matching. The goal of the online algorithm $\mathcal{A}$ is to maximize the total value of the selected edges in expectation over $\mathbf{v}\sim\mathcal{F}$.

Every online policy at each time $\tau\in[T]$ may be described as a function of the past selection decisions (not the edge values), since the posterior distribution of values and feasibility constraint for the edges coming after time τ (for a fixed sequence of past decisions) do not depend on the realized values before time τ. We use $\mathcal{A}(\mathbf{v},\sigma)$ to denote the set of all edges accepted by an online selection algorithm $\mathcal{A}$. When it is clear from the context, we sometimes will drop the dependency on the arrival order σ. Without loss of generality, one may restrict attention to *monotone* selection policies, i.e., policies that for any given history before time τ select the edge $e(\tau)$ with higher probability for larger values of $v_{e(\tau)}$. It means that any monotone *deterministic* selection policy can be described by a sequence of thresholds $(\delta_\tau(\sigma,\mathbf{v}))_{\tau=1}^{\tau=T}$, where each threshold δ_τ only depends on the prior history $(v_{e(t)})_{t=1}^{\tau-1}$, such that edge $e(\tau)$ is accepted if and only if $v_{e(\tau)}\geq\delta_\tau$ and neither of vertices i,j incident to $e(\tau)=(i,j)$ was covered by previously selected edges.

Approximation guarantees. The algorithm $\mathcal{A}$ is agnostic to the order of edge arrivals, that is $\mathcal{A}$ should perform well regardless of the arrival order of the edges. We study performance of $\mathcal{A}$ in the worst-case for a *fixed-order* adversary which is the standard assumption in the prior work on BOSP. Formally, we assume that an adversary selects the order σ of edge arrivals (or distribution of different orders) that does not change with the choices of algorithm and/or realized edge values. As the optimal solution is usually quite complex even in the most basic settings, the performance of the

[3] The algorithm $\mathcal{A}$ can be randomized, but it must be independent of the randomness (if any) in the choice of σ. Any randomized algorithm is a distribution over deterministic policies.

online policy $\mathcal{A}$ is compared against a stronger benchmark of the expected *prophet*'s performance who knows all the realized values $\mathbf{v}$ in advance before selecting any edges, i.e., the benchmark is the offline optimum solution. *Prophet inequality* refers to the approximation guarantee of ρ that online algorithm can achieve compared to the prophet, i.e., for any arrival order σ

$$\rho \cdot \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathcal{F}}\left[\sum_{e\in\mathcal{A}(\mathbf{v},\sigma)} v_e\right] \geq \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathcal{F}}\left[\mathtt{opt}(\mathbf{v})\right].$$

3. Online Algorithm Our algorithm belongs to a natural class of simple non-adaptive threshold policies that we call vertex-additive policies. Any vertex-additive policy (denoted as `V-add`) is described by two positive real vectors $\mathbf{l} = (l_1, \ldots, l_n)$ and $\mathbf{r} = (r_1, \ldots, r_m)$ that accepts a new arriving edge $e(t) = (i, j)$ with value $v_{e(t)}$ if and only if

1. vertices i, j are available, i.e., both $i \in L$, $j \in R$ are not covered by previously accepted edges;
2. edge's value exceeds the sum of its vertex thresholds $v_{e(t)} \geq l_i + r_j$.

If at least one of the previous two conditions fails, then the edge $e(t)$ is rejected.

3.1. Analysis of the vertex-additive policies We assume that incoming edges $(e(t))_{t=1}^{t=T}$ of the graph G arrive in a fixed order σ, which is unknown to us. The total value of the accepted edges consists of two parts: (i) the *revenue* that is equal to the sum of accepted edges' thresholds, i.e., the total payment received by the algorithm if it takes from each accepted edge a price equal to its threshold; (ii) the *surplus* which is equal to the sum of extra values that every accepted edge contributes to the total value on top of its guaranteed threshold payment. We will use the price terminology interchangeably with the thresholds when referring to the vertex-additive policy. In the following we analyze separately the revenue and the surplus parts of any vertex-additive policy $\mathcal{A} = \mathtt{V\text{-}add}(\mathbf{l}, \mathbf{r})$. To simplify notations we use $\mathcal{A}(\mathbf{v})$ to denote the set of accepted edges for a particular run of $\mathtt{V\text{-}add}(\mathbf{l}, \mathbf{r})$ on the valuation profile $\mathbf{v}$.

Revenue. The important property of the vertex-additive prices is that we can conveniently attribute the expected revenue of our policy (denoted as **Rev**) to the average set of covered vertices instead of the set of accepted edges. We denote the final set of covered vertices of $\mathcal{A}(\mathbf{v})$ for a given valuation profile $\mathbf{v}$ as $X(\mathbf{v})$ that consists of the vertex sets $X_L(\mathbf{v}), X_R(\mathbf{v})$ respectively in the left and the right parts of G. The expected revenue can be written as follows.

$$\mathbf{Rev} = \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{(i,j)\in\mathcal{A}(\mathbf{v})} (l_i + r_j)\right] = \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{i\in X_L(\mathbf{v})} l_i + \sum_{j\in X_R(\mathbf{v})} r_j\right] \tag{2}$$

Surplus. We can give a lower bound on the expected surplus using the edges between *uncovered* vertices of $\texttt{V-add}(\mathbf{l},\mathbf{r})$. The expected surplus (denoted as **Surplus**) by definition is equal to

$$\textbf{Surplus} \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{e(t)\in\mathcal{A}(\mathbf{v})\\ e(t)=(i,j)}} \left[v_{e(t)} - l_i - r_j\right]^+\right], \tag{3}$$

where $\left[v_{e(t)} - l_i - r_j\right]^+$ denotes $\max\{(v_{e(t)} - l_i - r_j), 0\}$. Since our goal is to compare the surplus with the optimal matching selected by the prophet, we focus on the edges in the optimal matching $\texttt{opt}(\widehat{\mathbf{v}})$, i.e., the optimal matching for another independently drawn valuation profile $\widehat{\mathbf{v}} \sim \mathcal{F}$. Specifically, our lower bound will include all the edges in a random optimal matching $\texttt{opt}(\widehat{\mathbf{v}})$ between uncovered vertices for the valuation profiles $\mathbf{v}$. To deal with the specific edges (i,j) in the optimal matching $\texttt{opt}(\widehat{\mathbf{v}})$ we employ indicator random variable $\mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}$ for the event that there is an edge $e(t)\in\texttt{opt}(\widehat{\mathbf{v}})$ between vertices $i\in L, j\in R$ and random variable $\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}})$:

$$\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) \stackrel{\text{def}}{=} \begin{cases} \widehat{v}_{e(t)} & \text{if } \exists e(t)\in\texttt{opt}(\widehat{\mathbf{v}}) \text{ s.t. } e(t)=(i,j),\\ 0 & \text{if } (i,j)\notin\texttt{opt}(\widehat{\mathbf{v}}) \end{cases} \qquad \mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } (i,j)\in\texttt{opt}(\widehat{\mathbf{v}}),\\ 0 & \text{if } (i,j)\notin\texttt{opt}(\widehat{\mathbf{v}}). \end{cases} \tag{4}$$

LEMMA 1. *The expected surplus is at least*

$$\textbf{Surplus} \ge \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{i\notin X_L(\mathbf{v})\\ j\notin X_R(\mathbf{v})}} \left[\mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\chi^{(i,j)}_{opt}(\widehat{\mathbf{v}})\right] - (l_i + r_j)\mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\mathbb{I}^{(i,j)}_{opt(\widehat{\mathbf{v}})}\right]\right]^+\right], \tag{5}$$

where $\mathbf{E}_{\widehat{\mathbf{v}}}[\mathbb{I}^{(i,j)}_{opt(\widehat{\mathbf{v}})}]$ *can be also written as* $\mathbf{Pr}_{\widehat{\mathbf{v}}}[(i,j)\in opt(\widehat{\mathbf{v}})]$.

Proof. To analyze performance of a vertex-additive policy $\texttt{V-add}(\mathbf{l},\mathbf{r})$ we consider another indicator random variable $\mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)$ of $\mathbf{v}\sim\mathcal{F}$ that indicates whether vertices $i\in L, j\in R$ are free to take (not yet covered) on the valuation profile $\mathbf{v}$ right before the arrival of the edge $e(t)$ at time t. The surplus of $\texttt{V-add}$ is given by (3) which we further rewrite and bound as follows

$$\begin{aligned}\textbf{Surplus} &= \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{i\in L\\ j\in R}} \sum_{t:e(t)=(i,j)} \left[v_{e(t)} - l_i - r_j\right]^+ \cdot \mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)\right]\\ &= \mathop{\mathbf{E}}_{\mathbf{v},\widehat{\mathbf{v}}}\left[\sum_{i,j} \sum_{t:e(t)=(i,j)} \left[\widehat{v}_{e(t)} - l_i - r_j\right]^+ \cdot \mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)\right]\\ &\ge \mathop{\mathbf{E}}_{\mathbf{v},\widehat{\mathbf{v}}}\left[\sum_{i,j} \sum_{t:e(t)=(i,j)} \left[\widehat{v}_{e(t)} - l_i - r_j\right]^+ \cdot \mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},T+1)\right], \end{aligned} \tag{6}$$

where the second equality holds, since the indicator function $\mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)$ does not depend on $v_{e(t)}$ and thus is independent of the value of $\left[v_{e(t)} - l_i - r_j\right]^+$ which we substituted with another independent and identically distributed random variable $\left[\widehat{v}_{e(t)} - l_i - r_j\right]^+$; and the inequality holds simply because $\mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)$ is a decreasing function in time t[4]. We note that the right hand side (RHS) of (6) can be directly related to the sets of covered vertices $X_L(\mathbf{v}), X_R(\mathbf{v})$.

$$\text{RHS}(6) = \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{i\notin X_L(\mathbf{v}),\\ j\notin X_R(\mathbf{v})}} \sum_{t:e(t)=(i,j)} \mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\left[\widehat{v}_{e(t)} - l_i - r_j\right]^+\right]\right] \ge \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{i\notin X_L(\mathbf{v}),\\ j\notin X_R(\mathbf{v})}} \mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\left[\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) - l_i - r_j\right]^+\right]\right], \quad (7)$$

where to get the last inequality we ignored some non negative terms $\left[\widehat{v}_{e(t)} - l_i - r_j\right]^+$ in the previous summation and counted only the edges between $i \notin X_L(\mathbf{v})$ and $j \notin X_R(\mathbf{v})$ that appear in the optimal matching $\texttt{opt}(\widehat{\mathbf{v}})$. To conclude the proof we will use the following simple fact about function $[\cdot]^+$.

