On the Robust Communication Complexity of Bipartite Matching

Sepehr Assadi
Rutgers University, Piscataway, NJ, USA

Soheil Behnezhad
University of Maryland, College Park, MD, USA

Abstract

We study the *robust* – à la Chakrabarti, Cormode, and McGregor [STOC'08] – communication complexity of the maximum bipartite matching problem. The edges of an *adversarially* chosen n-vertex bipartite graph G are partitioned *randomly* between Alice and Bob. Alice has to send a single message to Bob, using which Bob has to output an approximate maximum matching of G. We are particularly interested in understanding the best approximation ratio possible by protocols that use a near-optimal message size of $n \cdot \text{polylog}\,(n)$.

The communication complexity of bipartite matching in this setting under an *adversarial* partitioning is well-understood. In their beautiful paper, Goel, Kapralov, and Khanna [SODA'12] gave a $2/3$-approximate protocol with $O(n)$ communication and showed that this approximation is tight unless we allow more than a near-linear communication. The complexity of the robust version, i.e., with a *random* partitioning of the edges, however remains wide open. The best known protocol, implied by a very recent random-order streaming algorithm of the authors [ICALP'21], uses $O(n \log n)$ communication to obtain a $(2/3 + \varepsilon_0)$-approximation for a constant $\varepsilon_0 \sim 10^{-14}$. The best known lower bound, on the other hand, leaves open the possibility of all the way up to even a $(1 - \varepsilon)$-approximation using near-linear communication for constant $\varepsilon > 0$.

In this work, we give a new protocol with a significantly better approximation. Particularly, our protocol achieves a 0.716 expected approximation using $O(n)$ communication. This protocol is based on a new notion of *distribution-dependent sparsifiers* which give a natural way of sparsifying graphs sampled from a *known* distribution. We then show how to lift the assumption on knowing the graph's distribution via minimax theorems. We believe this is a particularly powerful method of designing communication protocols and might find further applications.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Maximum Matching, Communication Complexity, Random-Order Streaming

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.48

Category RANDOM

Funding *Sepehr Assadi*: Research supported in part by the NSF CAREER award CCF-2047061 and a gift from Google Research.
Soheil Behnezhad: Research supported by Google PhD Fellowship.

1 Introduction

Consider the following communication game. We have an n-vertex bipartite graph $G = (L, R, E)$ whose edges are partitioned into E^A and E^B given to Alice and Bob, respectively (both players know L and R). The goal is to compute an approximate maximum matching of G by Alice sending a single message to Bob and Bob outputting the solution. What is the tradeoff between the size of Alice's message and the approximation ratio of the output matching, or in other words, the *one-way communication complexity* of bipartite matching?

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021).
Editors: Mary Wootters and Laura Sanità; Article No. 48; pp. 48:1–48:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

It is known that $\Omega(n^2)$ communication is necessary for finding a maximum matching [23] and this is clearly sufficient by Alice sending her entire input. But the situation is more interesting for approximate protocols. A $1/2$-approximation with $O(n)$ communication can be obtained by Alice sending a maximum matching of her input to Bob and $\Omega(n)$ communication is clearly needed for any constant factor approximation. More interestingly, Goel, Kapralov, and Khanna [25] showed that $O(n)$ communication even suffices to obtain a $2/3$-approximation and that this is the "right" answer: any better approximation requires $n^{1+\Omega(1/\log\log n)} \gg n \cdot \text{polylog}(n)$ communication.

In this paper, we study a *robust* variant of this problem – à la Chakrabarti, Cormode, and McGregor [18] – wherein the graph G is still chosen adversarially, but now its edges are instead *randomly partitioned* between the two players, i.e., each edge is independently given to one of the players chosen uniformly at random. This model of random partitioning was introduced in [18] to go beyond the "doubly worst case" analysis of communication games, namely, adversarial inputs and adversarial partitions, and sheds more light into the source of hardness: whether it is due to a pathological partitioning of inputs or rather it holds for most input partitions.

Our main result is a substantial improvement over the $2/3$-approximations for adversarial partitions [25] under this random partition model.

▶ **Result 1** (Formalized in Theorem 17). *There is a randomized one-way protocol with $O(n)$ communication that achieves an expected 0.716-approximation for the bipartite matching problem under a random partitioning of the input edges between Alice and Bob.*

Prior to our work, the best known approximation ratio achievable for this problem was $(2/3+\varepsilon_0)$ for some $\varepsilon_0 \sim 10^{-14}$, obtained via the very recent random-order streaming algorithm of the same authors of this paper in [2].

We note that our protocol in this result can be considered *non-explicit*: we show the existence of the protocol rather than explicitly designing the protocol itself (see Section 1.1 for details). Alternatively, the protocol can be found also via a brute-force search in doubly exponential time.

1.1 Our Techniques

The $2/3$-approximation protocol of [25] (and follow-ups in [3] that simplified it or [37] that extended it to the online batch-arrival model) are all based on finding a suitable subgraph of Alice's input that preserves large matchings approximately, namely, a matching sparsifier (similar-in-spirit to cut sparsifiers [13, 14]). These subgraphs are defined through a series of graph-theoretic constraints: a novel decomposition into expanding sets (matching skeleton) in [25, 37] (see also [33]), and edge-degree bounded subgraphs in [3] (defined first in [16, 17] for dynamic graph algorithms). We take an entirely different approach in this paper.

The first step of our approach is a way of introducing distributional assumptions about the input, while still solving the problem in its full generality. In particular, in this step, we reduce the general problem to the case that the input graph G is sampled from some arbitrary but known distribution $\mathcal{G}$ of graphs. We achieve this via combining several relatively standard ideas specific to the matching problem with an application of Yao's minimax principle [41] (the so-called "hard direction" of this principle; see Section 5). This is the main *conceptual* step of our approach.

The second step is to design a protocol for the problem assuming that it is additionally given an input distribution $\mathcal{G}$ of the input graph. We achieve this through a new notion of "distribution-dependent sparsifiers" described below. This is our main *technical* step.

Distribution-dependent sparsifiers. Distribution-dependent sparsifiers can be used whenever we know a distribution $\mathcal{G}$ for inputs of Alice and Bob. In particular, the knowledge of $\mathcal{G}$ allows us to determine the "importance" of each edge in Alice's input E^A: this is the probability that this particular edge belongs to a fixed maximum matching (say, the lexicographically-first maximum matching) of a graph sampled from $\mathcal{G} \mid E^A$, i.e., input graphs after conditioning on Alice's input. The main part of our argument is to show that these importances can be used to sparsify the Alice's graph to $O(n)$ edges, while allowing Bob to find a large matching of the entire graph in expectation.

