Noname manuscript No.
(will be inserted by the editor)

Communication complexity of approximate maximum matching in the message-passing model

Zengfeng Huang · Bozidar Radunovic · Milan Vojnovic · Qin Zhang

Abstract We consider the communication complexity of finding an approximate maximum matching in a graph in a multi-party message-passing communication model. The maximum matching problem is one of the most fundamental graph combinatorial problems, with a variety of applications.

The input to the problem is a graph G that has n vertices and the set of edges partitioned over k sites, and an approximation ratio parameter α. The output is required to be a matching in G that has to be reported by one of the sites, whose size is at least factor α of the size of a maximum matching in G.

We show that the communication complexity of this problem is $\Omega(\alpha^2 kn)$ information bits. This bound is shown to be tight up to a $\log n$ factor, by constructing an algorithm, establishing its correctness, and an upper bound on the communication cost. The lower bound also applies to other graph combinatorial problems in the message-passing communication model, including max-flow and graph sparsification.

Keywords Approximate maximum matching · Multi-party communication complexity · Message passing

1 Introduction

Complex and massive volume data processing requires to scale out to parallel and distributed computation platforms. Scalable distributed computation algorithms are needed that make efficient use of scarce system resources such as communication bandwidth between compute nodes in order to avoid the communication network becoming a bottleneck. A particular interest has been devoted to studying scalable computation methods for graph data, which arises in a variety of applications including online services, online social networks, biological, and economic systems.

In this paper, we consider the distributed computation problem of finding an approximate maximum matching in an input graph whose edges are partitioned over different compute nodes (we refer to as sites). Several performance measures are of interest including the communication complexity in terms of the number of bits or messages, the time complexity in terms of the number of rounds, and the storage complexity in terms of the number of bits. In this paper we focus on the communication complexity. Our main result is a tight lower bound on the communication complexity for approximate maximum matching.

We assume a multi-party message-passing communication model [11,32], we refer to as *message-passing model*, which is defined as follows. The message-passing model consists of $k \geq 2$ sites p^1, p^2, …, p^k. The input is partitioned across k sites, with sites p^1, p^2, …, p^k holding pieces of input data x^1, x^2, …, x^k, respectively.

This paper is a revised and extended version of a paper by the same authors that appeared in the Proceedings of the 32nd Symposium on Theoretical Computer Science (STACS), Munich, Germany, March 4-7, 2015.

Zengfeng Huang
School of Data Science, Fudan University, China
E-mail: huangzf@cse.ust.hk

Bozidar Radunovic
Microsoft Research, Cambridge, United Kingdom
E-mail: bozidar@microsoft.com

Milan Vojnovic
Department of Statistics, London School of Economics (LSE), London, United Kingdom
E-mail: m.vojnovic@lse.ac.uk

Qin Zhang
Computer Science Department, Indiana University, Bloomington, USA
E-mail: qzhangcs@indiana.com

Supported in part by NSF IIS-1633215 and CCF-1844234

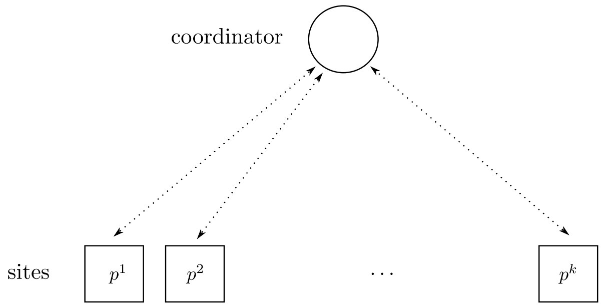

Fig. 1 Coordinator model.

The goal is to design a communication *protocol* for the sites to jointly compute the value of a given function $f : \mathcal{X}^k \to \mathcal{Y}$ at point $(x^1, x^2, \ldots, x^k)$. The sites are allowed to have point-to-point communications between each other. At the end of the computation, at least one site should return the answer. The goal is to find a protocol that minimizes the total communication cost between the sites.

For technical convenience, 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. We call this model *the coordinator model*. See Figure 1 for an illustration. Note that we have essentially replaced the clique communication topology with a star topology, which increases the total communication cost only by a factor of 2 and thus, it does not affect the order of the asymptotic communication complexity.

The *edge partition* of an input graph $G = (V, E)$ over k sites is defined by a partition of the set of edges E in k disjoint sets E^1, E^2, …, E^k, and assigning each set of edges E^i to site p^i. For bipartite graphs with a set of left vertices and a set of right vertices, we define an alternative way of an edge partition, referred to as the *left vertex partition*, as follows: the set of left vertices are partitioned in k disjoints parts, and all the edges incident to one part is assigned to a unique site. Note that left vertex partition is more restrictive, in the sense that any left vertex partition is an instance of an edge partition. Thus, lower bounds hold in this model are stronger as designing algorithms might be easier in this restrictive setting. Our lower bound is proved for left vertex partition model, while our upper bound holds for an arbitrary edge partition of any graph.

1.1 Summary of results

We study the approximate maximum matching problem in the message-passing model which we refer to as Distributed Matching Reporting (DMR) that is defined as follows: given as input is a graph $G = (V, E)$ with $|V| = n$ vertices and a parameter $0 < \alpha \leq 1$; the set of edges E is arbitrarily partitioned into $k \geq 2$ subsets $E^1, E^2, \cdots, E^k$ such that E^i is assigned to site p^i; the coordinator is required to report an α-approximation of the maximum matching in graph G.

In this paper, we show the following main theorem.