FACT 1. $\mathbf{E}[[s]^+] \ge [\mathbf{E}[s]]^+$ for any real random variables s.

Before we continue with the lower bound of (RHS) (7), we observe that $\left[\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) - l_i - r_j\right]^+ = \left[\left(\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) - l_i - r_j\right)\cdot \mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}\right]^+ = \left[\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) - (l_i + r_j)\cdot \mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}\right]^+$ for any fixed vertices $i \in L$, $j \in R$, and valuation profile $\widehat{\mathbf{v}}$. Indeed, by definition $\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) = 0$ and, therefore, $\left[\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) - l_i - r_j\right]^+ = 0$ whenever indicator $\mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})} = 0$. Finally, we obtain the required lower bound on the (RHS) of (7).

$$\begin{aligned}\textbf{Surplus} \ge \text{RHS}(7) &= \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{i\notin X_L(\mathbf{v}),\\ j\notin X_R(\mathbf{v})}} \mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\left[\left(\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}}) - l_i - r_j\right)\cdot \mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}\right]^+\right]\right] \\ &\ge \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{i\notin X_L(\mathbf{v}),\\ j\notin X_R(\mathbf{v})}} \left[\mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}})\right] - (l_i + r_j)\cdot \mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}\right]\right]^+\right],\end{aligned}$$

where the last inequality follows from the Fact 1 applied to the random variable $s_1 - s_2$, where $s_1 \stackrel{\text{def}}{=} \chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}})$ and $s_2 \stackrel{\text{def}}{=} (l_i + r_j)\cdot \mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}$. ☐

REMARK 1. Although we present the proof for the oblivious adversary (i.e., there is an unknown but fixed arrival order of the edges), our analysis extends to a stronger adaptive adversary that chooses the arrival order of the elements concurrently with the online algorithm. That is the adversary decides on the arrival of the edges one-by-one and picks the next edge $e(\mathbf{v},t)$ knowing the sequence of the previous values $(v_{e(1)}, v_{e(2)}, \cdots, v_{e(t-1)})$ and the previous choices of the online algorithm. Indeed, we only use the independence between $v_{e(t)}$ and $\mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)$ in the above derivations which still holds against the adaptive adversary regardless of the choice of $e(t)$ at time t.

[4] Time $T+1$ denotes the time after arrival of the last edge.

Formally, we just need to rewrite the equation (6) as follows.

$$\begin{aligned}\textbf{Surplus} = \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{\substack{i\in L\\ j\in R}} \sum_{e:e=(i,j)} \sum_{t} [v_e - l_i - r_j]^+ \cdot \mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)\cdot \mathbb{I}_{e=e(t)}\right] \\ = \mathop{\mathbf{E}}_{\mathbf{v},\widehat{\mathbf{v}}}\left[\sum_{i,j}\sum_{e:e=(i,j)}\sum_{t}[\widehat{v}_e - l_i - r_j]^+ \cdot \mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)\cdot \mathbb{I}_{e=e(\mathbf{v},t)}\right] \\ \geq \mathop{\mathbf{E}}_{\mathbf{v},\widehat{\mathbf{v}}}\left[\sum_{i,j}\sum_{e:e=(i,j)}[\widehat{v}_e - l_i - r_j]^+ \cdot \mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},T+1)\right],\end{aligned}$$

where the second equality holds because $\mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)\cdot \mathbb{I}_{e=e(\mathbf{v},t)}$ is independent from v_e; the inequality holds, since $\mathbb{I}^{i,j}_{\texttt{free}}(\mathbf{v},t)$ is a decreasing function of t and there is only one t such that $\mathbb{I}_{e=e(\mathbf{v},t)} = 1$.

3.2. Optimization of $\texttt{V-add}(\mathbf{l},\mathbf{r})$ parameters. The `V-add` policy has two variable parameters $\mathbf{l},\mathbf{r}$ representing the prices on the vertices in the left and right parts of G. We now show how to choose these parameters in order to achieve good approximation of the prophet. We cast our problem into a linear algebraic form. In this section, we derive this formulation and show how an implicitly described (via certain semi-linear equation) solution to this problem yields a 3-approximation to the prophet's expected value. We give a gradient-descent algorithm that efficiently calculates vectors $\mathbf{l},\mathbf{r}$ up to any given precision error in the following section.

Our linear algebraic formulation is based on the lower bounds (2) and (5) for the revenue and surplus terms of the vertex-additive policy $\texttt{V-add}(\mathbf{l},\mathbf{r})$. Namely, the expected value of the $\texttt{V-add}(\mathbf{l},\mathbf{r})$ is at least the sum of the right hand sides of (2) and (5)

$$\mathop{\mathbf{E}}_{\mathbf{v}}[\texttt{V-add}(\mathbf{l},\mathbf{r})] \geq \mathop{\mathbf{E}}_{\mathbf{v}}\left[\sum_{i\in X_L(\mathbf{v})} l_i + \sum_{j\in X_R(\mathbf{v})} r_j + \sum_{\substack{i\notin X_L(\mathbf{v})\\ j\notin X_R(\mathbf{v})}} \left[\mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}})\right] - (l_i + r_j)\mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}\right]\right]^+\right]. \quad (8)$$

We note that RHS of (8) is calculated in expectation over all valuation profiles $\mathbf{v}\sim\mathcal{F}$, which only determines the sets $X_L(\mathbf{v})$ and $X_R(\mathbf{v})$. We further relax the bound in RHS of (8) by letting the sets $X_L(\mathbf{v}), X_R(\mathbf{v})$ to be selected by an adversary who wants to minimize our performance guarantee. Let the worst-case sets be $S_L \subseteq L$ for $X_L(\mathbf{v})$ and $S_R \subseteq R$ for $X_R(\mathbf{v})$. Hence,

$$\mathop{\mathbf{E}}_{\mathbf{v}}[\texttt{V-add}(\mathbf{l},\mathbf{r})] \geq \min_{S_L,S_R}\left(\sum_{i\in S_L} l_i + \sum_{j\in S_R} r_j + \sum_{\substack{i\notin S_L\\ j\notin S_R}} \left[\mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\chi^{(i,j)}_{\texttt{opt}}(\widehat{\mathbf{v}})\right] - (l_i + r_j)\mathop{\mathbf{E}}_{\widehat{\mathbf{v}}}\left[\mathbb{I}^{(i,j)}_{\texttt{opt}(\widehat{\mathbf{v}})}\right]\right]^+\right). \quad (9)$$

We rewrite RHS of (9) by introducing two $n\times m$ matrices $\mathbf{M}$ and $\mathbf{Q}$ that respectively comprise expected contributions and selection probabilities to the optimal matching of the edges between all pairs of vertices $i\in L$ and $j\in R$. Formally,

$$\mathbf{M} \stackrel{\text{def}}{=} \begin{bmatrix} & \vdots & \\ \cdots & \mathop{\mathbf{E}}_{\mathbf{v}}[\chi^{(i,j)}_{\texttt{opt}}(\mathbf{v})] & \cdots \\ & \vdots & \end{bmatrix}_{i,j\in[n\times m]} \quad \text{and} \quad \mathbf{Q} \stackrel{\text{def}}{=} \begin{bmatrix} & \vdots & \\ \cdots & \mathop{\mathbf{Pr}}_{\mathbf{v}}[(i,j)\in\texttt{opt}(\mathbf{v})] & \cdots \\ & \vdots & \end{bmatrix}_{i,j\in[n\times m]} \tag{10}$$

Thus lower bound (9) can be written as

$$\mathop{\mathbf{E}}_{\mathbf{v}}[\texttt{V-add}(\mathbf{l},\mathbf{r})] \geq \min_{S_L, S_R} \left(\sum_{i\in S_L} l_i + \sum_{j\in S_R} r_j + \sum_{\substack{i\notin S_L \\ j\notin S_R}} [M_{i,j} - (l_i + r_j)\cdot Q_{i,j}]^+ \right). \tag{11}$$

On the other hand, the expected value of the optimal matching opt achieved by the prophet according to the definitions of matrix $\mathbf{M}$ (see (10),(4)) is equal to

$$\text{opt} = \mathbb{1}_L^\top \cdot \mathbf{M} \cdot \mathbb{1}_R, \tag{12}$$

where $\mathbb{1}_L, \mathbb{1}_R$ are n and m dimensional vectors with all coordinates equal to 1.