For our analysis of these sparsifiers, we need to show that the edges T communicated by Alice and the edges E^B given to Bob combined, include a large matching. We do so by constructing a large fractional matching $\mathbf{x}$ on the edges in $T \cup E^B$. Our construction of $\mathbf{x}$ is online, in the sense that we decide on the value of $\mathbf{x}$ induced on T before sampling Bob's subgraph E^B from $\mathcal{G} \mid E^A$. Thanks to the fact that Alice picks the edges of T according to their importance, we can construct $\mathbf{x}$ on T such that the fractional value around each vertex v is equal to the probability that v is matched in the optimum solution via an edge given to Alice. This is particularly useful because it implies that (i) the size of $\mathbf{x}$ induced on T equals the expected number of edges of Alice in an optimum matching, and that (ii) if a vertex is unlikely to be matched via an edge of Alice in the optimum solution, then the fractional matching $\mathbf{x}$ induced on T does not occupy this vertex by much, leaving room for the rest of the edges in E^B to use it.

As a warm-up in Section 3, we show how the ideas above lead to a very simple 0.656-approximate protocol under an *adversarial* partitioning of the input. This is only slightly worse than the optimal $2/3$-approximation for this problem, but more importantly, this warm-up conveys the key intuitions behind distribution-dependent sparsifiers and how they are extremely useful for matching in the communication setting. The protocol for our 0.716-approximation in Section 4 for the robust communication model is very similar, but its analysis is more involved and in particular is based on a careful examination of edge importance distributions under a random partitioning.

1.2 Further Aspects of Our Results

Random-order streams. The one-way communication model in general is strongly motivated by applications to *graph streaming algorithms* [23]. The *robust* communication model, in the same vain, is closely related to *random-order* streaming algorithms wherein the edges of the graph arrive in a random order. In particular, lower bounds in the (robust) communication model directly imply space lower bounds in the (random-order) streaming model [18] and upper bounds are sources of inspiration and stepping stones for designing streaming algorithms (see, e.g., [1, 25, 31] for instances of communication protocols that were turned into streaming algorithms in the context of the matching problem).

Maximum matchings have been studied extensively in random-order streams [1, 2, 15, 22, 24, 34, 35], leading to a $2/3$-approximation algorithm of [15] that hit a natural barrier for this problem, and the recent algorithm of [2] that improved this approximation to strictly more than $2/3$ (for a tiny constant improvement). We hope our ideas in this paper can lead to a significantly-better-than-$2/3$ approximation in random-order streams.

We remark that [2] proves the following robust communication lower bound (and thus a random-order streaming lower bound also): any $(1 - \Theta(1/\log n))$-approximation to maximum matching in the robust communication model requires $n^{1+\Omega(1/\log\log n)} \gg n \cdot \mathrm{polylog}(n)$ communication. Closing the gap between our upper bound and the lower bound of [2] remains a fascinating open question. Finally, we note that our improved protocol also has the following message: either one should be able to achieve a significantly-better-than-$^2/_3$ approximation (say, a 0.716-approximation) in random-order streams, or any lower bound technique for proving impossibility of such a result should deviate from the standard two-party communication complexity lower bound approach.

Non-explicit protocols. As remarked earlier, the protocol in our main result can be considered non-explicit. Alternatively, the players may need to first spend a doubly-exponential time to find the protocol, and only then they can use it to solve the problem (this is due to the arguments in the first step of our approach and in particular using Yao's minimax principle). From an algorithmic perspective, this is a weakness of our particular method of protocol design. On the other hand, we find our method particularly strong and insightful from a communication complexity point of view as we shall elaborate below.

Firstly, communication complexity is a non-uniform model of computation with players of unbounded computational power, and the only resource of interest is the communication cost of protocols. In this regard, our protocol uses the "full power" of this model to achieve its approximation ratio using the optimal $O(n)$ communication.

Secondly, and more importantly, there is a general gap in the study of communication complexity of graph problems: almost all protocols designed in the literature are based on algorithmic tools that are tailored to time-efficient protocols, while all known lower bounds are information-theoretic and hold even for protocols with computationally unbounded players. Can this inconsistent treatment be a contributing factor to the substantial gaps between known upper and lower bounds for various problems, including the robust communication complexity of bipartite matching? If so, then our approach in this paper allows us to explore a wider set of natural protocols for the problems at hand and move toward achieving tight(er) bounds on communication complexity. This will in turn suggest that purely information-theoretic complexity lower bounds cannot prove "strong enough" lower bounds for computationally-efficient algorithms as well. We leave the question of proving communication lower bounds for computationally-efficient protocols, which is the dual approach to our work in this paper, as a very interesting research direction for future work.

1.3 Further Related Work

The communication complexity of bipartite matching has been extensively studied from various angles including exact protocols [21,27,30], non-deterministic protocols [40], protocols with limited rounds of communication [3,5,23,25,26], or multi-party protocols [6,26,29,32,33] to name a few (this is by no means a comprehensive summary of previous results).

The one-way communication complexity of matching, in particular, is directly related to streaming algorithms. In fact, a key motivation in the work of Goel, Kapralov, and Khanna [25] was to determine whether there is a better-than-$^1/_2$-approximation algorithm for the matching problem in the streaming model that uses $\widetilde{O}(n)$ space, a longstanding open problem in this area. The lower bound in [25] implies that there is no semi-streaming algorithm with approximation ratio better than $^2/_3$; this lower bound was later improved by Kapralov to a $1 - {}^1/_e \approx 0.63$ in [31] and to $\frac{1}{1+\ln 2} \approx 0.59$ in [32]. Additionally, the communication protocols in [25] were also generalized in the same paper to achieve a $(1 - {}^1/_e)$-approximation in vertex-arrival streams.

Finally, we should point out that the work of [25] on the one-way communication complexity of bipartite matching has been quite instrumental and paved the path for various follow-ups including optimal algorithms for vertex-arrival streaming model [25,31], state-of-the-art lower bounds for streaming matching in both insertion-only streams [31,32] and dynamic streams [6, 20], online batch-arrival algorithms [37] and fault-tolerant algorithms [3] for maximum matching, stochastic matching problem [3, 4, 10, 11], and using RS graphs for proving communication lower bounds for other problems such as matrix rank [5, 9], independent sets [7, 19], and reachability [8].

2 Preliminaries

Notation. For any graph G, we use n to denote the number of vertices and $\mu(G)$ to denote the maximum matching size. A fractional matching $\mathbf{x}$ on a graph G is an assignment of values x_e to the edges e of G such that $x_e \geq 0$ for all edges e and for each vertex v, $x_v := \sum_{e \ni v} x_e \leq 1$. We use $|\mathbf{x}|$ as a shorthand for $\sum_e x_e$ which is the size of fractional matching $\mathbf{x}$.

The following standard fact implies that to show a bipartite graph has an integral matching of size μ, it suffices to construct a fractional matching of size μ on it.

▶ **Fact 1.** *Let* $\mathbf{x}$ *be a fractional matching of a bipartite graph* G*. Then* G *has an integral matching of size at least* $|\mathbf{x}|$*.*

Communication model. We study the standard two-party communication model of Yao [42] and in particular in the one-way model (see the excellent textbook by Kushilevitz and Nisan [36]). The only slight derivation is that we focus on randomly partitioned inputs, wherein the input graph is still chosen adversarially, but every edge in the graph is sent to one of the players chosen independently and uniformly at random. To our knowledge, this model was first introduced by Chakrabarti, Cormode, and McGregor in [18].

