Proportional Allocation: Simple, Distributed, and Diverse Matching with High Entropy

Shipra Agrawal [*1] **Vahab Mirrokni** [2] **Morteza Zadimoghaddam** [2]

Abstract

Inspired by many applications of bipartite matching in online advertising and machine learning, we study a simple and natural iterative proportional allocation algorithm: Maintain a priority score $\boldsymbol{\beta}_a$ for each node $a \in \mathbb{A}$ on one side of the bipartition, initialized as $\boldsymbol{\beta}_a = 1$. Iteratively allocate the nodes $i \in \mathbb{I}$ on the other side to eligible nodes in $\mathbb{A}$ in proportion of their priority scores. After each round, for each node $a \in \mathbb{A}$, decrease or increase the score $\boldsymbol{\beta}_a$ based on whether it is over- or under- allocated. Our first result is that this simple, distributed algorithm converges to a $(1-\epsilon)$-approximate fractional b-matching solution in $O(\frac{\log n}{\epsilon^2})$ rounds. We also extend the proportional allocation algorithm and convergence results to the maximum weighted matching problem, and show that the algorithm can be naturally tuned to produce maximum matching with *high entropy*. High entropy, in turn, implies additional desirable properties of this matching, e.g., it satisfies certain diversity and fairness (aka anonymity) properties that are desirable in a variety of applications in online advertising and machine learning.

1. Introduction

Generalized bipartite matching or bipartite b-matching is one of the fundamental problems in computer science. Canonical applications include resource allocation problems such as ad allocation in online advertising, job/server allocation in cloud computing, organ/donor matching, and product recommendation under resource constraints. It has also been utilized as an algorithmic tool in a variety of domains, including computer vision (Belongie et al., 2002), estimating text similarity (Pang et al., 2016), string matching for protein structure alignment (Krissinel & Henrick, 2004), document clustering (Dhillon, 2001); and as a subroutine in several machine learning tasks (Huang & Jebara, 2007; Jebara & Shchogolev, 2006).

The focus of this paper is on large-scale matching problems such as those arising in online advertising. In online advertising settings, a set of advertisers $\mathbb{A}$ provide their targeting domains to determine what subset of impressions $\mathbb{I}$ they are interested in. This can be modeled as a bipartite graph $G(\mathbb{A}, \mathbb{I}, \mathbb{E})$. The advertisers also set capacity/targeting constraints on the number of impressions they want their ads to be shown to, referred to as capacity (or budget) constraints. It is assumed that each advertiser a has a capacity constraint $\mathbf{C}_a$. The matching task is to assign each impression to at most one eligible advertiser based on the targeting information while respecting the capacity constraints. Typically, the goal is to maximize either the number of matched impressions, or the sum of values of the assignments if we are awarded different values for the assignment of every pair of impressions and advertisers.

The rapid growth in Internet advertising has introduced many large scale matching problems for assigning billions of impressions to advertisers on a daily basis. Classic centralized approaches to solve these problems may be irrelevant due to their computational and memory limitations. In online advertising, the number of impressions are usually much higher than the number of advertisers. Such bipartite graphs are called *lopsided bipartite graphs*. The number of impressions is often so large that these matching instances do not fit in the memory of a single machine, and there is a dire need of designing simple and scalable matching algorithms. This is true even if we treat similar impressions as identical copies because each impression type is "the Cartesian product of several features (such as geographic location, time of day/week), domains of which have sizes typically ranging from thousands to millions" (Bateni et al., 2017). Similar lopsided bipartite matching problems arise in many other domains, for example in product recommendation, where the number of users is typically much higher than the number of products, or document-word clustering, where

[*]Equal contribution [1]Columbia University, New York, NY [2]Google Research. Correspondence to: Shipra Agrawal <sa3305@columbia.edu>, Vahab Mirrokni <mirrokni@google.com>, Morteza Zadimoghaddam <zadim@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

the number of words is typically much larger compared to the number of documents (Dhillon, 2001).

All the above motivate the problem of designing simple and scalable algorithms for lopsided bipartite matching in practice. One such natural algorithm that has been used in practice is the *proportional allocation* algorithm: consider the bipartite matching problem on graph $G(\mathbb{A}, \mathbb{I}, \mathbb{E})$ with given capacity constraints $\mathbf{C}_a$ for $a \in \mathbb{A}$. Proportional allocation algorithm is as follows: Maintain a priority score $\boldsymbol{\beta}_a$ for each $a \in \mathbb{A}$, initialized as $\boldsymbol{\beta}_a = 1$. Iteratively allocate each node $i \in \mathbb{I}$ to an eligible node $a \in \mathbb{A}$ in proportion of its score $\boldsymbol{\beta}_a$. After each round, increase or decrease $\boldsymbol{\beta}_a$ based on over- or under- allocation of node a, for each $a \in \mathbb{A}$. Repeat until this algorithm converges to a stable solution. This is a natural and easy to implement algorithm, used in practice to compute b-matching in a distributed fashion for large-scale problems. This is especially useful when the graph is lop-sided, so that the number of advertisers, and hence, the number of priority scores to be maintained and communicated are relatively small.

Our first result is that this simple iterative algorithm converges to a $(1-\epsilon)$-approximate fractional b-matching solution in $O(\frac{\log n}{\epsilon^2})$ rounds. To this end, we first present a combinatorial proof of our result for the unweighted case. Then, we present a primal-dual interpretation via convex programing duality. We formulate a convex program for the problem of maximizing the cardinality/weight of matching in a bipartite graph, with the entropy of the matching as a regularizer in the objective. Interestingly, it turns out that the priority scores in the proportional allocation algorithm correspond to the dual variables of this convex program. And, the proportional allocation rule corresponds to the complimentary primal solution, for any given values of the dual variables. This formulation helps us extend the proportional allocation algorithm and convergence results to edge-weighted graphs.

More importantly, an implication of this formulation is that the proportional allocation algorithm naturally produces *high entropy* matchings. In fact, we formally demonstrate that we can set certain parameters of the algorithm to ensure convergence to an almost optimal matching with high entropy. High entropy, in turn, implies additional desirable properties of this matching. First of all, by maximizing entropy, the allocation achieves higher *diversity* both from the advertisers' point of view and from the users' perspective. From advertisers' perspective, they see a more diverse set of impressions which translates to reaching out to a more diverse demographics. From impressions' perspective, each user will also see a more diverse set of ads. The connection between entropy and various diversity measures has been confirmed by several papers (Qin & Zhu, 2013; Ahmed et al., 2017; Noia et al., 2017). Besides achieving higher diversity, high entropy allocations are also believed to be more fair (e.g., (Venkatasubramanian, 2010) and (Lan et al., 2010) propose high entropy as an important fairness criteria).

Furthermore, one can argue that the proportional allocation algorithm achieves better fairness due to its symmetry and anonymity properties (Lan et al., 2010). It is also likely to be more robust to changes in demand patterns (due to its increased randomized allocation criteria). Below, we further discuss the merits of proportional allocation in comparison to other related online and distributed algorithms for bipartite matching.

1.1. Related work

Graph matching and assignment problems are some of the most well studied problems in combinatorial optimization. There is considerable work on fast exact algorithms, as well as faster approximate algorithms, for matching problems. Notable examples include $(1-\epsilon)$ approximation in $O(\frac{m}{\epsilon}\log(1/\epsilon))$ time by (Duan & Pettie, 2014) for weighted graphs, where m is the number of edges in the graph. For maximum cardinality matching, many classic algorithms (e.g., (Hopcroft & Karp, 1971)) can achieve this.

