Communication Complexity of Approximate Maximum Matching in Distributed Graph Data

Zengfeng Huang[1], Božidar Radunović[2], Milan Vojnović[3], and Qin Zhang[4]

Technical Report
MSR-TR-2013-35

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

[1]Zengfeng Huang is with Hong Kong University of Science and Technology, Hong Kong (huangzf@cse.ust.hk). His work was performed in part while an intern with Microsoft Research.
[2]Božidar Radunović is with Microsoft Research, Cambridge, UK (bozidar@microsoft.com).
[3]Milan Vojnović is with Microsoft Research, Cambridge, UK (milanv@microsoft.com).
[4]Qin Zhang is with IBM Research Almaden, USA (qinzhang@cse.ust.hk).

Abstract– We consider the problem of computing an approximate maximum matching in a graph that consists of n vertices whose edges are stored across k distributed sites in a data center. We are interested in characterizing the communication complexity of this problem which is of primary concern in data centers where communication bandwidth is a scarce resource. Our main result is that any algorithm that finds an α-approximate maximum matching has a communication complexity of $\Omega(\alpha^2 kn)$. Perhaps surprisingly, we show that this lower bound matches an upper bound of a simple sequential algorithm, showing that no benefits can be obtained with respect to the communication cost despite the full flexibility allowed by the underlying computation model. Our lower bound for matching also implies lower bounds for other important graph problems in the distributed computation setting, including max-flow and graph sparsification. Other main contribution of this paper is a new technique for multi-party randomized communication complexity that is of wide applicability.

1 Introduction

Massive volumes of data are being collected in almost every type of industry posing challenges to the system architecture and algorithm design to best support analytics on big data. To scale up the computations on big data, the data is typically distributed and stored across various sites in a data center. Sites are interconnected with a communication network. The key challenge on the algorithm design side is how to process the data without putting too much of strain on the communication network, which is typically the bottleneck for many data analytics tasks. A particular interest has been devoted to large-scale graph data as it arises in many applications including online social networks, online services, biological and other networks. On the system architecture side, a lot of recent effort has been devoted to designing computation platforms that take as input large-scale input graphs, e.g. general iterative computations platforms such as Google's Pregel [24], Apache Giraph [6] and machine learning platforms such as GraphLab [23, 22]. There has been a recent surge of interest in distributed graph databases where a key challenge is to support efficient resolving of queries on big graphs that may consist of as many as billions of vertices and trillions of edges, e.g. semantic web knowledge graph, and Facebook Graph Search [11]. On the system architecture side, there are now many available graph database systems, e.g. see Neo4j [26] and Trinity [28] and the references therein. A key open challenge is to design efficient algorithms for processing of big graphs in distributed systems as also evidenced by recent focus of the theory community on this type of problems, e.g. [21, 19, 2, 3, 5, 4].

In this paper we consider the problem of approximate computation of a maximum matching in an input graph that is edge-partitioned across different sites in a distributed system. This is considered in the *message-passing* (or *coordinator*) model (precise definition given in Section 1.1) which is the model of interest in the view of current architecture of platforms for big data analytics. Our main result is a lower bound on the communication complexity that we show to be tight. To the best of our knowledge, this is the first characterization of the communication complexity for solving the approximate maximum matching problem, which is one of most elementary and studied combinatorial problems on graphs, in the message-passing model. This is also one of the first graph problems studied in the message-passing model. The only previous work, as far as we are concerned, is the graph connectivity problem studied in [27].

1.1 The Communication Model

We consider a natural distributed computational model that consists of k sites, $P^1, \ldots, P^k$, where each site P^i holds a piece of data input x^i, and they want to jointly compute a function $f(x^1, \ldots, x^k)$. In order for them to do this, we allow the sites to communicate among themselves. At the end of the computation, we require that at least one site should return the answer. We assume that each site has infinite communicational power, and the internal computation of each site is free of charge [1]. Our goal is to minimize the total communication between the k sites. We can distinguish three types of multi-site communication models:

1. *Blackboard model*: We assume that there is a central blackboard. Every time one site speaks, the message is written to the blackboard and everyone else can see it.

2. *Message-passing model*: Every time a site speaks, it specifies another site and it can only send a message to that site; the other $k-2$ sites cannot see the message.

3. *Coordinator model*: In this model, we introduce another special party called the coordinator. The coordinator does not have any input. We require that all sites can only talk with the coordinator, and at the end of the computation, the coordinator should output the answer.

The blackboard model corresponds to the situation where each site has the ability to broadcast the message, while the message-passing model corresponds to the point-to-point communication. The coordinator model is essentially the same as the message-passing model for the following reasons: Every time a site P^i wants to speak to another site P^j, it can first send the message to the coordinator, and then the coordinator forwards the message to P^j. By doing this, we only increase the total communication by a factor of 2, thus this will not affect the asymptotic communication complexity. [2] The coordinator model has an advantage that the coordinator can specify who speaks next based on all previous communication, making the model more rigorous for analysis.

In this paper, we study the message-passing model, and will interchangably refer to it as the coordinator model and the message-passing model for convenience. The message-passing model (and its dynamic version called the *distributed streaming model*) is well-studied in literature (mainly for statistical problems). We refer the reader to [12, 30] and the references therein for an overview of the development in this model.

We would like to comment that there are two general classes of the multi-party communication model, namely the number-in-hand (NIH) model (which includes all the three variants addressed above) and the number-on-forehead (NOF) model. In NIH, each player only knows its own input, while in NOF, as the name suggests, each player sees all the inputs (on the other $k-1$ players' foreheads) except its own. These two models are used in different applications: NOF is usually used for applications such as circuit complexity and data structure lower bounds, while NIH is suitable for proving lower bounds in the data stream model and distributed computation for big data. In this paper, we only consider the NIH model, in particular, the message-passing (or the coordinator) model. We refer the reader to the classic book in the area [18] for more information about the NOF model.

[1] This assumption makes perfect sense for the lower bound purpose. While for the upper bounds, it is desired that the internal computation time is also kept small (e.g., linear in the size of the input).

[2] A careful reader may find that in order to specify the destination site P^j, the message from P^i should include an index j which is $\log k$ bits. This might increase the communication complexity by a factor of $\log k$, but if each message is at least of size $\log k$ bits (which is usually the case), then this extra index will not affect the asymptotic communication complexity. For simplicity, we neglect this extra cost in the paper.

1.2 Problem Definition and Overview of Results

We study the approximate maximum matching problem in the message-passing model, which we refer to as DMR (Distributed Matching Reporting). Given a set of $k > 1$ sites, an input graph $G = (V, E)$ with $|V| = n$ vertices, and partitioning of edges $E = E^1 \cup E^2 \cup \cdots \cup E^k$ such that site P^i is assigned the subset of edges E^i, the output is defined to be an α-approximation of the maximum matching, for given $0 \leq \alpha \leqslant 1$, which must be reported by at least one of the sites. Notice that DMR is different from the counting version of the problem where the output corresponds to an α-approximate *size* of a maximum matching. In this paper, we prove the following main theorem.

Theorem 1 *Given $\alpha \in [0, 1]$, any protocol that computes an α-approximation for DMR in the message passing model with error probability $1/4$ has the communication complexity of $\Omega(\alpha^2 kn)$ bits, assuming that $k \leq n$. This lower bound actually holds for bipartite graphs.*

It is noteworthy that a simple greedy algorithm solves DMR for $\alpha = 1/2$ at the communication cost of $O(kn \log n)$ bits. This greedy algorithm is based on computing a maximal matching by a straightforward sequential procedure through the sites that we define as follows. Let $G(E')$ be the graph induced by a subset of edges $E' \subseteq E$. The first site P^1 computes a maximal matching M^1 in $G(E^1)$, and sends it to P^2. Then, P^2 computes a maximal matching M^2 in $G(E^1 \cap E^2)$ by greedily adding edges in E^2 to M^1, and then sends M^2 to P^3, and this continues until P^k. At the end, P^k outputs $M = M^k$ that is a maximal matching in the whole graph G, hence a $1/2$-approximation of the maximum matching in G. The communication cost of this protocol is $O(kn \log n)$ bits, as the size of each M^i is at most n. Thus, our lower bound matches the upper bound up to a $\log n$ factor. In Section 4, we give an upper bound that also matches our lower bound in the approximation factor α for any $\alpha \leq 1/2$ (up to a $\log n$ factor), which further shows the tightness of the lower bound.

The above upper bound uses a very simple sequential algorithm. Given the flexibility of our computation model, one may a priori contemplate that it might be possible to improve upon the upper bound $\tilde{O}(kn)$ for a constant $0 < \alpha \leqslant 1$.[3] The key insight from our paper is that the asymptotic communication cost of the sequential algorithm cannot be improved beyond a logarithmic factor.

We comment that in the blackboard model, a maximal matching can be obtained using $O(n \log n + k)$ bits of communication through a simple modification of our greedy algorithm proposed above: when player i speaks, instead of sending the partial matching M^i to player $i + 1$, it simply writes $M^i \setminus M^{i-1}$ to the blackboard. Thus our lower bound separates the complexities of the matching problem in the two models.

