Better bounds for matchings in the streaming model

Michael Kapralov[*]

July 26, 2012

Abstract

In this paper we present improved bounds for approximating maximum matchings in bipartite graphs in the streaming model. First, we consider the question of how well maximum matching can be approximated in a single pass over the input when $\tilde{O}(n)$ space is allowed, where n is the number of vertices in the input graph. Two natural variants of this problem have been considered in the literature: (1) the edge arrival setting, where edges arrive in the stream and (2) the vertex arrival setting, where vertices on one side of the graph arrive in the stream together with all their incident edges. The latter setting has also been studied extensively in the context of *online algorithms*, where each arriving vertex has to either be matched irrevocably or discarded upon arrival. In the online setting, the celebrated algorithm of Karp-Vazirani-Vazirani achieves a $1 - 1/e$ approximation by crucially using randomization (and using $\tilde{O}(n)$ space). Despite the fact that the streaming model is less restrictive in that the algorithm is not constrained to match vertices irrevocably upon arrival, the best known approximation in the streaming model with vertex arrivals and $\tilde{O}(n)$ space is the same factor of $1 - 1/e$.

We show that no (possibly randomized) single pass streaming algorithm constrained to use $\tilde{O}(n)$ space can achieve a better than $1 - 1/e$ approximation to maximum matching, even in the vertex arrival setting. This leads to the striking conclusion that no single pass streaming algorithm can get any advantage over online algorithms unless it uses significantly more than $\tilde{O}(n)$ space. Additionally, our bound yields the best known impossibility result for approximating matchings in the *edge arrival* model (improving upon the bound of $2/3$ proved by Goel at al[SODA'12]).

Second, we consider the problem of approximating matchings in multiple passes in the vertex arrival setting. We show that a simple fractional load balancing approach achieves approximation ratio $1 - e^{-k}k^{k-1}/(k-1)! = 1 - \frac{1}{\sqrt{2\pi k}} + o(1/k)$ in k passes using linear space. Thus, our algorithm achieves the best possible $1 - 1/e$ approximation in a single pass and improves upon the $1 - O(\sqrt{\log\log k/k})$ approximation in k passes due to Ahn and Guha[ICALP'11]. Additionally, our approach yields an efficient solution to the Gap-Existence problem considered by Charles et al[EC'10].

[*]Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, Email: kapralov@stanford.edu

1 Introduction

The need to process modern massive data sets necessitates rethinking classical solutions to many combinatorial optimization problems from the point of view of space usage and type of access to the data that algorithms assume. Applications in domains such as processing web-scale graphs, network monitoring or data mining among many others prohibit solutions that load the whole input into memory and assume random access to it. The streaming model of computation has emerged as a more realistic model for processing modern data sets. In this model the input is given to the algorithm as a stream, possibly with multiple passes allowed. The goal is to design algorithms that require small space and ideally one or a small constant number of passes over the data stream to compute a (often approximate) solution. For many problems with applications in network monitoring, it has been shown that space polylogarithmic in the size of the input is often sufficient to compute very good approximate solutions. On the other hand, even basic graph algorithms have been shown to require $\Omega(n)$ space in the streaming model[5], where n is the number of vertices. A common relaxation is to allow $O(n \cdot \text{polylog}(n))$ space, a setting often referred to as the *semi-streaming* model.

1.1 Matchings in the streaming model

The problem of approximating maximum matchings in bipartite graphs has received significant attention recently, and very efficient small-space solutions are known when multiple passes are allowed[6, 14, 4, 1, 2, 11]. The best known algorithm due to Ahn and Guha [1] achieves a $1 - O(\sqrt{\log\log k/k})$ in k passes for the weighted as well as the unweighted version of the problem using $\tilde{O}(kn)$ space.

Single pass algorithms. All algorithms mentioned above require at least two passes to achieve a nontrivial approximation. The problem of approximating matchings in a single pass has recently received significant attention[8, 11]. Two natural variants of this problem have been considered in the literature: (1) the edge arrival setting, where edges arrive in the stream and (2) the vertex arrival setting, when vertices on one side of the graph arrive in the stream together with all their incident edges. The latter setting has also been studied extensively in the context of *online algorithms*, where each arriving vertex has to either be matched irrevocably or discarded upon arrival.

In a single pass, the best known approximation in the edge arrival setting is still $1/2$, achieved by simply keeping a maximal matching (this was recently improved to $1/2 + \epsilon$ for a constant $\epsilon > 0$ under the additional assumption of random edge arrivals[11]). It was shown in [8] that no $\tilde{O}(n)$ space algorithm can achieve a better than $2/3$ approximation in this setting.

In the vertex arrival setting, the best known algorithms achieve an approximation of $1 - 1/e$. The assumption of vertex arrivals allows one to leverage results from online algorithms [10, 13, 9] . In the online model vertices on one side of the graph are known, and vertices on the other side arrive in an adversarial order. The algorithm has to either match a vertex irrevocably or discard upon arrival. The celebrated algorithm of Karp-Vazirani-Vazirani achieves a $1 - 1/e$ approximation for the online problem by crucially using randomization (additionally, this algorithm only uses $\tilde{O}(n)$ space). A *deterministic* single pass $\tilde{O}(n)$ space $1 - 1/e$ approximation in the vertex arrival setting was given in [8] (such a deterministic solution is provably impossible in the online setting). In [8], the authors also showed by analyzing a natural one-round communication problem that no single-pass streaming algorithm that uses $\tilde{O}(n)$ space can obtain a better than $3/4$ approximation in the vertex arrival setting. They also provided a protocol for this communication problem that matches the $3/4$ approximation ratio, suggesting that new techniques would be needed to prove a stronger impossibility result.

Lop-sided graphs. The techniques for matching problems outlined above yield efficient solutions that use $\tilde{O}(|P| + |Q|)$ space, where $|P|$ and $|Q|$ are the sizes of the sets in the bipartition. While this is a reasonable space bound to target, this can be prohibitively expensive for lop-sided graphs that arise, for example, in applications to ad allocations. Here the P side of the graph corresponds to the set of *advertisers*, and the Q

side to the set of *impressions* [3]. An important constraint is that the set of impressions Q may be so large that it is not feasible to represent it explicitly, ruling out algorithms that take $O(|P| + |Q|)$ space.

Data model for lop-sided graphs. Since the set Q cannot be represented explicitly, it is important to fix the model of access to Q. Here we assume the following scenario. Vertices in P arrive in the stream in an adversarial order, together with a representation of their edges. We make no assumptions on the way the edges are represented. For example, some edges could be stored explicitly, while others may be represented implicitly. We assume access to the following two functions:

1. LIST-NEIGHBORS(u, S) which, given a set of vertices $S \subseteq Q$ and a vertex $u \in P$, lists the neighbors of u in S;

2. NEW-NEIGHBOR(u, S) which, given a set of vertices $S \subseteq Q$ and a vertex $u \in P$ outputs a neighbor of u outside of the set S.

1.2 Our results

In this paper, we improve upon the best known bounds for both the single pass and multi-pass settings. In the single pass setting, we prove an optimal impossibility result for vertex arrivals, which also yields the best known impossibility result in the edge arrival model. For the multipass setting, we give a simple algorithm that improves upon the approximation obtained by Ahn and Guha in the vertex arrival setting, as well as yields an efficient solution to the Gap-Existence problem considered by Charles et al[3].

Lower bounds. In this paper we build upon the communication complexity approach taken in [8] to obtain lower bounds via what can be viewed as multi-party communication complexity. Our main result is an optimal bound on the best approximation ratio that a single-pass $\tilde{O}(n)$ space streaming algorithm can achieve in the vertex arrival setting:

Theorem 1 *No (possibly randomized) one-pass streaming algorithm that outputs a valid matching with probability at least $3/4$ can obtain a better than $1 - 1/e + \delta$-approximation to the maximum matching, for any constant $\delta > 0$, unless it uses at least $n^{1+\Omega(1/\log\log n)}$ space, even in the vertex arrival model.*

We note that this bound is matched by the randomized KVV algorithm[10] for the online problem and the deterministic $\tilde{O}(n)$ space algorithm of [8]. One striking consequence of our bound is that no single-pass streaming algorithm can improve upon the more constrained *online* algorithm of KVV, which has to make irrevocable decisions, unless is uses significantly more than $\tilde{O}(n)$ space. Our bound also improves upon the best known bound of $2/3$ for small space one-pass streaming algorithms in the *edge arrival model*.

Comparison with [8] It was shown in [8] via an analysis of the natural two-party communication problem that no one-pass streaming algorithm that uses $\tilde{O}(n)$ space can achieve approximation better than $2/3$ in the edge arrival setting and $3/4$ in the vertex arrival setting. Furthermore, the authors also gave a communication protocol that proves the optimality of both bounds for the communication problem, thus suggesting that a more intricate approach would be needed to prove better impossibility results.

In this paper we prove the optimal bound of $1 - 1/e$ on the best approximation that a single-pass $\tilde{O}(n)$ space algorithm can achieve *even in the vertex arrival setting*. While the lower bounds from [8] follow from a construction of a distribution on inputs that consists of two parts and hence yields a two-party communication problem, here we obtain an improvement by constructing hard input sequences that consist of k parts instead of two, getting a lower bound that approaches $1 - 1/e$ for large k. This can be viewed as multi-party communication complexity of bipartite matching, but we choose to present our lower bound in different terms for simplicity. We note that the approach of [8] to a multi-party setting requires a substantially different construction. We discuss the difficulties and our approach to overcoming them in section 2.

Upper bounds. We show that a simple algorithm based on fractional load balancing achieves the optimal $1-1/e$ approximation in a single pass and $1-\frac{1}{\sqrt{2\pi k}}+o(k^{-1/2})$ approximation in k passes, improving upon the best known algorithms for this setting:

Theorem 2 *There exists an algorithm for approximating the maximum matching M in a bipartite graph $G=(P,Q,E)$ with the P side arriving in the stream to factor $1-e^{-k}k^{k-1}/(k-1)! = 1-\frac{1}{\sqrt{2\pi k}}+O(k^{-3/2})$ in k passes using $O(|P|+|Q|)$ space and $O(m)$ time per pass.*

Remark 3 *Note that our algorithm extends trivially to the case when vertices in P have integral capacities $B_u, u\in P$, corresponding to advertiser budgets.*

The gap-existence problem. In [3] the authors give an algorithm for the closely related *gap-existence* problem. In this problem the algorithm is given a bipartite graph $G=(A,I,E)$, where A is the set of advertisers with budgets $B_a, a\in A$ and I is the set of impressions. The graph is lopsided in the sense that $|I|\gg|A|$. A matching M is *complete* if $|M\cap\delta(i)|=1$ for all $i\in I$ and $|M\cap\delta(a)|=B_a$ for all $a\in A$. The gap-existence problem consists of distinguishing between two cases:

(YES) there exists a complete matching with budgets B_a;

(NO) there does not exist a complete matching with budgets $\lfloor(1-\epsilon)B_a\rfloor$.

The approach of [3] is via sampling the I side of the graph, and yields a solution that allows for non-trivial subsampling when the budgets are large. In particular, they obtain an algorithm with runtime $O\left(\frac{|A|\log|A|}{\epsilon^2}\cdot\frac{|I|}{\min_a|B_a|}\right)$, which is sublinear in the size of the graph when all budgets are large. In section 5 we improve significantly upon their result, showing

Theorem 4 *Gap-Existence can be solved in $O(\log(|I|\cdot\sum_{a\in B_a}B_a)/\epsilon^2)$ passes using space $O(\sum_{a\in A}B_a/\epsilon)$. The time taken for each pass is linear in the representation of the graph.*

It should also be noted that the result of [3] could be viewed as a single pass algorithm, albeit with the stronger assumption that the arrival order in the stream is random.

Organization: In section 2 we present the framework of our lower bound, which relies on a special family of graphs that we refer to as (d,k,δ)-packing. We then give a construction of a (d,k,δ)-packing in section 3. Our basic multipass algorithm for approximating matchings is presented in section 4, and the algorithm for Gap-existence is given in section 5.

2 Single pass lower bound

In this section we define the notion of a (d,k,δ)*-packing*, our main tool in proving the lower bound. A (d,k,δ)-packing is a family of graphs parameterized by the set of root to leaf paths in a d-ary tree of height k, inspired by Ruzsa-Szemerédi graphs, i.e. graphs whose edge set can be partitioned into large *induced* matchings. In this section we will show that existence of a (d,k,δ)-packing with a large number of edges implies lower bounds on the space complexity of achieving a better than $1-1/e$ approximation to maximum matchings in a single pass over the stream.

We first recall the definition of induced matchings and ϵ-Ruzsa-Szemerédi graphs.