Motivated by the large-scale applications of matching in advertising and other e-commerce applications, recently there has been a focus on distributed algorithms. In these applications, it is desirable to have algorithms which run in potentially logarithmic rounds or phases, with each phase involving simple computations that can be distributed[1] and/or parallelized. Some recent literature includes the work by (Bahmani et al., 2014) in the MapReduce framework, which improves upon previous work of (Ahn & Guha, 2013) and (Bahmani et al., 2012). The proportional allocation algorithm provides a much simpler alternative approach for such distributed large-scale settings, especially in case of large lop-sided bipartite graphs. Arguably, this heuristic is comparable in its simplicity and ease of implementation to the greedy heuristic, which only allows a 2-factor approximation. In contrast, as proven in this paper, the proportional allocation converges to optimal solution in logarithmic number of rounds.

Another closely related work is by Charles et al. (2010) on fast streaming algorithms for bipartite matching in lopsided graphs. Proportional allocation has several significant benefits over the method proposed there, including high entropy matching, amenability to distributed implementation, and simple concise representation through priority scores of advertisers (i.e., one score for each node on the smaller side in the lop sided graphs) only.

We also note that iterative approximation algorithms have been developed for the more general problem class of pack-

[1] i.e., allow the graph to be stored in a distributed manner

ing and covering (Plotkin et al., 1995; Awerbuch & Khandekar, 2009; Garg & Konemann, 2007). In fact, many of these algorithms belong to the class of multiplicative weight update (MWU) methods(Arora et al., 2012). The MWU methods operate by maintaining weights w_a for each advertiser, similar to our priority scores. These weights are updated in a multiplicative manner based on the *amount* of over-allocation or under-allocation in every round. The weights are then used as Lagrangian dual variables to combine the packing (capacity) constraints, so that the packing problem reduces to a knapsack problem. The impression allocation then roughly reduces to greedily selecting impression-advertiser mappings with highest ratio $\mathbf{r}_{i,a}/w_a$. Besides having a simpler score update rule (constant factor updates) and a simpler, distributed assignment rule (proportional allocation), the proportional allocation algorithm is naturally designed to yield higher entropy solutions compared to these methods. Intuitively, this is because proportional allocation rule essentially does a softmax over $(\mathbf{r}_{i,a} - \boldsymbol{\beta}_a)$: imagine the case when weights $\mathbf{r}_{i,a}$ are distinct but infinitesimally close to each other, then the above-mentioned greedy approach will select the top $\mathbf{C}_a$ impressions for every advertiser, where as the softmax will give almost uniform distribution. In fact, we formally show that softmax is the optimal form of primal decision for maximizing entropy along with weight of the matching.

Extensions of such primal-dual approaches have also been proposed for *online* packing problems motivated by the *Display Ads Allocation (DA)* problem (Gupta & Molinaro, 2016; Agrawal & Devanur, 2015; Devanur et al., 2011; Agrawal et al., 2009; Feldman et al., 2009; 2010; Vee et al., 2010), and the *Budgeted Allocation (AdWords)* problem (Mehta et al., 2007; Devanur & Hayes, 2009). In the online setting, the impressions arrive one by one in sequential time steps, and should either be immediately assigned to one of the advertisers with remaining budget, or discarded. In these algorithms, the dual variables or advertiser weights are updated periodically over time either by solving an LP (Agrawal et al., 2009; Feldman et al., 2009; 2010; Devanur & Hayes, 2009), or by multiplicative weight updates (Gupta & Molinaro, 2016; Agrawal & Devanur, 2015; Devanur et al., 2011). These weights are then used as thresholds for making assignments of impressions arriving online. Besides the concerns mentioned above for MWU methods, the weight updates in these online algorithms must be performed sequentially, and therefore are not amenable to parallel implementations.

1.2. Organization of the paper.

In Section 2, we formulate the generalized bipartite matching problems considered in this paper. In Section 3, we present the proportional allocation algorithm for the maximum cardinality case, as well as its simple extension to the problem of finding maximum weighted matching with high-entropy. In Section 4, we prove our main results (Theorem 1 and Theorem 2) regarding efficient convergence of both these versions of the proportional allocation algorithm. The proof of Theorem 2 also provides an interesting primal-dual interpretation of the proportional allocation algorithm.

2. Problem Formulation

Here, we formulate the generalized bipartite matching problems, aka bipartite b-matching problems, considered in this paper. Throughout the paper, we use the terminology from online advertising, with the two sides of the bipartite graph being 'impressions' and 'advertisers'.

Maximum cardinality matching. A set $\mathbb{A}$ of advertisers and a set $\mathbb{I}$ of impressions are given. For each advertiser $a \in \mathbb{A}$, there is a set of impressions $\mathbf{N}_a \subseteq \mathbb{I}$ that can be potentially assigned to a. Similarly for any $i \in \mathbb{I}$, we define $\mathbf{N}_i \subseteq \mathbb{A}$ to be the set of advertisers that impression i can be matched to. These connections can be represented with a bipartite graph $\mathbb{G}$ of edge set $\mathbb{E} = \{(i,a) : i \in \mathbb{I}, a \in \mathbf{N}_i\} = \{(i,a) : a \in \mathbb{A}, i \in \mathbf{N}_a\}$. Each advertiser a has capacity $\mathbf{C}_a$ denoting maximum number of impressions she is interested to be matched to.

The goal is to find a subset of edges $M \subseteq \mathbb{E}$ such that:

- Each impression is incident to at most one edge in M. This property ensures that each impression is assigned to at most one advertiser.
- Each advertiser a is incident to at most $\mathbf{C}_a$ edges in M respecting its capacity.

while maximizing the cardinality of M. Such an edge set M is referred to as a maximum cardinality matching.

A *maximum cardinality fractional matching* is defined as an assignment $\{\mathbf{x}_{i,a}\} \in [0,1]^{\mathbb{E}}$ that maximizes $\sum_{(i,a)\in\mathbb{E}} \mathbf{x}_{i,a}$ while satisfying capacity constraints, i.e.,

$$\sum_{i\in\mathbb{I}} \mathbf{x}_{i,a} \quad \leq \mathbf{C}_a, \quad \forall a \in \mathbb{A}, \tag{1}$$

$$\sum_{a\in\mathbb{A}} \mathbf{x}_{i,a} \quad \leq 1, \quad \forall i \in \mathbb{I} \tag{2}$$

Maximum weighted matching. We also consider the more general problem of maximum weighted matching. Here, for each edge $e = (i,a) \in \mathbb{E}$, a weight $\mathbf{r}_{i,a}$ has been specified. The goal is to find a subset of edges $M \subseteq \mathbb{E}$ such that the capacity constraints for each advertiser and impression are satisfied, while maximizing total weight $\sum_{(i,a)\in M} \mathbf{r}_{i,a}$ of the matching. For fractional matching $\{\mathbf{x}_{i,a}\}_{(i,a)\in\mathbb{E}}$, similarly the goal is to maximize $\sum_{(i,a)\in\mathbb{E}} \mathbf{x}_{i,a}\mathbf{r}_{i,a}$, while satisfying constraints in (1) and (2).