Now RHS of (11) can be rewritten using linear algebraic notations. To this end we change vectors $\mathbf{l}, \mathbf{r}$ to diagonal $n\times n$ and $m\times m$ matrices $\mathbf{L}, \mathbf{R}$, and represent sets S_L, and S_R respectively as vectors $\boldsymbol{\alpha}\in\mathbb{R}^n$, $\boldsymbol{\beta}\in\mathbb{R}^m$ with

$$\mathbf{L} \stackrel{\text{def}}{=} \text{diag}(\mathbf{l}), \quad \mathbf{R} \stackrel{\text{def}}{=} \text{diag}(\mathbf{r}), \quad \boldsymbol{\alpha}[i] \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } i\in S_L \\ 0 & \text{if } i\notin S_L \end{cases}, \qquad \boldsymbol{\beta}[j] \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } j\in S_R \\ 0 & \text{if } j\notin S_R \end{cases}. \tag{13}$$

With these notations at hand, RHS of (11) is equal to

$$\min_{\substack{\boldsymbol{\alpha}\in\{0,1\}^n \\ \boldsymbol{\beta}\in\{0,1\}^m}} \left(\boldsymbol{\alpha}^\top \cdot \mathbf{L} \cdot \mathbb{1}_L + \mathbb{1}_R^\top \cdot \mathbf{R} \cdot \boldsymbol{\beta} + (\mathbb{1}_L - \boldsymbol{\alpha})^\top \cdot [\mathbf{M} - \mathbf{L}\cdot\mathbf{Q} - \mathbf{Q}\cdot\mathbf{R}]^+ \cdot (\mathbb{1}_R - \boldsymbol{\beta}) \right), \tag{14}$$

where $[\mathbf{A}]^+$ is a matrix operator that changes each (i,j) entry of the matrix $\mathbf{A} \stackrel{\text{def}}{=} \mathbf{M} - \mathbf{L}\cdot\mathbf{Q} - \mathbf{Q}\cdot\mathbf{R}$ to $[\mathbf{A}[i,j]]^+ = \max\{0, \mathbf{A}[i,j]\}$. Our optimization problem is to find vertex prices $\mathbf{l}, \mathbf{r}$ so as to maximize expression (14) with diagonal matrices $\mathbf{L} = \text{diag}(\mathbf{l}), \mathbf{R} = \text{diag}(\mathbf{r})$. Now we can state our main approximation guarantee.

THEOREM 1. *There are vertex prices* $\mathbf{l}, \mathbf{r}$ *such that vertex-additive policy* $\texttt{V-add}(\mathbf{l},\mathbf{r}) \geq \frac{opt}{3}$.

Proof. We will find non negative price vectors $\mathbf{l}, \mathbf{r}$, s.t. (14) $\geq \frac{\text{opt}}{3}$ for $\mathbf{L} = \text{diag}(\mathbf{l}), \mathbf{R} = \text{diag}(\mathbf{r})$. Let $\mathbf{A}^+ \stackrel{\text{def}}{=} [\mathbf{A}]^+ = [\mathbf{M} - \mathbf{L}\cdot\mathbf{Q} - \mathbf{Q}\cdot\mathbf{R}]^+$ which we call *virtual surplus* matrix. We view the algebraic expression under minimization in (14) as a bilinear function of $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and rewrite it as a separate sum of the constant, linear, and bilinear terms. Note that our optimization problem at hand is the max-min semi-bilinear (we say semi-bilinear and not bilinear, because the $[]^+$ operator breaks linearity) problem where minimization is taken over two vectors $\boldsymbol{\alpha}, \boldsymbol{\beta} \in [0,1]^n$ and maximization is taken over the matrices $\mathbf{L}$ and $\mathbf{R}$ that appear in the virtual surplus matrix $\mathbf{A}^+$. We relax and simplify the problem to be semi-linear in the minimization part by dropping positive bilinear term $\boldsymbol{\alpha}^\top \cdot \mathbf{A}^+ \cdot \boldsymbol{\beta} \geq 0$.

$$\begin{aligned}(14) &= \min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \left(\mathbb{1}_L^\top \cdot \mathbf{A}^+ \cdot \mathbb{1}_R - \boldsymbol{\alpha}^\top \cdot \left[\mathbf{A}^+ \cdot \mathbb{1}_R - \mathbf{L} \cdot \mathbb{1}_L\right] - \left[\mathbb{1}_L^\top \cdot \mathbf{A}^+ - \mathbb{1}_R^\top \cdot \mathbf{R}\right] \cdot \boldsymbol{\beta} + \boldsymbol{\alpha}^\top \cdot \mathbf{A}^+ \cdot \boldsymbol{\beta}\right) \\ &\geq \min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \left(\mathbb{1}_L^\top \cdot \mathbf{A}^+ \cdot \mathbb{1}_R - \boldsymbol{\alpha}^\top \cdot \left[\mathbf{A}^+ \cdot \mathbb{1}_R - \mathbf{L} \cdot \mathbb{1}_L\right] - \left[\mathbb{1}_L^\top \cdot \mathbf{A}^+ - \mathbb{1}_R^\top \cdot \mathbf{R}\right] \cdot \boldsymbol{\beta}\right), \quad (15)\end{aligned}$$

where the last inequality holds, since $\boldsymbol{\alpha}^\top \cdot \mathbf{A}^+ \cdot \boldsymbol{\beta} \geq 0$ for any choice of $\boldsymbol{\alpha}, \boldsymbol{\beta}$ vectors (recall that all entries in the virtual surplus matrix $\mathbf{A}^+$ and vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are non negative). We believe that the relaxation (15) admits an optimal solution that satisfies the following system (16) of semi-linear equations (it is not linear, since matrix $\mathbf{A}^+$ has a non linear dependency, e.g., on $\mathbf{L}$ due to $[\cdot]^+$ operator)[5].

$$\begin{cases} \mathbf{A}^+ \cdot \mathbb{1}_R = \mathbf{L} \cdot \mathbb{1}_L \\ \mathbb{1}_L^\top \cdot \mathbf{A}^+ = \mathbb{1}_R^\top \cdot \mathbf{R}. \end{cases} \quad (16)$$

We note that if (16) holds, then RHS of (15) is independent of $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and is equal to $\mathbb{1}_L^\top \cdot \mathbf{A}^+ \cdot \mathbb{1}_R$, which we call the *virtual surplus*. For the remainder of the proof, we assume that the system of equations (16) has a solution and prove the required bound on the expected value of the prophet, opt. We give an algorithmic proof for the existence of the solution to (16) in the next section.

We observe that the following two vector inequalities in Fact 2 hold true (recall that the matrix $\mathbf{Q}$ is the probability matrix for the optimal matching to have each edge (i,j)).

FACT 2. $\mathbb{1}_L \succeq \mathbf{Q} \cdot \mathbb{1}_R$ and $\mathbb{1}_R^\top \succeq \mathbb{1}_L^\top \cdot \mathbf{Q}$, where $\succeq$ stands for the coordinate-wise inequality $\geq$.

Finally, we show that the expression (14) is at least $\frac{1}{3}$opt of the prophet's expected value.

$$\begin{aligned}3 \cdot (14) \geq 3 \cdot \mathbb{1}_L^\top \cdot \mathbf{A}^+ \cdot \mathbb{1}_R &= \mathbb{1}_L^\top \cdot \mathbf{A}^+ \cdot \mathbb{1}_R + \mathbb{1}_L^\top \cdot \mathbf{L} \cdot \mathbb{1}_L + \mathbb{1}_R^\top \cdot \mathbf{R} \cdot \mathbb{1}_R \\ &\geq \mathbb{1}_L^\top \cdot \mathbf{A}^+ \cdot \mathbb{1}_R + \mathbb{1}_L^\top \cdot \mathbf{L} \cdot \left(\mathbf{Q} \cdot \mathbb{1}_R\right) + \left(\mathbb{1}_L^\top \cdot \mathbf{Q}\right) \cdot \mathbf{R} \cdot \mathbb{1}_R \\ &= \mathbb{1}_L^\top \cdot (\mathbf{A}^+ + \mathbf{L} \cdot \mathbf{Q} + \mathbf{Q} \cdot \mathbf{R}) \cdot \mathbb{1}_R \geq \mathbb{1}_L^\top \cdot \mathbf{M} \cdot \mathbb{1}_R = \text{opt}, \quad (17)\end{aligned}$$

where the first inequality and the first equality directly follow from (16) and (15); to get the second inequality we observe that all entries in $\mathbf{L}, \mathbf{R}, \mathbb{1}_R$, and $\mathbb{1}_L$ are non negative and also use the Fact 2; to get the last inequality, we notice that matrices $\mathbf{A}^+, \mathbf{L}, \mathbf{R}, \mathbf{Q}$ and vectors $\mathbb{1}_R, \mathbb{1}_L$ have non negative entries and observe that, since $\mathbf{A}^+ = [\mathbf{M} - \mathbf{L} \cdot \mathbf{Q} - \mathbf{Q} \cdot \mathbf{R}]^+$, $n \times m$, the matrix $(\mathbf{A}^+ + \mathbf{L} \cdot \mathbf{Q} + \mathbf{Q} \cdot \mathbf{R})$ dominates $n \times m$ matrix $\mathbf{M}$ in every entry; the last equality is (12). ☐

[5] We do not provide a proof for the optimality of the solution to (16), since we do not have a short formal argument and it is not required anywhere in the proof of Theorem 1.