An important contribution of our work is a new technique for studying communication complexity in the message-passing model that we believe is widely applicable. It is described in Section 1.3.

Direct Applications. Our result has also a wider applicability.

- Since bipartite matching can be solved using *max-flow* (find a feasible flow, not just approximate the value), our lower bound also holds for approximate max-flow.
- Our lower bound also implies a lower bound for *graph sparsification* (see the definition of graph sparsification, e.g., in [5]). This is because in our lower bound construction (see

[3] In $\tilde{O}$ we hide $\log^{O(1)}(kn)$ factors.

Section 3), the bipartite graph we constructed contains a lots of cuts of size 1, which have to be included in a sparsifier. By our construction, these edges form a good approximate matching. In [5], it is showed that there is a sketch based $O(1)$-approximate graph sparsification algorithm with sketch size $\tilde{O}(n)$, which directly translates to a protocol of $\tilde{O}(kn)$ communication in our model. Thus, our lower bound is tight up to a polylogarithmic factor.

- Any graph matching problem can be cast as a simple integer program [3]. When the graph is bipartite, the integrality gap is one, hence, our lower bound also holds for any general class of linear programs which contains the bipartite matching program as a special case.

1.3 New Technique for Lower Bounds

As far as we are concerned, there are currently two general techniques for randomized communication complexity lower bounds in the message-passing model. We briefly describe the high level ideas that underlie these two techniques, and then describe our new technique.

Symmetrization. The general idea of this technique, introduced in [27], is to reduce a 2-party problem to a k-party problem. Call the 2-party problem TWO and the k-party problem MULTI. Given an input (X, Y) for TWO, we proceed as follows.

1. Alice picks a random site P^I from the set of k sites, and assigns it an input $X^I = h(X)$, where X is Alice's input and h is a fixed function.
2. Bob constructs inputs for the other $k-1$ sites: Using a fixed function g, he computes an input $g(Y) = \{X^1, \ldots, X^{I-1}, X^{I+1}, \ldots, X^k\}$, and assigns X^i to site P^i for each $i \neq I$. In this construction, he guarantees that all $X^1, \ldots, X^{I-1}, X^{I+1}, \ldots, X^k$ have the same distribution as X^I (whose distribution is known to Bob), and $X^1, \ldots, X^k$ are independent given Y.
3. We show that a protocol that solves MULTI$(X^1, \ldots, X^k)$ can also solve TWO(X, Y).

Note that since the input distributions for the k sites are identical, in expectation, the communication cost of TWO should be at most a factor $2/k$ of that of MULTI. Thus, if we can show that TWO has a lower bound of $\Omega(n)$, then we will get an $\Omega(kn)$ lower bound for MULTI.

Composition. This technique was proposed in [30]. The high-level idea is to solve a complicated problem HARD under input $X^1, \ldots, X^k$, by first solving an easier problem EASY. Let $Z^1, \ldots, Z^k$ be the inputs to the k sites $P^1, \ldots, P^k$, respectively, for EASY. We assume that Z^i's are independent random variables such that each Z^i is the output of a 2-party problem TWO between the i-th site and the coordinator, i.e. $Z^i = \text{TWO}(X^i, Y)$, where X^i is the input of P^i and Y is something that the coordinator can construct after spending $T = o(kn)$ bits of communication with the k sites. Next, we show that to solve HARD, we must solve EASY, and to solve EASY, we must learn at least $\Omega(k)$ of Z^i's well. Now, in order to learn each Z^i well, we need to solve an instance of the problem TWO. Thus, if we can show that TWO has a lower bound of $\Omega(n)$, then we get an $\Omega(kn)$ bound for HARD.

Our New Technique. In this paper, we propose a new technique, which can be thought of as a combination of symmetrization and composition. At a high level, it is similar to symmetrization in that we want to perform a reduction from a 2-party problem TWO to a k-party problem MULTI. Given an input (A, B) for TWO, with $A, B \in \{0,1\}^{n/q}$,

1. Alice picks a random site P^I and assigns it an input $X^I = \{X^{I,1}, \ldots, X^{I,q}\} = h(A, R)$, where A is Alice's input, h is a fixed function, and R is some public randomness shared by Alice and Bob. Each $X^{I,j} \in \{0,1\}^{n/q}$.

2. Bob constructs inputs for the other $k-1$ sites and the coordinator. Concretely, he computes $g(B, R) = \{X^1, \ldots, X^{I-1}, X^{I+1}, \ldots, X^k, Y\}$ where $X^I = \{X^{I,1}, \ldots, X^{I,q}\}$, B is Bob's input, g is a fixed function, and R is some public randomness. Then, he assigns X^i to site P^i for each $i \neq I$, and assigns Y to the coordinator. In this construction, he guarantees that conditioned on Y, $X^{i,j}$ ($i \in [k]$, $j \in [q]$) are independent and identically distributed random variables. Let $Z^{i,j} = \text{TWO}(X^{i,j}, Y^j)$ ($i \in [k], j \in [q]$). Thus, conditioned on Y, $Z^{i,j}$'s are also i.i.d. random variables.

3. We show that, because of symmetry, we can recover the answer of $\text{TWO}(A, B)$ by simulating any protocol that solves $\text{MULTI}(X^1, \ldots, X^k)$. Then we show such a protocol must learn an $\Omega(1)$-fraction of $Z^{i,j}$'s well. Therefore, if $\text{TWO}(X^{i,j}, Y^j)$ has a lower bound of $\Omega(n/q)$, then we have an $\Omega(k \cdot q \cdot n/q) = \Omega(kn)$ bound for $\text{MULTI}(X^1, \ldots, X^k)$.

We briefly describe how we apply this framework to DMR. In the hard instance, we have a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n/2$. Each site P^i holds a set of $q = n/(2k)$ vertices which is a partition the set of left vertices U. The neighbors of each vertex in U is determined by a DISJ instance (that is, TWO is DISJ in our construction for DMR). In other words, we "embed" one DISJ instance into the neighborhood of each vertex in U. In total there are $q \times k = n/2$ DISJ instances, and we want to perform a direct-sum type of argument on these $n/2$ DISJ instances. To do this, we reduce DISJ to DMR in the way outlined above. We show that due to symmetry, the answer of DISJ can be recovered from a good matching reported. And then we use techniques from information complexity to establish the direct-sum theorem. We note that the use of symmetrization here is different from that in [27].

Notice that in this reduction, each site needs to solve q DISJ instances with the coordinator, thus for the purpose of a direct-sum argument, we have to use the *information cost* (see Section 2.3 for a definition) all the way through the proof, instead of simply using the communication cost as that in the previous works [27, 30]. For this purpose, we also need to give a new definition of information cost of a protocol in the message-passing (equivalently, the coordinator) model.

We believe that our techniques will have a wide applicability to prove distributed communication complexity for other graph problems. One reason is that for many graph problems whose solution certificates "span" the whole graph (e.g., connected components, vertex cover, dominating set, etc.), hard instances should be like ours for the matching problem, i.e., each of the k sites will contain roughly n/k vertices, and the neighborhood of each vertex defines an independent instance of a two-party problem. Thus, a direct-sum argument between each site and the coordinator using information cost may be necessary.

1.4 Related Work

The approximate maximum matching problem was studied extensively in the literature. In this section we only review the results obtained in some most related models, namely the streaming computation model [25, 1, 14, 2, 3, 4, 31, 17, 16] , distributed model of computation [29, 21, 20], and the Map-Reduce model [19].

In the streaming computation model, the maximum matching problem was presented as an open problem by McGregor [1] and a number of results have been established since then. Much

of previous work was devoted to the semi-streaming model, which allows $\tilde{O}(n)$ space. Recently, Ahn and Guha [3] obtained a $(1-\varepsilon)$-approximation for the maximum weight matching on graphs with space $\tilde{O}(n/\text{poly}(\varepsilon))$ bits and number of passes $\tilde{O}(1/\text{poly}(\varepsilon))$. Another recent work is that of Kapralov [16] studying approximate maximum matching in bipartite graphs in the streaming model. For the vertex arrival model (stream elements are vertices with all incident edges), he showed that no one-pass streaming algorithm (possibly randomized) that uses $\tilde{O}(n)$ bits of space can achieve better than $(1-1/e)$-approximation, and if r passes are allowed, a simple fractional load balancing achieves the approximating ratio $(1-1/\sqrt{2\pi r}+O(1/r))$ using $O(n\log n)$ bits of space. All these algorithms can be directly used to get an $\tilde{O}(kn)$ communication bound for $O(1)$-approximate matching in our message-passing model.