High entropy matching. The proportional allocation algorithm proposed in this paper naturally gives a high en-

Algorithm 1 PropAlloc : A proportional allocation algorithm for maximum cardinality matching

Input: $G = (\mathbb{A}, \mathbb{I}, \mathbb{E})$, $\{\mathbf{C}_a\}_{a\in\mathbb{A}}$; parameter $\epsilon \in (0,1)$, number of rounds R.
Initialization: Set $\boldsymbol{\beta}_a = 1$, for all $a \in \mathbb{A}$.
for rounds $\ell = 1, 2, \ldots, R$ **do**
Step 1: For each impression i, set assignment

$$\mathbf{x}_{i,a} = \frac{\boldsymbol{\beta}_a}{\sum_{a'\in\mathbf{N}_i} \boldsymbol{\beta}_{a'}}, \forall a \in \mathbf{N}_i$$

Step 2: For each advertiser a, update $\boldsymbol{\beta}_a$ as follows:

$$\mathbf{Alloc}_a \leq \frac{\mathbf{C}_a}{(1+\epsilon)} \implies \boldsymbol{\beta}_a \coloneqq (1+\epsilon)\boldsymbol{\beta}_a$$

$$\mathbf{Alloc}_a \geq (1+\epsilon)\mathbf{C}_a \implies \boldsymbol{\beta}_a \coloneqq \frac{\boldsymbol{\beta}_a}{(1+\epsilon)}$$

where $\mathbf{Alloc}_a \coloneqq \sum_{i\in\mathbf{N}_a} \mathbf{x}_{i,a}$.
end for
for each a with $\mathbf{Alloc}_a > \mathbf{C}_a$ **do**
Set $\mathbf{x}_{i,a} \coloneqq \frac{\mathbf{C}_a}{\mathbf{Alloc}_a}\mathbf{x}_{i,a}, \forall i \in \mathbf{N}_a$
end for

tropy fractional matching, while also maximizing cardinality/weight of the matching. To formally study this property of the algorithm, we consider an alternate objective of maximizing a combination of weight and entropy of the matching. Specifically, given a parameter $\lambda \geq 0$, the goal here is to find a fractional matching $\{\mathbf{x}_{i,a}\}_{(i,a)\in\mathbb{E}}$ that maximizes

$$\sum_{(i,a)\in\mathbb{E}} \mathbf{r}_{i,a}\mathbf{x}_{i,a} + \lambda \sum_{(i,a)\in\mathbb{E}} \mathbf{x}_{i,a}\log(1/\mathbf{x}_{i,a}) \tag{3}$$

while satisfying capacity constraints in (1) and (2). The second term in the above is the entropy of assignment $\{\mathbf{x}_{i,a}\}$.

3. Proportional allocation algorithm

We propose the multi round distributed algorithm PropAlloc that finds an almost optimum fractional matching, and then prove how the fractional matching can be transformed into an (integral) matching without much loss if the capacities of advertisers are large.

Algorithm PropAlloc intends to find priority score $\boldsymbol{\beta}_a$ for each advertiser $a \in \mathbb{A}$ such that if the impressions are assigned proportional to these priorities, we achieve an almost optimum allocation. Formally, impression i will be assigned to advertiser $a \in \mathbf{N}_i$ with probability $\frac{\boldsymbol{\beta}_a}{\sum_{a'\in\mathbf{N}_i}\boldsymbol{\beta}_{a'}}$. Algorithm PropAlloc then computes the expected number of impressions each advertiser a receives as follows.

$$\mathbf{Alloc}_a = \sum_{i\in\mathbf{N}_a} \frac{\boldsymbol{\beta}_a}{\sum_{a'\in\mathbf{N}_i}\boldsymbol{\beta}_{a'}} \tag{4}$$

Algorithm 2 PropAlloc $^+$: A proportional allocation algorithm for high-entropy maximum weight matching

Input: $G = (\mathbb{A}, \mathbb{I}, \mathbb{E})$, $\{\mathbf{C}_a\}_{a\in\mathbb{A}}$, weights $\{\mathbf{r}_{i,a}\}_{(i,a)\in\mathbb{E}}$, parameter λ; parameter $\epsilon \in (0,1)$, number of rounds R.
Initialization: Set $\boldsymbol{\beta}_a = (1+\epsilon)^{-R}$, for all $a \in \mathbb{A}$.
for rounds $\ell = 1, 2, \ldots, R$ **do**
Step 1: For each impression i, set assignment

$$\mathbf{x}_{i,a} = \begin{cases} \frac{\boldsymbol{\beta}_a \boldsymbol{D}_{i,a,\lambda}}{\sum_{a'\in\mathbf{N}_i}\boldsymbol{\beta}_{a'}\boldsymbol{D}_{i,a',\lambda}} & \text{if } \sum_{a'\in\mathbf{N}_i}\boldsymbol{\beta}_{a'}\boldsymbol{D}_{i,a',\lambda} \leq 1 \\ \frac{\boldsymbol{\beta}_a \boldsymbol{D}_{i,a,\lambda}}{\sum_{a'\in\mathbf{N}_i}\boldsymbol{\beta}_{a'}\boldsymbol{D}_{i,a',\lambda}} & \text{otherwise} \end{cases}$$

where $\boldsymbol{D}_{i,a,\lambda} = e^{\frac{\mathbf{r}_{i,a}}{\lambda}-1}$
Step 2: For each advertiser a, update $\boldsymbol{\beta}_a$ as follows:

$$\mathbf{Alloc}_a \leq \frac{\mathbf{C}_a}{(1+\epsilon)} \implies \boldsymbol{\beta}_a \coloneqq (1+\epsilon)\boldsymbol{\beta}_a$$

$$\mathbf{Alloc}_a \geq (1+\epsilon)\mathbf{C}_a \implies \boldsymbol{\beta}_a \coloneqq \frac{\boldsymbol{\beta}_a}{(1+\epsilon)}$$

where $\mathbf{Alloc}_a \coloneqq \sum_{i\in\mathbf{N}_a} \mathbf{x}_{i,a}$.
end for
for each a with $\mathbf{Alloc}_a > \mathbf{C}_a$ **do**
Reduce $\mathbf{x}_{i,a}$ for impressions $i \in \mathbf{N}_a$ with $\mathbf{x}_{i,a} \geq \frac{\mathbf{C}_a}{|\mathbf{N}_a|}$, until $\sum_{i\in\mathbf{N}_a} \mathbf{x}_{i,a} \leq \mathbf{C}_a$.
end for

Intuitively, if the expected allocation $\mathbf{Alloc}_a$ exceeds the capacity $\mathbf{C}_a$, it means advertiser a has been over-allocated, so the overflow of impressions $\mathbf{Alloc}_a - \mathbf{C}_a$ are going to be discarded without contributing anything to the objective function. On the other hand, if the expected allocation $\mathbf{Alloc}_a$ does not reach the capacity $\mathbf{C}_a$, it means advertiser a has been under-allocated, so the there is a $\mathbf{C}_a - \mathbf{Alloc}_a$ extra capacity left to be potentially exploited. Both of the above situations introduce some room for improving the priority variables. Algorithm PropAlloc initializes all $\boldsymbol{\beta}_a$ variables to the same value (for instance 1), and then updates $\boldsymbol{\beta}_a$ for each $a \in \mathbb{A}$ in each round as follows.

- If $\mathbf{Alloc}_a \leq \frac{\mathbf{C}_a}{(1+\epsilon)} \implies \boldsymbol{\beta}_a \coloneqq (1+\epsilon)\boldsymbol{\beta}_a$.
  In other words increase priority of a by a multiplicative factor of $1+\epsilon$.
- If $\mathbf{Alloc}_a \geq (1+\epsilon)\mathbf{C}_a \implies \boldsymbol{\beta}_a \coloneqq \frac{\boldsymbol{\beta}_a}{1+\epsilon}$.
  In other words decrease priority of a by a multiplicative factor of $1+\epsilon$.
- Otherwise, do not change the priority of a.