3.3. Algorithmic solution to the system of semi-linear equations (16) In this section, we describe a gradient descent type algorithm that solves system of equations (16) up to an arbitrary precision error ε in polynomial in n, m, and $\log(\varepsilon^{-1})$ time. We start by rewriting (16) as a system of equations in $\mathbf{l} = (l_1, \ldots, l_n)$ and $\mathbf{r} = (r_1, \ldots, r_m)$ variables.

$$\begin{cases} l_i = \sum_{j=1}^{m} [M_{i,j} - Q_{i,j}(l_i + r_j)]^+ & \forall i \in [n], \\ r_j = \sum_{i=1}^{n} [M_{i,j} - Q_{i,j}(l_i + r_j)]^+ & \forall j \in [m]. \end{cases} \tag{18}$$

Our algorithm iteratively updates the solution of $\mathbf{l}, \mathbf{r}$ to (18) reducing the total additive approximation error in every such iteration. Specifically, we look at each separate equation in (18) and write their approximation errors in two vectors $\boldsymbol{\delta}_L \in \mathbb{R}^n, \boldsymbol{\delta}_R \in \mathbb{R}^m$.

$$\boldsymbol{\delta}_L[i] \stackrel{\text{def}}{=} l_i - \sum_{j=1}^{m} [M_{i,j} - Q_{i,j}(l_i + r_j)]^+ \qquad \boldsymbol{\delta}_R[j] \stackrel{\text{def}}{=} r_j - \sum_{i=1}^{n} [M_{i,j} - Q_{i,j}(l_i + r_j)]^+ . \tag{19}$$

The update rule for $\mathbf{l}, \mathbf{r}$ in our iterative procedure is quite simple: we choose one of $\mathbf{l}$ or $\mathbf{r}$ vectors with the larger L_1-norm error in (19) and then correct this vector by subtracting half of its error vector. The description of our update rule for $\mathbf{l}, \mathbf{r}$ is summarized in Algorithm 1 below. The initialization of $\mathbf{l}$ and $\mathbf{r}$ vectors can be arbitrary, e.g., one can set both $\mathbf{l}, \mathbf{r}$ to be 0-vectors.

Algorithm 1 Iterative update algorithm solving system of equations (18).

Input: Matrices $(M_{i,j}, Q_{i,j})_{i,j \in [n] \times [m]}$ of non negative real numbers, error $\varepsilon > 0$.

Output: Vectors $\mathbf{l}, \mathbf{r}$ solving (18) up to total additive error ε.

repeat
  if $\|\boldsymbol{\delta}_L\|_1 \geq \|\boldsymbol{\delta}_R\|_1$ **then**
    update vector $\mathbf{l} \leftarrow \left(\mathbf{l} - \frac{1}{2} \cdot \boldsymbol{\delta}_L\right)$
  else
    update vector $\mathbf{r} \leftarrow \left(\mathbf{r} - \frac{1}{2} \cdot \boldsymbol{\delta}_R\right)$
  end if
  Update $\boldsymbol{\delta}_L, \boldsymbol{\delta}_R$ according to (19) with new $\mathbf{l}, \mathbf{r}$.
until $\|\boldsymbol{\delta}_L\|_1 + \|\boldsymbol{\delta}_R\|_1 \leq \varepsilon$

Theorem 2. *Algorithm 1 terminates in* $O\left(\log(\varepsilon^{-1}) + \log\left(\sum_{i,j} M_{i,j}\right)\right)$ *many steps. Moreover, the obtained solution is within* $O(\varepsilon)$ L_1*-distance to an exact solution of (18) and approximates each equality in (18) up to* $O(\varepsilon)$ *additive error.*

Proof. First, we show that our algorithm decreases the total L_1-norm of the combined error $\|\boldsymbol{\delta}_L\|_1 + \|\boldsymbol{\delta}_R\|_1$ in all iterations and thus converges to the solution of (18). Specifically, we show a constant fraction decrement in L_1-norm of the total error in each iteration.

Without loss of generality, let us assume that $\|\boldsymbol{\delta}_L\|_1 \geq \|\boldsymbol{\delta}_R\|_1$ in a given iteration. Thus we update $\mathbf{l}$ vector to a new value $\widehat{\mathbf{l}} \overset{\text{def}}{=} \mathbf{l} - \frac{1}{2} \cdot \boldsymbol{\delta}_L$ in this iteration with the respective new errors $\widehat{\boldsymbol{\delta}}_L(\widehat{\mathbf{l}}, \mathbf{r}), \widehat{\boldsymbol{\delta}}_R(\widehat{\mathbf{l}}, \mathbf{r})$ given by formula (19) for the updated $\mathbf{l} \leftarrow \widehat{\mathbf{l}}, \mathbf{r} \leftarrow \mathbf{r}$ vectors. Let us bound first the decrease in the L_1-norm of the error $\left\|\widehat{\boldsymbol{\delta}}_L\right\|_1$ compared to its previous value $\left\|\boldsymbol{\delta}_L\right\|_1$.

$$\begin{aligned}
\widehat{\boldsymbol{\delta}}_L[i] = \widehat{l}_i - \sum_{j=1}^{m} \left[M_{i,j} - Q_{i,j}\left(\widehat{l}_i + r_j\right)\right]^+ &= l_i - \frac{1}{2}\boldsymbol{\delta}_L[i] - \sum_{j=1}^{m} \left[M_{i,j} - Q_{i,j}\left(l_i - \frac{1}{2}\boldsymbol{\delta}_L[i] + r_j\right)\right]^+ \\
&= \boldsymbol{\delta}_L[i] - \frac{1}{2}\boldsymbol{\delta}_L[i] - \sum_{j=1}^{m} \left(\left[M_{i,j} - Q_{i,j}\left(l_i - \frac{1}{2}\boldsymbol{\delta}_L[i] + r_j\right)\right]^+ - \left[M_{i,j} - Q_{i,j}\left(l_i + r_j\right)\right]^+\right),
\end{aligned}$$

where to get the last equality we simply plugged in the definition of $\boldsymbol{\delta}_L[i]$ from (19). We let

$$d_{i,j} \overset{\text{def}}{=} \left[M_{i,j} - Q_{i,j}\left(l_i - \frac{1}{2}\boldsymbol{\delta}_L[i] + r_j\right)\right]^+ - \left[M_{i,j} - Q_{i,j}\left(l_i + r_j\right)\right]^+ \tag{20}$$

Hence, $\widehat{\boldsymbol{\delta}}_L[i] = \boldsymbol{\delta}_L[i] - \frac{1}{2}\boldsymbol{\delta}_L[i] - \sum_{j=1}^m d_{i,j}$ for every $i \in [n]$. We notice that $d_{i,j} = \left[A_{i,j} + \frac{1}{2}Q_{i,j} \cdot \boldsymbol{\delta}_L[i]\right]^+ - [A_{i,j}]^+$, where $A_{i,j} \overset{\text{def}}{=} M_{i,j} - Q_{i,j}(l_i + r_j)$ as per our definition of matrix $\mathbf{A}$. Now, the function $[\cdot]^+$ satisfies the following two simple properties: (i) $[x+y]^+ - [x]^+$ has the same sign as y; (ii) $\left|[x+y]^+ - [x]^+\right| \leq |y|$ for any real numbers x, y. Thus all $d_{i,j}$ for every $j \in [m]$ have the same sign as $\boldsymbol{\delta}_L[i]$ (property (i) for $A_{i,j}$ and $\frac{1}{2}Q_{i,j} \cdot \boldsymbol{\delta}_L[i]$). Moreover, by the second property

$$\left|\sum_{j=1}^{m} d_{i,j}\right| = \sum_{j=1}^{m} \left|d_{i,j}\right| \leq \sum_{j=1}^{m} \left|\frac{1}{2}Q_{i,j} \cdot \boldsymbol{\delta}_L[i]\right| \leq \frac{1}{2}\left|\boldsymbol{\delta}_L[i]\right|,$$

where the last inequality holds since $\sum_j Q_{i,j} \leq 1$ as the sum of the i-th row entries in the matching probability matrix $\mathbf{Q}$. Therefore, we have

$$\left|\widehat{\boldsymbol{\delta}}_L[i]\right| = \left|\boldsymbol{\delta}_L[i] - \frac{1}{2}\boldsymbol{\delta}_L[i] - \sum_{j=1}^{m} d_{i,j}\right| = \left|\boldsymbol{\delta}_L[i]\right| - \left(\frac{1}{2}\left|\boldsymbol{\delta}_L[i]\right| + \sum_{j=1}^{m}\left|d_{i,j}\right|\right).$$

We obtain a bound on the decrease of L_1-norm of $\widehat{\boldsymbol{\delta}}_L$ by adding previous equations over all $i \in [n]$.

$$\left\|\widehat{\boldsymbol{\delta}}_L\right\|_1 - \left\|\boldsymbol{\delta}_L\right\|_1 = \sum_{i=1}^{n}\left(\left|\widehat{\boldsymbol{\delta}}_L[i]\right| - \left|\boldsymbol{\delta}_L[i]\right|\right) = -\sum_{i=1}^{n}\frac{1}{2}\left|\boldsymbol{\delta}_L[i]\right| - \sum_{i=1}^{n}\sum_{j=1}^{m}\left|d_{i,j}\right|. \tag{21}$$