Unless specified otherwise, we assume that protocols are randomized and both players have access to the same *shared* source of randomness, referred to as *public coins*; however, one can always use Newman's theorem [39] to turn public coins into private coins with a negligible overhead. The communication cost of any protocol in this model is the worst-case length of the communicated messages; to be consistent with prior work on this problem in [3, 25, 31, 33], we measure the length of messages in $\Theta(\log n)$-bit words as opposed to the more standard convention of bits. Finally, we note that the main resource of interest in this model is the communication and the players are assumed to be *computationally unbounded.*

3 Warm-up: A 0.656-Approximation Under Adversarial Partitions

In this section, we describe a one-way protocol for the bipartite matching problem and prove that it achieves an approximation factor of $(4\sqrt{2}-5) \approx 0.656$ under an *adversarial* partitioning of the edges. While this protocol is slightly worse than the optimal $2/3$-approximate protocols in [3, 25] and its analysis shares some similarity with [12], we believe it is still instructive as it acts as a gentle introduction to the ideas used in our main protocol of Section 4.

A key technique introduced in this work is the notion of **distribution-dependent sparsifiers**. For now, let us assume that there is a *known* distribution $\mathcal{G}$ from which the inputs E^A and E^B of Alice and Bob are sampled. Now, suppose Alice has received E^A as input and plans to send a message to Bob. In order to do this, Alice considers the distribution of inputs conditioned on her input, i.e., $\mathcal{G} \mid E^A$. The message sent by Alice is then a subgraph of her input (the sparsifier), wherein each edge is included depending on the probability that this edge belongs to a fixed maximum matching of a graph sampled from $\mathcal{G} \mid E^A$.

Finally, we can lift the assumption on the knowledge of $\mathcal{G}$ using minimax theorems: distribution-dependent sparsifiers give us a deterministic protocol for *each* distribution of inputs with approximation ratio at least α for every distribution; thus, there should also exist a *single* randomized protocol that achieves the same α-approximation for *all* inputs. See Section 5 for this argument[1].

3.1 The Protocol

We now describe our new distribution-dependent protocol. For the rest of this proof, we assume that Alice and Bob are given the distribution of inputs $\mathcal{G}$. For each edge $e \in E^A$, we define:

$$a_e := \Pr_{G \sim \mathcal{G}}[e \in \mathsf{MM}(G) \mid E^A], \tag{1}$$

where function $\mathsf{MM}(\cdot)$ deterministically returns a fixed maximum matching of its input (for instance, the lexicographically-first one, or the one returned by the Hopcroft-Karp algorithm [28]). In words, a_e is the probability that e belongs to a fixed maximum matching of a graph G sampled from $\mathcal{G}$ conditioned on the input E^A given to Alice. We are going to treat a_e as the "importance" of edge e in E^A. Observe that since Alice is aware of $\mathcal{G}$, she can compute a_e for each edge $e \in E^A$.

Fractional matching interpretation. Consider the vector $\mathbf{a} := \{a_e\}_{e \in E^A}$. We claim that $\mathbf{a}$ is a feasible *fractional* matching of E^A: (i) for every edge $e \in E^A$, we have $a_e \geq 0$ as a_e is a probability, and (ii) for all vertices v, $a_v := \sum_{e \ni v} a_e \leq 1$ as it can be confirmed that:

$$a_v = \Pr_{G \sim \mathcal{G}}[v \text{ matched in } \mathsf{MM}(G) \text{ by edges of } E^A \mid E^A]. \tag{2}$$

This view of $\mathbf{a}$ presents a natural way of sparsifying Alice's input. Basically, we can sparsify the support of $\mathbf{a}$ via the standard cycle-canceling method (see Lemma 2 below) so that instead of (possibly up to) $\Omega(n^2)$ edges, it will only have $O(n)$ edges while still preserving the fractional matching of each *vertex* (but not necessarily the edges). This allows us to obtain another fractional matching $\mathbf{a}'$ that preserves key properties of $\mathbf{a}$ but is much sparser and thus Alice can simply send this fractional matching directly to Bob.

▶ **Lemma 2** (Cycle-Canceling Lemma – Folklore). *Let $\mathbf{f}$ be any fractional matching of E^A. There is another fractional matching $\mathbf{f}'$ on E^A such that:*

- **Sparsification property**: *There are at most $n-1$ edges e in E^A with $f'_e > 0$.*
- **Preserving marginals and size**: *For every vertex v, $f'_v = f_v$, also implying $|\mathbf{f}'| = |\mathbf{f}|$.*

Proof. Iteratively take a cycle in the support of $\mathbf{f}$, then alternately decrease and increase the value of edges in a way that the minimum value edge gets value zero. Since all cycles are even-length, the fractional matching around each vertices remains unchanged throughout the process. Once there are no more cycles, the remaining fractional matching is a forest with at most $n-1$ edges. ◀

We can now formalize the protocol as follows.

By Lemma 2 this protocol requires $O(n)$ communication (in fact, only $n-1$ edges). Thus, it only remains to analyze the approximation ratio of Algorithm 1 in the following.

[1] There is an important subtlety here: distribution-dependent sparsifiers approximate the matching *in expectation* over the choice of graphs in the distribution; in other words, the output matching is close to the optimal matching in expectation. To apply Yao's minimax principle however, one needs an *instance-wise* approximation for the input graph. Thus, the argument in this part is not a black-box application of minimax theorems.

Algorithm 1 A simple distribution-dependent sparsifier protocol.

(i) Given edges E^A as input to Alice, she computes the vector $\mathbf{a} = \{a_e\}_{e \in E^A}$ using Eq (1); as discussed above, $\mathbf{a}$ is a valid fractional matching of E^A.
(ii) Alice obtains fractional matching $\mathbf{a}'$ by running cycle canceling on $\mathbf{a}$ (Lemma 2) and then sends the edges T in the support of $\mathbf{a}'$ to Bob.
(iii) Bob, given message T from Alice and input E^B, returns a maximum matching of $E^B \cup T$.

▶ **Proposition 3.** *For any input distribution $\mathcal{G}$ on adversarial partitions, Algorithm 1 achieves a $4\sqrt{2} - 5 \approx 0.6568$ approximation in expectation and uses $O(n)$ communication.*

We prove this proposition in the next section.

3.2 The Analysis: Proof of Proposition 3

Recall that T is the support of the fractional matching $\mathbf{a}'$ that Alice sends to Bob. For the analysis, we only need to show that $T \cup E^B$ includes a large fractional matching (by Fact 1). To do so, we construct a fractional matching $\mathbf{x}$ supported on $T \cup E^B$ in the following way:

$$x_e = \begin{cases} a'_e & \text{if } e \in T, \\ 1 - \max\{a'_u, a'_v\} & \text{if } e = (u,v) \in \mathsf{MM}(G) \cap E^B \end{cases}. \tag{3}$$

Intuitively, once the subgraph E^A is given to Alice, we immediately commit her fractional matching $\mathbf{a}'$ to the final fractional matching $\mathbf{x}$. Then, after the subgraph E^B of Bob is revealed, on any edge $e = (u,v) \in \mathsf{MM}(G) \cap E^B$, we set $x_e = 1 - \max\{a'_u, a'_v\}$ which is the largest possible fractional value that does not violate its endpoints' fractional matching constraints due to $\mathbf{a}'$.