Algorithm PropAlloc consists of R rounds of computing **Alloc** variables based on the priorities, $\{\boldsymbol{\beta}_a\}_{a\in\mathbb{A}}$, and then updating the priorities with above rules. After all these rounds, PropAlloc computes the fractional matching respecting all capacity constraints as follows. For every impression $i \in \mathbb{I}$ and each advertiser $a \in \mathbf{N}_i$, we set the

assignment $\mathbf{x}_{i,a}$ to be the probability that i is assigned to a based on the current priority values. That is,

$$\mathbf{x}_{i,a} = \frac{\boldsymbol{\beta}_a}{\sum_{a' \in \mathbf{N}_i} \boldsymbol{\beta}_{a'}}$$

These assignments always respect the constraints (1) on impressions, since the total assignment of each impression $i \in \mathbb{I}$, given by $\sum_{a \in \mathbf{N}_i} \mathbf{x}_{i,a}$, is equal to 1. But, there might be advertisers that receive more total assignment than their capacities. To adjust for these over-allocations, at the end of R rounds, the assignments to these advertisers can be reduced *in any manner*. For each advertiser, $a \in \mathbb{A}$ with $\mathbf{Alloc}_a > \mathbf{C}_a$, we can scale down the assignments of all edges incident on a by a factor of $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a}$ to make sure that the capacity constraints are all respected. Therefore, the total weight of the fractional matching is equal to $\mathsf{MatchWeight} = \sum_{a \in \mathbb{A}} \min\{\mathbf{Alloc}_a, \mathbf{C}_a\}$.

The proportional allocation algorithm is summarized in Algorithm 1. We show that a logarithmic number of rounds suffices to converge to an almost optimum fractional allocation, and then find an integral assignment based on that. Further, among the maximum cardinality matching, the proportional allocation algorithm naturally finds matchings with high entropy. To formalize this observation, below we give a simple extension of the algorithm for the joint objective of maximizing entropy and weight of the matching, combined with a parameter λ. In fact Algorithm 1 will be a special case of the new algorithm for $\lambda \approx 0, \mathbf{r}_{i,a} = 1, \forall i, a$.

Algorithm for high-entropy weighted matching. Given weights $\{\mathbf{r}_{i,a}\}_{(i,a) \in \mathbb{E}}$, and a parameter $\lambda > 0$, a simple extension of the proportional allocation algorithm computes maximum weight matching with high entropy. This algorithm maintains and updates priority scores $\{\boldsymbol{\beta}_a\}$ in a similar manner to Algorithm 1. However, to account for weights and entropy parameter λ, given the priority scores, the assignments $\mathbf{x}_{i,a}$ are now computed as follows: let $\boldsymbol{D}_{i,a,\lambda} := e^{\frac{\mathbf{r}_{i,a}}{\lambda} - 1}$, then,

$$\mathbf{x}_{i,a} = \begin{cases} \boldsymbol{\beta}_a \boldsymbol{D}_{i,a,\lambda} & \text{if } \sum_{a' \in \mathbf{N}_i} \boldsymbol{\beta}_{a'} \boldsymbol{D}_{i,a',\lambda} \le 1 \\ \frac{\boldsymbol{\beta}_a \boldsymbol{D}_{i,a,\lambda}}{\sum_{a' \in \mathbf{N}_i} \boldsymbol{\beta}_{a'} \boldsymbol{D}_{i,a',\lambda}} & \text{otherwise} \end{cases}$$

The new algorithm is summarized in Algorithm 2. Note that the main change is in Step 2. Further, at the end of R rounds, earlier in Algorithm 1 we could allow any kind of adjustment to assignments of over-allocated advertisers. But, since entropy is of consideration here, in Algorithm 2 we make a slightly more careful adjustment: we only remove impressions with large assignment value, i.e., impressions $i \in \mathbf{N}_a$ with $\mathbf{x}_{i,a} \ge \frac{\mathbf{C}_a}{|\mathbf{N}_a|}$.

Note that these modifications keeps the simple distributed structure of the algorithm intact: given priority scores of advertisers, the impressions can be allocated in a distributed manner in proprtion of these scores.

4. Analysis

4.1. Main results

First, we show that after enough number of rounds, the fractional matching achieved by PropAlloc (refer to Algorithm 1 is almost optimal.

Theorem 1. *For any $\delta \in (0, 1]$, there exists[2] an $\epsilon > 0$ such that algorithm PropAlloc with parameter ϵ returns a $(1-\delta)$-approximate fractional matching after $R = O(\frac{\log(n/\delta)}{\delta^2})$ rounds. Here, n is the number of advertisers.*

Further, we provide a primal-dual interpretation of the proportional allocation algorithm to show that PropAlloc $^+$ (refer to Algorithm 2) can achieve any desired tradeoff between weight of the matching and entropy of the matching.

Theorem 2. *For any $\delta \in (0, 1]$, $\lambda > 0$, there exists an $\epsilon > 0$ such that algorithm PropAlloc $^+$ with parameter ϵ returns a fractional matching that achieves $(1-\delta)$-approximation for the weight-entropy objective in (3), after $R = O\left(\frac{\mathbf{r}_{\max}}{\mathbf{r}_{\min}} \frac{(1+\lambda \log \bar{N}))^2}{\lambda \delta}\right)$ rounds.*

Here, $\mathbf{r}_{\max} = \max_{(i,a) \in \mathbb{E}} \mathbf{r}_{i,a}, \mathbf{r}_{\min} = \min_{(i,a) \in \mathbb{E}} \mathbf{r}_{i,a}$, $\bar{N} = \max_{a \in \mathbb{A}} \frac{|\mathbf{N}_a|}{\mathbf{C}_a}$.

Remark 1. *Any feasible fractional allocation can be adapted as a randomized allocation algorithm since the sum of edge weights per impression does not exceed 1 and they can be interpreted as allocation probabilities. In expectation, this gives a feasible integral allocation. Further, using concentration bounds (e.g., Lemma 13 of (Bansal & Sviridenko, 2006)), with high probability, the capacity constraints will not be violated by more than a factor of $\tilde{O}(1 + \frac{1}{\sqrt{\boldsymbol{C}_a}})$ for any advertiser a. Therefore, if advertisers have large enough capacities, the fractional matching can be rounded to an integral solution with negligible loss.*

4.2. A combinatorial analysis of PropAlloc (Proof of Theorem 1)

We focus on the $\boldsymbol{\beta}$ variables when the algorithm terminates (after the end of round R). The minimum value the priority variables can take after R rounds is $\boldsymbol{\beta}_{min} = \frac{1}{(1+\epsilon)^R}$, and any $a \in \mathbb{A}$ can take one of the following $2R+1$ potential priority values:

$$\boldsymbol{\beta}_a \in \{\boldsymbol{\beta}_{min}, (1+\epsilon)\boldsymbol{\beta}_{min}, \cdots, (1+\epsilon)^{2R}\boldsymbol{\beta}_{min}\}$$

For each $0 \le k \le 2R$, let L_k be the set of advertisers with priority value $(1+\epsilon)^k \boldsymbol{\beta}_{min}$, i.e. $L_k := \{a | \boldsymbol{\beta}_a = (1+\epsilon)^k \boldsymbol{\beta}_{min}\}$. Since these sets form a hierarchy of priority

[2] It suffices to set $\epsilon = \delta/5$.

values, we call them level sets. We note that some of these sets may be empty. There are two main sources of possible suboptimality in the fractional matching that PropAlloc finds:

- Over-allocation: If $\mathbf{Alloc}_a$ is greater than $\mathbf{C}_a$, $\mathbf{Alloc}_a - \mathbf{C}_a$ matched impressions will not be counted towards the objective.
- Under-allocation: If $\mathbf{Alloc}_a$ is less than $\mathbf{C}_a$, an extra capacity of $\mathbf{C}_a - \mathbf{Alloc}_a$ is left to be exploited for advertiser a.

In the following, we show that for advertisers in most of the level sets, both of the above over-allocation and under-allocation losses are negligible.