Next we bound the increase in the L_1-norm of error $\widehat{\boldsymbol{\delta}}_R$ compared to $\left\|\boldsymbol{\delta}_R\right\|_1$. We note that

$$\begin{aligned}
\widehat{\boldsymbol{\delta}}_R[j] = r_j - \sum_{i=1}^{n} \left[M_{i,j} - Q_{i,j}\left(\widehat{l}_i + r_j\right)\right]^+ &= r_j - \sum_{i=1}^{n} \left[M_{i,j} - Q_{i,j}\left(l_i - \frac{1}{2}\boldsymbol{\delta}_L[i] + r_j\right)\right]^+ \\
&= \boldsymbol{\delta}_R[j] - \sum_{i=1}^{n} \left(\left[M_{i,j} - Q_{i,j}\left(l_i - \frac{1}{2}\boldsymbol{\delta}_L[i] + r_j\right)\right]^+ - \left[M_{i,j} - Q_{i,j}\left(l_i + r_j\right)\right]^+\right),
\end{aligned}$$

for every $j \in [m]$. Thus $\widehat{\boldsymbol{\delta}}_R[j] = \boldsymbol{\delta}_R[j] - \sum_{i=1}^n d_{i,j}$ and we get

$$\left\|\widehat{\boldsymbol{\delta}}_R\right\|_1 - \left\|\boldsymbol{\delta}_R\right\|_1 = \sum_{j=1}^m \left(\left|\widehat{\boldsymbol{\delta}}_R[j]\right| - \left|\boldsymbol{\delta}_R[j]\right|\right) = \sum_{j=1}^m \left(\left|\boldsymbol{\delta}_R[j] - \sum_{i=1}^n d_{i,j}\right| - \left|\boldsymbol{\delta}_R[j]\right|\right) \le \sum_{j=1}^m \sum_{i=1}^n \left|d_{i,j}\right|. \tag{22}$$

Finally, we combine bounds (21) and (22) to show constant factor drop in L_1-norm of errors $\widehat{\boldsymbol{\delta}}_L, \widehat{\boldsymbol{\delta}}_R$.

$$\begin{aligned}\left\|\widehat{\boldsymbol{\delta}}_L\right\|_1 + \left\|\widehat{\boldsymbol{\delta}}_R\right\|_1 - \left\|\boldsymbol{\delta}_L\right\|_1 - \left\|\boldsymbol{\delta}_R\right\|_1 &= \left(\left\|\widehat{\boldsymbol{\delta}}_L\right\|_1 - \left\|\boldsymbol{\delta}_L\right\|_1\right) + \left(\left\|\widehat{\boldsymbol{\delta}}_R\right\|_1 - \left\|\boldsymbol{\delta}_R\right\|_1\right) \\ &\le -\sum_i \frac{1}{2}\left|\boldsymbol{\delta}_L[i]\right| - \sum_{i,j}\left|d_{i,j}\right| + \sum_{i,j}\left|d_{i,j}\right| = -\frac{1}{2}\left\|\boldsymbol{\delta}_L\right\|_1 \le -\frac{1}{4}\left(\left\|\boldsymbol{\delta}_L\right\|_1 + \left\|\boldsymbol{\delta}_R\right\|_1\right),\end{aligned}$$

where the last inequality holds because we assumed $\|\boldsymbol{\delta}_L\|_1 \ge \|\boldsymbol{\delta}_R\|_1$. That means that the L_1-norm of the combined error decreases by a factor of $\frac{3}{4}$ in every round and after t steps the L_1-norm of the combined error $(\boldsymbol{\delta}_L(t), \boldsymbol{\delta}_R(t))$ will be $\|\boldsymbol{\delta}_L(t)\|_1 + \|\boldsymbol{\delta}_R(t)\|_1 \le (\frac{3}{4})^t(\|\boldsymbol{\delta}_L(0)\|_1 + \|\boldsymbol{\delta}_R(0)\|_1)$, where $\|\boldsymbol{\delta}_L(0)\|_1 = \sum_{i=1}^n\sum_{j=1}^m M_{i,j} = \|\boldsymbol{\delta}_R(0)\|_1$. Thus Algorithm 1 terminates in $O\left(\log(\varepsilon^{-1}) + \log\left(\sum_{i,j} M_{i,j}\right)\right)$ steps. Furthermore, if we let the Algorithm 1 to continue indefinitely, it will produce a sequence of states $\mathbf{s}_t \stackrel{\text{def}}{=} (\mathbf{l}(t), \mathbf{r}(t)) \in \mathbb{R}^{n+m}$ for each iteration t that converges to an exact solution of (18) as $t \to +\infty$. Indeed, we can show that $\mathbf{s}_t$ is a Cauchy sequence, i.e., for any $t_2 > t_1 \ge T$ we have

$$\|\mathbf{s}_{t_2} - \mathbf{s}_{t_1}\|_1 \le \sum_{t=T}^{+\infty}\|\mathbf{s}_t - \mathbf{s}_{t+1}\|_1 \le \sum_{t=T}^{+\infty}\left(\frac{\|\boldsymbol{\delta}_L(t)\|_1}{2} + \frac{\|\boldsymbol{\delta}_R(t)\|_1}{2}\right) \le \sum_{t=T}^{+\infty}\frac{\varepsilon}{2}\cdot\left(\frac{3}{4}\right)^{t-T} = O(\varepsilon),$$

since $\|\mathbf{s}_t - \mathbf{s}_{t+1}\|_1 = \frac{1}{2}\max\left(\|\boldsymbol{\delta}_L(t)\|_1, \|\boldsymbol{\delta}_R(t)\|_1\right)$ for any t and $\|\boldsymbol{\delta}_L(t)\|_1 + \|\boldsymbol{\delta}_R(t)\|_1 \le (\frac{3}{4})^{t-T}(\|\boldsymbol{\delta}_L(T)\|_1 + \|\boldsymbol{\delta}_R(T)\|_1) \le (\frac{3}{4})^{t-T}\cdot\varepsilon$ for any $t \ge T$. As $\mathbb{R}^{n+m}$ is a complete space, the Cauchy sequence $(\mathbf{s}_t)_{t=0}^{\infty}$ has a limit $\mathbf{s} \in \mathbb{R}^{n+m}$ which must be an exact solution to (18) as the combined error $(\boldsymbol{\delta}_L(t), \boldsymbol{\delta}_R(t))$ goes to 0 with $t \to +\infty$. The last equation also shows that the result $\mathbf{s}_T = (\mathbf{l}(T), \mathbf{r}(T))$ of Algorithm 1 is within $O(\varepsilon)$ distance from the exact solution $\mathbf{s}$. □

Influence of numerical errors. We showed in section 3.3 the existence of vertex prices $\mathbf{l}, \mathbf{r}$ and presented an efficient algorithm that computes these prices up to $O(\varepsilon)$ error. Note that matrices $\mathbf{M}$ and $\mathbf{Q}$ are not given precisely due to the estimation error in the Monte Carlo simulations (similar to the computation of the optimum in [27],[19],[20]). Thus it is important to understand how this computation and estimation errors affect approximation guarantees of the respective vertex additive policy `V-add`$(\mathbf{l}, \mathbf{r})$.

Suppose that our estimation of the matrices $\mathbf{M}$, $\mathbf{Q}$, and numerical solution to (16) have error of order $O(\varepsilon)$ per every matrix and vector entry. Following prior work, we can assume that all the values are normalized to be in $[0,1]$, i.e., the entries of the matrices $\mathbf{M}, \mathbf{Q}$ lie in $[0,1]$. We claim that these numerical errors translate into $O(\text{poly}(\varepsilon, n, m))$ additive error for the vertex additive policy.

THEOREM 3. *Suppose we have sample access to the distribution of edge values, then for any $\delta > 0$ we can compute vertex prices $\mathbf{l}, \mathbf{r}$ in poly$(1/\delta, n, m)$ time such that* `V-add`$(\mathbf{l}, \mathbf{r}) \ge \frac{opt}{3} - \delta$.

Proof. All the derivations up to equation (15) are done for the arbitrary price vectors $\mathbf{l}$ and $\mathbf{r}$, regardless of how we compute these vertex prices. We can assume that the Algorithm 1 receives imprecise estimations of $\widetilde{\mathbf{M}}$, $\widetilde{\mathbf{Q}}$ and then gets approximate solution to (16) which has $O(\varepsilon)$ error in each equality of (16). I.e., Algorithm 1 finds prices $\widetilde{\mathbf{L}}, \widetilde{\mathbf{R}}$ satisfying

$$\begin{cases} \left[\widetilde{\mathbf{M}} - \widetilde{\mathbf{L}} \cdot \widetilde{\mathbf{Q}} - \widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{R}}\right]^+ \cdot \mathbb{1}_R - \widetilde{\mathbf{L}} \cdot \mathbb{1}_L = O(\boldsymbol{\varepsilon}) \\ \mathbb{1}_L^\top \cdot \left[\widetilde{\mathbf{M}} - \widetilde{\mathbf{L}} \cdot \widetilde{\mathbf{Q}} - \widetilde{\mathbf{Q}} \cdot \widetilde{\mathbf{R}}\right]^+ - \mathbb{1}_R^\top \cdot \widetilde{\mathbf{R}} = O(\boldsymbol{\varepsilon}), \end{cases}$$

where $O(\boldsymbol{\varepsilon})$ denotes a vector with every entry of order $O(\varepsilon)$. We can assume that the prices $\widetilde{\mathbf{L}}, \widetilde{\mathbf{R}}$ also lie in $[0,1]$. If we substitute $\widetilde{\mathbf{M}}, \widetilde{\mathbf{Q}}$ with the correct estimates $\mathbf{M}, \mathbf{Q}$ we get

$$\begin{cases} \left[\mathbf{M} - \widetilde{\mathbf{L}} \cdot \mathbf{Q} - \mathbf{Q} \cdot \widetilde{\mathbf{R}}\right]^+ \cdot \mathbb{1}_R - \widetilde{\mathbf{L}} \cdot \mathbb{1}_L = n \cdot m \cdot O(\boldsymbol{\varepsilon}) \\ \mathbb{1}_L^\top \cdot \left[\mathbf{M} - \widetilde{\mathbf{L}} \cdot \mathbf{Q} - \mathbf{Q} \cdot \widetilde{\mathbf{R}}\right]^+ - \mathbb{1}_R^\top \cdot \widetilde{\mathbf{R}} = n \cdot m \cdot O(\boldsymbol{\varepsilon}), \end{cases}$$