Definition 5 *Let $G=(P,Q,E)$ denote a bipartite graph. A matching $F\subseteq E$ that matches a set $A\subseteq P$ to a subset $B\subseteq Q$ is* induced *if $E\cap(A\times B)=F$.*

Definition 6 *A bipartite graph $G = (P, Q, E)$ with $|P| = |Q| = n$ is an ϵ-Ruzsa-Szemerédi graph if one can write $E = \bigcup_{i=1}^{k} M_i$, where each M_i is an induced matching and $|M_i| = \epsilon n$ for all i.*

Several constructions of Ruzsa-Szemerédi graphs with a large number of edges are known. We will use the techniques pioneered in [7], where the authors construct ϵ-Ruzsa-Szemerédi graphs with constant $\epsilon < 1/3$, and the extensions developed in [8], where it is proved that

Theorem 7 *[8] For any constant $\delta \in (0, 1/2)$ there exist bipartite $(1/2 - \delta)$-Ruzsa-Szemerédi graphs on $2n$ nodes with $n^{1+\Omega(1/\log\log n)}$ edges.*

In the rest of the section we define a distribution on input instances for our problem of approximating maximum matchings in a single pass in the streaming model. We start by providing intuition for our distribution. It is useful to first recall how the best known lower bound of $3/4$ for the same setting is proved in [8]. The stream in [8] consists of two 'phases'. In the first phase, the algorithm is presented with a graph $G = (P, Q, E)$ such that $|P| = n, |Q| = 2n$ and the edge set E can be represented as a union of induced 2-matchings M_i, $i = 1, \ldots, k, k = n^{\Omega(1/\log\log n)}$, where M_i matches a subset $A_i \subseteq P$ such that $|A_i| \geq (1/2 - \delta)n$ to a subset $B_i \subseteq Q, |B_i| = (1 \pm \delta)n$. Then an index i is chosen uniformly at random from $[1 : k]$, and in the second part of the stream a matching arrives that matches a new set of vertices P^* to $Q^* = Q \setminus B_i$, making the edges of the (uniformly random) matching M_i crucial for constructing a better than $3/4$ approximation to the maximum matching in the whole instance. It is then shown, using an additional randomization trick, that the algorithm essentially needs to store $\Omega(1)$ bits for each edge in each induced matching M_i if it beats the $3/4$ approximation ratio.

We generalize this approach by constructing hard distributions on inputs that consist of *multiple phases*, for which any algorithm that achieves a better than $1 - 1/e$ approximation is essentially forced to remember $\Omega(1)$ bits per edge of the input graph. Ensuring that this is the case is the main challenge in generalizing the construction in [8] to a multiphase setting. We address this challenge using the notion of a (d, k, δ)-packing, which we now define.

2.1 (d, k, δ)-packing

Let $\mathcal{T}$ denote a d-ary tree of height k. A (d, k, δ)-packing will be defined as a function mapping root-to-leaf paths p in $\mathcal{T}$ to bipartite graphs on the vertex set (T, S), where T and S are the two sides of the bipartition. We will write $G(p)$ to denote the graph that a path p is mapped to by the packing.

The vertex set of $G(p)$ for each root-to-leaf path p will always be (T, S), so that the choice of p determines the set of edges of the graph. We partition the set S as $S = S_0 \cup \ldots \cup S_{k-1} \cup S_k$ (the sets $S_i, i = 0, \ldots, k$ are disjoint and correspond to $k + 1$ 'phases' of the input instance). We will always have $|T| = (1 + O(\delta))|S|$ for an arbitrarily small constant $\delta > 0$.

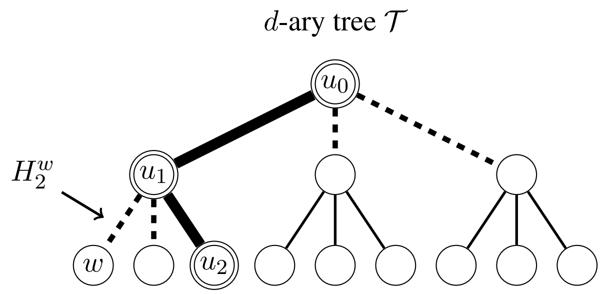

Figure 1: A root to leaf path in $\mathcal{T}$. Thick solid edges represent the edges of the path $(r = u_0, u_1, u_2)$. Dashed edges incident on nodes on the path $\mathcal{P}$ correspond to subgraphs H_i^w for $i = 0, 1$ and w a child of u_i.

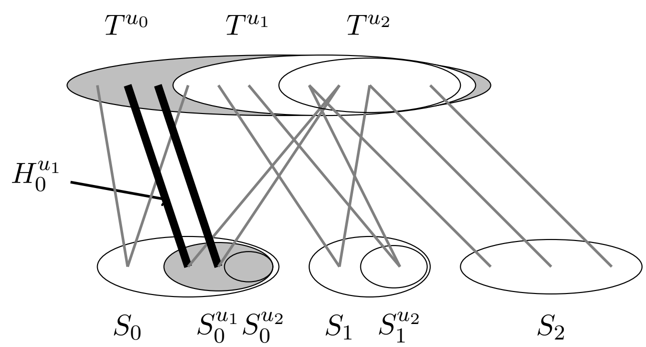

Figure 2: Subgraphs (T^{u_i}, S_i) that arrive in the stream. The edges of induced near-regular subgraph $H_0^{u_1}$ induced by $(T^{u_0} \setminus T^{u_1}) \cup S_0^{u_1}$ are shown in bold.

We now associate several sets of vertices on the T and S side of the bipartition with each node in the binary tree $\mathcal{T}$. Let $u \in \mathcal{T}$ be a node *at distance* $i \in [0, k]$ from the root. The following sets are associated with u:

1. a subset $T^u \subseteq T$, such that if w is a child of u in $\mathcal{T}$, one always has $T^w \subset T^u$;
2. for each $j \in [0 : i-1]$, a set S_j^u, such that if w is a child of u in $\mathcal{T}$, one always has $S_j^w \subset S_j^u$. To simplify notation, we set $S_i^{u_i} := S_i$.

We now describe the hard inputs that we will use. The input sequence is split into $k+1$ phases. The i-th phase corresponding to the i-th vertex on the path p from root to a leaf, where $i = 0, \ldots, k$ (see Fig. 1). During phase i the edges of the subgraph induced by $G_i(p) = (T^{u_i}, S_i, E_i(p))$ arrive in the stream. Crucially, the graph $G_i(p)$ will be a union of induced sparse subgraphs indexed by children of u_i.

This setup is illustrated in Fig. 1, where (a) all edges of the path $p = (r = u_0, u_1, u_2)$ are shown in bold and (b) all edges of $\mathcal{T}$ that are incident on nodes of p are dashed since the corresponding subgraphs H_i^w arrive in the stream. The path p yields a nested sequence $T = T^{u_0} \supset T^{u_1} \supset \ldots T^{u_k}$ shown in Fig. 2.

The reason behind the fact that this construction presents a hard instance for small space algorithms is as follows. At each step i the algorithm is presented with all the subgraphs H_i^w, of which all except the uniformly random one (corresponding to the next node on the path p, i.e. $H_i^{u_{i+1}}$) will be useful for constructing a large matching in the whole instance. Large here means a matching of size at least a $(1-(1-1/k)^k+\delta')$ fraction of the maximum for some constant $\delta' > 0$. To show that only these special subgraphs are useful for constructing a large matching, we will later exhibit a *directed cut* of appropriate size in the graph $G(p)$ that consists only of the edges of $H_i^{u_{i+1}}$, $i = 0, \ldots, d-1$ (see Lemma 13). The key to exhibiting such a cut is the special structure of the sets $S_i^{u_k}$ for $i = 0, \ldots, k-1$ that we define in property (2) of (d, k, δ)-packings below. An additional randomization trick will allow us to show that a construction of a (d, k, δ)-packing immediately yields a lower bound of essentially $\Omega(dn)$ on the space required for a single-pass algorithm to achieve an approximation ratio better than $1 - (1 - 1/k)^k + \delta'$ for a constant $\delta' > 0$.

We now transform the intuitive description above into a formal argument. We will use the following

Definition 8 *We call a bipartite graph $G = (P, Q, E)$ (a, b, δ)-*almost regular *if (1) at most a δ fraction of vertices in P has degree outside of $[(1-\delta)a, (1+\delta)a]$, and no vertex has degree larger than $(1+\delta)a$ and (2) at most a δ fraction of vertices in Q has degree outside of $[(1-\delta)b, (1+\delta)b]$, and no vertex has degree larger than $(1+\delta)b$.*

Definition 9 ((d, k, δ)-packing) *A mapping from the set of root-to-leaf paths p in a d-ary tree $\mathcal{T}$ to the set of bipartite graphs $G(p) = (T, S, E(p))$ is a (d, k, δ)-packing if the following conditions are satisfied.*

Let $p = (r = u_0, u_1, \ldots, u_k)$ be a root-to-leaf path in $\mathcal{T}$. Let $G(p) = (T, S, E(p))$ denote the graph that the path p is mapped to. Then the nested sequences of sets $T = T^{u_0} \supset T^{u_1} \supset \ldots \supset T^{u_{k-1}} \supset T^{u_k}$, and $S_i = S_i^{u_i} \supset S_i^{u_{i+1}} \supset \ldots \supset S_i^{u_k}$ satisfies the following properties for all $i = 0, \ldots, k-1$:

1. *For a constant $\gamma > 0$, one has for every child w of u_i in the tree $\mathcal{T}$ that the subgraph H_i^w induced by $(T^{u_i} \setminus T^w) \cup S_i^w$ is $((k-1)\gamma, k\gamma, \delta)$-almost regular.*

2. *there exists a set $Z^{u_i} \subset T$ such that $|Z^{u_i}| \le O(\delta/k^2)|T^{u_i}|$, and the subgraph induced by $(T^{u_i} \setminus (T^{u_k} \cup Z^{u_i})) \cup S_i^{u_k}$ contains only the edges of $H_i^{u_{i+1}}$.*

3. *there exists a matching of at least a $1 - \delta$ fraction of S_i to $T^{u_i} \setminus T^{u_{i+1}}$;*

4. *$|T^{u_i}| = (1 + O(\delta))(1 - 1/k)^{-k+i} n$ and $|S_i^{u_j}| = (1 + O(\delta))(1 - 1/k)^{-k+j} n/k$ for all $j = i, \ldots, k-1$.*

5. *there exists a matching of at least a $1 - \delta$ fraction of S_k to T^{u_k}.*

Furthermore, for each $i = 0, \ldots, k$ the edge set of the subgraph induced by $T^{u_i} \cup S_i$ only depends on the nodes of p at distance at most i from the root.

Remark 10 *One could replace property (1) with the requirement that H_i^w be a matching of a $1 - O(\delta)$ fraction of S_i^w to $T^{u_i} \setminus T^w$, and still get a lower bound that tends to $1 - 1/e$ for large k, albeit with slightly worse convergence. We prefer to use the more complicated definition to obtain the clean approximation ratio $1 - (1 - 1/k)^k + O(\delta)$, where δ can be chosen an arbitrarily small constant, for any $k > 1$.*

In what follows we will often refer to properties of (d, k, δ)-packings by number, without specifying each time that Definition 9 is meant.

In the rest of this section we will show that existence of large (d, k, δ)-packings implies space lower bounds for approximating matchings in one pass in the streaming model, thus proving

Theorem 11 *If a (d, k, δ)-packing with $\Theta(n)$ vertices exists for sufficiently large constant $k > 0$ and $\delta = O(1/k^3)$, then no one-pass streaming algorithm can obtain a better than $(1 - (1 - 1/k)^k + \delta')$-approximation for any constant $\delta' > 0$ in space $o(nd)$, even when vertices on one side of the graph arrive in the stream together with all their edges.*

Together with the construction of a (d, k, δ)-packing with $d = n^{\Omega(1/\log\log n)}$ and $\delta = O(1/k^3)$ given in section 3, this will yield a proof of Theorem 1.

2.2 Distribution over inputs

We now formally define the (random) input graph $\mathcal{I} = (P, Q, E)$ based on a (d, k, δ)-packing. We will always have $P = \bigcup_{i=0}^{k} S_i$ and $Q = T$, but it will be useful to have notation for the parts P and Q of the bipartition of $\mathcal{I}$. Let $p = (r = u_0, u_1, \ldots, u_k)$ denote the path from the root of $\mathcal{T}$ to a uniformly random leaf. Let $G = G(p)$ denote the graph that the path p is mapped to by our (d, k, δ)-packing.

Let $T = T^{u_0} \supset T^{u_1} \supset T^{u_2} \supset \ldots \supset T^{u_k}$ denote the sequence of subsets of T corresponding to p. For each $i = 0, \ldots, k-1$ and each child w of u_i let $H_i^w = (X_i^w, Y_i^w, E_i^w)$ denote the almost regular graph induced by $X_i^w \cup Y_i^w$, where $X_i^w = T^{u_i} \setminus T^w$ and $Y_i^w = S_i^w$.