Lemma 1. *For any $a \in \cup_{k=0}^{2R-1} L_k$, the under-allocation $\boldsymbol{C}_a - \boldsymbol{Alloc}_a$ is at most $3\epsilon C_a$. Similarly for any $a \in \cup_{k=1}^{2R} L_k$, the over-allocation $\boldsymbol{Alloc}_a - C_a$ is at most $3\epsilon C_a$.*

Proof. Due to the symmetry of the two claims, we only prove the former. Since a is not in level set L_{2R}, there was a time that we did not increase $\boldsymbol{\beta}_a$. Let t be the last round that $\boldsymbol{\beta}_a$ was not increased. At this point, $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a}$ was at least $\frac{1}{(1+\epsilon)}$. For $t = R$, this completes the proof. Otherwise, we focus on round $t+1 \le R$. Recall, $\mathbf{Alloc}_a = \sum_{i \in \mathbf{N}_a} \frac{\boldsymbol{\beta}_a}{\sum_{a' \in \mathbf{N}_i} \boldsymbol{\beta}_{a'}}$.

If $\boldsymbol{\beta}_a$ is unchanged at round t, the numerator of each term also remains unchanged. The denominator terms are increased at most by a factor of $(1+\epsilon)$. So in total, $\mathbf{Alloc}_a$ is not decreased by more than a factor of $(1+\epsilon)$ yielding the lower bound $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a} \ge \frac{1}{(1+\epsilon)^2}$ at round $t+1$. In the other case, $\boldsymbol{\beta}_a$ is decreased at round t, so the numerator of each term is also reduced by a factor of $(1+\epsilon)$. In total, the ratio $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a}$ is decreased by a factor of at most $\frac{1}{(1+\epsilon)^2}$ at round $t+1$. Note that the reduction of $\boldsymbol{\beta}_a$ at round t means the ratio $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a}$ was at least $1+\epsilon$, and therefore at least $\frac{1}{1+\epsilon}$ at round $t+1$. So independent of whether $\boldsymbol{\beta}_a$ was reduced or not, $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a}$ will be at least $\frac{1}{(1+\epsilon)^2}$ at round $t+1$.

By definition of t, $\boldsymbol{\beta}_a$ is increased in all rounds after t. With a similar argument, we know that $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a}$ does not decrease at any of these rounds. So the ratio $\frac{\mathbf{Alloc}_a}{\mathbf{C}_a}$ remains at least $\frac{1}{(1+\epsilon)^2} \ge 1 + 3\epsilon$ for $\epsilon \le 1$ till the last round. □

Lemma 1 shows that every advertiser is either changed in one direction (reducing or increasing $\boldsymbol{\beta}$) in all rounds, or its fractional allocation will be almost equal to its capacity. The latter helps us prove optimality, and the former only contains advertisers in level sets L_0 and L_{2R}. Next, we prove two main claims: on lower bounding the weight of the fractional matching, $\mathsf{MatchWeight} = \sum_{a \in \mathbb{A}} \min\{\mathbf{Alloc}_a, \mathbf{C}_a\}$, and on upper bounding the optimum value in terms of the level sets. These are stated as Claim 1.

Claim 1. *For any two indices $1 \le \ell$ and $\ell + \log(n/\epsilon)/\epsilon \le \ell' \le 2R$, we have:*

- MatchWeight, *is at least:*

$$(1-4\epsilon)\left(\sum_{k=0}^{\ell}\sum_{a\in L_k} \boldsymbol{C}_a + |\boldsymbol{N}(\cup_{k'=\ell'+1}^{2R} L_{k'})|\right) \quad (5)$$

  where $\boldsymbol{N}(S)$ for any subset S of advertisers is the union of their neighborhoods $\cup_{a\in S}\boldsymbol{N}_a$.
- *The weight of the optimum fractional matching does not exceed:*

$$\left(\sum_{k=0}^{\ell'}\sum_{a\in L_k} \boldsymbol{C}_a + |\boldsymbol{N}(\cup_{k'=\ell'+1}^{2R} L_{k'})|\right) \quad (6)$$

Proof. The proof of the second statement is very similar to folklore graph theoretic results like Konig's Theorem (Ahmadi & Hall). The number of matched impressions in the optimum allocation consists of two main classes: those matched to advertisers in $\cup_{k=0}^{\ell'} L_k$, and those assigned to advertisers in $\cup_{k=\ell'+1}^{2R} L_k$. The former cannot be more than the sum of capacities of the associated advertisers which is the first term in the upper bound. The latter is a subset of all neighbors of advertisers in $\cup_{k=\ell'+1}^{2R} L_k$ and therefore at most $|\mathbf{N}(\cup_{k'=\ell'+1}^{2R} L_{k'})|$.

To prove the first statement of the Claim, we categorize assigned impressions in MatchWeight into two categories. Using Lemma 1, the impressions assigned to advertisers in $L_0, L_1, \cdots, L_\ell$ almost fill up their capacities and therefore sum up to at least $(1-3\epsilon)\sum_{k=0}^{\ell}\sum_{a\in L_k} \mathbf{C}_a$ which is larger than the first term of the lower bound.

The second term represents all neighbors of advertisers in $L_{\ell'+1}, \cdots, L_{2R}$. To avoid double counting, we show that any impression that has a neighbor in $\cup_{k'=\ell'+1}^{2R} L_{k'}$ will not be assigned to any advertiser in $\cup_{k=0}^{\ell} L_k$ w.h.p. ($\ge 1-\epsilon$).

Consider impression i that is a neighbor of $a' \in L_{k'}$ for some $k' \ge \ell'+1$. Because ℓ' is at least $\ell + \log(n/\epsilon)/\epsilon$, we have $\boldsymbol{\beta}_{a'} \ge \frac{n}{\epsilon}\boldsymbol{\beta}_a$ for any $a \in L_k$ with $k \le \ell$. Therefore the probability of i being assigned to a is at most ϵ/n times the probability it being assigned to a'. Since there could be potentially at most n candidates like a, the probability of i being assigned to any advertiser in $\cup_{k=0}^{\ell} L_k$ is at most ϵ. So every impression in $\mathbf{N}(\cup_{k=\ell'+1}^{2R} L_k)$ will be assigned to some advertiser in $\cup_{k=\ell+1}^{2R} L_k$ with probability at least $1-\epsilon$. Using Lemma 1, at least $1-3\epsilon$ fraction of every such impression will be counted towards MatchWeight. So in total, we get at least $1-4\epsilon$ for each impression in $\mathbf{N}(\cup_{k'=\ell'+1}^{2R} L_{k'})$ which concludes the proof of the Claim. □

Proof of Theorem 1. Given Claim 1, there are two main gaps between the lower bound of (5) and the upper bound of (6): the $1-4\epsilon$ factor and the sum $\sum_{k=\ell+1}^{\ell'}\sum_{a\in L_k} \mathbf{C}_a$. We show that the latter gap is small for some value of ℓ and $\ell' = \ell + \log(n/\epsilon)/\epsilon$.

Summing this gap over different values of ℓ yields

$$\sum_{\ell=0}^{2R-\log(n/\epsilon)/\epsilon} \sum_{k=\ell+1}^{\ell+\log(n/\epsilon)/\epsilon} \sum_{a\in L_k} \mathbf{C}_a \leq (\log(n/\epsilon)/\epsilon) \sum_{k=1}^{2R} \sum_{a\in L_k} \mathbf{C}_a$$

Therefore there exists an $0 \leq \ell \leq 2R - \log(n/\epsilon)/\epsilon$ such that its associated gap $\sum_{k=\ell+1}^{\ell+\log(n/\epsilon)/\epsilon} \sum_{a\in L_k} \mathbf{C}_a$ is at most $\frac{\log(n/\epsilon)/\epsilon}{2R-\log(n/\epsilon)/\epsilon+1} \sum_{k=1}^{2R} \sum_{a\in L_k} \mathbf{C}_a \leq \epsilon \sum_{k=1}^{2R} \sum_{a\in L_k} \mathbf{C}_a$ where the last inequality holds when R is at least $\log(n/\epsilon)/\epsilon^2$.