In the context of distributed model of computation, Lotker et.al. [21, 20] considered the problem of approximate solving of maximum matching problem in a synchronous distributed computation model. In this computation model, each vertex is associated with a processor and edges represent bidirectional communication. The time is assumed to progress over synchronous rounds, where in each round each processor may send messages to its neighbors, which are then received and processed in the same round by their recipients. This model is different from ours: in their model, the input graph and the communication topology are the same. While in our model, the communication topology is essentially a complete graph which is different from the input graph, and in general, sites are not vertices in the topology graph. Their model is generally used for computation in a network while our model is more tailored to accommodate the architecture of big data analytics platforms. Their main results include a randomized algorithm that yields $(1-\epsilon)$-approximation for the maximum matching problem in $O(\log n)$ rounds. This implies the communication cost of $\tilde{O}(m)$ bits where $m=O(n^2)$ is the number of edges in the graph.

The maximum matching problem was also studied in the Map-Reduce model by Lattanzi et.al. [19] (see [19] for a description of the Map-Reduce model). Under certain assumptions on the model, they obtain a $\frac{1}{2}$-approximation algorithm in $O(1)$ rounds and $\tilde{O}(m)$ bits of communication.

1.5 Conventions

Let $[n]=\{1,2,\ldots,n\}$. All logarithms are with base of 2. We always use capital letters $X,Y,\ldots$ to denote random variables or sets, and the lower case letters $x,y,\ldots$ to denote specific values of random variables $X,Y,\ldots$. We write $x\sim\mu$ to mean that x is chosen randomly according to the distribution μ. When we say a protocol Π has success probability p we always mean Π has success probability *at least* p. On the other hand, when we say Π has error probability p, we always mean Π has error probability *at most* p. For convenience, we often abuse the notation by using Π for both a protocol and its transcript. We usually call a player a *site*, which we feel to be more suitable in the coordinator model that we consider in this paper.

2 Preliminaries

2.1 Information Theory

Here we review some basic definitions and inequalities from the information theory which we use in our proofs. We refer the reader to [13] for an introduction to the information theory.

For two random variables X and Y, we use $H(X)$ to denote the Shannon entropy of the random variable X, and $H(X|Y)$ to denote the conditional entropy of X given Y. Let $I(X;Y)=$

$H(X) - H(X|Y)$ denote the mutual information between X and Y, and $I(X;Y|Z)$ be the conditional mutual information given Z. We know that $I(X;Y) \geq 0$ for any X, Y. We will need the following inequalities from the information theory.

Data processing inequality: If random variables X and Z are conditionally independent given Y, then $I(X;Y \mid Z) \leq I(X;Y)$ and $I(X;Z) \leq I(X;Y)$.

Super-additivity of mutual information: If $X^1, \cdots, X^t$ are independent, then $I(X^1, \cdots, X^t; Y) \geq \sum_{i=1}^t I(X^i; Y)$.

Sub-additivity of mutual information: If $X^1, \cdots, X^t$ are conditional independent given Y, then $I(X^1, \cdots, X^t; Y) \leq \sum_{i=1}^t I(X^i; Y)$.

2.2 Communication Complexity

In a two party communication complexity model, we have two players Alice and Bob. Alice is given $x \in \mathcal{X}$ and Bob is given $y \in \mathcal{Y}$, and they want to jointly compute some function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, by exchanging messages according to a randomized protocol Π. We use Π_{xy} to denote the random transcript (i.e., the concatenation of messages) when Alice and Bob run Π on the input (x, y), and $\Pi(x, y)$ to denote the output of the protocol. When the input (x, y) is clear from the context, we will simply use Π to denote the transcript. We say Π is a δ-error protocol if for all (x, y), the probability that $\Pi(x, y) \neq f(x, y)$ is no larger than δ, where the probability is over the randomness used in Π. Let $|\Pi_{xy}|$ be the length of the transcript. The communication cost of Π is $\max_{x,y} |\Pi_{xy}|$. The δ-error randomized communication complexity of f, denoted by $R_\delta(f)$, is the minimal cost of any δ-error protocol for f.

The multi-party NIH communication complexity model is a natural generalization of the two-party model, where instead of two parties, we have k parties, each having a piece of input, and they want to compute some function together by exchanging messages. For more information about the communication complexity we refer the reader to [18].

2.3 Information Complexity

The communication complexity measures the number of bits needed to be exchanged by multiple players in order to compute some function together, while the information complexity studies the amount of information of the inputs that must be revealed by the bits exchanged. It was extensively studied in the last decade, e.g., [10, 7, 8, 30, 9]. One of the main reasons to study information complexity is to prove direct-sum type theorems, i.e, the information complexity of solving t independent copies of the same function simultaneously is t times the information complexity of solving one. There are several definitions of information complexity. In this paper, we will follow the definition used in [7].[4] In the two-party case, let Π be a two-party communication protocol and μ be a distribution on $\mathcal{X} \times \mathcal{Y}$, we define the information cost of Π measured under μ as $IC_\mu(\Pi) = I(XY; \Pi \mid R)$, where $(X, Y) \sim \mu$ and R is the public randomness used in Π. For any function f, we define the information complexity of f parameterized by μ and δ as $IC_{\mu,\delta}(f) = \min_{\delta\text{-error } \Pi} IC_\mu(\Pi)$.

2.4 Information Complexity in the Message-Passing Model

We can indeed extend the above definition of information complexity to k-party message-passing model. That is, let X^i be the input of i-th player with $(X^1, \ldots, X^k) \sim \mu$ and Π be the whole

[4] Usually called the *external* information cost, in contract with the *internal information cost* used in papers such as [8].

transcript, then we could define $IC_\mu(\Pi) = I(X^1, \ldots, X^k; \Pi \mid R)$. However, such a definition does not fully explore the point-to-point communication feature of the message-passing model. Indeed, the lower bound we can prove using such a definition is at most a lower bound we can prove under the blackboard model. For some functions, there could be a gap as large as $\tilde{\Omega}(k)$ between the complexities in these two models. For example, our problem admits a simple algorithm with communication $O(n \log n + k) = O(n \log n)$ (if we assume $k = O(n)$) in the blackboard model.

In this paper we give a new definition of information complexity for the message-passing model, which allows us to prove higher lower bounds compared with the simple generalization. Let Π^i be the transcript between i-th player and the coordinator, thus $\Pi = \Pi^1 \circ \Pi^2 \circ \ldots \circ \Pi^k$. We define the information cost of a problem f with respect to input distribution μ and error parameter δ ($0 \leq \delta \leq 1$) in the message passing model as $IC_{\mu,\delta}(f) = \min_{\delta\text{-error } \Pi} \sum_{i=1}^k I(X^1, \cdots, X^k; \Pi^i)$. The proof of the following theorems/lemmas can be found in Appendix C.

Theorem 2 *$R_\delta(f) \geq IC_{\mu,\delta}(f)$ for any distribution μ.*

Lemma 2.1 *If Y is independent of the random coins used by the protocol Π, then*

$$IC_{\mu,\delta}(f) \geq \min_\Pi \sum_{i=1}^k I(X^i, Y; \Pi^i).$$

In this paper we will measure information cost with respect to distributional errors. Given an input distribution ν, we say a protocol has distributional error δ under ν if the protocol errors with probability at most δ, where the probability is taken over both the randomness used in the protocol and the input distribution ν. Clearly any lower bounds proved for the distributional error also hold for the worst-case error.

3 The Lower Bound

We first overview the main ideas and intuition of the proof. Our lower bound is established by constructing a hard input distribution on bipartite graphs with n vertices.

A natural idea to approximately compute a maximum matching is to randomly sample a few edges from each site and hope that we can find a good matching using these edges. For the lower bound purpose, we have to make such a random sampling strategy difficult, meaning that no good matching can be constructed out of those sampled edges if we only sample an insufficient number of edges. One way to do this is to create a lot of *noisy* edges, entirely covered by a small set of vertices (denoted by W). Clearly, the size of the matching that can be formed by noisy edges must be no more than $|W|$. At the same time, we will also create a set of *important* edges in such a way that any large matching will have to include many of them. We will carefully craft the bipartite graph so that these edges cannot be found easily by a small amount of communication between the k sites. We implement this idea by embedding $n/2$ set disjointness problems (DISJ) into the graph, such that to find each important edge, the k sites have to solve a DISJ instance, which is communication intensive. In our reduction, each site is involved in solving $n/(2k)$ DISJ instances, which introduces some extra technical challenges. In particular, we have to use information cost instead of communication cost to accomplish a direct-sum type of argument.

In the remainder of this section we first investigate the information cost of the DISJ problem under certain input distributions, for which we have to first understand the information cost of

a primitive problem AND. After that, we reduce DISJ to DMR and prove an information cost lower bound for DMR.

3.1 The AND Problem

In the problem AND, Alice and Bob hold bits x and y, respectively, and they want to compute $\mathrm{AND}(x, y) = x \wedge y$.

Let A be Alice's input and B be Bob's input. We define two input distributions for (A, B). Let $p = c \cdot \alpha \in (0, 1/2]$, where c is some constant to be chosen later.