We now introduce some randomness into the graph H_i^w. Let $\bar{H}_i^w$ be obtained from H_i^w via the following subsampling process. For each i and w let K_i^w denote a uniformly random subset of X_i^w of size $\delta|X_i^w|$ for a small constant δ. Let $b_x^{i,w} = 1$ if $x \in K_i^w$ and 0 o.w. Then for each $x \in X_i^w$ the graph $\bar{H}_i^w$ contains all edges incident on x in H_i^w if $b_x^{i,w} = 0$ and none of the edges incident on x otherwise. For each $i = 0, \ldots, k-1$ let $\mathbf{b}_i = (b_x^{i,u_{i+1}})_{x \in X^i_{u_{i+1}}}$. Note that $\bar{H}_i^w$ is a $((k-1)\gamma, k\gamma, O(\delta))$-almost regular. For each $i = 0, \ldots, k-1$ let $\bar{G}_i(p) = (T^{u_i}, S_i, \bar{E}_i(p))$ denote the subgraph with bipartition (T^{u_i}, S_i) such that $\bar{E}_i(p)$ is the union of the edges of all graphs $\bar{H}_i^w$ over all children w of u_i. Let $G_k(p) = (T^{u_k}, S_k, \bar{E}_k(p))$ be a subgraph that consists of a perfect matching between S_k and T^{u_k} (see Fig. 2). The instance $\mathcal{I}$ is the union of $\bar{G}_i(p)$ over $i = 0, \ldots, k$.

We now specify the order in which the vertices appear in the stream. The stream will consist of $k+1$ *phases*. For each $i = 0, \ldots, k$ the vertices and edges of $\bar{G}_i(p)$ arrive in phase i in an arbitrary order.

This completes the description of the input. We now turn to proving Theorem 11. We will need the following claim

Claim 12 *G contains a matching of size at least $(1 - O(\delta))(1 - 1/k)^{-k} n$.*

Proof: It is sufficient to match a $1 - \delta$ fraction of S_i to $T^{u_i} \setminus T^{u_{i+1}}$ for all $i = 0, \ldots, k-1$, as guaranteed by property (3), and match the vertices in T^{u_k} to S_k. This matches a $1 - O(\delta)$ fraction of T, and hence yields the required matching. ■

2.3 Bounding performance of a small space algorithm

By Yao's minimax principle it is sufficient to upper bound the performance of a deterministic small space algorithm that succeeds with probability at least $1/2$. To do that, we bound the size of the matching that a small space algorithm can output at the end of the stream. Let E^* denote the set of edges that an algorithm outputs at the end of the stream. We first upper bound the approximation ratio that the algorithm obtains in terms of the number of edges in $E(H_i^{u_{i+1}}) \cap E^*$, where $p = (u_0, u_1, \ldots, u_k)$ is the uniformly random path from the root to a leaf in $\mathcal{T}$.

Lemma 13 *The size of the matching output by the algorithm is bounded by*

$$\Big((1 - 1/k)^{-k} - 1\Big) n + \sum_{i=0}^{k-1} |E(H_i^{u_{i+1}}) \cap E^*| + O(\delta k^2 n).$$

Proof: Consider the cut (A, B), where $A = \Big(T^0 \setminus (T^{u_k} \cup \bigcup_{i=0}^{k-1} Z^{u_i})\Big) \cup \bigcup_{i=0}^{k-1} (S_i \setminus S_i^{u_k})$ and $B = T^{u_k} \cup S^* \cup \bigcup_{i=0}^{k-1} S_i^{u_k} \cup \bigcup_{i=0}^{k-1} Z^{u_i}$. Here Z^{u_i} are the sets whose existence is guaranteed by property (2).

By the maxflow/mincut theorem, the size of the matching output by the algorithm is bounded by $|A \cap P| + |B \cap Q| + |((A \cap Q) \times (B \cap P)) \cap E^*|$. Furthermore, by property (2) in Definition 9 for the sets A and B one has, using the fact that there are no edges from S^* to $T \setminus T^{u_k}$ that $((A \cap Q) \times (B \cap P)) \cap E \subset \bigcup_{i=0}^{k-1} E(H_i^{u_{i+1}})$, and hence

$$|((A \cap Q) \times (B \cap P)) \cap E^*| \leq \sum_{i=0}^{k-1} |E(H_i^{u_{i+1}}) \cap E^*|. \tag{1}$$

Combining these estimates, we get that the size of the matching output by the algorithm is bounded by

$$\left| \bigcup_{i=0}^{k-1} (S_i \setminus S_i^{u_k}) \right| + |T^{u_k}| + \sum_{i=0}^{k-1} |Z^{u_i}| + \sum_{i=0}^{k-1} |E(H_i^{u_{i+1}}) \cap E^*|,$$

Recall that $|S_i| = (1+O(\delta))(1-1/k)^{-k+i}$ and $|S_i^{u_k}| = (1+O(\delta))n/k$ by property (4). Thus, the first term is at most

$$\begin{aligned}(1+O(\delta)) \left(\sum_{i=0}^{k-1} (1-1/k)^{-k+i} - 1 \right) n/k &= (1+O(\delta)) \left((1-1/k)^{-k} \frac{1-(1-1/k)^k}{1-(1-1/k)} - k \right) n/k \\ &= (1+O(\delta))((1-1/k)^{-k} - 2)n.\end{aligned}$$

Recalling that $T^{u_k} = (1+O(\delta))n$ by property (4) and $|Z^{u_i}| = O(k\delta)n$ by property (2) completes the proof. ■

We now show that no small space algorithm that is correct with probability at least $1/2$ can output more than a vanishingly small fraction of edges in $\bigcup_{i=0}^{k-1} E(H_i^{u_{i+1}})$. Recall that the vectors of bits flipped in the subsampling process that correspond to vertices (and their edge neighborhoods) in $\bar{H}_i^{u_{i+1}}$ are denoted by $\mathbf{b}_i$.

Lemma 14 *Let $\mathcal{I}$ denote the distribution on input graphs obtained from a (d, k, δ)-packing for constant k and $\delta = O(1/k^3)$. Let A be a $o(nd)$ space algorithm that is correct with probability at least $1/2$. Then for each $i = 0, \ldots, k-1$ the expected number of edges in $E(\bigcup_{i=0}^{k-1} H_i^{u_{i+1}})$ retained by A conditional on A being correct is $o(n)$.*

Proof:

We give the algorithm the following information for free. At the end of phase i the algorithm knows all vectors $\mathbf{u}_1, \ldots, \mathbf{u}_i$ on the path chosen in the distribution (of course, the algorithm does not know $\mathbf{u}_{i+1}$). This only makes the algorithm more powerful.

Let $\mathcal{G}_i$ denote the set of phase i graphs, i.e. the set of possible graphs on the vertices $T^{u_i} \cup S_i$. Since the algorithm knows all vectors $\mathbf{u}_1, \ldots, \mathbf{u}_i$, these graphs are solely determined by the choices made in the subsampling process in H_i^w for each w. Denote the state of the memory of the algorithm after i-th phase for $i = 0, \ldots, k-1$ by m_i. For each i between 0 and $k-1$ we denote the function that maps m_{i-1} and the graph $G_i = (T^{u_i}, S_i, E_i) \in \mathcal{G}_i$ to m_i by $\phi_i : \{0,1\}^s \times \mathcal{G}_i \to \{0,1\}^s$, where s is the number of bits of space that the algorithm uses. Wlog assume $m_{-1} = 0$.

Denote by E^* the set of edges that the algorithm outputs at the end of the stream. Denote the event that the algorithm is correct by $\mathcal{C}$. Let $E_i^* := E^* \cap (S_i \times Q)$. Let $M_i \in \{0,1\}^s$ denote the (random) state of the memory of the algorithm at the end of phase i. Let $\mathcal{D} := \{|E^*| = \Omega(n)\} \wedge \mathcal{C}$ and $\mathcal{D}_i = \{|E_i^*| = \Omega(1/k)n\} \wedge \mathcal{C}$.

We prove the lemma by contradiction. Suppose that conditional on being correct, the algorithm retains $\Omega(n)$ edges of $\bigcup_{i=0}^{k-1} E(H_i^{u_{i+1}})$. Then a simple averaging argument using the assumption that $\mathbf{Pr}[\mathcal{C}] \geq 1/2$ shows that $\mathbf{Pr}[\mathcal{D}] = \Omega(1)$ and there exists $j \in [0 : k-1]$ such that $\mathbf{Pr}[\mathcal{D}_j] \geq C/k$ for a constant $C > 0$. We will now concentrate on phase j. Denote the set of *good* memory configurations by $G = \{(m_{j-1}, m_j) \in \{0,1\}^s : \mathbf{Pr}[\mathcal{D}_j | M_{j-1} = m_{j-1}, M_j = m_j] \geq C/(2k))\}$. Thus, G is a set of memory configurations in the $j-1$-st and j-th phases such that conditional on $(M_{j-1}, M_j) \in G$ the algorithm is likely to output a lot of edges of $\bigcup_{i=0}^{k-1} E(H_i^{u_{i+1}})$. Then

$$\mathbf{Pr}[(M_{j-1}, M_j) \in G] + (C/(2k))\mathbf{Pr}[(M_{j-1}, M_j) \notin G] \geq \mathbf{Pr}[\mathcal{D}_j] \geq C/k,$$

so

$$\mathbf{Pr}[(M_{j-1}, M_j) \in G] \geq C/(2k). \tag{2}$$

Before proceeding, we prove an auxiliary lemma. Recall that in the definition of a (d,k,δ)-packing for all d children w of u_i the graph $\bar{H}_i^w$ is obtained from H_i^w by keeping edges incident to a uniformly random subset of a $1-\delta$ fraction of nodes in X_i^w. Thus, there are at least $\binom{|X_i^w|}{\delta|X_i^w|}^d = 2^{\eta dn}$ graphs in $\mathcal{G}_i$, where $\eta > 0$ is a constant. The following claim follows similarly to [8]. We give a proof here for completeness.

Claim 15 *Let $\alpha > 0$ be a constant and let F be any subset of $\mathcal{G}_i$. Let G_F denote a set of edges that are contained in at least $1/2$ of the graphs in F. Let $J \subseteq [1:d]$ be the set of indices such that G_F contains at least $\alpha|X_i^w|$ edges from H_{i-1}^w, where w is the j-th child of u_{i-1}, for each $j \in J$. Then if $|F| \geq 2^{(\eta - o(1))dn}$, $|J| = o(d)$.*

Proof: Let $|J| = d_1$. Recall that by property (1) the maximum degree in H_i^w is bounded above by $c := (1+O(\delta))\gamma k$. Thus, the number of graphs that can be in F is bounded by

$$\binom{(1-\alpha/c)|X_i^w|}{\delta|X_i^w|}^{d_1}\binom{|X_i^w|}{\delta|X_i^w|}^{d-d_1} = \left(2^{-\Omega(|X_i^w|)}\binom{|X_i^w|}{\delta|X_i^w|}\right)^{d_1}\binom{|X_i^w|}{\delta|X_i^w|}^{d-d_1} = 2^{-\Omega(d_1 n)}2^{\eta dn}.$$

It then follows that if $d_1 = \Omega(d)$, we have $|F| \leq 2^{(\eta - \Omega(1))dn}$, contradicting our assumption on the size of F. ■

Let $\mathcal{E}_j$ denote the event that $|\phi^{-1}_{M_{j-1}}(M_j)| \geq 2^{(\eta - o(1))dn}$. A simple counting argument shows that for a uniformly random graph $H \in \mathcal{G}_j$ we have $\mathbf{Pr}[\bar{\mathcal{E}}_j] = o(1)$ (here we use the fact that coin flips that determine which edges belong to H_{i-1}^w are independent of M_{j-1}). Combining this with (2), we get

$$\mathbf{Pr}[(M_{j-1}, M_j) \notin G] + \mathbf{Pr}[\bar{\mathcal{E}}_j] \leq 1 - C/(2k) + o(1) < 1.$$

Thus, there exists $m^*_{j-1}, m^*_j \in \{0,1\}^s$ such that the following properties hold

(P1) $\mathbf{Pr}[\mathcal{D}_j | M_{j-1} = m^*_{j-1}, M_j = m^*_j] \geq C/k$;

(P2) $|\phi^{-1}_{m^*_{j-1}}(m^*_j)| \geq 2^{(\eta - o(1))dn}$.

We can now complete the proof. For brevity let $\mathcal{M} = \{M_{j-1} = m^*_{j-1}, M_j = m^*_j\}$. Recall that E^*_j is the set of edges from $H_j^{u_{j+1}}$ that the algorithm outputs at the end of the stream. We have

$$\mathbf{E}_{E^*_j}\left[\mathbf{Pr}[\mathcal{D}_j | \mathcal{M}]\right] \geq C/k,$$

and so there exists $\hat{E}^*_j$ such that $\mathbf{Pr}[\mathcal{D}_j | \mathcal{M} \wedge E^*_j = \hat{E}^*_j] \geq C/k$.