In what follows, for any choice of E^A, we lower-bound the ratio $\mathbf{E}[|\mathbf{x}| \mid E^A]$ to $\mathbf{E}[\mu(G) \mid E^A]$ which implies the approximation ratio of our protocol. We emphasize that $\mathbf{x}$ is only constructed for the analysis and in the protocol, Bob simply returns a maximum matching of $T \cup E^B$.

Consider a maximum matching edge uv which belongs to the input of Bob, i.e. $uv \in \mathsf{MM}(G) \cap E^B$ and suppose that $a'_v > a'_u$. Observe that in $\mathbf{x}$, we set $x_{uv} = 1 - \max\{a'_u, a'_v\} = 1 - a'_v$. In this case, we say that vertex v is *responsible* for edge uv. Based on this, we define[2]:

$$b_v := \mathbf{Pr}[\exists uv \in \mathsf{MM}(G) \cap E^B \text{ such that } a'_v > a'_u \mid E^A], \tag{4}$$

i.e., b_v is the probability that v is responsible for some edge. We first bound the size of $\mathsf{MM}(G)$ based on the values a_v and b_v.

▷ Claim 4. $\mathbf{E}[\mu(G) \mid E^A] = \sum_v \frac{1}{2} a_v + b_v$.

Proof. We claim that,

(i) $\mathbf{E}[|\mathsf{MM}(G) \cap E^A| \mid E^A] = \frac{1}{2} \sum_v a_v$: by the definition of a_v in Eq (2) and the fact that the number of vertices matched in any matching is twice the size of the matching;
(ii) $\mathbf{E}[|\mathsf{MM}(G) \cap E^B| \mid E^A] = \sum_v b_v$: since each responsible vertex has an edge in $\mathsf{MM}(G) \cap E^B$ and for each such edge, exactly one of its neighbors is responsible.

The claim now follows by adding up the two equations above. ◁

[2] In case of ties, we break ties arbitrarily so that only one vertex is responsible for an edge.

We now also bound the size of $\mathbf{x}$ based on a_v and b_v values.

▷ Claim 5. For any vertex v, define $g_v := \frac{1}{2}a_v + (1-a_v)b_v$. Then, $\mathbf{E}[|\mathbf{x}| \mid E^A] = \sum_v g_v$.

Proof. By definition,

$$\sum_v g_v = \sum_v (\tfrac{1}{2}a_v + (1-a_v)b_v) = |\mathbf{a}| + \sum_v (1-a_v)b_v.$$

The first term $|\mathbf{a}|$ in the sum corresponds to the part of fractional matching $\mathbf{x}$ constructed on the edges T sent by Alice, using the fractional matching $\mathbf{a}'$, where we have $|\mathbf{a}'| = |\mathbf{a}|$ by Lemma 2.

It thus remains to prove that contribution of $\mathbf{x}$ on the remaining edges (i.e. those given to Bob in $\mathsf{MM}(G) \cap E^B$), has expected size $\sum_v (1-a_v)b_v$. This follows from the fact that each vertex v is responsible for some edge $uv \in \mathsf{MM}(G) \cap E^B$ with probability b_v by Eq (4), and that when this happens, we set $x_{uv} = 1 - a'_v = 1 - a_v$ (as $a'_v = a_v$ for all v by Lemma 2). Noting that exactly one of the endpoints of each edge $e \in \mathsf{MM}(G) \cap E^B$ is responsible for it, we get that $\mathbf{x}$ on the set of edges given to Bob has expected size exactly $\sum_v (1-a_v)b_v$, completing the proof. ◁

Claims 4 and 5 imply that the approximation factor of Algorithm 1 is

$$\frac{\mathbf{E}[|\mathbf{x}| \mid E^A]}{\mathbf{E}[\mu(G) \mid E^A]} = \frac{\sum_v g_v}{\sum_v \frac{1}{2}a_v + b_v}. \tag{5}$$

To lower bound this ratio, we use Fact 6 below.[3]

▶ **Fact 6.** *For all $a, b \geq 0$ satisfying $a + b \leq 1$, it holds that $\frac{0.5a+(1-a)b}{0.5a+b} \geq 4\sqrt{2} - 5$.*

Now to use Fact 6 to lower bound the approximation factor, first recall that for each vertex v, by the definition of a_v and b_v in Eq (2) and (4), we have,

$$\begin{aligned} a_v + b_v &\leq \mathbf{Pr}[v \text{ is matched in } \mathsf{MM}(G) \cap E^A \mid E^A] + \mathbf{Pr}[v \text{ is matched in } \mathsf{MM}(G) \cap E^B \mid E^A] \\ &= \mathbf{Pr}[v \text{ is matched in } \mathsf{MM}(G) \mid E^A] \leq 1. \end{aligned}$$

Thus, we can apply Fact 6 and get that for each vertex v, $\frac{g_v}{0.5a_v+b_v} = \frac{0.5a_v+(1-a_v)b_v}{0.5a_v+b_v} \geq 4\sqrt{2}-5$. This implies that

$$\frac{\mathbf{E}[|\mathbf{x}| \mid E^A]}{\mathbf{E}[\mu(G) \mid E^A]} \overset{(5)}{=} \frac{\sum_v g_v}{\sum_v \frac{1}{2}a_v + b_v} \geq \frac{\sum_v (4\sqrt{2}-5)(\frac{1}{2}a_v + b_v)}{\sum_v \frac{1}{2}a_v + b_v} = 4\sqrt{2} - 5,$$

which proves Proposition 3 that Algorithm 1 achieves a $(4\sqrt{2}-5)$-approximation.

▶ Remark 7. There are distributions for which the inequality above is actually equality. That is, we have $\mathbf{E}[|\mathbf{x}| \mid E^A] = (4\sqrt{2}-5)\mathbf{E}[\mu(G) \mid E^A]$. Therefore, this analysis based on the construction of fractional matching $\mathbf{x}$ cannot show an approximation factor better than $(4\sqrt{2}-5)$ for this protocol.

That being said, by "scaling" the fractional matching $\mathbf{a}$ of Alice before sparsifying it, one can in fact achieve a ($2/3$)-approximation which is optimal for adversarial partitions with $O(n)$ communication [25]. We use this scaling idea in our protocol in Section 4.

[3] Mathematica can verify Fact 6; see e.g., this page on WolframAlpha.

4 A 0.7167-Approximation Under Random Partitions

In this section, we show that a properly "scaled" variant of our distribution-dependent sparsifier of Section 3 – formalized as Algorithm 2 – achieves a significantly better approximation factor of 0.7167 in expectation, under a *random* partitioning of the edges between the players.