Theorem 1 *For every $0 < \alpha \leq 1$ and the number of sites $1 < k \leq n$, any α-approximation randomized algorithm for* DMR *in the message-passing model with the error probability of at most $1/4$ has a communication complexity of $\Omega(\alpha^2 kn)$ bits.*

Moreover, this communication complexity holds for an instance of a bipartite graph.

In this paper we are more interested in the case when $k \gg \log n$, since otherwise the trivial lower bound of $\Omega(n \log n)$ bits (the number of bits to describe a maximum matching) is already near-optimal.

For DMR, a seemingly weaker requirement is that, at the end of the computation, each site p^i outputs a set of edges $M^i \subseteq E^i$ such that $M^1 \cup M^2 \cup \cdots \cup M^k$ is a matching of size that is at least factor α of a maximum matching. However, given such an algorithm, each site might just send M^i to the coordinator after running the algorithm, which will increase the total communication cost by at most an additive term of $n \log n$. Therefore, our lower bound also holds for this setting.

A simple greedy distributed algorithm solves DMR for $\alpha = 1/2$ with the communication cost of $O(kn \log n)$ bits. This algorithm is based on computing a maximal matching in graph G. A maximal matching is a matching whose size cannot be enlarged by adding one or more edges. A maximal matching is computed using a greedy sequential procedure defined as follows. Let $G(E')$ be the graph induced by a subset of edges $E' \subseteq E$. Site p^1 computes a maximal matching M^1 in $G(E^1)$, and sends it to p^2 via the coordinator. Site p^2 then 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 site p^3. This procedure is continued and it is completed once site p^k computed M^k and sent it to the coordinator. Notice that M^k is a maximal matching in graph G, hence it is a 1/2-approximation of a maximum matching in G. The communication cost of this protocol is $O(kn \log n)$ bits because the size of each M^i is at most n edges and each edge's identifier can be encoded with $O(\log n)$ bits. This shows that our lower bound is tight up to a $\log n$ factor. This protocol is essentially sequential and takes $O(k)$ rounds in total. We show that Luby's classic parallel algorithm for maximal matching [29] can be easily adapted to our model with

$O(\log n)$ rounds of computation and $O(kn \log^2 n)$ bits of communication.

In Section 4, we show that our lower bound is also tight with respect to the approximation ratio parameter α for any $0 < \alpha \leq 1/2$ up to a $\log n$ factor. It was shown in [36] that many statistical estimation problems and graph combinatorial problems require $\Omega(kn)$ bits of communication to obtain an *exact* solution. Our lower bound shows that for DMR even computing a constant approximation requires this amount of communication.

The lower bound established in this paper applies also more generally for a broader range of graph combinatorial problems. Since a bipartite maximum matching problem can be found by solving a *max-flow* problem, our lower bound also holds for *approximate max-flow*. Our lower bound also implies a lower bound for the *graph sparsification* problem; see [4] for definition. This is because in our lower bound construction (see Section 3), the bipartite graph under consideration contains many cuts of size $\Theta(1)$ which have to be included in any sparsifier. By our construction, these edges form a good approximate maximum matching, and thus any good sparsifier recovers a good matching. In [4], it was shown that there is a sketch-based $O(1)$-approximate graph sparsification algorithm with the sketch size of $\tilde{O}(n)$ bits, which directly translates to an approximation algorithm of $\tilde{O}(kn)$ communication in our model. Thus, our lower bound is tight up to a poly-logarithmic factor for the graph sparsification problem.

We briefly discuss the main ideas and techniques of our proof of the lower bound for DMR. As a hard instance, we use a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n/2$. Each site p^i holds a set of $r = n/(2k)$ vertices which is a partition of the set of left vertices U. The neighbors of each vertex in U is determined by a two-party set-disjointness instance (DISJ, defined formally in Section 3.2). There are in total $rk = n/2$ DISJ instances, and we want to perform a direct-sum type of argument on these $n/2$ DISJ instances. We show that due to symmetry, the answer of DISJ can be recovered from a reported matching, and then use information complexity to establish the direct-sum theorem. For this purpose, we use a new definition of the information cost of a protocol in the message-passing model.

We believe that our techniques would prove useful to establish the communication complexity for other graph combinatorial problems in the message-passing model. The reason is that for many graph problems whose solution certificates "span" the whole graph (e.g., connected components, vertex cover, dominating set, etc), it is natural that a hard instance would be like for the maximum matching problem, i.e., each of the k sites would hold roughly n/k vertices and the neighbourhood of each vertex would define an independent instance of a two-party communication problem.

1.2 Related work

The problem of finding an approximate maximum matching in a graph has been studied for various computation models, including the streaming computation model [5], MapReduce computation model [21,16], and a traditional distributed computation model known as LOCAL computation model.

In [31], the maximum matching was presented as one of open problems in the streaming computation model. Many results have been established since then by various authors [1], [2], [3], [7], [15], [24], [23], [19], [20], [30], and [37]. Many of the studies were concerned with a streaming computation model that allows for $\tilde{O}(n)$ space; referred to as the semi-streaming computation model. The algorithms developed for the semi-streaming computation model can be directly applied to obtain a constant-factor approximation of maximum matching in a graph in the message-passing model that has a communication cost of $\tilde{O}(kn)$ bits.

For the approximate maximum matching problem in the MapReduce computation model, [26] found an 1/2-approximation algorithm, which requires a constant number of rounds and uses $\tilde{O}(m)$ bits of communication, for any input graph with m edges.

The approximate maximum matching has been studied in the LOCAL computation model by various authors [17,27,28,33]. In this computation model, each processor corresponds to a unique vertex of the graph and edges represent bidirectional communications between processors. The time advances over synchronous rounds. In each round, every processor sends a message to each of its neighbours, and then each processor performs a local computation using as input its local state and the received messages. Notice that in this model, the input graph and the communication topology are the same, while in the message-passing model the communication topology is essentially a complete graph which is different from the input graph and, in general, sites do not correspond to vertices of the topology graph.