Now recall that $\mathcal{D}_j = \{|E^*_j| = \Omega(1/k)n\} \wedge \mathcal{C}$. Thus, we have isolated memory configurations m^*_{j-1} and m^*_j and a set of edges $\hat{E}^*_j$ of size $\Omega(1/k)n$ such that the algorithm can output $\hat{E}^*_j$ and be correct with probability at least C/k conditional on $M_{j-1} = m^*_{j-1}$ and $M_j = m^*_j$!

Finally, note that conditional on $\mathcal{M} \wedge \{E^*_j = \hat{E}^*_j\}$ all graphs $H \in \phi_i^{-1}(m^*)$ are equiprobable. Now using property **P2** above together with Claim 15 we conclude that $|\hat{E}^*_j| = o(n)$, which is a contradiction.

■

We can now give

Proof of Theorem 11: The proof of Theorem 11 now follows by combining Claim 12, Lemma 13 and Lemma 14 after setting $\delta = c\delta'/k^2$ for a small constant $c > 0$. ■

3 Construction of a (d, k, δ)-packing

In this section we give a construction of a (d, k, δ)-packing on $\Theta(n)$ nodes with $d = n^{\Omega\left(\frac{1}{\log\log n}\right)}$ for any constant k and sufficiently small constant $\delta > 0$. Our construction will use many of the techniques introduced in [7] and (the full version of) [8].

We first introduce notation. As before, the sides of the bipartition of the graph $G(p)$ that we need to construct are denoted by T and $S = S_0 \cup \ldots \cup S_k$. We use the notation $[a] = \{1, \ldots, a\}$ for integer $a \geq 1$. In our construction the $T = T^0$ side of the graph is identified with a hypercube $[m^4]^m$ for a value of m to be chosen later, and the sets $S_i, i = 0, \ldots, k-1$ are identified with a subsampled version of the hypercube $[m^4]^m$. The vertices of the last set S_k do not have any special structure. Vertices $x \in T$ or $y \in S_i$ will often be treated as points $x, y \in [m^4]^m$. Each node u of $\mathcal{T}$ (except the root) will be labeled with a binary vector $\mathbf{u} \in \{0,1\}^m$. We will write $|\mathbf{u}|$ to denote the Hamming weight of $\mathbf{u}$. For $x \in T$ and $u \in \mathcal{T}$ we use the dot product notation $(x, \mathbf{u}) = \sum_{i=1}^m x_i \cdot \mathbf{u}_i \in \mathbb{Z}$. For an interval $[a, b]$, where a, b are integers, and an integer number W we will write $[a, b] \cdot W$ to denote the interval $[a \cdot W, b \cdot W]$. Finally, for an integer i and an integer W we will write $i \mod W$ to denote the residue of i modulo W that belongs to $[0, W-1]$.

For convenience of the reader, we first give an informal outline of the construction. Given a path $p = (u_0, u_1, \ldots, u_k)$ from the root of $\mathcal{T}$ to a uniformly random leaf, we construct the packing as follows. First, we associate with each node of $\mathcal{T}$ other than the root a subset of $\{0,1\}^m$ (i.e. a binary vector) from a family of subsets of fixed cardinality and with small intersections. Since the subsets corresponding to nodes of $\mathcal{T}$ have small intersections, one can think of them as nearly orthogonal vectors.

We then traverse the path p from the root to the leaf and at step i, $i = 0, \ldots, k-1$ we essentially set[1]

$$T^{u_{i+1}} := \{x \in T^{u_i} : (x, \mathbf{u}_{i+1}) \mod W \in [1/k, 1] \cdot W\},$$

where W is an appropriately chosen parameter. Thus, traversing a root to leaf path amounts to repeatedly cutting the hypercube with hyperplanes whose normal vectors are almost orthogonal. At step i the set S_i is identified with an appropriately subsampled copy of T^{u_i}, and a Ruzsa-Szemerédi graph is constructed on (T^{u_i}, S_i). At step i, besides defining the new set $T^{u_{i+1}}$, the vector $\mathbf{u}_{i+1}$ (corresponding to the next vertex on the path) is used to define a subset $S_j^{u_{i+1}} \subseteq S_j^{u_i}$ for all $j \leq i$ by similarly cutting $S_j^{u_i}$ with a hyperplane. The most important property of our construction will be the fact that when we reach the leaf u_k, most of the edges going out of $S_j^{u_k}$ for $j = 0, \ldots, k-1$ will be contained in T^{u_k}, yielding property (2) of (d, k, δ)-packings. We note that the idea of using nearly orthogonal vectors to construct Ruzsa-Szemerédi graphs was introduced in [7] and further generalized in [8], so this part of our construction adapts known techniques to our setting. Our main contribution here is the approach of constructing a recursive sequence of graphs by cutting the hypercube by nearly orthogonal hyperplanes, which allows us to derive property (2).

We now give the details of the construction. We will use the following lemma from [8], which is a convenient formulation of the construction of error correcting codes with fixed weight in [12]

Lemma 16 *[8] For sufficiently large $m > 0$, any constant $\epsilon \in (0,1)$ and constant $\gamma \in (0,2)$ there exists a family $\mathcal{F}$ of subsets of $[m]$ of size ϵm with intersection at most $\gamma\epsilon^2 m$ such that $\frac{1}{m}\log|\mathcal{F}| \geq c_{\epsilon,\gamma} - o(1)$.*

Our main lemma is

Lemma 17 *For any constants $k, \delta' > 0$ there exists a (d, k, δ')-packing on $\Theta(n)$ nodes with $d = n^{\Omega\left(\frac{1}{\log\log n}\right)}$.*

Proof: We associate with each node of the d-ary tree $\mathcal{T}$ of height k a vector $\mathbf{v}$ from a family of almost orthogonal binary vectors of equal weight whose existence is guaranteed by Lemma 16. Since the number of nodes in such a tree is at most d^{k+1}, we can afford to set $d = 2^{\Omega(m)}$ since k is constant. Besides associating

[1]This statement is slightly imprecise in the interest of clarity.

with each node $u \in \mathcal{T}$ a vector $\mathbf{u}$, we also associate with u a random variable U_u that is uniformly distributed over the integers between 0 and $W-1$, where W is a parameter that will be chosen later. The variables U_u and $U_{u'}$ are independent for $u \neq u'$.

Let $X' = Y = [m^4]^m$ for some integer $m > 0$. Let X be a uniformly random subset of X' where each point of X' appears independently with probability $1/k$. We will refer to vertices in X and Y as points in $[m^4]^m$. We now specify how a graph satisfying the properties in definition 9 is constructed for a given path $p = (u_0, u_1, \ldots, u_k)$ denote a path from the root of $\mathcal{T}$ to a leaf of $\mathcal{T}$.

The path p induces a decomposition of the vertex set T as follows. For all $i = 0, \ldots, k-1$

$$\begin{aligned} T^{u_i} &= \{y \in Y : (y, \mathbf{u}_j) \mod W \in [1/k, 1) \cdot W, \text{ for all } j \in [1:i]\} \\ S_i &= \{x \in X' : (x, \mathbf{u}_j) \mod W \in [1/k, 1) \cdot W, \text{ for all } j = [1:i]\}. \end{aligned} \tag{3}$$

Also, let

$$S_j^{u_i} = \{x \in S_j : (x, \mathbf{u}_l) \mod W \in [1/k, 1) \cdot W, \text{ for all } l \in [1:i]\}, \text{ for all } j = 0, \ldots, i-1 \tag{4}$$

The set S_k is a disjoint set of vertices connected to T^{u_k} by a perfect matching.

Consider fixed i between 0 and $k-1$. For all children w of u_i let

$$\begin{aligned} R^Y(w) &= \{y \in T^{u_i} : ((y, \mathbf{w}) + U_w) \mod W \in [0, 1/k] \cdot W\} \\ W^Y(w) &= \{y \in T^{u_i} : ((y, \mathbf{w}) + U_w) \mod W \in ([1/k, 1/k + \delta] \cup [1-\delta, 1)) \cdot W\} \\ B^Y(w) &= \{y \in T^{u_i} : ((y, \mathbf{w}) + U_w) \mod W \in [1/k + \delta, 1-\delta] \cdot W\} \end{aligned} \tag{5}$$

Define $R^X(w), W^X(w), B^X(w)$ similarly (note that these sets are defined only for S_i):

$$\begin{aligned} R^X(w) &= \{x \in S_i : ((x, \mathbf{w}) + U_w) \mod W \in [0, 1/k] \cdot W\} \\ W^X(w) &= \{x \in S_i : ((x, \mathbf{w}) + U_w) \mod W \in ([1/k, 1/k + \delta] \cup [1-\delta, 1)) \cdot W\} \\ B^X(w) &= \{x \in S_i : ((x, \mathbf{w}) + U_w) \mod W \in [1/k + \delta, 1-\delta] \cdot W\} \end{aligned} \tag{6}$$

We note here that the random shift U_w is not necessary for most properties that we establish, and will only be useful to establishing property (3). First, we analyze

Size of the sets $T^{u_i}, S_j, S_j^{u_i}, R, B, W$ and property (4). We will need

Claim 18 *Let $\delta > 0$ be a constant such that $1/\delta$ and $\delta W/|\mathbf{w}|$ are integers, and let $U \in [0 : W-1]$ be an integer. Define for $q = 0, \ldots, 1/\delta - 1$*

$$A_q = |\{y \in Y : ((y, \mathbf{u}_j) + U) \mod W \in [\delta q, \delta(q+1)] \cdot W| \, . \tag{7}$$

Then $|A_q| \in (1 \pm o(1))\delta|Y|$.

Proof: Consider the mapping $\psi : y \to y - \frac{\delta W}{|\mathbf{u}_j|} \cdot \mathbf{u}_j$. This is a well defined mapping into Y for all $y \in Y$ except those that have at least one coordinate smaller than $\frac{\delta W}{|\mathbf{u}_j|} = O(1)$. We denote this set by R. But for any fixed l one has $|\{y \in Y : y_l < \frac{\delta W}{|\mathbf{u}_j|}\} = \frac{\delta W}{m^4 |\mathbf{u}_j|} = o(|Y|/m^2)$, and hence by the union bound over all $l = 1, \ldots, m$ one has $|R| = o(|Y|)$. For all $q = 1, \ldots, 1/\delta - 1$ the mapping ϕ maps A_q injectively into A_{q-1}, and A_0 into $A_{1/\delta - 1}$, everywhere except R. Thus, one has $|A_q| = \delta(1 \pm o(1))|Y|$, and the conclusion of the lemma follows. ■

We first prove

Lemma 19 *Consider any set $\mathcal{S}$ defined by $\mathcal{S} = \{y \in Y : (y, \mathbf{u}) \mod W \in [a_{\mathbf{u}}, b_{\mathbf{u}}] \cdot W, \mathbf{u} \in \mathcal{U}\}$, where $\mathcal{U}$ is a collection of binary vectors and $a_{\mathbf{u}}, b_{\mathbf{u}}$ are constants. Let $\mathbf{v}$ be a vector such that $|\mathbf{u}| = |\mathbf{v}|$ for all $\mathbf{u} \in \mathcal{U}$ and $\max_{\mathbf{u} \in \mathcal{U}} (\mathbf{u}, \mathbf{v})/|\mathbf{v}| \leq \delta'$, and $A, B \in [0, 1], A \leq B$ are rational constants. Let*

$$\mathcal{S}' = \{y \in \mathcal{S} : (y, \mathbf{v}) \mod W \in [A, B] \cdot W\}.$$

Then for sufficiently large $W = O(m)$ one has $||\mathcal{S}'| - (B - A)|\mathcal{S}|| = O(|\mathcal{U}|\delta')$.

Proof: Consider the mapping $\psi_{\mathbf{v},j} : y \to y - \frac{j \cdot \delta(B-A)W}{|\mathbf{v}|} \cdot \mathbf{v}$, where δ is a sufficiently small rational constant such that $1 - (B - A)$ is an integer multiple of $\delta(B - A)$. Note that the mapping is well-defined as long as W is an integer multiple of $1/(\delta(B - A))$, which is admissible under our assumption that $W = O(m)$.

Let $y \in \mathcal{S}$. Then

$$(\psi_{\mathbf{v},j}(y), \mathbf{u}) = (y, \mathbf{u}) + \frac{j \cdot \delta(B - A)W}{|\mathbf{v}|} \cdot (\mathbf{u}, \mathbf{v}) \leq (y, \mathbf{u}) + j \cdot \delta(B - A)W\delta',$$

so $\psi_{\mathbf{v},j}$ for $|j| \leq 1/(\delta(B - A))$ maps points $y \in \mathcal{S}$ into $\mathcal{S}$ unless either

$$(y, \mathbf{u}) \mod W \in [a_{\mathbf{u}}, a_{\mathbf{u}} + \delta'] \cup [b_{\mathbf{u}} - \delta', b_{\mathbf{u}}] \cdot W \tag{8}$$

for at least one $\mathbf{u} \in \mathcal{U}$ or y has at least one coordinate smaller than W. We call such points *bad* and denote this set by R. For a fixed $\mathbf{u}$ the fraction of $y \in Y$ that do not satisfy (8) is $O(\delta')$ by Claim 18 and hence by the union bound over all $\mathbf{u} \in \mathcal{U}$ we get that the fraction of such points in Y is $O(|\mathcal{U}|\delta')$. The fraction of points with at least one coordinate smaller than W is at most W/m^4, and hence by the union bound the fraction of points with at least one coordinate smaller than W is $o(1)$, so $|R| = O(\mathcal{U}\delta') \cdot |Y|$.