ν_1: We first choose a random bit $W \in \{0, 1\}$ such that $\mathsf{Pr}[W = 0] = p$ and $\mathsf{Pr}[W = 1] = 1 - p$. If $W = 0$, we set $B = 0$, and $A = 0$ or 1 with equal probability. If $W = 1$, we set $A = 0$, and set $B = 1$ with probability $1 - p$ and $B = 0$ with probability p. Thus we have

$$(A, B) = \begin{cases} (0,0) & \text{w. pr.} \quad 3p/2 - p^2, \\ (0,1) & \text{w. pr.} \quad 1 - 2p + p^2, \\ (1,0) & \text{w. pr.} \quad p/2. \end{cases}$$

W here is served as an auxiliary random variable to break the dependence between A and B, since ν_1 is not a production distribution. The use of W will be clear in the reduction. Let τ be the distribution of W. Note that τ partitions ν_1, i.e, given τ, ν_1 is a product-form distribution.

μ_1: We first choose W according to τ, and then choose (A, B) according to ν_1 given W. Next, we reset A to be 0 or 1 with equal probability. Let δ_1 be the probability that $(A, B) = (1, 1)$ under distribution μ_1. We have $\delta_1 = (1 - 2p + p^2)/2$.

For $p = 1/2$, it is proved in [7] that if a private coin protocol Π has worst case error $1/2 - \beta$, then $I(A, B; \Pi \mid W) \geq \Omega(\beta^2)$, where the information cost is measured with respect to μ_1. Here we extend this to any $p \leq 1/2$ and distributional error.

We say a protocol has a one-sided error δ for AND under μ, if it has distributional error δ and always answers correctly for inputs (x, y) satisfying $\mathrm{AND}(x, y) = 0$.

Theorem 3 *Let Π be the transcript of any public coin protocol for AND on input distribution μ_1 with error probability $\delta_1 - \beta$ for a $\beta \in (0, \delta_1)$. We have $I(A, B; \Pi \mid W, R) = \Omega(\beta^2 p / {\delta_1}^2)$, where the information is measured when $W \sim \tau$, $(A, B) \sim \nu_1$, and R is the public randomness. If Π has a one-side error $\delta_1(1 - \beta)$, then $I(A, B; \Pi \mid W, R) = \Omega(\beta p)$.*

Proof. The proof is somewhat technical and is deferred to Appendix A. □

3.2 The DISJ Problem

In the problem DISJ, Alice holds $s = \{s_1, \ldots, s_k\} \in \{0, 1\}^k$ and Bob holds $t = \{t_1, \ldots, t_k\} \in \{0, 1\}^k$, and they want to compute $\mathrm{DISJ}(s, t) = \bigvee_{\ell=1}^{k} \mathrm{AND}(s_\ell, t_\ell)$.

Let $S = \{S_1, \ldots, S_k\}$ be Alice's input and $T = \{T_1, \ldots, T_k\}$ be Bob's input. We again define two input distributions for (S, T).

ν_k: We first choose $W = \{W_1, \ldots, W_k\} \sim \tau^k$, and then choose $(S_\ell, T_\ell) \sim \nu_1$ given W_ℓ, for each $1 \leq \ell \leq k$. For notation convenience, let $\nu_{k|*w}$ be the distribution of S conditioned on $W = w$, and let $\nu_{k|w*}$ be the distribution of T conditioned on $W = w$.

μ_k: We first choose $W = \{W_1, \ldots, W_k\} \sim \tau^k$, and then choose $(S_\ell, T_\ell) \sim \nu_1$ given W_ℓ, for each $1 \le \ell \le k$. Next, we pick a special coordinate D uniformly at random from $\{1, \ldots, k\}$, and reset S_D to be 0 or 1 with equal probability. Note that $(S_D, T_D) \sim \mu_1$, and the probability that $\mathrm{DISJ}(S, T) = 1$ is also δ_1. For notation convenience, let $\mu_{k|S=s}$ be the distribution of T conditioned on $S = s$, and let $\mu_{k|T=t}$ be the distribution of S conditioned on $T = t$.

Similar to AND, we say a protocol has a one-sided error δ for DISJ under μ_k, if it has distributional error δ and always answers correctly for inputs (s, t) satisfying $\mathrm{DISJ}(s, t) = 0$.

Theorem 4 *Let Π be the transcript of any public coin protocol for DISJ on input distribution μ_k with error probability $\delta_1 - \gamma$ for a $\gamma \in (0, \delta_1)$. We have $I(S, T; \Pi \mid W, R) = \Omega(\gamma^2 pk/{\delta_1}^2)$, where the information is measured when $W \sim \tau^k$, $(S, T) \sim \mu_k$, and R is the public randomness used by the protocol. If Π has a one-sided error $\delta_1(1 - \gamma)$, then $I(S, T; \Pi \mid W, R) = \Omega(\gamma pk)$.*

Proof. The proof is deferred to Appendix B. □

3.3 Proof of Theorem 1

In this section we first reduce DISJ to DMR, and then give a proof for Theorem 1.

Before going to the detailed reduction, we first provide an overview of the hard input distribution that we construct for DMR. The whole graph is a random bipartite graph consisting of $q = n/(2k)$ i.i.d. random bipartite graphs $G^1, \ldots, G^q$, where $G^j = (U^j, V^j, E^j)$ with $U^j = \{u^{j,1}, \ldots, u^{j,k}\}$ and $V^j = \{v^{j,1}, \ldots, v^{j,k}\}$. The set of neighbors of each vertex $u^{j,i} \in U^j$, for $i \in [k]$, is determined by a k-bit random vector $X^{j,i}$, that is, $(u^{j,i}, v^{j,\ell}) \in E^j$ if $X^{j,i}_\ell = 1$. The k (k-bit) random vectors $\{X^{j,1}, \ldots, X^{j,k}\}$ are chosen as follows: we first choose $(X^{j,1}, Y^j) \sim \mu_k$, and then independently choose for each $i \in \{2, \ldots, k\}$, a k-bit vector $X^{j,i}$ according to the conditional distribution $\mu_{k|T=Y^j}$. Finally, the input for the i-th site is simply vertices $\{u^{1,i}, \ldots, u^{q,i}\}$ and all their incident edges, which is actually determined by $X^i = \{X^{1,i}, \ldots, X^{q,i}\}$. Note that $Y = \{Y^1, \ldots, Y^k\}$ is *not* part of the input for DMR, but it will be helpful to think Y as a *virtual* input for the coordinator.

Input Reduction. Let $s \in \{0, 1\}^k$ be Alice's input and $t \in \{0, 1\}^k$ be Bob's input for DISJ. Alice and Bob construct an input $\{X^1, \ldots, X^k\}$ for DMR, where $X^i = \{X^{1,i}, \ldots, X^{q,i}\}$ with $X^{j,i} \in \{0, 1\}^k$ ($j \in [q]$) is the input for site i.

1. Alice and Bob use public coins to sample an index I uniformly at random from $\{1, \ldots, k\}$. Alice will construct the input X^I for the I-th site, and Bob will construct the inputs $X^1, \cdots, X^{I-1}, X^{I+1}, \cdots, X^k$ for the other $k - 1$ sites.

2. Alice and Bob use public coins to sample an index J uniformly at random from $\{1, \ldots, q\}$.

3. Alice sets $X^{J,I} = s$, and Bob sets $Y^J = t$. For each $i \in [k] \wedge i \neq I$, Bob privately samples $X^{J,i}$ according to $\mu_{k|T=t}$. This finishes the construction of G^J.

4. For each $j \in [q] \wedge j \neq J$, they construct G^j as follows,

   (a) Alice and Bob first use public coins to sample $W^j = \{W^j_1, \ldots, W^j_k\} \sim \tau^k$ (see the definition of τ in Section 3.1).

(b) Alice and Bob privately sample $X^{j,I}$ and Y^j according to conditional distributions $\nu_{k|*W^j}$ and $\nu_{k|W^j*}$, respectively. Bob also privately samples $X^{j,1}$, ..., $X^{j,I-1}$, $X^{j,I+1}$, ..., $X^{j,k}$ independently according to the conditional distribution $\nu_{k|T=Y^j}$.

(c) Alice privately samples $D^{j,I}$ uniformly at random from $\{1,\ldots,k\}$, and resets $X^{j,I}_{D^{j,I}}$ to be 0 or 1 with equal probability. This makes $\{X^{j,I}, Y^j\} \sim \mu_k$. Bob does the same for all $i \in [k] \wedge i \neq I$. That is, for each $i \in [k] \wedge i \neq I$, he privately samples $D^{j,I}$ uniformly at random from $\{1,\ldots,k\}$, and resets $X^{j,i}_{D^{j,I}}$ to be 0 or 1 with equal probability.

Note that the I-th site's input X^I is determined by the public coins, Alice's input s and her private coins. And the remaining $k-1$ sites' inputs $\{X^1,\cdots,X^{I-1},X^{I+1},\cdots,X^k\}$ are determined by the public coins, Bob's input t and his private coins. Let ϕ denote the distribution of $\{X^1,\ldots,X^k\}$ when (s,t) is chosen according to the distribution μ_k.