A variety of graph and statistical computation problems have been recently studied in the message-passing model [22], [32], [34], [36], [35]. A wide range of graph and statistical problems has been shown to be hard in the sense of requiring $\Omega(kn)$ bits of communication, including *graph connectivity* [32,36], *exact counting of distinct elements* [36], and *k-party set-disjointness* [11]. Some of these problems have been shown to be hard even for random order inputs [22].

In [11], it has been shown that the communication complexity of the *k-party set-disjointness* problem in the message-passing model is $\Omega(kn)$ bits. This work was independent and concurrent to ours. Incidentally, it uses a similar but different input distribution to ours. Similar input distributions were also used in previous work such as [32] and [34]. This is not surprising because of the nature of the message-passing model. There may exist a reduction between the k-party set-disjointness and DMR but showing this is non-trivial and would require a formal proof. The proof of our lower bound is different in that we use a reduction of the k-party DMR to a 2-party set-disjointness using a symmetrisation argument, while [11] uses a coordinative-wise direct-sum theorem to reduce the k-party set-disjointness to a k-party 1-bit problem.

The approximate maximum matching has been recently studied in the coordinator model under additional condition that the sites send messages to the coordinator simultaneously and once, referred to as the simultaneous-communication model. The coordinator then needs to report the output that is computed using as input the received messages. It has been shown in [7] that for the vertex partition model, our lower bound is achievable by a simultaneous protocol for any $\alpha \leq 1/\sqrt{k}$ up to a poly-logarithmic factor.

The communication/round complexity of approximate maximum matching has been studied in the context of finding efficient economic allocations of items to agents, in markets that consist of unit-demand agents in a distributed information model where agents' valuations are unknown to a central planner, which requires communication to determine an efficient allocation. This amounts to studying the communication or round complexity of approximate maximum matching in a bipartite graph that defines preferences of agents over items. In a market with n agents and n items, this amounts to approximate maximum matching in the n-party model with a left vertex partition. [14] and [6] studied this problem in the so called blackboard communication model, where messages sent by agents can be seen by all agents. For one-round protocols, [14] established a tight trade-off between message size and approximation ratio. As indicated by the authors in [14], their randomized lower bound is actually a special case of ours. In a follow-up work, [6] obtained the first non-trivial lower bound on the number of rounds for general randomized protocols.

1.3 Roadmap

In Section 2 we present some basic concepts of probability and information theory, communication and information complexity that are used throughout the paper. Section 3 presents the lower bound and its proof, which is the main result of this paper. Section 4 establishes the tightness of the lower bound up to a poly-logarithmic factor. Finally, in Section 5, we conclude.

2 Preliminaries

2.1 Basic facts and notation

Let $[q]$ denote the set $\{1, 2, \ldots, q\}$, for given integer $q \geq 1$. All logarithms are assumed to have base 2. We use capital letters $X, Y, \ldots$ to denote random variables and the lower case letters $x, y, \ldots$ to denote specific values of respective random variables $X, Y, \ldots$.

We write $X \sim \mu$ to mean that X is a random variable with distribution μ, and $x \sim \mu$ to mean that x is a sample from distribution μ. For a distribution μ on a domain $\mathcal{X} \times \mathcal{Y}$, and $(X, Y) \sim \mu$, we write $\mu(x|y)$ to denote the conditional distribution of X given $Y = y$.

For any given probability distribution μ and positive integer $t \geq 1$, we denote with μ^t the t-fold product distribution of μ, i.e. the distribution of t independent and identically distributed random variables according to distribution μ.

We will use the following distances between two probability distributions μ and ν on a discrete set $\mathcal{X}$: (a) the *total variation distance* defined as

$$d(\mu, \nu) = \frac{1}{2} \sum_{x \in \mathcal{X}} |\mu(x) - \nu(x)| = \max_{S \subseteq \mathcal{X}} |\mu(S) - \nu(S)|$$

and, (b) the *Hellinger distance* defined as

$$h(\mu, \nu) = \sqrt{\frac{1}{2} \sum_{x \in \mathcal{X}} \left(\sqrt{\mu(x)} - \sqrt{\nu(x)}\right)^2}.$$

The total variation distance and Hellinger distance satisfy the following relation:

Lemma 1 *For any two probability distributions μ and ν, the total variation distance and the Hellinger distance between μ and ν satisfy*

$$d(\mu, \nu) \leq \sqrt{2} h(\mu, \nu).$$

With a slight abuse of notation for two random variables $X \sim \mu$ and $Y \sim \nu$, we write $d(X, Y)$ and $h(X, Y)$ in lieu of $d(\mu, \nu)$ and $h(\mu, \nu)$, respectively.

We will use the the following two well-known inequalities.

Hoeffding's inequality Let X be the sum of $t \geq 1$ independent and identically distributed random variables that take values in $[0,1]$. Then, for any $s \geq 0$,

$$\mathbf{Pr}[X - \mathbf{E}[X] \geq s] \leq e^{-2s^2/t}.$$

Chebyshev's inequality Let X be a random variable with variance $\sigma^2 > 0$. Then, for any $s > 0$,

$$\mathbf{Pr}[|X - \mathbf{E}[X]| \geq s] \leq \frac{\sigma^2}{s^2}.$$

2.2 Information theory

For two random variables X and Y, let $H(X)$ denote the *Shannon entropy* of the random variable X, and let $H(X|Y) = \mathbf{E}_y[H(X|Y=y)]$ 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 let $I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$ denote the *conditional mutual information* given Z. The mutual information between any X and Y is non negative, i.e. $I(X;Y) \geq 0$, or equivalently, $H(X|Y) \leq H(X)$.