▶ **Theorem 8.** *There is a deterministic one-way protocol that given any arbitrary but known distribution $\mathcal{G}$ of input graphs, and a graph G sampled from $\mathcal{G}$ partitioned randomly between Alice and Bob, outputs a matching $M(G)$ in G such that $\mathbf{E}\,|M(G)| \geq 0.7167 \cdot \mathbf{E}[\mu(G)]$. The protocol requires communicating at most $n-1$ edges from Alice to Bob.*

4.1 The Protocol

Recall from our Algorithm 1 in Section 3 that Alice, given her subgraph E^A, first defines a fractional matching $\mathbf{a}$ on E^A where for each edge $e \in E^A$, $a_e = \mathbf{Pr}_{G\sim\mathcal{G}}[e \in \mathsf{MM}(G) \mid E^A]$, and then applies cycle canceling on $\mathbf{a}$ and sends the support of the resulting fractional matching $\mathbf{a}'$ to Bob. Our protocol in this section is very similar, except that instead of applying cycle-canceling on $\mathbf{a}$, we first "scale" $\mathbf{a}$ to obtain another fractional matching $\mathbf{z}$ and then send the support of cycle-canceled version $\mathbf{z}'$ of $\mathbf{z}$ to Bob. To be more precise about what we mean by scaling $\mathbf{a}$, let us define:

$$h(x,y) := \min\left\{\frac{3}{2}, \frac{1}{x}, \frac{1}{y}\right\}. \tag{6}$$

Now for each edge $e=(u,v) \in E^A$ we define

$$z_e := h(a_v, a_u) \cdot a_e. \tag{7}$$

Noting that $\mathbf{a}$ is a fractional matching, we get that $a_v \leq 1, a_u \leq 1$, which implies $h(a_u, a_v) \geq 1$ and thus $z_e \geq a_e$. This means that indeed $\mathbf{z} = \{z_e\}_{e\in E^A}$ is entry-wise larger than $\mathbf{a}$. But can this scaling violate fractional matching constraints, i.e., for some v, $z_v := \sum_{e\ni v} z_e > 1$? As a simple consequence of our definition of function h, it turns out that indeed $\mathbf{z}$ is still a fractional matching.

▶ **Observation 9.** *Let $\mathbf{z}$ be obtained as above, then $\mathbf{z}$ is a fractional matching of E^A.*

Proof. It is clear that $\mathbf{z} \geq 0$ since $z_e \geq a_e \geq 0$ for each edge e. To see why $z_v \leq 1$ for all v, observe that for each edge $e=(u,v)$, $z_e = h(a_u,a_v)a_e \leq \frac{1}{a_v}a_e$; hence $z_v \leq \frac{1}{a_v}\sum_{e\ni v} a_e = a_v/a_v = 1$. ◀

Note that the proof of Observation 9 only uses $h(x,y) \leq \min\{\frac{1}{x}, \frac{1}{y}\}$. The reason that we defined h to be $\min\{\frac{3}{2}, \frac{1}{x}, \frac{1}{y}\}$ will be apparent later when analyzing the approximation.

Our scaled protocol can thus be formalized as follows.

Since the support of $\mathbf{z}'$ has $n-1$ edges, Algorithm 2 only requires communicating $n-1$ edges. It thus only remains to analyze its approximation ratio.

4.2 The Analysis of Algorithm 2

As in Section 3, to analyze the size of matching $\mathsf{MM}(T \cup E^B)$ reported by Bob, we construct a large fractional matching $\mathbf{x}$ on $T \cup E^B$ and then use the fact that the maximum matching of this graph is at least as large as any fractional matching on it. Our construction of this fractional matching $\mathbf{x}$ is also in fact the same as our construction in Section 3 with the

Algorithm 2 A scaled distribution-dependent sparsifier protocol.

```
(i)   Given edges $E^A$ as input to Alice, she computes the vector $\mathbf{a} = \{a_e\}_{e \in E^A}$
      using Eq (1).
(ii)  Alice then constructs $\mathbf{z} = \{z_e\}_{e \in E^A}$ using Eq (7); by Observation 9 $\mathbf{z}$ is a valid
      fractional matching of $E^A$.
(iii) Alice obtains a fractional matching $\mathbf{z}'$ by running cycle canceling on $\mathbf{z}$ (Lemma 2)
      and then sends the edges $T$ in the support of $\mathbf{z}'$ to Bob.
(iv)  Bob, given message $T$ from Alice and input $E^B$, returns a maximum matching of
      $T \cup E^B$.
```

difference that we first commit the sparsified version $\mathbf{z}'$ of the *scaled* fractional matching $\mathbf{z}$ to $\mathbf{x}$. More formally, we have:

$$x_e := \begin{cases} z'_e & \text{if } e \in T, \\ 1 - \max\{z'_u, z'_v\} & \text{if } e = (u, v) \in \mathsf{MM}(G) \cap E^B \end{cases}.$$

To analyze the size of $\mathbf{x}$, we need a few definitions. Definition 11 below for b_v is equivalent to the definition of b_v in Section 3, but instead of vector $\mathbf{a}$, for each edge $e \in \mathsf{MM}(G) \cap E^B$ the vertex with higher $\mathbf{z}$ is made responsible. To be more formal and to avoid ties (for pairs of vertices with $z_u = z_v$) we first define an ordering over the vertices in Definition 10 below and then define b_v.

▶ **Definition 10.** *Based on fractional matching $\mathbf{z}$, we define a total ordering over the vertex set V as follows. For any pair of vertices u and v with $z_u \neq z_v$, we say $v \succ u$ if $z_v > z_u$. For pairs u, v with $z_v = z_u$ we break the tie arbitrarily; say $v \succ u$ if the ID of v is larger than u.*

▶ **Definition 11.** *For each vertex v, define $b_v := \mathbf{Pr}[\exists u : uv \in \mathsf{MM}(G) \cap E^B$ and $v \succ u \mid E^A]$.*

Based on this definition of b_v and similar to Claim 4 of Section 3, we get that:

▷ Claim 12. $\mathbf{E}[\mu(G) \mid E^A] = \sum_v \frac{1}{2} a_v + b_v$.

Proof. Follows from the same argument in the proof of Claim 4. ◁

The next step is where we start to substantially deviate from the analysis of Section 3. We first give an informal explanation of why a different approach might be needed to analyze Algorithm 2 (the reader may choose to skip this informal explanation and jump to the new analysis after). After that, we formally describe our actual analysis which is based on a notion of "contribution sharing".

Informal explanation: why a different analysis is needed. In Claim 5 of Section 3 we showed $\mathbf{E}[|\mathbf{x}| \mid E^A] = \sum_v \frac{1}{2} a_v + (1 - a_v) b_v$, implying intuitively that each vertex v contributes an expected size of $g_v = \frac{1}{2} a_v + (1 - a_v) b_v$ to $\mathbf{x}$. We then proved the claimed approximation ratio by comparing this contribution g_v of each vertex v with $\frac{1}{2} a_v + b_v$, which can be thought of as the portion of the benchmark $\mathbf{E}[\mu(G) \mid E^A] = \sum_v \frac{1}{2} a_v + b_v$ charged to vertex v.