In this reduction, in each bipartite graph G^j, we carefully embed k instances of DISJ in random positions, and the output of a DISJ instance determines whether a specific edge in the graph exists or not. In the whole graph, we embed a total of $k \times q = n/2$ DISJ instances. The input of one such DISJ instance is just the original input of Alice and Bob, and the other $(n/2-1)$ instances are sampled by Alice and Bob using public and private random coins. Such a symmetric construction can be used to argue that if the original DISJ instance is solved, then with a good probability, at least $\Omega(n)$ of embedded DISJ instances are solved. We note that this use of symmetrization is different from that in [27]: The proof that the original DISJ instance can be solved by solving DMR (that is, obtaining a good matching) is rely on the symmetric property.

Let $p = \alpha/20 \leq 1/20$, where recall that p is a parameter in distribution μ_k and α is the approximation parameter. Now, given a protocol $\mathcal{P}'$ for DMR that achieves an α-approximation and error probability $1/4$ with respect to ϕ, we construct a protocol $\mathcal{P}$ for DISJ with one-sided error probability $\delta_1(1-\alpha/10)$ with respect to μ_k, as follows.

Protocol $\mathcal{P}$

1. Given an input $(S,T) \sim \mu_k$, Alice and Bob construct an input $(X^1,\ldots,X^k) \sim \phi$ for DMR as described by the input reduction above. Let $Y = \{Y^1,\ldots,Y^q\}$ be the virtual input created for the coordinator. Let I, J be the two indices sampled by Alice and Bob during the reduction.

2. Alice plays the I-th site, and Bob plays the other $k-1$ sites and the coordinator. They run $\mathcal{P}'$ for DMR. Any communication between the I-th site and the other $k-1$ sites and the coordinator will be exchanged between Alice and Bob. For any communication between the other $k-1$ sites and the coordinator, Bob will just simulate it without any actual communication. At the end the coordinator (that is, Bob) gets a matching M.

3. Bob outputs 1 if and only if there exists an edge $(u^{J,I}, v^{J,\ell})$ in the matching M for some $\ell \in [k]$, such that $Y^J_\ell \equiv T_\ell = 1$, and 0 otherwise.

In the rest of this section, we first show the correctness of the reduction, and then use an information cost lower bound of DISJ to prove an information cost lower bound of DMR.

Correctness. First, suppose $\mathrm{DISJ}(S,T) = 0$, i.e., $S_\ell \wedge T_\ell = 0$ for all $\ell \in [k]$. Then, for each $\ell \in [k]$, we must have either $Y^J_\ell \equiv T_\ell = 0$ or $X^{J,I}_\ell \equiv S_\ell = 0$, but $X^{J,I}_\ell = 0$ means no edge

between $u^{J,I}$ and $v^{J,\ell}$. Thus $\mathcal{P}$ will always answer correctly when $\mathrm{DISJ}(S,T)=0$, i.e., it has a one-sided error.

Now suppose that $S_\ell = T_\ell = 1$ for a certain $\ell \in [k]$ (note that there is at most one such ℓ according to our construction), which we denoted by L. The output of $\mathcal{P}$ is correct if $(u^{J,I}, v^{J,L}) \in M$. In the rest of the analysis we estimate the probability that this event happens.

For each $G^j = \{U^j, V^j\}$ ($j \in [q]$), let $U_1^j = \{u^{j,i} \mid \mathrm{DISJ}(X^{j,i}, Y^j) = 1\}$ and $U_0^j = U^j \setminus U_1^j$. Let $V_1^j = \{v^{j,\ell} \mid Y_\ell^j = 1\}$ and $V_0^j = V^j \setminus V_1^j$. Let $U_0 = \cup_{j=1}^q U_0^j$, $U_1 = \cup_{j=1}^q U_1^j$, $V_0 = \cup_{j=1}^q V_0^j$ and $V_1 = \cup_{j=1}^q V_1^j$. Intuitively, edges between $U_0 \cup U_1$ and V_0 can be seen as *noisy* edges, since the total number of such edges is large but the maximum matching they can form is small (at most $|V_0| \le 2pn$ according to Lemma 3.1, see below). On the contrary, we say the edges between U_1 and V_1 the *important* edges, since the maximum matching they can form is large, though the total number of such edges is small. Note that there is no edge between U_0 and V_1. Therefore, to find a good matching we must choose many edges from the important edges. A key feature here is that all important edges are *symmetric*, that is, each important edge is equally likely to be the edge $(u^{J,I}, v^{J,L})$. Thus with a good probability $(u^{J,I}, v^{J,L})$ will be included in the matching returned by $\mathcal{P}'$. Using this we can answer whether $X^{J,I}$ $(= S)$ and Y^J $(= T)$ intersect or not, thus solving the original DISJ problem.

We first estimate the size of the maximum matching in graph $G = \{G^1, \ldots, G^q\}$. Recall we set $p = \alpha/20 \le 1/20$ and $\delta_1 = (1 - 2p + p^2)/2$, thus $9/20 < \delta_1 < 1/2$.

Lemma 3.1 *With probability* 0.99*, the following events happen.*

1. $|V_0| \le 2pn$*. In this case the size of the maximum matching formed by edges between* V_0 *and* $U_0 \cup U_1$ *is no more than* $2pn$*.*
2. *The maximum matching of the graph* G *is at least* $0.2n$*.*

Proof. The first item follows simply by a Chernoff bound. Note that each vertex in $\bigcup_{j\in[q]} V^j$ is included in V_0 independently with probability $(2p-p^2)$, and $\mathsf{E}[|V_0|] = (2p-p^2)n/2$, therefore $\mathsf{Pr}[|V_0| \ge 2pn] \ge \mathsf{Pr}[|V_0| - \mathsf{E}[|V_0|] \ge pn] \le e^{-\Omega(p^2 n)}$.

For the second item, we first consider the size of the matching in G^j for a fixed $j \in [q]$, that is, a matching between vertices in U^j and V^j. For each $i \in [k]$, let L^i be the coordinate ℓ where $X_\ell^{j,i} = Y_\ell^j = 1$ if such an ℓ exists (note that by our construction at most one such coordinate exists), and NULL otherwise.

We use a greedy algorithm to construct a matching between U^j and V^j. For i from 1 to k, we connect $u^{j,i}$ to v^{j,L^i} if L^i is not NULL and v^{j,L^i} is not connected by any $u^{j,i'}$ $(i' < i)$. At the end, the size of the matching is essentially the number of distinct elements in $\{L^1, \ldots, L^k\}$, which we denote by R. We have the following claim. The proof is similar to Lemma 4 in [30], and we leave it to Appendix C.3.

Claim 1 *It holds* $R \ge 0.25k$ *with probability* $1 - O(1/k)$*.*

Therefore, for each $j \in [k]$, with probability $1 - O(1/k)$, we can find a matching in G^j of size at least $0.25k$. If $q = n/(2k) = o(k)$, then by a simple union bound it holds that with probability at least 0.99, the size of the maximum matching in $G = \{G^1, \ldots, G^q\}$ is at least $0.25n$. Otherwise, since $G^1, \ldots, G^q$ are constructed independently, by another application of Chernoff bound, we have that with probability $1 - e^{-\Omega(q)} \ge 0.99$, the size of the maximum matching in $G = \{G^1, \ldots, G^q\}$ is at least $0.2n$. ◻

Now let us make our intuition above more precise. First, if $\mathcal{P}'$ is an α-approximation protocol with error probability $1/4$, then by Lemma 3.1 we have that with probability at least $3/4 - 0.01 \geq 2/3$, $\mathcal{P}'$ will output a matching M containing at least $(\alpha \cdot 0.2n - 2pn)$ important edges. We know that there are at most $n/2$ important edges and the edge $(u^{J,I}, v^{J,L})$ is one of them. We say (i, j, ℓ) is important for G, if $(u^{j,i}, v^{j,\ell})$ is an important edge in G. Since our construction is totally symmetric, for any G in the support, we have

$$\mathsf{Pr}[I = i, J = j, L = \ell \mid G] = \mathsf{Pr}[I = i', J = j', L = \ell' \mid G].$$

for any (i, j, ℓ) and (i', j', ℓ') which are important in G. In other words, given an input G, the protocol can not distinguish between any two important edges. Then we can apply the principle of deferred decisions to decide the value (I, J) after the matching has already been computed, i.e., the probability $(u^{J,I}, v^{J,L}) \in M$ is at least

$$2/3 \cdot \frac{\alpha \cdot 0.2n - 2pn}{n/2} \geq \alpha/10.$$

Recall that we have chosen $p = \alpha/20$.

To sum up, protocol $\mathcal{P}$ solves DISJ correctly with one-sided error at most $(\delta_1(1 - \alpha/10))$, where δ_1 is the probability that $\mathrm{DISJ}(S, T) = 1$ when (S, T) is distributed according to μ_k.