Let $\widetilde{\mathbf{A}}^+ \stackrel{\text{def}}{=} \left[\mathbf{M} - \widetilde{\mathbf{L}} \cdot \mathbf{Q} - \mathbf{Q} \cdot \widetilde{\mathbf{R}}\right]^+$. Then we can write similar derivation to (17) and keep track of the errors.

$$\begin{aligned} 3 \cdot (14) \geq 3 \min_{\boldsymbol{\alpha} \in [0,1]^n, \boldsymbol{\beta} \in [0,1]^m} & \left(\mathbb{1}_L^\top \cdot \widetilde{\mathbf{A}}^+ \cdot \mathbb{1}_R - \boldsymbol{\alpha}^\top \cdot \left[\widetilde{\mathbf{A}}^+ \cdot \mathbb{1}_R - \widetilde{\mathbf{L}} \cdot \mathbb{1}_L\right] - \left[\mathbb{1}_L^\top \cdot \widetilde{\mathbf{A}}^+ - \mathbb{1}_R^\top \cdot \widetilde{\mathbf{R}}\right] \cdot \boldsymbol{\beta}\right) \\ &= 3 \cdot \mathbb{1}_L^\top \cdot \widetilde{\mathbf{A}}^+ \cdot \mathbb{1}_R - \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \left(\boldsymbol{\alpha}^\top \cdot nm \cdot O(\boldsymbol{\varepsilon}) + nm \cdot O(\boldsymbol{\varepsilon})^\top \cdot \boldsymbol{\beta}\right) \\ &\geq \mathbb{1}_L^\top \cdot \widetilde{\mathbf{A}}^+ \cdot \mathbb{1}_R + \mathbb{1}_L^\top \cdot \widetilde{\mathbf{L}} \cdot \mathbb{1}_L + \mathbb{1}_R^\top \cdot \widetilde{\mathbf{R}} \cdot \mathbb{1}_R - 3nm \cdot (n+m) \cdot O(\varepsilon) \\ &\geq \mathbb{1}_L^\top \cdot (\widetilde{\mathbf{A}}^+ + \widetilde{\mathbf{L}} \cdot \mathbf{Q} + \mathbf{Q} \cdot \widetilde{\mathbf{R}}) \cdot \mathbb{1}_R - \text{poly}\,(n,m)\, O(\varepsilon) \geq \text{opt} - \text{poly}\,(n,m)\, O(\varepsilon). \end{aligned}$$

By setting ε to be $\delta/poly(n,m)$ we conclude the proof of the theorem. □

4. Lower Bound In this section we prove a lower bound of 2.25. We construct a sequence of bipartite graphs $G_n(L,R,E)$ with $L = R = [3n]$ such that the ratio between offline and online optimum converges to 2.25 as n goes to infinity. Edges of the graph G_n arrive in three consecutive batches: first arrive edges of E_1, then E_2, then E_3. The distributions of the edge values in E_1, E_2, and E_3 are as follows.

$$\begin{aligned} E_1 &= \{(i, i+n) | i \in [n]\} \cup \{(j+n, j) | j \in [n]\} && v_e = {}^1/_2 \\ E_2 &= \{(i, i+2n) | i \in [n]\} \cup \{(j+2n, j) | j \in [n]\} && \{\mathbf{Pr}\left[v_e = {}^3/_4\right] = 0.5, \quad \mathbf{Pr}\left[v_e = 0\right] = 0.5\} \\ E_3 &= \{(i,j) | i,j \in [n]\} && \{\mathbf{Pr}\left[v_e = n\right] = {}^1/_{n^2}, \quad \mathbf{Pr}\left[v_e = 0\right] = 1 - {}^1/_{n^2}\} \end{aligned}$$

The idea is that the online algorithm needs to decide whether to take edges with guaranteed low value of ${}^1/_2$ from E_1 without seeing the realization of the future edges, then the online algorithm receives middle value edges $v_e = {}^3/_4$ from E_2 without knowing which large value rare edges $v_e = n$

from E_3 will arrive at the end. The optimum offline solution should take as many large edges from E_3 with $v_e = n$ as possible, then it matches each edge from E_2 with $v_e = 3/4$ (if $i \in [n]$ or $j \in [n]$ was not covered by an edge from E_3), and finally match all unmatched vertices $i, j \in [n]$ using low value edges in E_1.

Before going into technical details of the analysis, let us first give a small warm up example with a weaker lower bound of 2.11 that illustrates the key features of our more general asymptotic example.

EXAMPLE 1. Consider a bipartite graph with the left part $L = \{1, 2, 3\}$ and the right part $R = \{a, b, c\}$. The edges $(1, c)$ and $(3, a)$ arrive first, both having deterministic value of 1. After that edges $(1, b)$ and $(2, a)$ arrive, each edge having higher value of $M = 1.5$ with probability 0.5 and value 0 otherwise ($(1, b)$ arrives before $(2, a)$). Finally, edge $(1, a)$ arrives having a large value $1/\varepsilon$ with small probability 2ε and value 0 otherwise (we think of ε being arbitrary small number).

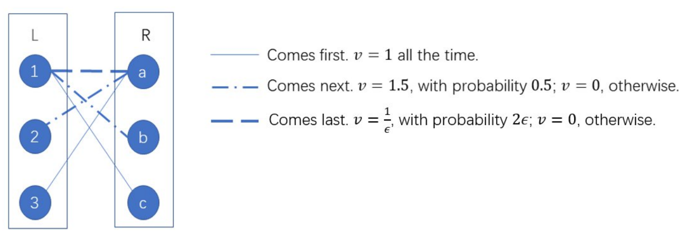

FIGURE 1. Example that shows a lower bound of 2.11.

The best the online algorithm can do is to skip edges $(1, c)$ and $(3, a)$ and then look at the value of the edge $(1, b)$. If $v(1, b) = 1.5$, then the algorithm takes the edge $(1, b)$ and the next edge $(2, a)$ regardless of its value; if $v(1, b) = 0$ the algorithm skips both edges $(1, b)$, $(2, a)$, and takes the edge $(1, a)$. The expected value obtained by such online policy is $\mathbf{Pr}[v(1, b) = 1.5] \cdot (1.5 + \mathbf{Pr}[v(2, a) = 1.5] \cdot 1.5) + \mathbf{Pr}[v(1, b) = 0] \cdot \mathbf{Pr}[v(1, a) = 1/\varepsilon] \cdot 1/\varepsilon = 0.5 \cdot (1.5 + 1.5 \cdot 0.5) + 0.5 \cdot 2\varepsilon \cdot 1/\varepsilon = 2.125$. This is indeed the best policy, because if the algorithm decides to take any of the $(1, c)$ and $(3, a)$ edges, then it would better take both of them with the total value of $2 < 2.125$ (after that the algorithm can only hope to get value $M = 1.5$ for the remaining edge with probability 0.5, i.e., expected value is $0.75 < 1$). If the algorithm passes on the deterministic edges and also decides to pass on the middle value edges $(1, b)$ and $(2, a)$, then its expected gain comes only from the last edge $\mathbf{Pr}[v(1, a) = 1/\varepsilon] \cdot 1/\varepsilon = 2 < 2.125$. Now, the expected value of the optimal matching is $\text{opt} = \mathbf{Pr}[v(1, a) = 1/\varepsilon] \cdot 1/\varepsilon + \mathbf{Pr}[v(1, a) = 0] \cdot (\mathbf{Pr}[v(1, b) = 1.5] \cdot 1.5 + \mathbf{Pr}[v(1, b) = 0] \cdot 1 + \mathbf{Pr}[v(2, a) = 1.5] \cdot 1.5 + \mathbf{Pr}[v(2, a) = 0] \cdot 1) = 4.5$. This gives us a lower bound of $4.5/2.125 = 2.117$ for the above simple example with only 5 edges.

We use a similar although more involved argument to obtain our stronger lower bound of 2.25.