Similarly to Claim 18, define

$$A_q = |\{y \in \mathcal{S} : (y, \mathbf{v}) \mod W \in [(B - A)\delta q, (B - A)\delta(q + 1)] \cdot W| \,. \tag{9}$$

Now let $D = [0 : \frac{1}{(B-A)\delta})$ denote the set of indices such that $\mathcal{S} = \bigcup_{d \in D} A_d$, and let $D' = [\frac{A}{(B-A)\delta} : \frac{B}{(B-A)\delta}]$ denote the set of indices such that $\mathcal{S}' = \bigcup_{d \in D'} A_d$.

Define a bipartite graph $F = (\mathcal{S}', \mathcal{S} \setminus \mathcal{S}', E_F)$ by including an edge $(x, y), x \in \mathcal{S}', y \in \mathcal{S} \setminus \mathcal{S}'$ to E_F whenever $\psi_{\mathbf{v},j}(x) = y$ for some $j \in D$. Thus, each $x \in \mathcal{S}' \setminus R$ has degree $|D \setminus D'|$ in F, and $x \in (\mathcal{S} \setminus \mathcal{S}') \setminus R$ have degree $|D'|$ in F. Furthermore, the degree of each $x \in \mathcal{S}'$ is bounded by $|D \setminus D'|$ and the degree of each $x \in \mathcal{S} \setminus \mathcal{S}'$ is bounded by $|D'|$.

Putting these estimates together, we have $|\mathcal{S}' \setminus R| \cdot |D \setminus D'| \leq |\mathcal{S} \setminus \mathcal{S}'| \cdot |D'|$, i.e.

$$|\mathcal{S}'| \leq (|\mathcal{S}| - |\mathcal{S}'|) \cdot \frac{|D'|}{|D \setminus D'|} + |R| = (|\mathcal{S}| - |\mathcal{S}'|) \cdot \frac{B - A}{1 - (B - A)} + |R|.$$

Thus, $|\mathcal{S}'| \leq (B-A) \cdot |\mathcal{S}| + (1-(B-A))|R|$. On the other hand, we also have $|(\mathcal{S} \setminus \mathcal{S}') \setminus R| \cdot |D'| \leq |\mathcal{S}'| \cdot |D \setminus D'|$, i.e.

$$|\mathcal{S} \setminus \mathcal{S}'| \leq |\mathcal{S}'| \cdot \frac{|D \setminus D'|}{|D'|} + |R| = |\mathcal{S}'| \cdot \frac{1 - (B - A)}{B - A} + |R|$$

Thus, $(B - A)(|\mathcal{S}| - |\mathcal{S}'|) \leq |\mathcal{S}'| \cdot (1 - (B - A)) + (B - A)|R|$, so $|\mathcal{S}'| \geq (B - A)|\mathcal{S}| - (B - A)|R|$. The conclusion of the lemma follows. ■

Estimates on the size of sets T^{u_i} now follow by noting that one has $|\mathcal{U}| \leq k$ in all cases, and that the maximum dot product δ' can be chosen to be $1/\text{poly}(k)$. The bounds on the size of $S_i^{u_j}, R, B, W$ follow in a similar way with the additional application of Chernoff bounds to the sampling of points that are included in X'.

We now define the edges of the $((k - 1)\gamma, k\gamma, O(\delta))$-almost regular induced subgraph H_i^w, for a constant $\gamma > 0$ (the induced property will be shown later). The subgraph H_i^w will consist of disjoint copies of small complete bipartite graphs.

Constructing H_i^w. Fix a child w of u_i. For the purposes of constructing H_i^w we *condition on the values of all shifts U_w*. In what follows we omit the parameter w when referring to sets $R^Y(w), W^Y(w), B^Y(w)$. For two vertices $b, b' \in R^Y$ such that $|(b-b', \mathbf{w})| \le W/k$ we say that $b \sim b'$ if $b - b' = \lambda \cdot \mathbf{w}$ for some λ. Note that we have $\lambda \in \left[-\frac{W}{k|\mathbf{w}|}, \frac{W}{k|\mathbf{w}|}\right]$. We write $\mathcal{B}_b \subseteq Y$ to denote the equivalence class of b. It follows directly from the definition of $\mathcal{B}_b$ and (5) that $|\mathcal{B}_b| = W/(k|\mathbf{w}|)$ for all b. Also, let

$$\mathcal{A}_b = B^X \cap \left(\bigcup_{\lambda \in [0,(1-1/k)W/|\mathbf{w}|]} (\mathcal{B}_b + \lambda \cdot \mathbf{w}) \right).$$

Note that $\mathcal{A}_b$ is a random set (determined by the random choice of $X \subset X'$). Since each element of X' is included in X independently with probability $1/k$, we have that $\mathbf{E}[|\mathcal{A}_b|] = (1 \pm O(\delta))(1-1/k)|\mathcal{B}_b|$.

We now define a set of edges of a $((k-1)\gamma, k\gamma, \delta)$-almost regular subgraph between (a subset of) $\mathcal{B}_b$ and $\mathcal{A}_b$. First note that $\mathbf{E}[|\mathcal{B}_b|] = (1 \pm O(\delta))(1-1/k)|\mathcal{A}_b|$. Furthermore, since X is obtained from X' by independent sampling at rate $1/k$, standard concentration inequalities yield

$$\mathbf{Pr}\left[|\mathcal{A}_v| \notin (1 \pm \delta)(1-1/k)|\mathcal{B}_v|\right] \le e^{-\delta^2(1/2)|\mathcal{B}_v|/4} \le \delta^2 \tag{10}$$

for $|\mathcal{A}_b| > \gamma = 16\ln(8/\delta)/\delta^2$. To ensure this, it is sufficient to ensure that $W \ge \frac{16k\ln(8/\delta)}{\delta^2} \cdot |\mathbf{w}|$. We note here that we are thinking of δ as being smaller than $1/k$. In particular, we will set $\delta = O(1/\text{poly}(k))$ at the end of the construction. Now for each $c \in \mathcal{A}_b, d \in \mathcal{B}_b$ include an edge (c,d) in H_i^w. We will define a complete bipartite graph on each such equivalence class $\mathcal{A}_b, \mathcal{B}_b$, i.e. for each $c \in \mathcal{A}_b, d \in \mathcal{B}_b$ include an edge (c,d) in H_i^w. However, since we used randomness to chose the set X', some of these classes may be too small due to stochastic fluctuations. We deal with this problem next.

We now classify points $b \in R^Y$ as good or bad depending on the how close $|B_b|$ is to its expectation. In particular, mark a b *bad* if $|\mathcal{B}_b| \notin (1 \pm \delta)(1-1/k)|\mathcal{A}_b|$ and *good* otherwise. Note that in fact this is a well-defined property of an equivalence class. Let $J_{\mathcal{B}}$ denote the indicator random variable that equals 1 if $\mathcal{B}$ is bad and 0 otherwise, where $\mathcal{B}$ is an equivalence class. Note that $J_{\mathcal{B}}$ is independent of $J_{\mathcal{B}'}$ for $\mathcal{B} \neq \mathcal{B}'$, since J is determined by the random choice of $X \subset X'$ and we are conditioning on the values of all shifts $U_w, w \in \mathcal{T}$. By (10) one has $\mathbf{E}[J_{\mathcal{B}}] \le \delta^2$ for all equivalence classes $\mathcal{B}$. Note that each equivalence class contains a constant number of points, and hence there are $\Omega(m^{4m})$ equivalence classes for every i and w child of u_i.

An application of Chernoff bounds shows that for fixed i and fixed w a child of u_i

$$\mathbf{Pr}\left[\sum_{\mathcal{B}} J_{\mathcal{B}} > 2\mathbf{E}\left[\sum_{\mathcal{B}} J_{\mathcal{B}}\right]\right] \le e^{-\Omega(m^{4m})}. \tag{11}$$

Note that by (10) one has that (11) bounds the probability of there being more than $2\delta^2$ fraction of bad classes for fixed $w \in \mathcal{T}$. Taking a union bound over $2^{O(m)}$ nodes of $\mathcal{T}$, we conclude that there will be no more than $2\delta^2$ fraction of bad equivalence classes in H_i^w *for any i, and w a child of u_i*.

If b is good, let $\mathcal{A}'_b$ denote an arbitrary subset of $\mathcal{A}_b$ of cardinality $(1-\delta)(1-1/k)|\mathcal{B}_b|$. Similarly, let $\mathcal{B}'_b$ denote an arbitrary subset of $\mathcal{B}_b$ of cardinality $(1-\delta)|\mathcal{B}_b|$, so that $|\mathcal{A}'_b| = (1-1/k)|\mathcal{B}'_b|$. Now for each $c \in \mathcal{A}'_b, d \in \mathcal{B}'_b$ include an edge (c,d) in H_i^w. Note that each such graph is a $((k-1)\gamma, k\gamma, \delta)$-almost regular graph, as required by property (1). Note that all matched edges are of the form (c,d), where

$$c = d - \lambda \cdot \mathbf{w}, \lambda \in (0, W/|\mathbf{w}|]. \tag{12}$$

The union of the small complete graphs that we constructed yields the graph H_i^w for a fixed child w of u_i. We also showed that on such graph H_i^w contains more than a $2\delta^2$ fraction of bad classes whp, which completes the construction of the graphs H_i^w.

Induced property (property (1)). Graphs H_i^w constructed in this way are induced for the same reason as in [7, 8] when the vectors $\mathbf{w}, \mathbf{w}'$ corresponding to two distinct nodes of $\mathcal{T}$ are chosen in such a way that $|\mathbf{w}| = |\mathbf{w}'| = \epsilon m$ (recall that $X' = Y = [m^4]^m$) and

$$(\mathbf{w}, \mathbf{w}') \leq (5/2)\epsilon|\mathbf{w}| \tag{13}$$

for sufficiently small constant ϵ. Indeed, consider a fixed i and suppose that an edge $(a, b) \in E(H_i^w)$ is induced by $H_i^{w'}$ for $w' \neq w$. But then it must be that either $c \in R^Y(w'), d \in B^X(w')$ or $d \in R^Y(w'), c \in B^X(w')$. In either case one has

$$|(c - d, \mathbf{w}')| \geq \delta \cdot W. \tag{14}$$

However, by (13) together with (12) one has

$$|(c - d, \mathbf{w}')| \leq \frac{W}{|\mathbf{w}|}(\mathbf{w}, \mathbf{w}') \leq \frac{W}{|\mathbf{w}|}(5/2)\epsilon|\mathbf{w}| = (5/2)\epsilon W,$$

which is a contradiction with (14) for $\epsilon < \delta/10$.

Existence of a large matching (property (3)) We now show that for any i and w a child of u_i there exists a matching of $1 - O(\delta)$ fraction of S_i to $T^{u_i} \setminus T^w$. We will do this by exhibiting a fractional matching of appropriate size.

Consider a point $x \in T^{u_i}$. We need to analyze the degree of x in the graph $T^{u_i} \cup S_i$. Note that the degree of x depends on (1) the number of vectors w for which $x \in R^Y(w)$ and (2) on the size of the equivalence classes that x belongs to for different w. We first analyze (1).

For a fixed w it follows by Claim 18 and the definition of U_w that $\mathbf{Pr}_{U_w}[x \in R^Y(w)] \in (1 \pm o(1))\frac{1}{k}$. Next note that each vertex $x \in R^Y(w)$ has degree $(k-1)\gamma$ in H_i^w. Furthermore, since the random shifts U_w are independent for different w, we obtain using Chernoff bounds that for a fixed $x \in T^{u_i}$

$$\mathbf{Pr}\left[\sum_{w \text{ child of } u_i} \mathbf{1}_{x \in R^Y(w)} \notin (1 \pm O(\delta))d/k\right] \leq e^{-\Omega(\delta^2 d/k)}. \tag{15}$$

A similar argument shows that the expected degree of each vertex in $S_i \setminus S_i^w$ has similar concentration around $k\gamma d$. Since there are only $O(m^{4m})$ vertices and $2^{O(m)}$ nodes in the tree $\mathcal{T}$, and $d = 2^{\Omega(m)}$, a union bound shows that vertex degrees are concentrated in each T^{u_i}, S_i pair with high probability. Now it remains to handle the loss of edges due to $x \in T^{u_i}$ belonging to small equivalence classes for some w. However, it follows from the analysis in (11) that at most an $O(\delta^2)$ fraction of the edge mass can be lost because of this, yielding the following fractional matching. Put weight $1/(k\gamma)$ on each edge in H_i^w, and put weight $\frac{1}{(1+O(\delta))k(1-1/k)\gamma d}$ on each edge going from $T^{u_i} \setminus T^w$ to $S_i \setminus S_i^w$. Since degrees in T^{u_i} are bounded by $(1 + O(\delta))(1 - 1/k)\gamma d$, and degrees is S_i are bounded by $(1+O(\delta))k\gamma d$, this is feasible and yields a matching of size $(1-O(\delta+\delta^2))|S_i|$, proving property (3).