Information Cost. Now we analyze the information cost of DMR. Let $\Pi = \Pi^1 \circ \Pi^2 \circ \cdots \circ \Pi^k$ be the best protocol for DMR with respect to input distribution ϕ and one-sided error probability $\delta = \delta_1(1 - \alpha/10)$. By Lemma 2.1, we have $IC_{\phi,\delta}(\mathrm{DMR}) \geq \sum_{i=1}^{k} I(X^i, Y; \Pi^i)$. Let $W^{-J} = \{W^1, \ldots, W^q\} \setminus W^J$, and $W = W^J W^{-J}$. Recall that in our input reduction I, J, W^{-J} are public coins used by Alice and Bob.

$$\begin{aligned}
2/n \cdot IC_{\phi,\delta}(\mathrm{DMR}) &\geq 1/(qk) \cdot \sum_{i=1}^{k} I(X^i, Y; \Pi^i) \\
&\geq 1/(qk) \cdot \sum_{i=1}^{k} I(X^i, Y; \Pi^i \mid W) \quad \text{(data processing inequality)} \\
&\geq 1/(qk) \cdot \sum_{i=1}^{k} \sum_{j=1}^{q} I(X^{j,i}, Y^j; \Pi^i \mid W^{-j}, W^j) \quad \text{(super-additivity)} \quad (1) \\
&= 1/(qk) \cdot \sum_{i=1}^{k} \sum_{j=1}^{q} I(S, T; \Pi^i \mid I = i, J = j, W^{-j}, W_{S,T}) \quad (2) \\
&= I(S, T; \Pi^I \mid I, J, W^{-J}, W_{S,T}) \\
&\geq I(S, T; \Pi^* \mid W_{S,T}, R) \quad (3) \\
&= \Omega(\alpha^2 k) \quad (4)
\end{aligned}$$

where

1. $W_{S,T} \sim \tau^k$ is the random variable used to sample (S, T) from μ_k. Eq. (2) holds because the distribution of W^j is the same as that of $W_{S,T}$, and the conditional distribution of $(X^{j,i}, Y^j, \Pi^i \mid W^{-j}, W^j)$ is the same as $(S, T, \Pi^i \mid I = i, J = j, W^{-j}, W_{S,T})$.

2. In Eq. (3), Π^* is the best protocol for DISJ with one-sided error probability at most $(\delta_1(1 - \alpha/10))$ and R is the public randomness used in Π^*. The information is measured according to μ_k.

3. Eq. (4) holds by Theorem 4. Recall that we have set $p = \alpha/20$.

Therefore, we have $R_{1/4}(\text{DMR}) \geq IC_{\phi,1/4}(\text{DMR}) \geq \Omega(\alpha^2 kn)$, proving our Theorem 1.

4 The Upper Bound

In this section we present a simple α-approximation algorithm for $\alpha \leq 1/2$, which matches the lower bound . The algorithm consists of two steps. In the first step, each site computes a local maximum matching and sends its size to the coordinator. The coordinator compares these sizes, and then sends a message to the site that has the largest local maximum matching. This site then sends the local maximum matching to the coordinator. We can assume that the size of this matching is not larger than αn, as otherwise, the local matching of that site can be declared to be the output of the algorithm, since it is already an α-approximation. Note that the communication cost of this step is at most $O((k + \alpha n) \log n)$ bits.

In the second step, the coordinator picks each site randomly with probability $\alpha' = 4\alpha$, and computes a maximal matching among the sites picked using the straightforward algorithm that we described in the introduction. The communication cost of this step is at most $O((k + \alpha^2 kn) \log n)$ bits in expectation. We now show the correctness of the algorithm.

Let X_i be the random variable indicating the event that the i-th site is picked in the second step, and we have $\mathsf{E}[X_i] = \alpha'$ and $\mathsf{Var}[X_i] = \alpha'(1 - \alpha')$. Let M be the global maximum matching and $m = |M|$. We use m_i to denote the number of edges in M which are incident to the vertices in the i-th site, thus $\sum_i m_i = m$ (recall that we assume edge partitioning where edges are partitioned disjointly across the set of k sites). For the same reason as in the first step, we can again assume that $m_i \leq \alpha' m$ for all $i \in [k]$, since otherwise, we will already get an α-approximation. Let Y be the size of the maximal matching that is obtained in the second step. Recall that a maximal matching is at least $1/2$ of a maximum matching, thus we have $Y \geq \frac{1}{2} \cdot \sum_{i=1}^{k} m_i X_i$. Let $Y' = \sum_{i=1}^{k} m_i X_i$. So we have $\mathsf{E}[Y'] = \alpha' m$ and

$$\mathsf{Var}[Y'] = \alpha'(1 - \alpha') \sum_{i=1}^{k} m_i^2 \leq \alpha' \cdot \alpha' m^2 = \alpha'^2 m^2.$$

The inequality holds since we assume that $m_i \leq \alpha' m$ for all $i \in [k]$. Now, we can apply Chebyshev's inequality to bound the error probability. We have

$$\mathsf{Pr}[|Y' - \alpha' m| \geq \alpha' m/2] \leq 1/4.$$

Therefore, with probability at least $3/4$, it holds $Y \geq 1/2 \cdot Y' \geq 1/2 \cdot \alpha' m/2 = \alpha m$.

Theorem 5 *For $\alpha \leq 1/2$, there exists a randomized algorithm that computes an α-approximation of the maximum matching with probability at least $3/4$ at the communication cost of $O((k + \alpha^2 nk + \alpha n) \log n)$ bits.*

Note that $\Omega(\alpha n)$ is a trivial lower bound, simply because the size of the output could be as large as $\Omega(\alpha n)$. And obviously $\Omega(k)$ is also a lower bound, since the coordinator has to talk to each of the sites at least one. Thus, together with the lower bound $\Omega(\alpha^2 kn)$ in Theorem 1, the upper bound above is tight up to a $\log n$ factor.

5 Concluding Remarks

In this paper we have shown tight bounds on the communication complexity for solving approximate maximum matching problem in the message-passing communication model. An important problem left open after this work is the complexity of the counting version of the problem, i.e., what is the communication complexity if we only require the k sites to compute an approximation of the *size* of a maximum matching, instead of reporting the matching itself? Note that our lower bound proof crucially relies on the fact that the protocol has to return a certificate of the matching. Thus, in order to prove a lower bound for the counting version of the problem, we need new ideas, and it is also possible that a better upper bound exists. Another interesting direction for future research is to investigate other important graph problems, for example, connected components, minimum spanning tree, vertex cover and dominating set. We believe that our technique can possibly to applied to those problems as well (see the discussions at the end of Section 1.3).

Acknowledgments

The authors would like to thank Ke Yi for useful discussions.

References

[1] Question 16: Graph matchings (Andrew McGregor) in open problems in data streams and related topics IITK workshop on algorithms for data streams, 2006. `http://www.cse.iitk.ac.in/users/sganguly/data-stream-probs.pdf`.

[2] K. J. Ahn and S. Guha. Laminar families and metric embeddings: Non-bipartite maximum matching problem in the semi-streaming model. *CoRR*, abs/1104.4058, 2011.

[3] K. J. Ahn and S. Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. In *Proceedings of the 38th international conference on Automata, languages and programming - Volume Part II*, ICALP'11, pages 526–538, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] K. J. Ahn, S. Guha, and A. McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 459–467. SIAM, 2012.

[5] K. J. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st symposium on Principles of Database Systems*, PODS '12, pages 5–14, New York, NY, USA, 2012. ACM.

[6] Apache. Giraph. `http://incubator.apache.org/giraph`, Mar. 2013.

[7] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68:702–732, June 2004.

[8] B. Barak, M. Braverman, X. Chen, and A. Rao. How to compress interactive communication. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 67–76. ACM, 2010.

[9] M. Braverman. Interactive information complexity. In *Proceedings of the 44th symposium on Theory of Computing*, pages 505–524. ACM, 2012.

[10] A. Chakrabarti, Y. Shi, A. Wirth, and A. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 270–278, 2001.

[11] J. Clark. Facebook rides Unicorn to graph search nirvana. The Register, `http://www.theregister.co.uk/2013/03/07/facebook_unicorn_helps_graph_search`, Jan. 2013.

[12] G. Cormode, S. Muthukrishnan, K. Yi, and Q. Zhang. Continuous sampling from distributed streams. *J. ACM*, 59(2):10, 2012.

[13] T. Cover and J. Thomas. *Elements of information theory*. Wiley-interscience, 2006.

[14] L. Epstein, A. Levin, J. Mestre, and D. Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM Journal on Discrete Mathematics*, 25(3):1251–1265, 2011.

[15] D. M. Kane, J. Nelson, and D. P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proc. ACM Symposium on Principles of Database Systems*, 2010.

[16] M. Kapralov. Improved lower bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, 2013.