Using Lemma 1, for every $a \in \cup_{k=1}^{2R} L_k$, the over-allocation $\mathbf{Alloc}_a - \mathbf{C}_a$ is at most $3\epsilon\mathbf{C}_a$. Therefore MatchWeight is at least $(1-3\epsilon)\sum_{k=1}^{2R}\sum_{a\in L_k}\mathbf{C}_a$. This means the gap associated for some ℓ is at most $\epsilon\mathsf{MatchWeight}/(1-3\epsilon)$. Using Claim 1, we have $\mathsf{MatchWeight} \geq (1-4\epsilon)(OPT - \epsilon\mathsf{MatchWeight}/(1-3\epsilon))$ yielding a final approximation factor of at least $1-5\epsilon$. Then, the theorem statement can be obtained by setting $\delta = \epsilon/5$, $R = \log(n/\epsilon)/\epsilon^2 = O(\log(n/\delta)/\delta^2)$.

4.3. Primal-dual interpretation: Proof of Theorem 2

Consider the matching problem with weights $\{\mathbf{r}_{i,a}\}_{(i,a)\in\mathbb{E}}$. Given any $\lambda > 0$, let OPT_λ denote the optimal value of the following convex optimization problem that maximizes a combination of weight of matching and entropy:

$$\begin{array}{lll} \text{maximize} & \sum_{(i,a)\in\mathbb{E}} \mathbf{r}_{i,a}\mathbf{x}_{i,a} + & \lambda \sum_{i,a} \mathbf{x}_{i,a}\log(1/\mathbf{x}_{i,a}) \\ \text{subject to} & \sum_{a\in\mathbf{N}_i} \mathbf{x}_{i,a} \leq 1, & i \in \mathbb{I} \\ & \sum_{i\in\mathbf{N}_a} \mathbf{x}_{i,a} \leq \mathbf{C}_a, & a \in \mathbb{A} \\ & \mathbf{x}_{i,a} \geq 0 & \forall (i,a) \in \mathbb{E} \end{array} \tag{7}$$

We show that in $R = \frac{\mathbf{r}_{\max}}{\mathbf{r}_{\min}} \frac{(1+\lambda\log(\bar{N}))^2}{\delta\lambda}$ iterations, where $\bar{N} = \max_a \frac{|\mathbf{N}_a|}{\mathbf{C}_a}$, the proportional allocation algorithm with

$$\epsilon \leq \frac{\mathbf{r}_{\min}}{\mathbf{r}_{\max}} \frac{1}{8(2+\lambda\log(\bar{N}))}\delta,$$

finds an assignment $\{\mathbf{x}_{i,a}\}_{i,a}$ satisfying:

$$\mathrm{OPT}_\lambda \leq (1+\delta)\sum_{i,a} \mathbf{r}_{i,a}\mathbf{x}_{i,a} + \lambda \sum_{i,a} \mathbf{x}_{i,a}\log(1/\mathbf{x}_{i,a})$$

Following upper bound on OPT_λ can be obtained using Lagrangian duality for the convex program (7). This also provides a dual-based interpretation of the decision $\mathbf{x}_{i,a}$ with priority scores $\{\boldsymbol{\beta}_a\}$ emerging as an exponential function of the corresponding dual variables for the advertisers' capacity constraints.

Lemma 2. *Given* **any** $\{\gamma_a \geq 0\}_a$, *let*

$$\boldsymbol{x}^*_{i,a} = \begin{cases} \frac{e^{\frac{-\gamma_a}{\lambda}}\boldsymbol{D}_{i,a,\lambda}}{\sum_{a'\in N_i} e^{\frac{-\gamma_{a'}}{\lambda}}\boldsymbol{D}_{i,a',\lambda}}, & \sum_{a'\in N_i} e^{\frac{-\gamma_{a'}}{\lambda}}\boldsymbol{D}_{i,a',\lambda} \geq 1 \\ e^{\frac{-\gamma_a}{\lambda}}\boldsymbol{D}_{i,a,\lambda} & \textit{otherwise.} \end{cases} \tag{8}$$

(recall $\boldsymbol{D}_{i,a,\lambda} = e^{\frac{\mathbf{r}_{i,a}}{\lambda}-1}$). Then,

$$\begin{aligned} \mathit{OPT}_\lambda \leq & \sum_{(i,a)\in\mathbb{E}} \boldsymbol{r}_{i,a}\boldsymbol{x}^*_{i,a} - \lambda \boldsymbol{x}^*_{i,a}\log(\boldsymbol{x}^*_{i,a}) \\ & + \sum_{a\in\mathbb{A}} \gamma_a(\boldsymbol{C}_a - \sum_{i\in\boldsymbol{N}_a} \boldsymbol{x}^*_{i,a}) \end{aligned} \tag{9}$$

Proof. Using Lagrangian duality for (7)

$$\mathrm{OPT}_\lambda = \min_{\gamma\geq 0, z\geq 0} \max_{\mathbf{x}\geq 0} L(\mathbf{x},\gamma,z)$$

where

$$L(\mathbf{x},\gamma,z) := \begin{pmatrix} \sum_{i,a} \mathbf{r}_{i,a}\mathbf{x}_{i,a} - \lambda\mathbf{x}_{i,a}\log(\mathbf{x}_{i,a}) \\ + \sum_i z_i(1 - \sum_{a\in\mathbf{N}_i} \mathbf{x}_{i,a}) \\ + \sum_a \gamma_a(\mathbf{C}_a - \sum_{i\in\mathbf{N}_a} \mathbf{x}_{i,a}) \end{pmatrix}$$

Also, for any $\{\gamma_a \geq 0, z_i \geq 0\}$

$$\mathrm{OPT}_\lambda \leq \max_{\mathbf{x}\geq 0} L(\mathbf{x},\gamma,z)$$

Now, $\frac{\partial}{\partial \mathbf{x}_{i,a}} L(\mathbf{x},\gamma,z) = \mathbf{r}_{i,a} - \lambda - \lambda\log(\mathbf{x}_{i,a}) - z_i - \gamma_a$

so for any $z_i \geq 0, \gamma_a \geq 0$,

$$\mathbf{x}^*_{i,a} = e^{-\gamma_a/\lambda - z_i/\lambda} e^{\frac{\mathbf{r}_{i,a}}{\lambda}-1}$$

satisfies $\mathbf{x}^*_{i,a} \geq 0$, $\frac{\partial}{\partial \mathbf{x}_{i,a}} L(\mathbf{x},\gamma,z) = 0$, and therefore it is a maximizer of $L(\mathbf{x},\gamma,z)$, and from above we have $\mathrm{OPT}_\lambda \leq L(\mathbf{x}^*,\gamma,z)$. Now, set z_i as follows: IF $\sum_{a\in\mathbf{N}_i} e^{-\gamma_a/\lambda}\boldsymbol{D}_{i,a,\lambda} \geq 1$, set $e^{\frac{z_i}{\lambda}} = \sum_{a\in\mathbf{N}_i} e^{\frac{-\gamma_a}{\lambda}}\boldsymbol{D}_{i,a,\lambda}$ where $\boldsymbol{D}_{i,a,\lambda} = e^{\frac{\mathbf{r}_{i,a}}{\lambda}-1}$. Otherwise, set $z_i = 0$. Then, substituting z_i, $\mathbf{x}^*_{i,a}$ is as given in (8). Further, for all i, $z_i(\sum_{a\in\mathbf{N}_i} \mathbf{x}^*_{i,a} - 1) = 0$, substituting which we get

$$\begin{aligned} L(\mathbf{x}^*,\gamma,z) = & \sum_{i,a} \mathbf{r}_a\mathbf{x}^*_{i,a} - \lambda\mathbf{x}^*_{i,a}\log(\mathbf{x}^*_{i,a}) \\ & + \sum_a \gamma_a(\mathbf{C}_a - \sum_{i\in\mathbf{N}_a} \mathbf{x}^*_{i,a}) \end{aligned}$$

and therefore, using $\mathrm{OPT}_\lambda \leq L(\mathbf{x}^*,\gamma,z)$ we obtain the upper bound in (9). □