A straightforward generalization of this framework for analyzing Algorithm 2 would be as follows: It is not hard to see that $\mathbf{E}[|\mathbf{x}| \mid E^A] = \sum_v \frac{1}{2} z_v + (1 - z_v) b_v$ (the proof follows from a similar argument to Claim 5); thus it suffices to show that the contribution $g_v = \frac{1}{2} z_v + (1 - z_v) b_v$ of each vertex is large compared to the portion $\frac{1}{2} a_v + b_v$ of the optimum charged to this vertex. The problem with this type of argument, however, is that it is hard to measure exactly how the scaling part of Algorithm 2 is useful. In particular, take a vertex v and suppose that for every neighbor u of v in E^A, it holds that $a_u = 1$. This way, for each edge $e = (v, u) \in E^A$ we would have $h(a_v, a_u) = 1$ and thus $z_e = h(a_v, a_u)a_e = a_e$. That is, the edges of vertex v are in fact not scaled at all. This would mean that $z_v = a_v$ and thus $g_v = \frac{1}{2}z_v + (1 - z_v)b_v = \frac{1}{2}a_v + (1 - a_v)b_v$, which is not any different from the guarantee we would get for vertex v without any scaling.

The issue discussed above intuitively implies that in defining the contribution g_v of each vertex, not only we should take into account the values of z_v and b_v, but that in fact the values of a_u for neighbors u of v are also important. Motivated by this, we define g_v such that intuitively we share the contribution of each vertex with its neighbors. That is, each vertex passes a portion of its contribution to its neighbors, and as a result also receives a portion of the contribution of them. This dynamic allows us to argue that scaling does indeed help our protocol.

The formal analysis via "contribution sharing". Consider function $\ell(x)$ defined as

$$\ell(x) := \max\left\{\frac{x - 2/3}{6}, 0\right\}. \tag{8}$$

This function ℓ is the sharing function and the reason that is defined this way will be apparent later in the analysis. For each vertex v, define

$$g_v := \frac{1}{2}z_v + (1 - z_v)b_v - \ell(a_v)a_v + \sum_u \ell(a_u)a_{uv}. \tag{9}$$

The following lemma whose proof is deferred to the full version, states that the expected size of fractional matching $\mathbf{x}$ conditioned on E^A, is equal to $\sum g_v$. Therefore, intuitively, we can think of g_v as the amount that vertex v contributes to the size of $\mathbf{x}$ in expectation.[4]

▶ **Lemma 13.** $\mathbf{E}\left[|\mathbf{x}| \mid E^A\right] = \sum_v g_v$.

To show that $\mathbf{x}$ tends to be large, Lemma 13 above implies that it suffices to show g_v is large. The next definition and the lemma that follows it are used for this purpose.

▶ **Definition 14.** *Let $a, b \in [0, 1]$. We define:*

$$f(a, b, x) := b + \Big(\big(\tfrac{1}{2} - b\big)h(a, x) + \ell(x) - \ell(a)\Big) \cdot a \qquad \text{and} \qquad f(a, b) := \min_{x \in [0,1]} f(a, b, x),$$

▶ **Lemma 15.** *For any vertex v, it holds that $g_v \geq f(a_v, b_v)$.*

The proof of Lemma 15 is also deferred to the full version due to space constraints. This lower bound is particularly useful since $f(a_v, b_v)$ only depends on the values of a_v and b_v, whereas g_v also depends on a_u of neighbors u of v. Having this, if we in fact prove that $f(a_v, b_v) \geq \alpha(\frac{1}{2}a_v + b_v)$ for all v, then we get that Algorithm 2 achieves an approximation ratio of at least α since

$$\frac{\mathbf{E}[|\mathbf{x}| \mid E^A]}{\mathbf{E}[\mu(G) \mid E^A]} \overset{\text{Claim 12, Lemma 13}}{=} \frac{\sum_v g_v}{\sum_v \frac{1}{2}a_v + b_v} \overset{\text{Lemma 15}}{\geq} \frac{\sum_v f(a_v, b_v)}{\sum_v \frac{1}{2}a_v + b_v} \geq \frac{\sum_v \alpha(\frac{1}{2}a_v + b_v)}{\sum_v \frac{1}{2}a_v + b_v} \geq \alpha.$$

[4] We note that in fact Lemma 13 holds for *any* possible definition of function ℓ. That is, in the proof of Lemma 13, we do not use the value of $\ell(x)$ defined in Eq (8).

Note that up to this point of the analysis, we have not used the fact that the edges are partitioned randomly between Alice and Bob. Therefore, in light of the lower bound of [25] which proves achieving a better-than-(2/3)-approximation for requires $n^{1+\Omega(1/\log\log n)}$ communication, we get that Algorithm 2 cannot achieve a better-than-(2/3)-approximation under an adversarial partitioning of the input graph. As a result, there should be a choice of a_v, b_v such that $f(a_v, b_v) \leq \frac{2}{3}(\frac{1}{2}a_v + b_v)$. Indeed one can confirm that for $a_v = \frac{1}{2}$ and $b_v = \frac{1}{2}$, $f(\frac{1}{2}, \frac{1}{2}) = f(\frac{1}{2}, \frac{1}{2}, 0) = 0.5 = \frac{2}{3}(\frac{1}{2}a_v + b_v)$.

How random partitioning helps. Our main insight in bypassing the 2/3-barrier highlighted above is that for an average vertex v, it cannot always occur that $a_v = b_v = \frac{1}{2}$ under a random partitioning. Formally, for a vertex u chosen uniformly at random from V,

$$\mathbf{E}_{u\sim V}[a_u] = \frac{1}{n}\sum_v \mathbf{E}[a_v] = \frac{2}{n}\mathbf{E}|\mathbf{a}| = \frac{2}{n}\mathbf{E}|\mathsf{MM}(G) \cap E^A| \overset{(\star)}{=} \frac{2}{n} \cdot \frac{\mathbf{E}[\mu(G)]}{2} = \frac{1}{n}\mathbf{E}[\mu(G)],$$

$$\mathbf{E}_{u\sim V}[b_u] = \frac{1}{n}\sum_v \mathbf{E}[b_v] \overset{\text{Definition 11}}{=} \frac{1}{n}\mathbf{E}|\mathsf{MM}(G) \cap E^B| \overset{(\star)}{=} \frac{1}{n} \cdot \frac{\mathbf{E}[\mu(G)]}{2} = \frac{1}{2n}\mathbf{E}[\mu(G)],$$

where the equalities marked with $(\star)$ use the fact that each edge is given to Alice/Bob with probability 1/2. This implies that $\mathbf{E}[a_u] = 2\mathbf{E}[b_u]$ which formalizes our earlier claim that $a_u = b_u = \frac{1}{2}$ cannot always happen for an average vertex u.

To turn the intuition above into an actual analysis of the approximation factor for Algorithm 2 under a random partitioning, we write a factor revealing program formalized as Program 1. We prove that the solution to Program 1 is indeed a lower bound for the approximation ratio of Algorithm 2. The proof can be found in the full version of the paper and is based on our intuition above regarding the relation between $\mathbf{E}[a_u]$ and $\mathbf{E}[b_u]$ for a vertex u chosen at random.

We note that for generality Program 1 is written with a parameter p which is 1/2 (more generally p can be thought of as the probability that each edge is given to Alice).