We will use the following relations from information theory:

Chain rule for mutual information For any jointly distributed random variables $X^1, X^2, \ldots, X^t$, Y and Z,

$$I(X^1, X^2, \ldots, X^t; Y|Z) = \sum_{i=1}^{t} I(X_i; Y|X_1, \ldots, X_{i-1}, Z).$$

Data processing inequality If X and Z are conditionally independent random variables given Y, then

$$I(X;Y|Z) \leq I(X;Y) \text{ and } I(X;Z) \leq I(X;Y).$$

Super-additivity of mutual information If $X^1, X^2, \ldots, X^t$ are independent random variables, then

$$I(X^1, X^2, \ldots, X^t; Y) \geq \sum_{i=1}^{t} I(X^i; Y).$$

Sub-additivity of mutual information If $X^1, X^2, \ldots, X^t$ are conditionally independent random variables given Y, then

$$I(X^1, X^2, \ldots, X^t; Y) \leq \sum_{i=1}^{t} I(X^i; Y).$$

We will use the follow concavity property of mutual information, whose proof can be found in the book [13] (Theorem 2.7.4).

Lemma 2 *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 $p(y|x)$.*

2.3 Communication complexity

In the two party communication complexity model two players, Alice and Bob, are required to jointly compute a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Alice is given $x \in \mathcal{X}$ and Bob is given $y \in \mathcal{Y}$, and they want to jointly compute the value of $f(x,y)$ 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 that Π is a *γ-error protocol* if for every input (x,y), the probability that $\Pi(x,y) \neq f(x,y)$ is not larger than γ, where the probability is over the randomness used in Π. We will refer to this type of error as *worst-case error*. An alternative and weaker type of error is the *distributional error*, which is defined analogously for an input distribution, and where the error probability is over both the randomness used in the protocol and the input distribution.

Let $|\Pi_{xy}|$ denote the length of the transcript in information bits. The communication cost of Π is

$$\max_{x,y,r} |\Pi^r_{xy}|,$$

where r is the randomness used in Π. The *γ-error randomized communication complexity* of f, denoted by $R_\gamma(f)$, is the minimal cost of any γ-error protocol for f.

The multi-party communication complexity model is a natural generalization to $k \geq 2$ parties, where each party has a part of the input, and the parties are required to jointly compute a function $f : \mathcal{X}^k \to \mathcal{Z}$ by exchanging messages according to a randomized protocol.

For more information about communication complexity, we refer the reader to [25].

2.4 Information complexity

The *communication complexity* quantifies the number of bits that need to be exchanged by two or more players in order to compute some function together, while the *information complexity* quantifies the amount of information of the inputs that must be revealed by the protocol. The information complexity has been extensively studied in the last decade, e.g., [12,8,9,34,10]. There are several definitions of information complexity. In this paper, we follow the definition used in [8]. In the two-party case, let μ be a distribution on $\mathcal{X} \times \mathcal{Y}$, we define the information cost of Π measured under μ as

$$IC_\mu(\Pi) = I(X, Y; \Pi_{XY} | R)$$

where $(X, Y) \sim \mu$ and R is the public randomness used in Π. For notational convenience, we will omit the subscript of Π_{XY} and simply use $I(X, Y; \Pi|R)$ to denote the information cost of Π. It should be clear that $IC_\mu(\Pi)$ is a function of μ for a fixed protocol Π. Intuitively, this measures how much information of X and Y is revealed by the transcript Π_{XY}. For any function f, we define the information complexity of f parametrized by μ and γ as

$$IC_{\mu,\gamma}(f) = \min_{\gamma\text{-error } \Pi} IC_\mu(\Pi).$$

Remark. For a public coin protocol, we implicitly allow it to use private randomness unless otherwise specified.

2.5 Information complexity and coordinator model

We can indeed extend the above definition of information complexity to the k-party coordinator model. That is, let X^i be the input of player i with $(X^1, \ldots, X^k) \sim \mu$ and Π be the whole transcript, then we could define $IC_\mu(\Pi) = I(X^1, X^2, \ldots, X^k; \Pi|R)$. However, such a definition does not fully explore the point-to-point communication feature of the coordinator model. Indeed, the lower bound we can prove using such a definition is at most what we can prove under the blackboard model and our problem admits a simple algorithm with communication $O(n \log n + k)$ in the blackboard model. In this paper we give a new definition of information complexity for the coordinator model, which allows us to prove higher lower bounds compared with the simple generalization. Let Π^i be the transcript between player i and the coordinator, thus $\Pi = \Pi^1 \circ \Pi^2 \circ \ldots \circ \Pi^k$. We define the information cost for a function f with respect to input distribution μ and the error parameter $\gamma \in [0, 1]$ in the coordinator model as

$$IC_{\mu,\gamma}(f) = \min_{\gamma\text{-error}\Pi} \sum_{i=1}^{k} I(X^1, X^2, \cdots, X^k; \Pi^i).$$

The next theorem is an extension of a similar result from [8] to the multi-party setting.

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

Proof For any protocol Π, the expected size of its transcript is (we abuse the notation by using Π also for the transcript) $\mathbf{E}[|\Pi|] = \sum_{i=1}^{k} \mathbf{E}[|\Pi^i|] \geq \sum_{i=1}^{k} H(\Pi^i) \geq IC_{\mu,\gamma}(\Pi)$. The theorem then follows because the worst-case communication cost is at least the average-case communication cost.