THEOREM 4. *For any $\varepsilon > 0$ there is an instance such that no online algorithm has competitive ratio better than $2.25 - \varepsilon$ against the prophet.*

Proof We begin by first writing a lower bound on the expected offline optimum opt.

$$\text{opt} = n \sum_{i,j\in[n]} \mathbf{Pr}\left[(i,j) \in \texttt{opt}(\mathbf{v})\right] + \sum_{i\in[n]} \left(\frac{3}{4}\mathbf{Pr}\left[(i,i+2n) \in \texttt{opt}(\mathbf{v})\right] + \frac{1}{2}\mathbf{Pr}\left[(i,i+n) \in \texttt{opt}(\mathbf{v})\right]\right) + \sum_{j\in[n]} \left(\frac{3}{4}\mathbf{Pr}\left[(j+2n,j) \in \texttt{opt}(\mathbf{v})\right] + \frac{1}{2}\mathbf{Pr}\left[(j+n,j) \in \texttt{opt}(\mathbf{v})\right]\right) \quad (23)$$

For the first term corresponding to the large edges $e \in E_3$ in $\texttt{opt}(\mathbf{v})$ we have

$$n \sum_{i,j\in[n]} \mathbf{Pr}\left[(i,j) \in \texttt{opt}(\mathbf{v})\right] \geq n \sum_{i,j} \mathbf{Pr}\left[v_{(i,j)} = n\right] \cdot \mathbf{Pr}\left[\forall e \in E_3 \text{ s.t. } e \neq (i,j), e \cap (i,j) \neq \emptyset : v_e = 0\right] \geq n \cdot n^2 \cdot \frac{1}{n^2} \cdot \left(1 - \frac{2n-2}{n^2}\right) = n - O(1), \quad (24)$$

where the first inequality is due to the fact that $\texttt{opt}(\mathbf{v})$ must include every large edge $v_{(i,j)} = n$ that does not share a vertex i or j with any other large edge $v_e = n$; to get the second inequality we apply union bound to the edges $e \in E_3$ that share a vertex with (i,j). For the second term we get

$$\sum_{i\in[n]} \left(\frac{3}{4}\mathbf{Pr}\left[(i,i+2n) \in \texttt{opt}(\mathbf{v})\right] + \frac{1}{2}\mathbf{Pr}\left[(i,i+n) \in \texttt{opt}(\mathbf{v})\right]\right) = \sum_{i\in[n]} \mathbf{Pr}\left[\forall j \in [n]\ (i,j) \notin \texttt{opt}(\mathbf{v})\right] \cdot \left(\frac{3}{4}\mathbf{Pr}\left[v_{(i,i+2n)} = \frac{3}{4}\right] + \frac{1}{2}\mathbf{Pr}\left[v_{(i,i+2n)} = 0\right]\right) \geq n \cdot \left(1 - \frac{n}{n^2}\right) \cdot \left(\frac{3}{4}\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2}\right) = \frac{5}{8}n - O(1), \quad (25)$$

where the first equality holds, as the prophet should take $(i, i+2n)$ with $v_{(i,i+2n)} = \frac{3}{4}$ if i is not covered by the edges of E_3 in $\texttt{opt}(\mathbf{v})$; to get the first inequality we use union bound for n edges $(i,k) \in E_3$. We get the same bound as (25) for the third term in (23). We combine lower bound (24), and two lower bounds (25) to get

$$\text{opt} \geq \frac{9}{4}n - O(1) \quad (26)$$

Now, we derive an upper bound on the expected payoff of any online algorithm $\mathcal{A}$. First, notice that there is no randomness in the values of the edges $e \in E_1$. Thus we can assume without loss of generality that $\mathcal{A}$ chooses some fixed number of edges from E_1. Then the algorithm picks some random number of edges $\mathbb{Y}$ from the second group E_2, and finally takes a random number $\mathbb{X}$ of edges from the third group E_3:

$$z \stackrel{\text{def}}{=} |\{e \in E_1 | e \notin \mathcal{A}(\mathbf{v})\}| \qquad \mathbb{Y} \stackrel{\text{def}}{=} |\{e \in E_2 | e \in \mathcal{A}(\mathbf{v})\}| \qquad \mathbb{X} \stackrel{\text{def}}{=} |\{e \in E_3 | e \in \mathcal{A}(\mathbf{v})\}|$$

Then the expected payoff of the algorithm is as follows

$$\mathbf{E}\left[\sum_{e\in\mathcal{A}(\mathbf{v})} v_e\right] = \frac{1}{2}\cdot(2n - z) + \frac{3}{4}\cdot\mathbf{E}\left[\mathbb{Y}\right] + n\cdot\mathbf{E}\left[\mathbb{X}\right]. \quad (27)$$

First, we get an upper bound on $\mathbf{E}[\mathbb{X}]$ via $\mathbb{Y}$ and z. Indeed, if z_L, z_R are the number of vertices not covered by $\mathcal{A}$ in the left $i \in [n]$ and the right $j \in [n]$ sides of G_n after arrival of E_1 edges and $\mathbb{Y}_L, \mathbb{Y}_R$ are the number of E_2 edges taken by $\mathcal{A}$ of the form $(i, i+2n), (j+2n, j)$, respectively, then

$$\mathbf{E}[\mathbb{X}] \le \sum_{i,j\in[n]} \mathbf{Pr}\left[v_{(i,j)} = n\right] \cdot \mathbb{I}(i, j \text{ not covered}) = \frac{1}{n^2} \cdot (z_L - \mathbb{Y}_L)(z_R - \mathbb{Y}_R) \le \left(\frac{z - \mathbb{Y}}{2n}\right)^2,$$

where the last inequality holds since $z = z_L + z_R$ and $\mathbb{Y} = \mathbb{Y}_L + \mathbb{Y}_R$. We continue with the upper bound on the expected payoff (27) as follows

$$\begin{aligned}\mathbf{E}\left[\sum_{e\in\mathcal{A}(\mathbf{v})} v_e\right] &\le n - \frac{z}{2} + \mathbf{E}\left[\frac{3}{4}\mathbb{Y} + n\left(\frac{z-\mathbb{Y}}{2n}\right)^2\right] \\ &= n\sum_{y=0}^{2n}\left(1 - \frac{1}{2}\frac{z}{n} + \frac{3}{4}\frac{y}{n} + \frac{z^2 - 2yz + y^2}{4n^2}\right)\mathbf{Pr}\left[\mathbb{Y} = y\right] = n\sum_{y=0}^{2n} f\left(\frac{y}{n}, \frac{z}{n}\right)\mathbf{Pr}\left[\mathbb{Y} = y\right], \quad (28)\end{aligned}$$

where $f(\beta,\gamma) \stackrel{\text{def}}{=} 1 - \frac{\gamma}{2} + \frac{3\beta}{4} + \frac{(\gamma-\beta)^2}{4}$ for $\gamma = \frac{z}{n}$ and $\beta = \frac{y}{n}$ with $0 \le \beta \le \gamma \le 2$, as $0 \le \mathbb{Y} \le z \le 2n$. We observe that the random variable $\mathbb{Y}$ is stochastically dominated by $\texttt{Binomial}(\frac{1}{2}, z)$, since $\mathcal{A}$ should not take the edges $e \in E_2$ with $v_e = 0$ ($\mathcal{A}$ may or may not take $v_e = \frac{3}{4}$). Therefore, we have by Chebyshev's inequality

$$\mathbf{Pr}\left[\frac{\mathbb{Y}}{n} \ge \frac{z}{2n} + \frac{n^{3/4}}{n}\right] \le \mathbf{Pr}_{y\sim\texttt{Bernoulli}(\frac{1}{2},z)}\left[\frac{y}{n} \ge \frac{\mathbf{E}[y]}{n} + \frac{n^{3/4}}{n}\right] = o(1) \quad \text{as} \quad n \to +\infty$$

We note that $f(\beta,\gamma) < 3$ is bounded[6] for all values of z and y in the support of $\mathbb{Y}$. We plug this and Chebyshev's bound in (28) and get

$$\mathbf{E}\left[\sum_{e\in\mathcal{A}(\mathbf{v})} v_e\right] \le n\sum_{y=0}^{\frac{z}{2}+n^{3/4}} f\left(\frac{y}{n}, \frac{z}{n}\right)\mathbf{Pr}\left[\mathbb{Y} = y\right] + n \cdot 3 \cdot o(1). \quad (29)$$

We have $\frac{y}{n} - \frac{z}{2n} = \beta - \frac{\gamma}{2} \le o(1)$ for all y in the summation range of RHS(29) ($y \le \frac{z}{2} + n^{3/4}$). One can rewrite $f(\beta,\gamma)$ as follows

$$f(\beta,\gamma) = 1 + \underbrace{\left(\beta - \frac{\gamma}{2}\right)}_{\le o(1)} \cdot \underbrace{\left(\frac{3}{8}(2-\gamma) + \frac{\beta}{4}\right)}_{1\ge \text{ also } \ge 0} - \underbrace{\frac{\gamma}{16}(2-\gamma)}_{\ge 0} \quad \text{where} \quad \gamma \le 2.$$

We have $f(\beta,\gamma) = 1 + o(1)$ for any $\beta = \frac{y}{n} \le \frac{\gamma}{2} + n^{-1/4}$ where $y \le \frac{z}{2} + n^{3/4}$. Thus (29) implies

$$\mathbf{E}\left[\sum_{e\in\mathcal{A}(\mathbf{v})} v_e\right] \le n\Big(1 + o(1)\Big)\mathbf{Pr}\left[\mathbb{Y} \le z/2 + n^{3/4}\right] + n \cdot o(1) \le n\Big(1 + o(1)\Big).$$

Therefore, any online algorithm $\mathcal{A}$ cannot achieve better guarantee than $\frac{9}{4} - o(1)$ fraction of the prophet (see (26)) on the series of graphs G_n as n goes to infinity. This gives us the desired lower bound. □

[6] Indeed $f(\beta,\gamma) = 1 + \frac{\beta}{4} + \frac{(\gamma-\beta)^2}{4} - \frac{\gamma-\beta}{2} \le 1 + \frac{1}{2} + \frac{2^2}{4} - 0 < 3$ for $0 \le \beta \le \gamma \le 2$.

References

[1] Aggarwal G, Goel G, Karande C, Mehta A (2011) Online vertex-weighted bipartite matching and single-bid budgeted allocations. *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, 1253–1264, URL http://dx.doi.org/10.1137/1.9781611973082.95.

[2] Alaei S (2014) Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.* 43(2):930–972, URL http://dx.doi.org/10.1137/120878422.

[3] Alaei S, Hajiaghayi M, Liaghat V (2012) Online prophet-inequality matching with applications to ad allocation. *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, 18–35, URL http://dx.doi.org/10.1145/2229012.2229018.