We now prove property (2). For $i = 0, \ldots, k-1$ let

$$Z^{u_i} = \{y \in Y : (y, \mathbf{u}_j) \mod W \in ([1/k - \delta, 1/k] \cup [0, \delta]) \cdot W \text{ for some } j \in [1:k]\}. \tag{16}$$

We need to show that the subgraph H^* induced by $(T^{u_i} \setminus (T^{u_k} \cup Z^{u_i})) \cup S_i^{u_k}$ only contains the edges of $H_i^{u_{i+1}}$. First note that if an edge $(c, d), c \in P, d \in Q$ belongs to H^*, then $c \in S_i^{u_k}$ and $d \in T^{u_i}$, so (c, d) necessarily belongs to some graph H_i^w, where w is a child of u_i. Then we have by (12) that

$$d - c = q \cdot \mathbf{w}, \text{ where } |q| \leq W/|\mathbf{w}|.$$

On the other hand, we have for all $j = 1, \ldots, k$ using the orthogonality condition (13)

$$|(c - d, \mathbf{u}_j)| \leq \frac{W}{|\mathbf{w}|} |(\mathbf{w}, \mathbf{u}_j)| \leq (5/2)\epsilon W. \tag{17}$$

Now recall that $a \in S_i^{u_k}$, so by (3) and (4)

$$(c, \mathbf{u}_j) \mod W \in [1/k, 1] \cdot W, \forall j = 1, \ldots, k.$$

Thus, by (17) one has

$$(d, \mathbf{u}_j) \mod W \in ([1/k - \delta, 1] \cup [0, \delta]) \cdot W, \forall j \leq k,$$

i.e. $d \in Z^{u_i} \cup T^{u_k}$, if we set ϵ to smaller than $\delta/10$.

It remains to bound the size of Z^{u_i}. First note that it follows from Claim 18 that for sufficiently small constant δ (e.g. $\delta < 1/k^2$) one has

$$|\{y \in Y : (y, \mathbf{u}_j) \mod W \in ([1/k - \delta, 1/k] \cup [0, \delta]) \cdot W| \leq 2\delta |Y|. \tag{18}$$

Now by a union bound over all $j \in [1 : k]$ we conclude that $|Z^{u_i}| \leq 2\delta k |Y| = O(\delta k n)$.

It remains to set parameters. First, inspection of the bounds obtained so far reveals that setting $\delta = c\delta'/k^4$ for a sufficiently small constant $c > 0$ is sufficient to obtain a (d, k, δ')-packing, where we set $\epsilon = \delta/10$. Finally, the size of the graphs obtained is essentially the same as in [7] and [8]. In particular, the number of vertices is $n = \Theta(m^{4m})$ and $d = 2^{\Omega(m)}$. Thus, we get a graph on n vertices with $d = n^{\Omega\left(\frac{1}{\log \log n}\right)}$ edges. ■

Proof of Theorem 1: The proof follows by combining Theorem 11 and Lemma 17. ■

4 Multipass approximation for matchings

In this section we present the basic version of our algorithm for approximating matchings in multiple passes in the vertex arrival setting. Let $G = (P, Q, E)$ denote a bipartite graph. We assume that vertices in P arrive in the stream together with all their edges. At each step the algorithm maintains a fractional matching $\{f_e\}_{e \in E}$, where the capacity of each vertex in Q is infinite and the capacity of each vertex $u \in P$ is equal to the number of times it has appeared in so far (i.e. always between 1 and k). The capacity of an edge $e = (u, v), u \in P, v \in Q$ is equal to the capacity of u. For a vertex $u \in P$ we write $\delta(u)$ to denote the set of neighbors of u in G.

4.1 Algorithm

We now give the algorithm and show how to implement each pass in linear time.

Algorithm 1: PROCESS-VERTEX(G, u, $\delta(u)$)

1: Augment capacity of u and all edges in $\delta(u)$ by 1.
2: WATER-FILLING($G', u, \delta(u)$)
3: REMOVE-CYCLES(G', f).

The function WATER-FILLING($G', u, \delta(u)$) increases the load of the least loaded neighbors of u simultaneously (with other neighbors joining if the load reaches their level) until one unit of water is dispensed out of u. Here the support of the fractional matching $\{f_e\}_{e \in E}$ maintained by the algorithm is denoted by G'. The function REMOVE-CYCLES(G', f) reroutes flow among cycles that could have emerged in the process, ensuring that the flow is supported on at most $|P| + |Q| - 1$ edges. We note that as stated, Algorithm 1 does not necessarily take $O(m)$ time per pass due to the runtime of cycle removal. However, simply buffering incoming vertices until the number of edges received is $\Theta(n)$ and only then removing cycles yields a linear time implementation. Here we can use DFS to reroute flow along cycles in time linear in the number of nodes.

Remark 20 *We note that a single pass of this algorithm is different from the one-pass algorithm that achieves* $1 - 1/e$ *approximation from [8]. However, we will later show that our algorithm in fact also achieves the ratio of* $1 - 1/e$ *in a single pass.*

We now turn to analyzing the approximation ratio. We first give a sketch of the proof under additional assumptions on the graph G, and then proceed to give the relevant definitions and the complete argument.

4.2 Analysis for a simple case

We now assume that $G = (P, Q, E)$ has a perfect matching M. For each $k \geq 1$ and all $x \geq 0$ denote by $b^k(x)$ the number of vertices in Q that have load *at least* x after k passes. We start by pointing out some useful properties of the function $b^k(x)$. First, note that $b^k(0) = |M|$, $b^k(x)$ is non-increasing in x and $b^k(x) - b^{k-1}(x) \geq 0$ for all x. Furthermore, we have

$$\int_0^\infty b^k(x)dx = k|M|, \tag{19}$$

since every vertex $u \in P$ contributed 1 unit of water, amounting to $|M|$ amount of water overall, and (19) calculates the sum of loads on all $v \in Q$. Furthermore, note that the size of the matching constructed by the algorithm after k passes is exactly equal to

$$\frac{1}{k}\int_0^k b^k(x)dx, \tag{20}$$

since every vertex $v \in Q$ with load x contributed $\frac{1}{k} \cdot \min\{k, x\}$ to the matching. Hence the approximation ratio after k passes is at least

$$1 - \frac{1}{k}\int_k^\infty b^k(x)dx, \tag{21}$$

where we used (19) to convert (20) into (21). Thus, it is sufficient to lower bound $\int_0^k b^k(x)dx$ in order to analyze the approximation ratio, and we turn to bounding this quantity.

First consider the case $k = 1$. Fix $x \geq 0$ and consider vertices $v \in Q$ that have load at least x – there are at least $\int_x^\infty b^1(s)ds$ of them. For each such vertex u consider its match $M(u)$. Since u ended up at level at least x after the first pass, its match $M(u)$ must have been at level at least x when u arrived, and levels are monotone increasing. Hence, we have

$$b^1(x) \geq \int_x^\infty b^1(s)ds \tag{22}$$

for all $x \geq 0$. This, however, together with (19) can be shown to imply that $\int_x^\infty b^1(s)ds \leq |M| \cdot e^{-x}$ for all x. We immediately get using (21) that the approximation ratio after one pass is at least $1 - 1/e$.

Now suppose that $k > 1$ and consider vertices $v \in Q$ that are at level at least x after k-th pass, but were at a lower level after $(k-1)$-st pass. There are exactly $b^k(x) - b^{k-1}(x)$ such vertices. Since these vertices u were at level at least x after k-th pass, their matches $M(u)$ must have also been at level at least x when they arrived, implying that

$$b^k(x) \geq \int_x^\infty (b^k(s) - b^{k-1}(s))ds \tag{23}$$

for all $x \geq 0$. Solving (23), we get that for all $k \geq 1$

$$\int_x^\infty b^k(s)ds \leq |M| \cdot \int_x^\infty F^k(s)ds, \tag{24}$$

where $1-F^k(x)$ is the cdf of the Gamma distribution with scale 1 and shape k, i.e. $F^k(x) = \int_x^\infty e^{-s}s^{k-1}/(k-1)!ds$. Using this in (21) yields the desired bound on the approximation ratio, i.e. $1 - e^{-k}k^{k-1}/k!$.

4.3 General case

The proof sketch we gave in the previous subsection works under the assumption that G has a perfect matching. The general case turns out to be substantially more involved. Interestingly, while the analysis above proceeds by showing that not too much mass will be in the tail $\int_k^\infty b^k(x)dx$, here we find it more convenient to show that substantial mass will be in the head of the distribution, i.e. bound $\int_0^k b^k(x)dx$ from below. We extend the argument using a careful reweighting of vertices and scaling of levels guided by the structure of the *canonical decomposition* of G introduced in [8], which we now define.

Let $G = (P, Q, E)$ denote a bipartite graph. For a set $S \subseteq P$ we denote the set of neighbors of S by $\Gamma(S)$. For a number $\alpha > 0$ the graph G is said to have vertex expansion at least α if $|\Gamma(S)| \geq \alpha|S|$ for all $S \subseteq P$.

Definition 21 (Canonical decomposition) *Let $G = (P, Q, E)$ denote a bipartite graph. A partition of $Q = \bigcup_{j\in\mathcal{I}} T_j, T_j \cap T_i = \emptyset, j \neq i$ and $P = \bigcup_{j\in\mathcal{I}} S_j, S_j \cap S_i = \emptyset, j \neq i$ together with numbers $\alpha_j > 0$, where $\alpha_j \leq 1$ for $j \leq 0$ and $\alpha_j > 1$ for $j > 0$ is called a* canonical partition *if*

1. *for all i one has $\Gamma\left(\bigcup_{j\in\mathcal{I}, j\leq i} S_j\right) \subseteq \bigcup_{j\in\mathcal{I}, j\leq i} T_j$;*
2. *$|\Gamma(S) \cap T_j| \geq \alpha_j|S|$ for all $S \subseteq S_j$ for all $j \in \mathcal{I}$;*
3. *$|T_j|/|S_j| = \alpha_j, j \in \mathcal{I}$.*

Here $\mathcal{I} \subset \mathbb{Z}$ is a set of indices.

Please see Fig. 3 for an illustration.

Remark 22 *For $k = 1$, the analysis is inspired by the analysis of the round-robin algorithm in [15]. We note that the difference in our case is that we essentially consider a fractional version of their process, and obtain significantly better bounds on the quality of approximation. In particular, the best approximation factor that follows from the result of [15] is* $1/8$ even after any k passes*, while here we get the optimal $1 - 1/e$ factor for $k = 1$, and an approximation of the form $1 - O(1/k^{1/2})$ for all $k > 0$.*

We now introduce some definitions. For a node $v \in Q$ let $l^k(v)$ denote the load of v after the k-th pass. Note that $l^k(v) \geq 0$ and may in general grow with $|P|$ for the most loaded vertices in Q. The core of our analysis will consist of bounding the distribution of water levels among vertices in Q, showing that there cannot be too many highly overloaded vertices. It will be convenient to assume that water is allocated in multiples of some $\Delta_{org} > 0$ (such a Δ_{org} always exists since we are dealing with a finite process).

Shadow allocation and density function $\phi_v^k(x)$. First, define

(Source capacities) Define $w_s(u), u \in P$ by setting $w_s(u) = \min\{1, \alpha_j\}$ for $u \in S_j$. Note that one has $\sum_{u\in P} w_s(u) = |M|$. Also, for $v \in T_j$ let $w_s(v) := \min\{1, \alpha_j\}$ for convenience.

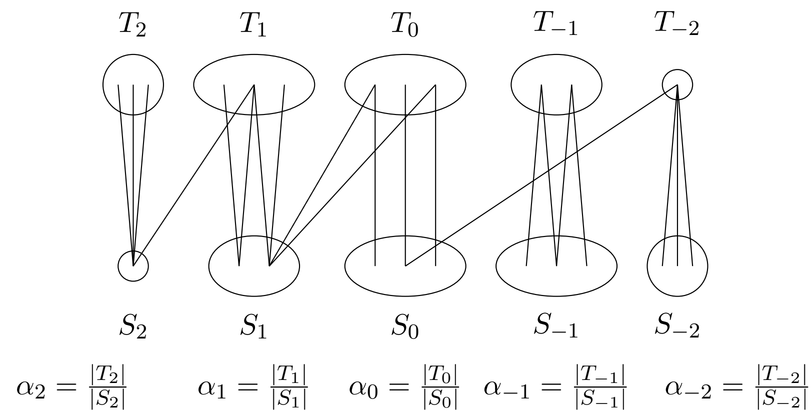

Figure 3: Canonical decomposition of a bipartite graph. Note that edges from S_i only go to T_j with $j \leq i$ (property (1)).