Lemma 3 *If Y is independent of the random coins used by the protocol Π, then*

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

Proof The statement directly follows from the data processing inequality because given $X^1, X^2, \ldots, X^k$, Π is fully determined by the random coins used, and is thus independent of Y.

3 Lower Bound

The lower bound is established by constructing a hard distribution for the input bipartite graph $G = (U, V, E)$ such that $|U| = |V| = n/2$.

We first discuss the special case when the number of sites k is equal to $n/2$, and each site is assigned one unique vertex in U together with all its adjacent edges. We later discuss the general case.

A natural approach to approximately compute a maximum matching in a graph is to randomly sample a few edges from each site, and hope that we can find a good matching using these edges. To rule out such strategies, we construct random edges as follows.

We create a large number of *noisy edges* by randomly picking a small set of nodes $V_0 \subseteq V$ of size roughly $\alpha n/10$ and connect each node in U to each node in V_0 independently at random with a constant probability. Note that there are $\Theta(\alpha n^2)$ such edges and the size of any matching that can be formed by these edges is at most $\alpha n/10$, which we will show to be asymptotically $\frac{\alpha}{2}$OPT, where OPT is the size of a maximum matching.

We next create a set of *important edges* between U and $V_1 = V \setminus V_0$ such that each node in U is adjacent to at most one random node in V_1. These edges are important in the sense that although there are only $\Theta(|U|) = \Theta(n)$ of them, the size of a maximum matching they can form is large, of the order OPT. Therefore, to compute a matching of size at least αOPT, it is necessary to find and include $\Theta(\alpha\text{OPT}) = \Theta(\alpha n)$ important edges.

We then show that finding an important edge is in some sense equivalent to solving a set-disjointness (DISJ) instance, and thus, we have to solve $\Theta(n)$ DISJ instances. The concrete implementation of this intuition is via an embedding argument.

In the general case, we create $n/(2k)$ independent copies of the above random bipartite graph, each with $2k$ vertices, and assign $n/(2k)$ vertices to each site (one from each copy). We then prove a direct-sum theorem using information complexity.

In the following, we introduce the two-party AND problem and the two-party DISJ problem. These two problems have been widely studied and tight bounds are known (e.g. [8]). For our purpose, we need to prove stronger lower bounds for them. We then give a reduction from DISJ to DMR and prove an information cost lower bound for DMR in Section 3.3.

3.1 The two-party AND problem

In the two-party AND communication problem, Alice and Bob hold bits a and b respectively, and they want to compute the value of the function $\text{AND}(a, b) = a \wedge b$.

Next we define input distributions for this problem. Let A, B be random variables corresponding to the inputs of Alice and Bob respectively. Let $p \in (0, 1/2]$ be a parameter. Let τ_q denote the probability distribution of a Bernoulli random variable that takes value 0 with probability q or value 1 with probability $1-q$. We define two input probability distributions ν and μ for (A, B) as follows.

ν: Sample $w \sim \tau_p$, and then set the value of (a, b) as follows: if $w = 0$, let $a \sim \tau_{1/2}$ and $b = 0$; otherwise, if $w = 1$, let $a = 0$, and $b \sim \tau_p$. Thus, we have

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

μ: Sample $w \sim \tau_p$, and then choose (a, b) as above (i.e. sample (a, b) according to ν). Then, reset the value of a to be 0 or 1 with equal probability (i.e. set $a \sim \tau_{1/2}$).

Here w is an auxiliary random variable to break the dependence of A and B, as we can see A and B are not independent, but conditionally independent given w.

Definition 1 We use δ to denote the probability that $(A, B) = (1, 1)$ under distribution μ, which is $(1-p)^2/2$.

For the special case $p = 1/2$, by [8], it is shown that, for any *private* coin protocol Π with worst-case error probability γ, the information cost $I(A, B; \Pi | W) = \Omega(1 - 2\sqrt{\gamma})$ for $\gamma \le 1/4$, where the information cost is measured with respect to ν and W is the random variable corresponding to w. If we write $\gamma = 1/2 - \beta$ for $\beta > 1/4$, then for any private coin protocol Π with worst-case error probability $1/2 - \beta$, the information cost

$$I(A, B; \Pi | W) = \Omega(1 - 2\sqrt{0.5 - \beta}) = \Omega(\beta^2).$$

This is because $\sqrt{0.5 - \beta} \le 0.5 - c\beta^2$ for some constant c as long as β is strictly larger than 1/4. Note that the above mutual information is different from the definition of information cost; it is referred to as *conditional information cost* in [8]. It is smaller than the standard information cost by the data processing inequality (Π and W are conditionally independent given A, B). For a fixed randomized protocol Π, the value of the conditional mutual information $I(A, B; \Pi | W)$ is determined once the joint distribution of (A, B, W) is given. Therefore, when we say the (conditional) information cost is measured w.r.t. ν, it means that the mutual information, $I(A, B; \Pi | W)$, is calculated for $(A, B, W) \sim \nu$.

The above lower bound might seem counterintuitive, as the answer to AND is always 0 under the input distribution ν and a protocol can just output 0 which does not reveal any information. However, such a protocol will have worst-case error probability 1, i.e., it is always wrong when the input is $(1, 1)$, contradicting the assumption. When distributional error is considered, the (distributional) error and information cost can be measured w.r.t. different input distributions. In our case, the error will be measured under μ and the information cost will be measured under ν, and we will prove that any protocol having small error under μ must incur high information cost under ν.

We next derive an extension that generalizes the result of [8] to any $p \in (0, 1/2]$ and distributional errors. We will also use the definition of *one-sided error.*

Definition 2 For a two-party binary function $f(x, y)$, we say that a protocol has a *one-sided error* γ for f under a distribution if it is always correct when the correct answer is 0, and is correct with probability at least $1 - \gamma$ conditional on $f(x, y) = 1$.