Corollary 1. *Let $\{\boldsymbol{x}^R_{i,a}\}_{(i,a)\in\mathbb{E}}$ be the assignments and $\{\boldsymbol{\beta}^R_a\}_{a\in\mathbb{A}}$ be the priority scores at the end of R iterations of Algorithm 2, then*

$$\begin{aligned} \mathit{OPT}_\lambda \leq & \sum_{(i,a)\in\mathbb{E}} \boldsymbol{r}_{i,a}\boldsymbol{x}^R_{i,a} - \lambda\boldsymbol{x}^R_{i,a}\log(\boldsymbol{x}^R_{i,a}) \\ & - \sum_{a\in\mathbb{A}} \lambda\log(\boldsymbol{\beta}^R_a)(\boldsymbol{C}_a - \sum_{i\in\boldsymbol{N}_a} \boldsymbol{x}^R_{i,a}) \end{aligned} \tag{10}$$

Proof. We can observe this using Lemma 2, by substituting $\gamma_a = \lambda\log(1/\boldsymbol{\beta}^R_a)$. Since initial value of $\boldsymbol{\beta}_a$ is $(1+\epsilon)^{-R}$, and there is a increase of at most $(1+\epsilon)^R$ factor, we have that $\boldsymbol{\beta}_a \leq 1$, so that $\gamma_a = \lambda\log(1/\boldsymbol{\beta}^R_a) \geq \lambda\log(1) = 0$. Therefore, it is a valid assignment of γ_a. □

Primal-dual interpretation of PropAlloc $^+$. From the above discussion, observe that there is a one-to-one mapping between the priority scores $\boldsymbol{\beta}_a$ and dual variables γ_a. On setting $\boldsymbol{\beta}_a = e^{\frac{-\gamma_a}{\lambda}}$, we obtained that the assignments made by our algorithm are same as complimentary solution $\{\mathbf{x}^*_{i,a}\}$ given by (8). This provides a primal dual interpretation of the proportional allocation algorithm. The proportional allocation algorithm is essentially updating the dual variables based on the feasibility (over-allocation/under-allocation) of the primal complimentary solution.

Now, using observations similar to those made in Lemma 1 in the previous section, it is easy to see that algorithm PropAlloc $^+$ satisfies the following property.

Lemma 3. *For any $a \in \mathbb{A}$, unless $\boldsymbol{\beta}_a$ was increased in all iterations or decreased in all iterations, at the end of R iterations of Algorithm 2, **Alloc**$_a := \sum_{i \in \boldsymbol{N}_a} \boldsymbol{x}_{i,a} \in [(1+\epsilon)^{-2}\boldsymbol{C}_a, (1+\epsilon)^2\boldsymbol{C}_a]$.*

We are now ready to prove Theorem 2. Here we provide an outline, with detailed proof in the supplementary material.

Proof of Theorem 2 (Sketch). Without loss of generality, let's assume that $\mathbf{r}_{\max}$ is 1. This can be obtained by dividing all $\mathbf{r}_{i,a}$ by $\mathbf{r}_{min}$. $\mathbf{r}_{min}$ in the processed instance is then in fact the ratio of $\mathbf{r}_{\min}$ and $\mathbf{r}_{\max}$ of the original instance. Let $\mathbf{x}^R_{i,a}$ and $\boldsymbol{\beta}^R_a$ denote the value of assignments and priority scores at the end of R iterations of Algorithm 2 (before the processing in the last step was done to handle over-allocated advertisers). And, let $\mathbf{x}^M_{i,a}$ denote the feasible assignments obtained after the processing in the last step of the algorithm. Let $\text{weight}(M) := \sum_{i,a \in \mathbb{E}} \mathbf{r}_{i,a}\mathbf{x}^M_{i,a}$ denote the weight of this feasible fractional matching M.

Initially, $\boldsymbol{\beta}_a = (1+\epsilon)^{-R}$. From Lemma 3, for every a, either $\sum_{i \in \mathbf{N}_a} \mathbf{x}^R_{i,a} \in [(1+\epsilon)^{-2}\mathbf{C}_a, (1+\epsilon)^2\mathbf{C}_a]$, i.e., the advertiser budget constraint is approximately satisfied; or, we will have that $\boldsymbol{\beta}_a$ was continuously increased/decreased by $(1+\epsilon)$ factor for all R iterations, so that $\boldsymbol{\beta}^R_a$ is either 1 or $(1+\epsilon)^{-2R}$. Let us call the first set of advertisers where the budget constraint is approximately satisfied as $\mathcal{E}$. For these advertisers, $|\mathbf{C}_a - \sum_{i \in \mathbf{N}_a} \mathbf{x}_{i,a}| \le 3\epsilon\mathbf{C}_a$ for any $\epsilon \le 1$. Also, $\boldsymbol{\beta}^R_a \ge (1+\epsilon)^{-2R}$. Among the second set, let $\mathcal{O}$ be the set of advertisers $a \in \mathbb{A}$ with $\boldsymbol{\beta}^R_a = (1+\epsilon)^{-2R}$. Here, $\boldsymbol{\beta}_a$ was continuously decreased in order to decrease the allocation, and these advertisers will be over-allocated in the end. For the remaining $a \notin \mathcal{E}, a \notin \mathcal{O}$, we have $\boldsymbol{\beta}^R_a = 1$.

Using the upper bound from (10), and substituting the value of $\boldsymbol{\beta}^R_a$,

$$\begin{aligned}\text{OPT}_\lambda \quad \le \quad & \sum_{i,a} \mathbf{r}_{i,a}\mathbf{x}^R_{i,a} + \sum_{a \in \mathcal{O}} 2R\epsilon\lambda(\mathbf{C}_a - \sum_{i \in \mathbf{N}_a} \mathbf{x}^R_{i,a}) \\ & + \sum_{a \in \mathcal{E}} 2R\epsilon\lambda(3\epsilon\mathbf{C}_a) - \lambda \sum_{i,a} \mathbf{x}^R_{i,a} \log(\mathbf{x}^R_{i,a})\end{aligned}$$

The terms for $a \notin \mathcal{O}, a \notin \mathcal{E}$ do not appear in above because $\log(1/\boldsymbol{\beta}^R_a) = \log(1) = 0$ for those a. Next, we relate the above upper bound to the weight and entropy of the feasible fractional matching M. First, we substitute R as:

$$R = \tfrac{1}{2\epsilon\lambda}\left(1 + \lambda \log(\bar{N})\right), \tag{11}$$

(where $\bar{N} = \max_a \frac{\mathbf{C}_a}{|\mathbf{N}_a|}$) to decompose the above upper bound on OPT_λ as:

$$\text{OPT}_\lambda \le \sum_{i,a} \mathbf{r}_{i,a}\mathbf{x}^R_{i,a} + \sum_{a \in \mathcal{O}} (\mathbf{C}_a - \sum_{i \in \mathbf{N}_a} \mathbf{x}^R_{i,a}) \tag{12}$$

$$+\lambda \log(\bar{N}) \sum_{a \in \mathcal{O}} (\mathbf{C}_a - \sum_{i \in \mathbf{N}_a} \mathbf{x}^R_{i,a}) - \lambda \sum_{i,a} \mathbf{x}^R_{i,a} \log \mathbf{x}^R_{i,a} \tag{13}$$

$$+ \sum_{a \in \mathcal{E}} 3\epsilon \left(1 + \lambda \log(\bar{N})\right) \mathbf{C}_a \tag{14}$$