[4] Alaei S, Hajiaghayi M, Liaghat V (2013) The online stochastic generalized assignment problem. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, 11–25, URL http://dx.doi.org/10.1007/978-3-642-40328-6_2.

[5] Ashlagi I, Burq M, Jaillet P, Saberi A (2018) Maximizing efficiency in dynamic matching markets. *CoRR* abs/1803.01285, URL http://arxiv.org/abs/1803.01285.

[6] Azar P, Kleinberg R, Weinberg M (2014) Prophet inequalities with limited information. *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, 1358–1377, URL http://dx.doi.org/10.1137/1.9781611973402.100.

[7] Buchbinder N, Jain K, Naor J (2007) Online primal-dual algorithms for maximizing ad-auctions revenue. *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, 253–264, URL http://dx.doi.org/10.1007/978-3-540-75520-3_24.

[8] Buchbinder N, Segev D, Tkach Y (2017) Online algorithms for maximum cardinality matching with edge arrivals. *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, 22:1–22:14, URL http://dx.doi.org/10.4230/LIPIcs.ESA.2017.22.

[9] Chawla S, Hartline J, Kleinberg R (2007) Algorithmic pricing via virtual valuations. *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*, 243–251, URL http://dx.doi.org/10.1145/1250910.1250946.

[10] Chawla S, Hartline J, Malec D, Sivan B (2010) Multi-parameter mechanism design and sequential posted pricing. *STOC*, 311–320.

[11] Chiplunkar A, Tirodkar S, Vishwanathan S (2015) On randomized algorithms for matching in the online preemptive model. *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, 325–336, URL http://dx.doi.org/10.1007/978-3-662-48350-3_28.

[12] Cohen IR, Eden A, Fiat A, Jez L (2015) Pricing online decisions: Beyond auctions. *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, 73–91, URL http://dx.doi.org/10.1137/1.9781611973730.7.

[13] Cohen-Addad V, Eden A, Feldman M, Fiat A (2016) The invisible hand of dynamic market pricing. *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, 383–400, URL http://dx.doi.org/10.1145/2940716.2940730.

[14] Correa J, Cristi A, Epstein B, Soto J (2020) The two-sided game of googol and sample-based prophet inequalities. Chawla S, ed., *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, 2066–2081 (SIAM), URL http://dx.doi.org/10.1137/1.9781611975994.127.

[15] Correa J, Dütting P, Fischer F, Schewior K (2019) Prophet inequalities for I.I.D. random variables from an unknown distribution. Karlin A, Immorlica N, Johari R, eds., *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, 3–17 (ACM), URL http://dx.doi.org/10.1145/3328526.3329627.

[16] Correa J, Foncea P, Pizarro D, Verdugo V (2019) From pricing to prophets, and back! *Oper. Res. Lett.* 47(1):25–29, URL http://dx.doi.org/10.1016/j.orl.2018.11.010.

[17] Devanur N, Hayes T (2009) The adwords problem: online keyword matching with budgeted bidders under random permutations. *Proceedings 10th ACM Conference on Electronic Commerce (EC-2009), Stanford, California, USA, July 6–10, 2009*, 71–78, URL http://dx.doi.org/10.1145/1566374.1566384.

[18] Devanur N, Jain K, Sivan B, Wilkens C (2019) Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *J. ACM* 66(1):7:1–7:41, URL https://dl.acm.org/citation.cfm?id=3284177.

[19] Duetting P, Feldman M, Kesselheim T, Lucier B (2017) Prophet inequalities made easy: Stochastic optimization by pricing non-stochastic inputs. *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, 540–551, URL http://dx.doi.org/10.1109/FOCS.2017.56.

[20] Dütting P, Kesselheim T (2019) Posted pricing and prophet inequalities with inaccurate priors. Karlin A, Immorlica N, Johari R, eds., *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, 111–129 (ACM), URL http://dx.doi.org/10.1145/3328526.3329576.

[21] Dütting P, Kesselheim T, Lucier B (2020) An o(log log m) prophet inequality for subadditive combinatorial auctions. *FOCS*, 306–317 (IEEE).

[22] Dütting P, Kleinberg R (2015) Polymatroid prophet inequalities. *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, 437–449, URL http://dx.doi.org/10.1007/978-3-662-48350-3_37.

[23] Epstein L, Levin A, Segev D, Weimann O (2013) Improved bounds for online preemptive matching. *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany*, 389–399, URL http://dx.doi.org/10.4230/LIPIcs.STACS.2013.389.

[24] Ezra T, Feldman M, Gravin N, Tang Z (2020) Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. Biró P, Hartline J, Ostrovsky M, Procaccia A, eds., *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, 769–787 (ACM), URL http://dx.doi.org/10.1145/3391403.3399513.

[25] Feldman J, Mehta A, Mirrokni V, Muthukrishnan S (2009) Online stochastic matching: Beating 1-1/e. *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, 117–126, URL http://dx.doi.org/10.1109/FOCS.2009.72.

[26] Feldman M, Fiat A, Roytman A (2017) Makespan minimization via posted prices. *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, 405–422, URL http://dx.doi.org/10.1145/3033274.3085129.

[27] Feldman M, Gravin N, Lucier B (2015) Combinatorial auctions via posted prices. *SODA*, 123–135, URL http://dx.doi.org/10.1137/1.9781611973730.10.

[28] Feldman M, Svensson O, Zenklusen R (2016) Online contention resolution schemes. *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, 1014–1033, URL http://dx.doi.org/10.1137/1.9781611974331.ch72.

[29] Goel G, Mehta A (2008) Online budgeted matching in random input models with applications to adwords. *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, 982–991, URL http://dl.acm.org/citation.cfm?id=1347082.1347189.

[30] Haeupler B, Mirrokni V, Zadimoghaddam M (2011) Online stochastic weighted matching: Improved approximation algorithms. *Internet and Network Economics - 7th International Workshop, WINE 2011, Singapore, December 11-14, 2011. Proceedings*, 170–181, URL http://dx.doi.org/10.1007/978-3-642-25510-6_15.

[31] Hajiaghayi M, Kleinberg R, Sandholm T (2007) Automated online mechanism design and prophet inequalities. *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, 58–65, URL http://www.aaai.org/Library/AAAI/2007/aaai07-009.php.

[32] Hill TP, Kertz RP (1992) A survey of prophet inequalities in optimal stopping theory. *Contemporary Mathematics* 125:191–207.

[33] Huang Z, Kang N, Tang Z, Wu X, Zhang Y, Zhu X (2018) How to match when all vertices arrive online. *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, 17–29, URL http://dx.doi.org/10.1145/3188745.3188858.

[34] Huang Z, Peng B, Tang Z, Tao R, Wu X, Zhang Y (2019) Tight competitive ratios of classic matching algorithms in the fully online model. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, 2875–2886, URL http://dx.doi.org/10.1137/1.9781611975482.178.

[35] Karande C, Mehta A, Tripathi P (2011) Online bipartite matching with unknown distributions. *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 587–596, URL http://dx.doi.org/10.1145/1993636.1993715.

[36] Karp R, Vazirani U, Vazirani V (1990) An optimal algorithm for on-line bipartite matching. *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, 352–358, URL http://dx.doi.org/10.1145/100216.100262.

[37] Kennedy DP (1985) Optimal stopping of independent random variables and maximizing prophets. *Annals of Probability* 13(2):566–571, URL http://dx.doi.org/10.1214/aop/1176993009.

[38] Kennedy DP (1987) Prophet-type inequalities for multi-choice optimal stopping. *Stoch. Proc. Appl.* 24:77–88.

[39] Kertz RP (1986) Comparison of optimal value and constrained maxima expectations for independent random variables. *Adv. Appl. Prob.* 18:311–340.

[40] Kleinberg R, Weinberg M (2012) Matroid prophet inequalities. *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, 123–136.

[41] Krengel U, Sucheston L (1977) Semiamarts and finite values. *Bull. Amer. Math. Soc.* 83(4):745–747, URL https://projecteuclid.org:443/euclid.bams/1183538915.

[42] Mahdian M, Yan Q (2011) Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 597–606, URL http://dx.doi.org/10.1145/1993636.1993716.

[43] Manshadi VH, Gharan SO, Saberi A (2011) Online stochastic matching: Online actions based on offline statistics. *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, 1285–1294, URL http://dx.doi.org/10.1137/1.9781611973082.98.

[44] McGregor A (2005) Finding graph matchings in data streams. *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, 170–181, URL http://dx.doi.org/10.1007/11538462_15.

[45] Mehta A, Saberi A, Vazirani U, Vazirani V (2007) Adwords and generalized online matching. *J. ACM* 54(5):22, URL http://dx.doi.org/10.1145/1284320.1284321.

[46] Mirrokni V, Gharan S, Zadimoghaddam M (2012) Simultaneous approximations for adversarial and stochastic online budgeted allocation. *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, 1690–1701, URL http://portal.acm.org/citation.cfm?id=2095250&CFID=63838676&CFTOKEN=79617016.

[47] Rubinstein A (2016) Beyond matroids: secretary problem and prophet inequality with general constraints. *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 324–332, URL http://dx.doi.org/10.1145/2897518.2897540.

[48] Rubinstein A, Vardi S (2017) Sorting from noisier samples. *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, 960–972, URL http://dx.doi.org/10.1137/1.9781611974782.60.

[49] Rubinstein A, Wang J, Weinberg M (2020) Optimal single-choice prophet inequalities from samples. Vidick T, ed., *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, 60:1–60:10 (Schloss Dagstuhl - Leibniz-Zentrum für Informatik), URL http://dx.doi.org/10.4230/LIPIcs.ITCS.2020.60.

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