▶ **Lemma 16.** *Let r be the solution of Program 1 below for $p = \frac{1}{2}$; then $\mathbf{E}|\mathbf{x}| \geq r \cdot \mathbf{E}[\mu(G)]$.*

▶ **Program 1.** A factor revealing (non-linear) program for the performance of Algorithm 2.

$$\begin{array}{rl}
\text{find} & \text{a distribution } \mathcal{S} \text{ for } (a,b) \text{ over } [0,1]\times[0,1] \\
\text{minimizing} & \mathbf{E}_{\mathcal{S}}[f(a,b)]/\mathbf{E}_{\mathcal{S}}[\frac{1}{2}a + b] \\
\text{subject to} & \mathbf{E}_{\mathcal{S}}[a] = \frac{2p}{1-p}\mathbf{E}_{\mathcal{S}}[b] \\
 & \mathbf{Pr}_{\mathcal{S}}[a + b \leq 1] = 1 \\
 & \mathbf{Pr}_{\mathcal{S}}[a, b \geq 0] = 1
\end{array}$$

In order to find the solution of Program 1, we simplify it and then write a factor revealing LP, showing that $r \geq 0.7167$ which implies the same bound on the approximation ratio of Algorithm 2. Due to space limits, we omit the details of this step, referring interested readers to the full version of the paper.

5 Lifting Knowledge of Distribution via Minimax Theorems

As discussed before, our protocol of Section 4 achieves its claimed approximation guarantee assuming that the input graph G is drawn from some distribution $\mathcal{G}$ that is known to the algorithm a priori. In the standard communication complexity model, however, we do not have access to distribution $\mathcal{G}$ and the algorithm should work against every possible input graph. In this section, we show how one can use minimax theorems to lift the assumption on knowledge of the distribution $\mathcal{G}$ in our protocols, without incurring any loss to the approximation guarantee. The following theorem formalizes our main result in the Introduction.

▶ **Theorem 17.** *There is a randomized one-way protocol that given any arbitrary input graph G partitioned randomly between Alice and Bob, outputs a matching $M(G)$ in G such that $\mathbf{E}\,|M(G)| \geq 0.716 \cdot \mu(G)$. The protocol requires $O(n)$ communication from Alice to Bob.*

Consider a *deterministic* protocol $\mathcal{A}$ and let us use $\mathcal{A}(G^A, G^B)$ to denote the size of the matching returned by the protocol $\mathcal{A}$ when Alice receives subgraph G^A and Bob receives subgraph G^B. Recall that in our discussion of Section 4, we say protocol $\mathcal{A}$ obtains an α-approximation if

$$\mathop{\mathbf{E}}_{G\sim\mathcal{G},(G^A,G^B)}[\mathcal{A}(G^A, G^B)] \geq \alpha \cdot \mathop{\mathbf{E}}_{G\sim\mathcal{G}}[\mu(G)], \tag{10}$$

where here and throughout this section, by subscript (G^A, G^B) we mean the random process of partitioning the edges of G into G^A and G^B independently and uniformly at random. This guarantee is inherently different from that of Theorem 17. In the following, we first show how one can remedy this part and then give the argument for lifting the assumption on the knowledge of $\mathcal{G}$.

Step 1: Getting an Instance-Wise Approximation Guarantee. In order to remove the assumption on the knowledge of the distribution $\mathcal{G}$ we first show that we can slightly modify our protocols to get an *instance-wise* expected approximation guarantee.

▶ **Lemma 18.** *Suppose that given any input distribution $\mathcal{G}$ on n-vertex graphs, there is an α-approximate maximum matching protocol $\mathcal{A}$ (i.e., $\mathcal{A}$ satisfies Eq (10)) with communication cost $O(n)$. For any input distribution $\mathcal{G}$ and any parameter $\varepsilon > 0$, there is another deterministic protocol $\mathcal{A}'$ with communication cost $O(\frac{n}{\varepsilon})$ such that*

$$\mathop{\mathbf{E}}_{G\sim\mathcal{G},(G^A,G^B)}\left[\mathcal{A}'(G^A, G^B)/\mu(G)\right] \geq (1 - \varepsilon - o(1)) \cdot \alpha.$$

Due to space constraints, we omit the proof of Lemma 18. However, we note that it is easy to prove if one allows $O(n \log n/\varepsilon)$ communication instead of $O(n/\varepsilon)$. To see this, note that if distribution $\mathcal{G}$ was such that $\mu(G)$ was the same in all outcomes of G, then Lemma 18 would be trivial. Therefore, one can make $O(\log n/\varepsilon)$ geometrically increasing guesses for the value of $\mu(G)$, condition on each separately, and run protocol $\mathcal{A}$ on them in parallel. In the full version of the paper, we show how one can avoid the extra $O(\log n)$ factor and achieve this with just $O(n/\varepsilon)$ communication.

Step 2: Using Yao's Minimax. Now that we have an instance-wise approximation guarantee using Lemma 18, we show how one can use Yao's minimax principle [41] to give a single randomized protocol that works against all possible input graphs without knowledge of the distribution $\mathcal{G}$ from which the graph is drawn. The discussion of this section is essentially a straightforward extension of Yao's minimax principle [41] (see, e.g., [38, Section 2.2] or [36]) for the random partition model. The proof of this proposition is almost identical to that of the original Yao's minimax principle and we claim no novelty for this proof.

▶ **Proposition 19.** *Let C and α be two parameters. Suppose for every distribution $\mathcal{G}$ on n-vertex graphs, there exists a deterministic protocol $\mathcal{A}_{\mathcal{G}}$ with communication cost C with an instance-wise approximation guarantee $\mathbf{E}_{G\sim\mathcal{G},(G^A,G^B)\sim G}\left[\mathcal{A}_{\mathcal{G}}(G^A, G^B)/\mu(G)\right] \geq \alpha$ where here*

(G^A, G^B) is a random partitioning of G. Then, there exists a randomized protocol $\mathcal{A}_\star$ with communication cost C such that for every *graph G, $\mathbf{E}_{\mathcal{A}_\star,(G^A,G^B)\sim G}[\mathcal{A}_\star(G^A, G^B)] \geq \alpha\mu(G)$, where the expectation here is taken over both the randomness of the protocol and the random partitioning of the edge-set of G between the players.*

Proof. Consider a game between two players called the Input player and the Algorithm player. The set of strategies of the Input player are all bipartite graphs on n vertices, denoted by $\mathbb{G}(n)$, and the set of strategies of the Algorithm player are all deterministic one-way protocols with communication cost C, denoted by $\mathbb{P}(C)$; for fixed n and C, both sets are finite.

For any graph $G \in \mathbb{G}(n)$ as a strategy of the Input player and deterministic protocol $\mathcal{A} \in \mathbb{P}(C)$ as the strategy of the Algorithm player, we define:

$$\mathsf{val}(G, \mathcal{A}) := \mathop{\mathbf{E}}_{(G^A,G^B)\sim G}\left[\mathcal{A}(G^A, G^B)/\mu(G)\right].$$

On a choice of (pure) strategies G and $\mathcal{A}$ by the players, we define the payoff of the Algorithm player as $\mathsf{val}(G, \mathcal{A})$ and for the Input player as $-\mathsf{val}(G, \mathcal{A})$. Alternatively, the Algorithm player would like to maximize $\mathsf{val}(G, \mathcal{A})$ (by choosing $\mathcal{A}$), while the Input player tries to minimize it (by choosing G). Thus, this is a zero-sum game.