[17] C. Konrad, F. Magniez, and C. Mathieu. Maximum matching in semi-streaming with few passes. In *APPROX-RANDOM*, pages 231–242, 2012.

[18] E. Kushilevitz and N. Nisan. Communication complexity. 1997.

[19] S. Lattanzi, B. Moseley, S. Suri, and S. Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In *Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures*, SPAA '11, pages 85–94, New York, NY, USA, 2011. ACM.

[20] Z. Lotker, B. Patt-Shamir, and S. Pettie. Improved distributed approximate matching. In *SPAA*, pages 129–136, 2008.

[21] Z. Lotker, B. Patt-Shamir, and A. Rosen. Distributed approximate matching. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, PODC '07, pages 167–174, New York, NY, USA, 2007. ACM.

[22] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5(8):716–727, Apr. 2012.

[23] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. Graphlab: A new framework for parallel machine learning. In *UAI*, pages 340–349, 2010.

[24] G. Malewicz, M. H. Austern, A. Bik, J. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *SIGMOD '10*, pages 135–146, 2010.

[25] A. McGregor. Finding graph matchings in data streams. In *Proceedings of the 8th international workshop on Approximation, Randomization and Combinatorial Optimization Problems, and Proceedings of the 9th international conference on Randomization and Computation: algorithms and techniques*, APPROX'05/RANDOM'05, pages 170–181, Berlin, Heidelberg, 2005. Springer-Verlag.

[26] Neo4j. The world's leading graph database. `http://www.neo4j.org`, Mar. 2013.

[27] J. M. Phillips, E. Verbin, and Q. Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 486–501. SIAM, 2012.

[28] B. Shao, H. Wang, and Y. Li. Trinity: A distributed graph engine on a memory cloud. In *ACM SIGMOD*, 2013.

[29] M. Wattenhofer and R. Wattenhofer. Distributed weighted matching. In R. Guerraoui, editor, *Distributed Computing*, volume 3274 of *Lecture Notes in Computer Science*, pages 335–348. Springer Berlin Heidelberg, 2004.

[30] D. P. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 941–960, New York, NY, USA, 2012. ACM.

[31] M. Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, Feb. 2012.

A Proof of Theorem 3

We will use Π_{ab} to denote the transcript when the input is a, b. By definition,

$$\begin{aligned} I(A,B;\Pi_{AB} \mid W) &= pI(A,0;\Pi_{A0} \mid W=0) + (1-p)I(0,B;\Pi_{0B} \mid W=1) \\ &= pI(A;\Pi_{A0}) + (1-p)I(B;\Pi_{0B}). \end{aligned} \tag{5}$$

In (5) A distributed uniformly in $\{0,1\}$, $\mathsf{Pr}[B=0] = p$ and $\mathsf{Pr}[B=1] = 1-p$. It is proved in [7] that if the U and V are random variables with uniform distribution in $\{0,1\}$, then

$$I(U;\Pi_{U0}) \geq h^2(\Pi_{00},\Pi_{10}),$$

and

$$I(V;\Pi_{0V}) \geq h^2(\Pi_{00},\Pi_{01})$$

where $h(X,Y)$ is the Hellinger distance between two random variables X, Y. However now the distribution of B is not uniform. To bound the second part of (5), we need to use the following lemma, the proof of which can be found in the book [13] (Theorem 2.7.4).

Lemma A.1 *Let $(X,Y) \sim p(x,y) = p(x)p(y|x)$. The mutual information $I(X,Y)$ is a concave function of $p(x)$ for fixed $p(y|x)$.*

In our case, x is B and y is Π_{0B}, and it is easy to see the conditional probability $\mathsf{Pr}[\Pi_{0B} = \pi|B=b]$ is fixed for any π and b. So the mutual information $I(B;\Pi_{0B})$ is a concave function of the distribution of B. Let μ be the uniform distribution in $\{0,1\}$, and υ be the distribution always taking value 1. Here we have $\mathsf{Pr}[B=0] = p$ and $\mathsf{Pr}[B=1] = 1-p$, which can be

expressed as a convex combination of μ and υ as $2p\mu + (1-2p)\upsilon$ (In this paper we always assume $p \leq 1/2$). Then the second part of the mutual information can be bounded

$$I(B;\Pi_{0B}) \geq 2pI_\mu(B;\Pi_{0B}) + (1-2p)I_\upsilon(B;\Pi_{0B}) \geq 2p \cdot h^2(\Pi_{00},\Pi_{01})$$

as mutual information is non-negative. So

$$\begin{aligned} I(A,B;\Pi_{AB} \mid W) &= pI(A;\Pi_{A0} \mid W=0) + (1-p)I(B;\Pi_{0B} \mid W=1) \\ &\geq p \cdot h^2(\Pi_{00},\Pi_{10}) + (1-p) \cdot 2p \cdot h^2(\Pi_{00},\Pi_{01}) \\ &\geq p \cdot (h^2(\Pi_{00},\Pi_{10}) + h^2(\Pi_{00},\Pi_{01})). \end{aligned} \tag{6}$$

We next show that if Π is a protocol with error probability no larger than $(\delta_1 - \beta)$ under distribution μ_1, then

$$h^2(\Pi_{00},\Pi_{10}) + h^2(\Pi_{00},\Pi_{01}) = \Omega(\beta^2/\delta_1^2),$$

from which the theorem follows.

By the triangle inequality,

$$h(\Pi_{00},\Pi_{10}) + h(\Pi_{00},\Pi_{01}) \quad \geq \quad h(\Pi_{01},\Pi_{10}) = h(\Pi_{00},\Pi_{11})$$

The last equality is from the *cut-and-paste* lemma in [7] (Lemma 6.3). Thus

$$\begin{aligned} h(\Pi_{00},\Pi_{10}) + h(\Pi_{00},\Pi_{01}) &\geq 1/2 \cdot (h(\Pi_{00},\Pi_{10}) + h(\Pi_{00},\Pi_{01}) + h(\Pi_{00},\Pi_{11})) \\ &\geq 1/2 \cdot (h(\Pi_{00},\Pi_{10}) + h(\Pi_{00},\Pi_{11})) \\ &\geq 1/2 \cdot h(\Pi_{10},\Pi_{11}). \quad \text{(Triangle inequality)} \end{aligned}$$

Similarly we have,

$$h(\Pi_{00},\Pi_{10}) + h(\Pi_{00},\Pi_{01}) \geq 1/2 \cdot h(\Pi_{01},\Pi_{11}).$$

So for any $a,b,c \in [0,1]$ with $a+b+c=1$,

$$h(\Pi_{00},\Pi_{10}) + h(\Pi_{00},\Pi_{01}) \geq 1/2 \cdot (ah(\Pi_{00},\Pi_{11}) + bh(\Pi_{01},\Pi_{11}) + ch(\Pi_{10},\Pi_{11})) \tag{7}$$

Let $e_{00}, e_{01}, e_{10}, e_{11}$ be the error probability of Π when the input is $(0,0),(0,1),(1,0),(1,1)$ respectively. Recall that $\delta_1 = \mu_1(1,1) \leq 1/2$. By assumption,

$$\begin{aligned} (\delta_1 - \beta) &\geq \mu_1(0,0)e_{00} + \mu_1(1,0)e_{10} + \mu_1(0,1)e_{01} + \delta_1 e_{11} \\ &\geq \delta_1 \left(\frac{(\mu_1(0,0)e_{00} + \mu_1(1,0)e_{10} + \mu_1(0,1)e_{01})}{1-\delta_1} + e_{11} \right) \quad \text{(since } \delta_1 \leq 1/2\text{)} \\ &= \delta_1 \left(\frac{\mu_1(0,0)}{1-\delta_1}(e_{00}+e_{11}) + \frac{\mu_1(0,1)}{1-\delta_1}(e_{01}+e_{11}) + \frac{\mu_1(1,0)}{1-\delta_1}(e_{10}+e_{11}) \right) \\ &= \delta_1(a(e_{00}+e_{11}) + b(e_{01}+e_{11}) + c(e_{10}+e_{11})) \end{aligned} \tag{8}$$

where $a+b+c=1$.

Let $\Pi(x,y)$ be the output of Π when the input is (x,y). Let us analyze the value of $e_{00}+e_{11}$. The other two are similar.

$$\begin{aligned} e_{00}+e_{11} &= \mathsf{Pr}[\Pi(0,0)=1] + \mathsf{Pr}[\Pi(1,1)=0] \\ &= 1 - (\mathsf{Pr}[\Pi(0,0)=0] - \mathsf{Pr}[\Pi(1,1)=0]) \\ &\geq 1 - V(\Pi_{00},\Pi_{11}). \end{aligned}$$

Here $V(X,Y)$ is the total variation distance between X, Y. We also have $e_{01} + e_{11} \geq 1 - V(\Pi_{01}, \Pi_{11})$ and $e_{10} + e_{11} \geq 1 - V(\Pi_{10}, \Pi_{11})$. It is known (see, e.g., [7], Section 6) that

$$V(X,Y) \leq h(X,Y)\sqrt{2 - h^2(X,Y)} \leq \sqrt{2}h(X,Y),$$

Thus by (8) we get

$$a \cdot h(\Pi_{00}, \Pi_{11}) + b \cdot h(\Pi_{10}, \Pi_{11}) + c \cdot h(\Pi_{01}, \Pi_{11}) \quad \geq \quad \beta/(\sqrt{2}\delta_1).$$

It follows from (7) that

$$h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01}) \geq \beta/(2\sqrt{2}\delta_1).$$

So we have

$$\begin{aligned} h^2(\Pi_{00}, \Pi_{10}) + h^2(\Pi_{00}, \Pi_{01}) &\geq 1/2 \cdot (h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01}))^2 \quad \text{(by Cauchy-Schwarz)} \\ &\geq \beta^2/(16\delta_1^2). \end{aligned}$$

Then the first part of the theorem follows from (6).