Now, matching M was created by removing $\sum_{i \in \mathbf{N}_a} \mathbf{x}^R_{i,a} - \mathbf{C}_a$ edges from $\{\mathbf{x}^R_{i,a}\}$ for every over-allocated advertiser $a \in \mathcal{O}$. Therefore, the second term in (12) accounts for the weight (since $\mathbf{r}_{\max} = 1$) of all edges removed, except for those in $a \in \mathcal{E}$. Since $a \in \mathcal{E}$ can be over-allocated by at most $3\epsilon\mathbf{C}_a$, we can show that for small ϵ, almost all the decrease in the weight is accounted for, and (12) is close to $\text{weight}(M)$:

$$(12) \le (1 + \tfrac{\delta}{2})\text{weight}(M), \text{ when}$$

$$\epsilon = \frac{\mathbf{r}_{\min}}{8(2 + \lambda \log(\bar{N}))}\delta, \tag{15}$$

Similarly, we show that the first term in (13) accounts for any increase in entropy due to removal of edges from $\mathbf{x}^R_{i,a}$, so that

$$(13) \le \lambda\text{Entropy}(\mathbf{x}^M_{i,a}) := \lambda \textstyle\sum_{i,a} \mathbf{x}^R_{i,a} \log(1/\mathbf{x}^R_{i,a})$$

Here, we utilize the fact that in Algorithm 2 does not decrease very small assignments: it only decreases assignment of edges while $\mathbf{x}^R_{i,a} \ge \mathbf{C}_a/|\mathbf{N}_a|$. Finally, for small ϵ, the last part (14) is negligible compared to $\text{weight}(M)$. Specifically, for the choice of ϵ in (15),

$$(14) \le \tfrac{\delta}{2}\text{weight}(M).$$

Combining these observations,

$$\text{OPT}_\lambda \quad \le \quad (1+\delta)\text{weight}(M) + \lambda\text{Entropy}(M)$$

Finally, from (11), substituting value of ϵ from (15), we have the number of iterations

$$R = \frac{1}{2\epsilon\lambda}\left(1 + \lambda \log(\bar{N})\right) \le \frac{8}{\mathbf{r}_{min}} \frac{(1 + \lambda \log(\bar{N}))^2}{\lambda\delta}$$

Then, the theorem statement is obtained on substituting back $\mathbf{r}_{\min}/\mathbf{r}_{\max}$ for $\mathbf{r}_{\min}$.

References

Agrawal, S. and Devanur, N. R. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1405–1424. Society for Industrial and Applied Mathematics, 2015.

Agrawal, S., Wang, Z., and Ye, Y. A dynamic near-optimal algorithm for online linear programming. *Computing Research Repository*, 2009.

Ahmadi, A. and Hall, G. Bipartite matching and vertex covers. http://www.princeton.edu/~amirali/Public/Teaching/ORF523/S16/ORF523_S16_Lec6_gh.pdf. Accessed: 2018-02-09.

Ahmed, F., Dickerson, J. P., and Fuge, M. Diverse weighted bipartite b-matching. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 35–41, 2017.

Ahn, K. J. and Guha, S. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput.*, 222:59–79, January 2013.

Arora, S., Hazan, E., and Kale, S. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012. doi: 10.4086/toc.2012.v008a006. URL https://doi.org/10.4086/toc.2012.v008a006.

Awerbuch, B. and Khandekar, R. Stateless distributed gradient descent for positive linear programs. *SIAM J. Comput.*, 38(6):2468–2486, 2009.

Bahmani, B., Kumar, R., and Vassilvitskii, S. Densest subgraph in streaming and mapreduce. *Proc. VLDB Endow.*, 5(5):454–465, January 2012.

Bahmani, B., Goel, A., and Munagala, K. Efficient primal-dual graph algorithms for mapreduce. In *International Workshop on Algorithms and Models for the Web-Graph*, pp. 59–78. Springer, 2014.

Bansal, N. and Sviridenko, M. The santa claus problem. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pp. 31–40, New York, NY, USA, 2006. ACM.

Bateni, M., Esfandiari, H., Mirrokni, V. S., and Seddighin, S. A study of compact reserve pricing languages. pp. 363–368, 2017.

Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, April 2002.

Charles, D., Chickering, M., Devanur, N. R., Jain, K., and Sanghi, M. Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads. In *Proceedings of the 11th ACM conference on Electronic commerce*, pp. 121–128. ACM, 2010.

Devanur, N. and Hayes, T. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *EC*, pp. 71–78, 2009.

Devanur, N. R., Jain, K., Sivan, B., and Wilkens, C. A. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pp. 29–38, New York, NY, USA, 2011. ACM.

Dhillon, I. S. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pp. 269–274, New York, NY, USA, 2001. ACM.

Duan, R. and Pettie, S. Linear-time approximation for maximum weight matching. *J. ACM*, 61(1):1:1–1:23, January 2014.

Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., and Pal, M. Online ad assignment with free disposal. In *WINE*, 2009.

Feldman, J., Henzinger, M., Korula, N., Mirrokni, V. S., and Stein, C. Online stochastic packing applied to display ad allocation. In *ESA*, pp. 182–194. Springer, 2010.

Garg, N. and Konemann, J. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007.

Gupta, A. and Molinaro, M. How the experts algorithm can help solve lps online. *Math. Oper. Res.*, 41(4):1404–1431, 2016.

Hopcroft, J. E. and Karp, R. M. A n5/2 algorithm for maximum matchings in bipartite. In *Proceedings of the 12th Annual Symposium on Switching and Automata Theory (Swat 1971)*, SWAT '71, pp. 122–125, Washington, DC, USA, 1971. IEEE Computer Society.

Huang, B. and Jebara, T. Loopy belief propagation for bipartite maximum weight b-matching. In *Artificial Intelligence and Statistics*, pp. 195–202, 2007.

Jebara, T. and Shchogolev, V. B-matching for spectral clustering. In Fürnkranz, J., Scheffer, T., and Spiliopoulou, M. (eds.), *Machine Learning: ECML 2006*, pp. 679–686, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

Krissinel, E. and Henrick, K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D*, 60 (12 Part 1):2256–2268, Dec 2004.

Lan, T., Kao, D. T. H., Chiang, M., and Sabharwal, A. An axiomatic theory of fairness in network resource allocation. In *INFOCOM 2010. 29th IEEE International Conference on Computer Communications*, pp. 1343–1351, 2010.

Mehta, A., Saberi, A., Vazirani, U., and Vazirani, V. Adwords and generalized online matching. *J. ACM*, 54(5): 22, 2007.

Noia, T. D., Rosati, J., Tomeo, P., and Sciascio, E. D. Adaptive multi-attribute diversity for recommender systems. *Inf. Sci.*, 382-383:234–253, 2017.

Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., and Cheng, X. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2793–2799. AAAI Press, 2016.

Plotkin, S. A., Shmoys, D. B., and Tardos, É. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20(2):257–301, 1995.

Qin, L. and Zhu, X. Promoting diversity in recommendation by entropy regularizer. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pp. 2698–2704, 2013.

Vee, E., Vassilvitskii, S., and Shanmugasundaram, J. Optimal online assignment with forecasts. In *EC*, pp. 109–118, 2010.

Venkatasubramanian, V. Fairness is an emergent self-organized property of the free market for labor. *Entropy*, 12(6):1514–1531, 2010.