Recall that δ is the probability that $(A, B) = (1, 1)$ when $(A, B) \sim \mu$, which is $(1-p)^2/2$ (see Definition 1). Recall that $p \in (0, 1/2]$, and thus $\delta < 1/2$. Note that a distributional error of δ under μ is trivial, as a protocol that always outputs 0 achieves this (but it has one-sided error 1). Therefore, for two-sided error, we will consider protocols with error probability slightly better than the trivial protocol, i.e., with error probability $\delta - \beta$ for some $\beta \le \delta$.

Theorem 3 *Suppose that Π is a public coin protocol for* AND *which has distributional error $\delta - \beta$, for $\beta \in (0, \delta)$, under input distribution μ; let R denote its public randomness. Then*

$$I(A, B; \Pi | W, R) = \Omega(p(\beta/\delta)^2)$$

where the information is measured with respect to ν.

If Π has a one-sided error $1 - \beta$, then

$$I(A, B; \Pi | W, R) = \Omega(p\beta).$$

If we set $p = 1/2$, the first part of Theorem 3 recovers the result of [8].

Proof (of Theorem 3) We first prove that theorem for *private* coin protocols. Let Π_{ab} denote the transcript when the input is a, b. By definition,

$$\begin{aligned}&I(A, B; \Pi_{AB}|W)\\ &= p \cdot I(A, 0; \Pi_{A0}|W = 0) +\\ &\quad (1 - p) \cdot I(0, B; \Pi_{0B}|W = 1)\\ &= p \cdot I(A; \Pi_{A0}) + (1 - p) \cdot I(B; \Pi_{0B}). \qquad (1)\end{aligned}$$

With a slight abuse of notation, in (1), A and B are random variables with distributions $\tau_{1/2}$ and τ_p, respectively.

For any random variable U with distribution $\tau_{1/2}$, the following two inequalities were established in [8]:

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

and

$$I(U; \Pi_{0U}) \geq h^2(\Pi_{00}, \Pi_{01}) \qquad (3)$$

where $h(X, Y)$ is the Hellinger distance between two random variables X and Y.

We can apply these bounds to lower bound the term $I(A; \Pi_{A0})$. However, we cannot apply them to lower bound the term $I(B; \Pi_{0B})$ when $p < 1/2$ because then the distribution of B is not $\tau_{1/2}$. To lower bound the term $I(B; \Pi_{0B})$, we will use Lemma 2, which claims that the mutual information $I(B; \Pi_{0B})$ is a concave function of the distribution τ_p of B, since the distribution of Π_{0B} is fixed given B.

Recall that τ_p is the probability distribution that takes value 0 with probability p and takes value 1 with probability $1 - p$. Note that τ_p can be expressed as a convex combination of $\tau_{1/2}$ and τ_0 (always taking value 1) as follows: $\tau_p = 2p\tau_{1/2} + (1 - 2p)\tau_0$. (Recall that p is assumed to be smaller than 1/2.) Let $B_0 \sim \tau_{1/2}$ and $B_1 \sim \tau_0$. Then, using Lemma 2, we have

$$\begin{aligned}I(B; \Pi_{0B}) &\geq 2p \cdot I(B_0; \Pi_{0B_0}) + (1 - 2p) \cdot I(B_1; \Pi_{0B_1})\\ &\geq 2p \cdot h^2(\Pi_{00}, \Pi_{01})\end{aligned}$$

where the last inequality holds by (3) and non-negativity of mutual information.

Thus, we have

$$\begin{aligned}&I(A, B; \Pi_{AB}|W)\\ &= p \cdot I(A; \Pi_{A0}) + (1 - p) \cdot I(B; \Pi_{0B})\\ &\geq p \cdot h^2(\Pi_{00}, \Pi_{10}) + (1 - p)2p \cdot h^2(\Pi_{00}, \Pi_{01})\\ &\geq p \cdot (h^2(\Pi_{00}, \Pi_{10}) + h^2(\Pi_{00}, \Pi_{01})) \qquad (4)\end{aligned}$$

where the last inequality holds because $p \leq 1/2$.

We next show that if Π is a protocol with error probability smaller than or equal to $\delta - \beta$ under distribution μ, then

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

which together with other above relations implies the first part of the theorem.

By the triangle inequality,

$$\begin{aligned}h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01}) &\geq h(\Pi_{01}, \Pi_{10})\\ &= h(\Pi_{00}, \Pi_{11}) \qquad (5)\end{aligned}$$

where the last equality is from the *cut-and-paste lemma* in [8] (Lemma 6.3).

Thus, we have

$$\begin{aligned}&h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01})\\ &\geq \frac{1}{2}h(\Pi_{00}, \Pi_{10}) + \frac{1}{2}(h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01}))\\ &\geq \frac{1}{2}(h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{11}))\\ &\geq \frac{1}{2}h(\Pi_{10}, \Pi_{11}) \qquad (6)\end{aligned}$$

where the last inequality is by the triangle inequality.