Next let us consider public coin protocol. Let R denote the public randomness. Let Π_r be the private coin protocol when we fix $R = r$. Recall that $\delta_1 \leq 1/2$ is the probability of $(A,B) = (1,1)$. We assume that the error probability of Π_r is at most δ_1, since otherwise we can just answer $\mathrm{AND}(A,B) = 0$. Let $(\delta_1 - \beta_r)$ be the error probability of Π_r. We have already shown that

$$I(A,B;\Pi_r \mid W) = \Omega({\beta_r}^2 p/{\delta_1}^2).$$

And we also have $\sum_r(\mathsf{Pr}[R=r] \cdot (\delta_1 - \beta_r)) = \delta_1 - \beta$, or

$$\textstyle\sum_r(\mathsf{Pr}[R=r] \cdot \beta_r) = \beta. \tag{9}$$

Thus we have

$$\begin{aligned} I(A,B;\Pi \mid W,R) &= \textstyle\sum_r \mathsf{Pr}[R=r] I(A,B;\Pi_r \mid W, R=r) \\ &\geq \textstyle\sum_r \mathsf{Pr}[R=r] \Omega({\beta_r}^2 pk/{\delta_1}^2) \\ &\geq \Omega(\beta^2 pk/{\delta_1}^2). \end{aligned}$$

The last inequality is due to the Jensen's inequality and (9).

If Π has a one-sided error $\delta_1(1-\beta)$, i.e., it will output 1 with probability β when the input is $(1,1)$, then we can run l instances of the protocol and answer 1 if and only if there exists one instance which outputs 1. Let Π' be this new protocol. The transcript Π' is the concatenation of l instances of Π, that is, $\Pi = \Pi_1 \circ \Pi_2 \circ \ldots \circ \Pi_l$. To make the distributional error smaller than $0.1\delta_1 = \Theta(1)$ under μ_1, it is enough to set $l = O(1/\beta)$. Thus by the first part of this theorem, we have $I(A,B;\Pi' \mid W,R) = \Omega(p)$.

$$\begin{aligned} I(A,B;\Pi' \mid W,R) &= I(A,B;\Pi_1,\Pi_2,\cdots,\Pi_l \mid W,R) \\ &\leq \sum_{i=1}^{l} I(A,B;\Pi_i \mid W,R) \qquad (10) \\ &= l \cdot I(A,B;\Pi \mid W,R), \end{aligned}$$

where (10) follows from the sub-additivity and the fact that $\Pi_1, \Pi_2, \cdots, \Pi_l$ are conditional independent of each other given A, B and W. So $I(A,B;\Pi \mid W,R) \geq \Omega(p) \cdot 1/l = \Omega(\beta p)$.

B Proof of Theorem 4

We first consider the two-sided error case. Consider the following reduction from AND to DISJ. Alice has input u, and Bob has input v. They want to decide the value of $u \wedge v$. They first publicly sample $J \in_R [n]$, and embed u, v in the J-th position, i.e. setting $S[J] = u$ and $T[J] = v$. Then they publicly sample $W[j]$ according to τ for each $j \neq J$. Let $W[-J] = \{W[1], \ldots, W[J-1], W[J+1], \ldots, W[n]\}$. Conditioning on $W[j]$, they further privately sample $(S[j], T[j]) \sim \nu_1$ for each $j \neq J$. Then they run the protocol Π on the input (S, T), and output whatever Π outputs. Let Π' denote this protocol for AND. It easy to see if $(U, V) \sim \mu_1$, the distributional error of Π' is the same as that of Π under input distribution μ_k. The public coins of Π' include J, $W[-J]$ and the public coins R of Π. We first analyze the information cost when (S, T) is distributed according to ν_k.

$$\begin{aligned}
\frac{1}{k} \cdot I(S, T; \Pi \mid W, R) &\geq \frac{1}{k} \cdot \sum_{j=1}^{k} I(S[j], T[j]; \Pi \mid W[j], W[-j], R) \quad \text{(super-additivity)} \\
&= \frac{1}{k} \cdot \sum_{j=1}^{k} I(U, V; \Pi \mid W[j], J = j, W[-j], R) \\
&= I(U, V; \Pi \mid W[J], J, W[-J], R) \\
&= \Omega(\gamma^2 p / {\delta_1}^2). \qquad (11)
\end{aligned}$$

The last equality is from Theorem 3. Thus $I(S, T; \Pi \mid W, R) = \Omega(\gamma^2 pk/{\delta_1}^2)$ when $(S, T) \sim \nu_k$.

Now we consider the information cost when $(S, T) \sim \mu_k$. Recall that to sample from μ_k, we first sample $(S, T) \sim \nu_k$, and then randomly pick $D \in_R [k]$ and set $S[D]$ to 0 or 1 with equal probability. Let $\mathcal{E}$ be the indicator random variable of the event that the last step does not change the value of $S[D]$. Thus $(\mu_k | \mathcal{E} = 1) = \nu_k$, and $\mathsf{Pr}[\mathcal{E} = 1] = \mathsf{Pr}[\mathcal{E} = 0] = 1/2$. We get

$$\begin{aligned}
I_{\mu_k}(S, T; \Pi \mid W, R) &\geq I_{\mu_k}(S, T; \Pi \mid W, R, \mathcal{E}) - H(\mathcal{E}) \\
&= \frac{1}{2} \cdot I_{\mu_k}(S, T; \Pi \mid W, R, \mathcal{E} = 1) + \frac{1}{2} \cdot I_{\mu_k}(S, T; \Pi \mid W, R, \mathcal{E} = 0) - 1 \\
&\geq \frac{1}{2} \cdot I_{\nu_k}(S, T; \Pi \mid W, R) - 1 \\
&= \Omega(\gamma^2 pk/{\delta_1}^2).
\end{aligned}$$

The proof for the one-sided error case is the same, except that we use the one-sided error lower bound $\Omega(\gamma p)$ in theorem 3 to bound (11).

C Other Omitted Proofs

C.1 Proof of Theorem 2

Proof. For any protocol Π, the expected size of its transcript is (we abuse the notation by using Π also for the transcript)

$$\mathsf{E}[|\Pi|] = \sum_{i=1}^{k} \mathsf{E}[|\Pi^i|] \geq \sum_{i=1}^{k} H(\Pi^i) \geq IC_{\mu,\delta}(\Pi).$$

The theorem then follows since the worst-case cost is at least the average. □

C.2 Proof of Lemma 2.1

Proof. It follows directly from the data processing inequality, since Π and Y are conditionally independent given $X^1, \ldots, X^k$. ◻

C.3 Proof of Claim 1

Proof. The proof is similar to Lemma 4 in [30]. By our construction, we have $\mathsf{E}[|U_1^j|] = \delta_1 k$ and $\mathsf{E}[|V_1^j|] = (1-2p+p^2)k$. Similar to the first item we have that with probability $\left(1 - e^{-\Omega(k)}\right)$, $|V_1^j| \geq 0.9 \cdot \mathsf{E}[|V_1^j|] = 0.9 \cdot (1-2p+p^2)k \geq 0.8k$ (recall $p \leq 1/20$) and $|U_1^j| \geq 0.9 \cdot \mathsf{E}[|U_1^j|] \geq 0.4k$. Therefore with probability $\left(1 - e^{-\Omega(k)}\right)$, R must be at least the value R' of the following bin-ball game: We throw each of $0.4k$ balls to one of the $0.8k$ bins uniformly at random, and then count the number of non-empty bins at the end of the process. By Fact 1 and Lemma 1 in [15], we have $\mathsf{E}[R'] = (1-\lambda) \cdot 0.4k$ for some $\lambda \in [0, 1/4]$ and $\mathsf{Var}[R'] < 4(0.4k)^2/(0.8k) = 0.8k$. Thus by Chebyshev's Inequality we have

$$\mathsf{Pr}[R' < \mathsf{E}[R'] - 0.05k] \leq \frac{\mathsf{Var}[R']}{(0.05k)^2} < 320/k.$$

Thus with probability $1 - O(1/k)$, we have $R \geq R' \geq 0.25k$. ◻