(Sink capacities) Define $w_t(v), v \in Q$ by setting $w_t(v) = \min\{1, 1/\alpha_j\}$ for $v \in T_j$. Note that one has $\sum_{v\in Q} w_t(v) = |M|$. Also, for $u \in S_j$ let $w_t(u) := \min\{1, 1/\alpha_j\}$ for convenience.

We will use the concept of a *shadow allocation*, in which whenever a units of water are added to a vertex $v \in Q$ in the original allocation, $a/w_t(v)$ units of water are added to v in the shadow allocation. Now whenever water from a vertex $u \in P$ is added to vertex $v \in Q$ at level x during the j-th pass *in the shadow allocation*, we let $\phi_v^j(x) := w_s(u)$, where ϕ is the *density function*. It will be crucial that

$$\sum_{v\in Q} w_t(v) \int_0^\infty \phi_v^j(x)dx = |M| \tag{25}$$

for all $j = 1, \ldots, k$. We assume that water in the shadow allocation is allocated in multiples of some $\Delta > 0$.
Then

Lemma 23 *One has for all $x \geq 0$ and all $k \geq 1$*

$$b^k(x) \geq \int_x^\infty \sum_{v\in Q} w_t(v)\phi_v^k(s)ds.$$

Proof: Recall that the pairs in the canonical decomposition of G are denoted by (S_j, T_j), where the expansion factors α_j are increasing with j. We need to that

$$b^k(x) \geq \int_x^\infty \sum_{v\in Q} w_t(v) \cdot \phi_v^k(s)ds \tag{26}$$

By definition of the canonical decomposition $(S_j, T_j)_{j\in\mathcal{I}}$ for each $j \in \mathcal{I}$ there exists a (possibly fractional) matching M_j in G that matches each $u \in S_j$ exactly α_j times and each $v \in T_j$ exactly once. Let $M_j(u, v) \in [0, 1]$ denote the extent to which u is matched to v, so that $\sum_{v\in T_j} M_j(u, v) = \alpha_j$ for all $u \in S_j$ and $\sum_{u\in S_j} M_j(u, v) = 1$ for all $v \in T_j$.

Consider a node $u \in S_j$ and suppose that a Δ_{org} amount of its water was allocated to level $[i\cdot\Delta, (i+1)\cdot\Delta]$ of a vertex $v \in T_r$ *in the original allocation*. Note that $r \leq j$ since there are no edges from S_j to $T_r, r > j$. By the definition of the shadow allocation Δ_{org} amount of water in the original allocation corresponds to

$\frac{\Delta_{org}}{w_t(u)}$ water placed contiguously in the shadow allocation. Let $t := \frac{\Delta_{org}}{w_t(u)\cdot\Delta}$ and let $\Delta \cdot j, \ldots, \Delta \cdot (j+t-1)$ denote the t contiguous levels that this water occupies in the shadow allocation.

By definition of the water-filling algorithm all neighbors w of u must have been at level at least $\frac{w_t(v)}{w_t(u)}(i+1)\Delta_{org} \geq (i+1)\Delta_{org}$ when the node u was allocated since $w_t(v) = \min\{1, 1/\alpha_r\} \geq w_t(u) = \min\{1, 1/\alpha_j\}$.

Thus, we have that for each such $u \in S_j$

$$\text{contribution to rhs of (26)} = \Delta_{org} \cdot \phi_v^k(s)$$

since the Δ_{org} amount of water corresponds to $t = \frac{\Delta_{org}}{w_t\Delta}$ slabs of size Δ, and the contribution is then weighted by $w_t(v)$ in the rhs of (26). We now calculate the contribution of $u \in S_j$ to the lhs. We let each $u_j \in S_j$ contribute $M_j(u, w)$ to each $w \in T_j$, so that the total contribution to each w is 1 and total contribution of each $u \in S_j$ is α_j. Thus,

$$\text{contribution of } u \text{ to lhs of (26)} \geq \Delta_{org} \cdot \min\{1, 1/\alpha_j\} \cdot \alpha_j$$

since u has α_j matches in T_j, whose contributions are weighted by $\min\{1, 1/\alpha_j\}$. As before, it remains to note that $\min\{1, 1/\alpha_j\} \cdot \alpha_j = \min\{1, \alpha_j\} = \phi_v^k(s)$. ■

We now get

Lemma 24 *One has for all $x \geq 0$ and all $k \geq 1$*

$$|M| - b^k(x) \leq \int_0^x (b^k(s) - b^{k-1}(s))ds.$$

Proof: By Lemma 23 we have

$$b^k(x) \geq \int_x^\infty \sum_{v \in Q} w_t(v)\phi_v^k(s)ds.$$

Putting this together with (25) we get

$$|M| - b^k(x) \leq \int_0^x \sum_{v \in Q} w_t(v)\phi_v^k(s)ds$$

for all $x \geq 0$ and $k \geq 1$. To complete the proof, we note that

$$\int_0^x \sum_{v \in Q} w_t(v)\phi_v^k(s)ds \leq \int_0^x (b^k(s) - b^{k-1}(s))ds$$

for all $k \geq 1$ and $x \geq 0$, where we let $b^0 \equiv 0$ for convenience. ■

We also need

Lemma 25 *Algorithm 1 constructs a matching of size at least*

$$\frac{1}{k}\int_0^k b^k(x)dx.$$

Proof: A vertex $v \in Q$ contributes $\frac{1}{k}\min\{k, l^k(v)\} \geq w_t(v)\frac{1}{k}\min\{k, l^k(v)/w_t(v)\}$ to the matching, implying that the size of the constructed matching is at least

$$\frac{1}{k}\int_0^k b^k(x)dx.$$

■

We now prove lower bounds on $b^k(x)$. Recall that for integer $k \geq 1$

$$F^k(x) = \int_x^\infty e^{-s} s^{k-1}/(k-1)! ds = \sum_{i=0}^{k-1} e^{-x} x^i / i!. \tag{27}$$

Note that $1 - F^k(x)$ is the cdf of the Gamma distribution with scale 1 and shape k.

Lemma 26 *For every $k \geq 1$ one has for all $x \geq 0$*

$$\int_0^x b^k(s) ds \geq |M| \cdot \int_0^x F^k(s) ds.$$

Proof:

We prove the lemma by induction on k.

Base: $k = 1$ This follows immediately since by Lemma 24 one has

$$\int_0^x b^1(s) ds \geq |M| - b^1(x). \tag{28}$$

Letting $f(x) = \int_0^x b^1(s) ds$, we get that $f'(x) \geq |M| - f(x)$ for all $x \geq 0$, $f(0) = 0$ and $f'(0) = |M|$, which implies that $f(x) \geq |M| \cdot (1 - e^{-x})$, as required.

Inductive step: $k - 1 \to k$ We need to prove that

$$\int_0^x b^k(s) ds \geq |M| \cdot \int_0^x F^k(s) ds. \tag{29}$$

By Lemma 24 for all $x \geq 0$

$$b^k(x) \geq |M| - \int_0^x (b^k(s) - b^{k-1}(s)) ds \geq |M| - \int_0^x b^k(s) ds + |M| \cdot \int_0^x F^{k-1}(s) ds, \tag{30}$$

where we used the inductive hypothesis to replace $\int_0^x b^{k-1}(s) ds$ with $|M| \cdot \int_0^x F^{k-1}(s) ds$.

Thus,

$$\int_0^x b^k(s) ds \geq |M| - b^k(x) + |M| \cdot \int_0^x F^{k-1}(s) ds. \tag{31}$$

Let $f(x) = \int_0^x b^k(s) ds$. We have from (31) that

$$f'(x) = |M| - f(x) + |M| \cdot \int_0^x F^{k-1}(s) ds, f(0) = 0, f'(0) = |M|.$$

Thus, $f'(x)$ is given by the solution of

$$g(x) = -g'(x) + |M| \cdot F^{k-1}(x), g(0) = |M|. \tag{32}$$

The solution of (32) is given by

$$g(x) = e^{-x} \left(|M| \int_0^x e^s F^{k-1}(s) ds + |M| \right). \tag{33}$$

Calculating the integral in (33) yields

$$\int_0^x e^s F^{k-1}(s) ds = \int_0^x e^s \int_s^\infty \frac{1}{(k-1)!} z^{k-1} e^{-z} dz ds = \int_0^x \sum_{j=0}^{k-1} \frac{1}{j!} s^j ds = \sum_{j=1}^{k} \frac{1}{j!} x^j, \tag{34}$$

and hence $g(x) = |M| \cdot F^k(x)$. Thus, $\int_0^x b^k(s) ds \geq f(x) = |M| \cdot \int_0^x F^k(s) ds$ as required.

■

Given Lemma 26, we immediately obtain

Theorem 27 *Algorithm 1 achieves a $(1 - e^{-k}\frac{k^{k-1}}{(k-1)!})$-approximation to maximum matchings in k passes over the input stream.*

Proof: The approximation ratio is at least

$$\frac{1}{k}\int_0^k b^k(x)dx \geq \frac{1}{k}\int_0^k F^k(x)dx = 1 - \frac{1}{k}\int_k^\infty F^k(x)dx.$$

Recalling that $F^k(x) = \sum_{j=0}^{k-1} e^{-x}x^j/j!$ and using integration by parts

$$\int e^{-x}x^j/j!dx = -e^{-x}x^j/j!\big|_k^\infty + \int e^{-x}x^{j-1}/(j-1)!dx,$$

we get

$$\begin{aligned}\int_k^\infty F^k(x)dx &= \int_k^\infty \sum_{j=0}^{k-1} e^{-x}x^j/j!dx = \sum_{j=0}^{k-1}(k-j)e^{-k}k^j/j! \\ &= \sum_{j=0}^{k-1} e^{-k}k^{j+1}/j! - \sum_{j=1}^{k-1} e^{-k}k^j/(j-1)! = e^{-k}k^k/(k-1)!\end{aligned} \tag{35}$$

Thus,

$$\frac{1}{k}\int_k^\infty F^k(x)dx = \frac{e^{-k}k^{k-1}}{(k-1)!} = \frac{1}{\sqrt{2\pi k}} + O(k^{-3/2})$$

■

5 Gap-existence

In this section we show how our techniques yield an efficient algorithm for Gap-existence, thereby proving Theorem 4. Recall that we are given a graph $G = (A, I, E)$ and integral budgets B_a. Note that integral budgets can be simulated implicitly by creating B_a copies of a for all $a \in A$. For simplicity, this is the approach that we take.

We now present a discretized version of Algorithm 1. We will explicitly maintain a subset $I^* \subset I$ of size $O(|A|/\epsilon)$, relying on the following two oracles:

1. an oracle LIST-NEIGHBORS(a, I^*) that, given a node $a \in A$ and a set I^* outputs the set of nodes $I^{**} \subseteq I^*$ that a is connected to;
2. an oracle NEW-NEIGHBOR(a, I^*) that, given any set $I^* \subseteq I$, outputs any node $i \in I \setminus I^*$ that a is connected to or $\emptyset$ if all neighbors of a are in I^*.

Algorithm 2: DISCRETIZED-WATERFILLING(G, a, ϵ, k)

1: $I^* \leftarrow \emptyset$
2: **while** exists a neighbor i of a in I^* with level $< (\epsilon/4)k$ **do**
3: Allocate water to i until it is at level $(\epsilon/4)k$
4: **end while**
5: $I^* \leftarrow I^* \cup$ NEW-NEIGHBOR(a, I^*)
6: Perform water filling on neighbors in I^*.
7: REMOVE-CYCLES(G')

First we prove

Lemma 28 *The space used by Algorithm 2 is $O(|A|/\epsilon)$.*

Proof: Call a vertex *saturated* if the amount of water in it is at least ϵk. The number of saturated vertices is $O(|A|/\epsilon)$ since there are $k|A|$ units of water in the system, and each saturated vertex accounts for at least ϵk. We say that an unsaturated vertex i belongs to $a \in A$ if i was added to I^* when NEW-NEIGHBOR was called from a. Note that for each $a \in A$ only one $i \in I$ belongs to a. Thus, this amounts to at most $|A|$ additional vertices. ■

Our algorithm for Gap-Existence is as follows:

Algorithm 3: GAP-EXISTENCE(G, ϵ)

1: Run DISCRETIZED-WATERFILLING(G) with $k = O(\log(|I| \cdot \sum_{a\in A} B_a)/\epsilon^2)$.
2: Let G' denote the support of the fractional solution.
3: Output **YES** if a complete matching with budgets $\lfloor(1-\epsilon)B_a\rfloor$ exists in G', **NO** otherwise.