Similarly, it holds that

$$h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01}) \geq \frac{1}{2}h(\Pi_{01}, \Pi_{11}). \qquad (7)$$

From (5), (6) and (7), for any positive real numbers a, b, and c such that $a + b + c = 1$, we have

$$\begin{aligned}&h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01})\\ &\geq \frac{1}{2}(a \cdot h(\Pi_{00}, \Pi_{11}) + b \cdot h(\Pi_{01}, \Pi_{11})\\ &\quad + c \cdot h(\Pi_{10}, \Pi_{11})). \qquad (8)\end{aligned}$$

Let p^e denote the distributional error probability of Π over μ and p^e_{xy} denote the error probability of Π conditioned on that the input is (x, y). Recall $\delta = \mu(1, 1) \leq 1/2$. We have

$$\begin{aligned}p^e &= \mu(0,0)p^e_{00} + \mu(1,0)p^e_{10} + \mu(0,1)p^e_{01} + \delta p^e_{11}\\ &\geq \delta\left(\frac{\mu(0,0)p^e_{00} + \mu(1,0)p^e_{10} + \mu(0,1)p^e_{01}}{1 - \delta} + p^e_{11}\right)\\ &= \delta(a^*(p^e_{00} + p^e_{11}) + b^*(p^e_{01} + p^e_{11})\\ &\quad + c^*(p^e_{10} + p^e_{11})) \qquad (9)\end{aligned}$$

where

$$a^* = \frac{\mu(0,0)}{1 - \delta}, b^* = \frac{\mu(0,1)}{1 - \delta} \text{ and } c^* = \frac{\mu(1,0)}{1 - \delta},$$

and clearly $a^* + b^* + c^* = 1$. Let $\Pi(x, y)$ be the output of Π when the input is (x, y), which is also a random variable. Note that

$$\begin{aligned}p^e_{00} + p^e_{11} &= \mathbf{Pr}[\Pi(0,0) = 1] + \mathbf{Pr}[\Pi(1,1) = 0]\\ &= 1 - (\mathbf{Pr}[\Pi(0,0) = 0] - \mathbf{Pr}[\Pi(1,1) = 0])\\ &\geq 1 - d(\Pi_{00}, \Pi_{11}) \qquad (10)\end{aligned}$$

where $d(X, Y)$ denote the total variation distance between probability distributions of random variables X and Y. Using Lemma 1, we have

$$p^e_{00} + p^e_{11} \geq 1 - \sqrt{2}h(\Pi_{00}, \Pi_{11}). \tag{11}$$

By the same arguments, we also have

$$p^e_{01} + p^e_{11} \geq 1 - \sqrt{2}h(\Pi_{01}, \Pi_{11}) \tag{12}$$

and

$$p^e_{10} + p^e_{11} \geq 1 - \sqrt{2}h(\Pi_{10}, \Pi_{11}). \tag{13}$$

Combining (11), (12) and (13) with (9) and the assumption that $p^e \leq \delta - \beta$, we obtain

$$a^* h(\Pi_{00}, \Pi_{11}) + b^* h(\Pi_{10}, \Pi_{11}) + c^* h(\Pi_{01}, \Pi_{11}) \geq \frac{\beta}{\sqrt{2}\delta}.$$

By (8), we have

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

From the Cauchy-Schwartz inequality, it follows

$$\begin{aligned} & h^2(\Pi_{00}, \Pi_{10}) + h^2(\Pi_{00}, \Pi_{01}) \\ \geq & \frac{1}{2}(h(\Pi_{00}, \Pi_{10}) + h(\Pi_{00}, \Pi_{01}))^2. \end{aligned}$$

Hence, we have

$$h^2(\Pi_{00}, \Pi_{10}) + h^2(\Pi_{00}, \Pi_{01}) \geq \frac{\beta^2}{16\delta^2}$$

which combined with (4) establishes the first part of the theorem for private coin protocols.

Public coin protocols. Let R denote the public randomness. Let Π_r be the private coin protocol when we fix $R = r$. Recall that $\delta \leq 1/2$ is the probability that $(A, B) = (1, 1)$. We assume that the error probability of Π_r is at most δ, since otherwise we can just answer AND$(A, B) = 0$. Let $(\delta - \beta_r)$ be the error probability of Π_r. We have already shown that

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

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

$$\textstyle\sum_r (\mathbf{Pr}[R = r] \cdot \beta_r) = \beta. \tag{14}$$

Thus we have

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

The last inequality is due to the Jensen's inequality (since $f(x) = x^2$ is a convex function) and (14).

We now go on to prove the second part of the theorem. Assume Π has a one-sided error $1 - \beta$, i.e., it outputs 1 with probability at least β if the input is $(1, 1)$, and always outputs correctly otherwise. To boost the success probability, we can run m parallel instances of the protocol and answer 1 if and only if there exists one instance which outputs 1. Let Π' be this new protocol, and it is easy to see that it has a one-sided error of $(1 - \beta)^m$. By setting $m = O(1/\beta)$, it is at most $1/10$, and thus the (two-sided) distributional error of Π' under μ is smaller than $\delta/10$. By the first part of the theorem, we know $I(A, B; \Pi' | W, R) = \Omega(p)$. We also have

$$\begin{aligned} I(A, B; \Pi' | W, R) &= I(A, B; \Pi_1, \Pi_2, \ldots, \Pi_m | W, R) \\ &\leq \sum_{i=1}^{m} I(A, B; \Pi_i | W, R) \\ &= mI(A, B; \Pi | W, R), \end{aligned}$$

where the inequality follows from the sub-additivity and the fact that $\Pi_1, \Pi_2, \ldots, \Pi_m$ are conditionally independent of each other given A, B and W. Thus, we have $I(A, B; \Pi | W, R) \geq \Omega(p/m) = \Omega(p\beta)$. □

3.2 The two-party DISJ communication problem

The two-party DISJ communication problem with two players, Alice and Bob, who hold strings of bits $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y, \ldots, y_k)$, respectively, and they want to compute

$$\text{DISJ}(x, y) = \text{AND}(x_1, y_1) \vee \cdots \vee \text{AND}(x_k, y_k).$$

By interpreting x and y as indicator vectors that specify subsets of $[k]$, DISJ$(x, y) = 0$ if and only if the two sets represented by x and y are disjoint. Note that this accommodates the AND problem as a special case when $k = 1$.