Let Δ_G denote the set of all distributions on strategies (graphs) of the Input player and Δ_P denote the set of all distributions on strategies (deterministic protocols) of the Algorithm player. This way, Δ_G and Δ_P denote the set of all mixed strategies for the Input player and Algorithm player, respectively. Considering this is a zero-sum game, Von Neumann's Minimax Theorem asserts that,

$$\min_{\mathcal{G}\in\Delta(G)} \max_{\mathcal{A}\in\mathbb{P}(C)} \mathop{\mathbf{E}}_{G\sim\mathcal{G}}[\mathsf{val}(G, \mathcal{A})] = \max_{\mathcal{A}_R\in\Delta_P} \min_{G\in\mathbb{G}(n)} \mathop{\mathbf{E}}_{\mathcal{A}\sim\mathcal{A}_R}[\mathsf{val}(G, \mathcal{A})].$$

Replacing the value of $\mathsf{val}(G, \mathcal{A})$ with its definition on both sides, we have

$$\min_{\mathcal{G}\in\Delta(G)} \max_{\mathcal{A}\in\mathbb{P}(C)} \mathop{\mathbf{E}}_{G,(G^A,G^B)}\left[\frac{\mathcal{A}(G^A, G^B)}{\mu(G)}\right] = \max_{\mathcal{A}_R\in\Delta_P} \min_{G\in\mathbb{G}(n)} \mathop{\mathbf{E}}_{\mathcal{A}\sim\mathcal{A}_R,(G^A,G^B)}\left[\frac{\mathcal{A}(G^A, G^B)}{\mu(G)}\right]. \tag{11}$$

The LHS in Eq (11) corresponds to picking any possible distribution on inputs and then running the "best" deterministic protocol on this distribution and measuring the instance-wise expected approximation ratio of the protocol. Thus, by the statement of the proposition, the LHS is $\geq \alpha$.

The RHS in Eq (11) corresponds to picking any distribution over deterministic protocols, i.e., a (public-coin) randomized protocol, and then running this protocol on the "worst" input graph and measure the expected ratio of the protocol. By the lower bound on LHS and Eq (11), this is at least α, which means that there exists a randomized protocol $\mathcal{A}_\star$ with communication cost C (the $\arg\max$ of RHS in Eq (11)) that achieves an α-approximation in expectation for every input graph partitioned randomly between Alice and Bob. This concludes the proof. ◀

Theorem 17 now follows immediately from Theorem 8, Lemma 18 and Proposition 19.

References

1 Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In

Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019, pages 1616–1635, 2019.

2 Sepehr Assadi and Soheil Behnezhad. Beating two-thirds for random-order streaming matching. *CoRR*, abs/2102.07011. To appear in ICALP 2021, 2021.

3 Sepehr Assadi and Aaron Bernstein. Towards a unified theory of sparsification for matching problems. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, January 8-9, 2019 - San Diego, CA, USA*, pages 11:1–11:20, 2019.

4 Sepehr Assadi, Sanjeev Khanna, and Yang Li. The stochastic matching problem with (very) few queries. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 43–60, 2016.

5 Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1723–1742, 2017.

6 Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364, 2016.

7 Sepehr Assadi, Gillat Kol, and Rotem Oshman. Lower bounds for distributed sketching of maximal matchings and maximal independent sets. In Yuval Emek and Christian Cachin, editors, *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 79–88. ACM, 2020.

8 Sepehr Assadi and Ran Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 342–353. IEEE, 2020.

9 Maria-Florina Balcan, Yi Li, David P. Woodruff, and Hongyang Zhang. Testing matrix rank, optimally. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 727–746, 2019.

10 Soheil Behnezhad and Mahsa Derakhshan. Stochastic weighted matching: (1-ε) approximation. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1392–1403, 2020.

11 Soheil Behnezhad, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi. Stochastic matching with few queries: (1-ε) approximation. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1111–1124, 2020.

12 Soheil Behnezhad, Alireza Farhadi, MohammadTaghi Hajiaghayi, and Nima Reyhani. Stochastic matching with few queries: New algorithms and tools. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2855–2874, 2019.

13 András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 47–55, 1996.

14 András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM J. Comput.*, 44(2):290–319, 2015.

15 Aaron Bernstein. Improved bounds for matching in random-order streams. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, pages 12:1–12:13, 2020.

16 Aaron Bernstein and Cliff Stein. Fully dynamic matching in bipartite graphs. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, July 6-10, 2015, Proceedings, Part I*, pages 167–179, 2015.

17 Aaron Bernstein and Cliff Stein. Faster fully dynamic matchings with small approximation ratios. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, January 10-12, 2016*, pages 692–711, 2016.

18 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, May 17-20, 2008*, pages 641–650, 2008.

19 Graham Cormode, Jacques Dark, and Christian Konrad. Independent sets in vertex-arrival streams. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 45:1–45:14, 2019.

20 Jacques Dark and Christian Konrad. Optimal lower bounds for matching and vertex cover in dynamic graph streams. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

21 Shahar Dobzinski, Noam Nisan, and Sigal Oren. Economic efficiency requires interaction. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 233–242, 2014.

22 Alireza Farhadi, Mohammad Taghi Hajiaghayi, Tung Mai, Anup Rao, and Ryan A. Rossi. Approximate maximum matching in random streams. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1773–1785, 2020.

23 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005.

24 Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola Svensson. Weighted matchings via unweighted augmentations. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 491–500, 2019.

25 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 468–485. SIAM, 2012. URL: http://dl.acm.org/citation.cfm?id=2095116.2095157.

26 Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 287–298, 2013.

27 András Hajnal, Wolfgang Maass, and György Turán. On the communication complexity of graph properties. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 186–191, 1988.

28 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.

29 Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. Communication complexity of approximate matching in distributed graphs. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 460–473, 2015.

30 Gábor Ivanyos, Hartmut Klauck, Troy Lee, Miklos Santha, and Ronald de Wolf. New bounds on the classical and quantum communication complexity of some graph properties. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, pages 148–159, 2012.

31 Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697, 2013. doi:10.1137/1.9781611973105.121.

32 Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on*

Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021, pages 1874–1893. SIAM, 2021.

33 Michael Kapralov, Gilbert Maystre, and Jakab Tardos. Communication efficient coresets for maximum matching. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 156–164. SIAM, 2021.

34 Christian Konrad. A simple augmentation method for matchings with applications to streaming algorithms. In *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, pages 74:1–74:16, 2018.

35 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 231–242, 2012.

36 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

37 Euiwoong Lee and Sahil Singla. Maximum matching in the online batch-arrival model. In *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, pages 355–367, 2017.

38 Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

39 Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

40 Ran Raz and Boris Spieker. On the "log rank"-conjecture in communication complexity. In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 168–176, 1993.

41 Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 222–227, 1977.

42 Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 209–213, 1979.