We now assume that we are in the **YES** case and prove that the algorithm will find a matching with budgets $\lfloor(1-\epsilon)B_a\rfloor$. We refer to vertices $i \in I$ that have a nonzero amount of water as *active*. Let $p_i = 1$ for active vertices and $p_i = 0$ o.w. Abusing notation somewhat, for an active vertex $i \in I$ let $l^k(i)$ denote the level of water in i minus ϵk and 0 otherwise. The Gap-Existence case is in fact somewhat simpler than the general case of approximating matchings that we just discussed, so we will use the more lightweight techniques from the analysis of the simple case for matchings.

For each $k \geq 1$ and all $x \geq 0$ denote by $b^k(x)$ the number of vertices in I that have load *at least* $x+\epsilon k$ after k passes. We start by pointing out some useful properties of the function $b^k(x)$. First, note that $b^k(0) \leq |I|$, $b^k(x)$ is non-increasing in x and $b^k(x) - b^{k-1}(x) \geq 0$ for all x. Recall that we are interested in recovering a $1-\epsilon/2$-matching of the A side. To do that, we scale all allocations by $1-\epsilon/2$. The size of the matching recovered is

$$(1-\epsilon/2)(\epsilon/4)k\sum_{i\in I} p_i + (1-\epsilon/2)\int_0^\infty b^k(x)dx = (1-\epsilon/2)k|M|, \tag{36}$$

since every vertex $a \in A$ contributed k units of water, one in each round, amounting to $k|M|$ amount of water overall, except for the water that was allocated below ϵk, and (36) calculates the sum of loads on all $i \in I$. Furthermore, note that the size of the matching constructed by the algorithm after k passes is at least

$$(1-\epsilon/2)(\epsilon/4)\sum_{i\in I} p_i + \frac{1}{k}\int_0^{k(1-\epsilon/4)/(1-\epsilon/2)} b^k(x)dx, \tag{37}$$

since every vertex $i \in I$ with load x contributes at least $\frac{1}{k}\cdot\min\{k(1-\epsilon/4), x\}$ to the matching before scaling, and hence $\frac{1}{k}\cdot\min\{k(1-\epsilon/4)/(1-\epsilon/2), x\}$ after scaling. Hence the approximation ratio after k passes is at least

$$1 - \frac{1}{k}\int_{k(1-\epsilon/4)/(1-\epsilon/2)}^\infty b^k(x)dx, \tag{38}$$

where we used (36) to convert (37) into (38). Thus, it is sufficient to lower bound $\int_0^k b^k(x)dx$ in order to analyze the approximation ratio, and we turn to bounding this quantity.

Lemma 29 *One has for all $k \geq 1$*

$$b^k(x) \geq \int_x^\infty (b^k(s) - b^{k-1}(s))ds. \tag{39}$$

for all $x \geq 0$, where $b^0 \equiv 0$.

Proof: For each such vertex $a \in A$ consider its match $M(a)$. If a ended up allocating water at level at least x during the k-th pass, its match $M(a)$ must have been at level at least x when a arrived. Together with the fact that levels are monotone increasing this gives the result. We omit the details since they would essentially repeat the proof of Lemma 23 with minor changes due to the absence of weights w_t on the I side. ■

We now get

Lemma 30 *For all $k \geq 1$ and all $x \geq 0$*

$$\int_x^\infty b^k(s)ds \leq |I| \cdot \int_x^\infty F^k(s)ds. \tag{40}$$

Proof: We prove the lemma by induction on k.

Base: $k = 1$ We prove the statement by contradiction. Suppose that

$$\int_{x_0}^\infty b^1(s)ds > |I| \int_{x_0}^\infty e^{-s}ds = |I|e^{-x_0} \tag{41}$$

for some $x_0 \geq 0$. Recall that by Lemma 29 one has

$$b^1(x) \geq \int_x^\infty b^1(s)ds, \tag{42}$$

for all $x \geq 0$. Let $g(x) = \left(\int_{x_0}^\infty b^1(s)ds\right) \cdot e^{-x+x_0}$ for $x \in [0, x_0]$. Then $g(x)$ satisfies (42) with equality, and hence $b^1(x) \geq g(x)$ for all $x \in [0, x_0]$. But $g(0) = \left(\int_{x_0}^\infty b^1(s)ds\right) \cdot e^{x_0} > |I|$, a contradiction with $b^1(0) = |I|$.

Inductive step: $k - 1 \to k$ We need to prove that

$$\int_x^\infty b^k(s)ds \leq |I| \cdot \int_x^\infty F^k(s)ds. \tag{43}$$

Recall that by Lemma 29 for all $x \geq 0$

$$b^k(x) \geq \int_x^\infty (b^k(s) - b^{k-1}(s))ds = \int_x^\infty b^k(s)ds - |I| \cdot \int_x^\infty F^{k-1}(s)ds, \tag{44}$$

where we used the inductive hypothesis to replace $\int_x^\infty b^{k-1}(s)ds$ with $|I| \cdot \int_x^\infty F^{k-1}(s)ds$.

Fix any point $x_0 \geq 0$ and denote

$$\gamma := \int_{x_0}^\infty b^k(s)ds.$$

We will show that one necessarily has $\gamma > |I| \cdot \int_{x_0}^\infty F^k(s)ds$.

It now follows from (44) that $b^k(x)$ is lower bounded by the solution of

$$g(x) = \int_x^\infty (g(s) - |I| \cdot F^{k-1}(s))ds$$
$$g(x_0) = \gamma.$$

Thus, $g(x)$ satisfies

$$g'(x) = -g(x) + |I| \cdot F^{k-1}(x), g(x_0) = \gamma. \tag{45}$$

The solution of (45) is given by

$$g(x) = e^{-x}\left(-|I|\int_0^x e^s F^{k-1}(s)ds + c\right), \tag{46}$$

where the constant c depends on γ. Note that $g(0) = c$, and recalling that g lower bounds $b^k(x)$, which is at most $|I|$ at $x = 0$, we have that $c \le |I|$.

Calculating the integral in (46) yields

$$\int_0^x e^s F^{k-1}(s)ds = \int_0^x e^s \int_s^\infty \frac{1}{(k-1)!} z^{k-1} e^{-z} dz ds = \int_0^x \sum_{j=0}^{k-1} \frac{1}{j!} s^j ds = \sum_{j=1}^{k} \frac{1}{j!} x^j, \tag{47}$$

and hence

$$g(x) = (c + |I| \sum_{j=1}^{k} e^{-x} x^j / j!) = |I| \cdot F^k(x) + (c - |I|).$$

In particular, it follows that $\gamma = g(x_0) = |I| \cdot F^k(x) + (c - |I|) \le |I| \cdot F^k(x)$, completing the proof of the inductive step.

■

We now ready to prove correctness. Suppose that we are in the **YES** case, i.e. there exists a complete matching with budgets B_a. Consider the fractional allocation returned by DISCRETIZED-WATERFILLING(G), and multiply it by $(1 - \epsilon/2)/k$. Recalling that each active vertex can take at least $1 - \epsilon/4$ units of water, we get that every vertex in $i \in I$ now contributes $\frac{1-\epsilon/2}{k(1-\epsilon/4)} \min\{k(1 - \epsilon/4)/(1 - \epsilon/2), l^k(v)\}$ to the matching.

Thus, by Lemma 30 together with (38) shows that the amount of water lost is at most

$$(1 - \epsilon/2)|I| \frac{1}{k} \int_{k(1-\epsilon/4)/(1-\epsilon/2)}^\infty b^k(x)dx. \tag{48}$$

We will need

Lemma 31 *For all $k \ge 1$ and $\epsilon^* \ge 0$*

$$\frac{1}{k}\int_{k(1+\epsilon^*)}^\infty F^k(x)dx \le e^{-\epsilon^* k}(1+\epsilon^*)^k \cdot e^{-k} k^{k-1}/(k-1)!$$

Proof: Recalling that $F^k(x) = \sum_{j=0}^{k-1} e^{-x} x^j / j!$ and using integration by parts

$$\int e^{-x} x^j / j! dx = \left[-e^{-x} x^j / j!\right]_k^\infty + \int e^{-x} x^{j-1} / (j-1)! dx,$$

we get

$$\begin{aligned}\int_{k(1+\epsilon^*)}^\infty F^k(x)dx &= \int_{k(1+\epsilon^*)}^\infty \sum_{j=0}^{k-1} e^{-x} x^j / j! dx = \sum_{j=0}^{k-1} (k-j) e^{-k(1+\epsilon^*)} (k(1+\epsilon^*))^j / j! \\ &\le e^{-\epsilon^* k}(1+\epsilon^*)^k \sum_{j=0}^{k-1} (k-j) e^{-k} k^j / j! = e^{-\epsilon^* k}(1+\epsilon^*)^k \cdot e^{-k} k^{k-1}/(k-1)!\end{aligned} \tag{49}$$

■

Let $\epsilon^* = (1-\epsilon/4)/(1-\epsilon/2) - 1$. By Lemma 31 we have

$$\frac{1}{k}\int_{k(1+\epsilon^*)}^{\infty} b^k(x)dx \le e^{-k(\epsilon^* - \ln(1+\epsilon^*))}\cdot [e^{-k}k^{k-1}/(k-1)!] \le e^{-(\epsilon^*)^2 k/3} \tag{50}$$

for sufficiently small $\epsilon^* > 0$. Also note that $\epsilon/5 \le \epsilon^* \le \epsilon$ for sufficiently small $\epsilon > 0$. Hence, letting $k = \gamma \log(|I| \cdot \sum_{a\in A} B_a)/\epsilon^2$ for a sufficiently large constant $\gamma > 0$ yields a $(1-(\sum_{a\in A} B_a)^{-2})$-approximate fractional matching with budgets $\lfloor (1-\epsilon)B_a \rfloor$. We now argue that the set of edges that this fractional matching is supported on admits a complete matching with budgets $\lfloor (1-\epsilon)B_a \rfloor$. We will need

Lemma 32 *Let $G = (P, Q, E)$ denote a bipartite graph. Suppose that there exists a fractional matching of size $|P|(1-|P|^{-2})$ in G. Then the support of the fractional matching contains a perfect matching of the $|P|$ side.*

Proof: Consider the subgraph G' that supports a fractional $1-|P|^{-2}$ matching. Recall that a graph supports an α-matching of the P-side iff $|\Gamma(S)| \ge \alpha|S|$ for all $S \subseteq P$. Now note that the ratio $|\Gamma(S)|/|S|$ is a rational number of the form i/j where $j \le |P|$. The existence of the fractional matching implies that $|\Gamma(S)|/|S| \ge (1-|P|^{-2})$ for all $S \subseteq P$. Since $|\Gamma(S)|/|S|$ can only have denominator at most $|P|$, this implies that in fact $|\Gamma(S)| \ge |S|$ for all $|S|$. ■

Since the budgets $\lfloor (1-\epsilon)B_a \rfloor$ are integral, finding a complete matching with budgets $\lfloor (1-\epsilon)B_a \rfloor$ is equivalent to finding a complete matching in a graph with $\sum_{a\in A} \lfloor (1-\epsilon)B_a \rfloor$ vertices on the A side. Lemma 32 now implies the existence of a complete matching in the set of edges that the fractional matching is supported on. This completes the proof of Theorem 4.

6 Acknowledgements

The author is grateful to Nikhil Devanur for bringing the Gap-Existence problem to his attention.

References

[1] K. Ahn and S. Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *ICALP*, pages 526–538, 2011.

[2] K. Ahn and S. Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *CoRR*, abs/1104.2315, 2011.

[3] Denis Xavier Charles, Max Chickering, Nikhil R. Devanur, Kamal Jain, and Manan Sanghi. Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads. *ACM Conference on Electronic Commerce*, pages 121–128, 2010.

[4] Sebastian Eggert, Lasse Kliemann, and Anand Srivastav. Bipartite graph matchings in the semi-streaming model. *ESA 2009*, pages 492–503, 2009.

[5] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the streaming model: the value of space. *SODA*, pages 745–754, 2005.

[6] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348:207–216, 2005.

[7] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. *STOC*, 2002.

[8] A. Goel, M. Kapralov, and S. Khanna. On the communication and streaming complexity of maximum bipartite matching. `http://www.stanford.edu/~kapralov/papers/matching-covers-full.pdf` *(preliminary version appeared in SODA 2012)*.

[9] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. *STOC*, pages 587–596, 2011.

[10] R. Karp, U. Vazirani, and V. Vazirani. An optimal algorithm for online bipartite matching. *STOC*, 1990.

[11] Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. *CoRR*, abs/1112.0184, 2011.

[12] V. I. Levenstein. Upper bounds for codes with a fixed weight of vectors (in russian). *Problems of information transmission*, pages 3–12, 1971.

[13] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. *STOC*, pages 597–606, 2011.

[14] A. McGregor. Finding graph matchings in data streams. *APPROX-RANDOM*, pages 170–181, 2005.

[15] Rajeev Motwani, Rina Panigrahy, and Ying Xu. Fractional matching via balls-and-bins. *APPROX-RANDOM*, pages 487–498, 2006.