Let $A = (a_1, a_2, \ldots, a_k)$ be Alice's input and $B = (b_1, b_2, \ldots, b_k)$ be Bob's input. We define two input distributions ν_k and μ_k for (A, B) as follows.

ν_k: For each $i \in [k]$, independently sample $(a_i, b_i) \sim \nu$, and let w_i be the corresponding auxiliary random variable (see the definition of ν). Define $w = (w_1, w_2, \cdots, w_k)$.

μ_k: Let $(a, b) \sim \nu_k$, then pick d uniformly at random from $[k]$, and reset a_d to be 0 or 1 with equal probability. Note that $(a_d, b_d) \sim \mu$, and the probability that DISJ$(A, B) = 1$ is equal to δ.

We will use $\mu_k(a)$ and $\mu_k(b)$ to denote the marginal distribution of a and b respectively under μ_k. Similarly we use $\mu_k(a|b)$ to denote the conditional distribution of a given the value of b.

We define the one-sided error for DISJ similarly: A protocol has a one-sided error δ for DISJ if it is always correct when DISJ$(x, y) = 0$, and is correct with probability at least $1 - \delta$ when DISJ$(x, y) = 1$.

Theorem 4 *Let Π be any public coin protocol for* DISJ *with error probability $\delta - \beta$ on input distribution μ_k, where $\beta \in (0, \delta)$, and let R denote the public randomness used by the protocol. Then*

$$I(A, B; \Pi | W, R) = \Omega(kp(\beta/\delta)^2)$$

where the information is measured w.r.t. μ_k.

If Π has a one-sided error $1 - \beta$, then

$$I(A, B; \Pi | W, R) = \Omega(kp\beta).$$

Proof We first consider the two-sided error case. Let Π be a protocol for DISJ with distributional error $\delta - \beta$ under μ_k. 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 $j \in [k]$, and embed u, v in the j-th position, i.e. set $a_j = u$ and $b_j = v$. Then they publicly sample $w_{j'}$ according to τ_p for all $j' \neq j$. Let $w_{-j} = (w_1, \ldots, w_{j-1}, w_{j+1}, \ldots, w_k)$. Conditional on $w_{j'}$, they sample $(a_{j'}, b_{j'})$ such that $(a_j, b_j) \sim \nu$ for each $j' \neq j$. Note that this step can be done using only private randomness, since, in the definition of ν, $a_{j'}$ and $b_{j'}$ are independent given $w_{j'}$. Then they run the protocol Π on the input (a, b) and output whatever Π outputs. Let Π' denote this protocol for AND. Let U, V, A, B, W, J be the corresponding random variables of u, v, a, b, w, j respectively. It is easy to see that if $(U, V) \sim \mu$, then $(A, B) \sim \mu_k$, and thus the distributional error of Π' is $\delta - \beta$ under μ. The public coins used in Π' include J, W_{-J} and the public coins R of Π.

We first analyze the information cost of Π' under $(A, B) \sim \nu_k$. We have

$$\frac{1}{k} I(A, B; \Pi | W, R)$$

$$\geq \frac{1}{k} \sum_{j=1}^{k} I(A_j, B_j; \Pi | W_j, W_{-j}, R) \tag{15}$$

$$= \frac{1}{k} \sum_{j=1}^{k} I_\nu(U, V; \Pi' | W_j, J = j, W_{-j}, R) \tag{16}$$

$$= I(U, V; \Pi' | W_J, J, W_{-J}, R) \tag{17}$$

$$= \Omega(p(\beta/\delta)^2) \tag{18}$$

where (15) is by the supper-additivity of mutual information, (16) holds because when $(U, V) \sim \nu$ the conditional distribution of $(U, V, \Pi, W_j, W_{-j}, R)$ given $J = j$ is the same as the distribution of $(A_j, B_j, \Pi, W_j, W_{-j}, R)$, (17) is by the definition of conditional mutual information and the fact that J is uniformly sampled from $[k]$, and (18) follows from Theorem 3 using the fact that Π' has error $\delta - \beta$ under μ.

We have established that when $(A, B) \sim \nu_k$, it holds

$$I(A, B; \Pi | W, R) = \Omega(kp(\beta/\delta)^2). \tag{19}$$

We now consider the information cost when $(A, B) \sim \mu_k$. Recall that to sample from μ_k, we first sample $(a, b) \sim \nu_k$, and then pick d uniformly at random from $[k]$ and reset a_d to 0 or 1 with equal probability. Let ξ be the indicator random variable of the event that the last step does not change the value of a_d.

We note that for any jointly distributed random variables X, Y, Z and W,

$$I(X; Y | Z) \geq I(X; Y | Z, W) - H(W). \tag{20}$$

To see this note that by the chain rule for mutual information, we have

$$I(X, W; Y | Z) = I(X; Y | Z) + I(W; Y | X, Z)$$

and

$$I(X, W; Y | Z) = I(W; Y | Z) + I(X; Y | W, Z).$$

Combining the above two equalities, (20) follows by the facts $I(W; Y | X, Z) \geq 0$ and $I(W; Y | Z) \leq H(W | Z) \leq H(W)$.

Let $(A, B) \sim \mu_k$ and $(A', B') \sim \nu_k$. We have

$$\begin{aligned} I(A, B; \Pi | W, R) &\geq I(A, B; \Pi | W, R, \xi) - H(\xi) \\ &= \frac{1}{2} I(A, B; \Pi | W, R, \xi = 1) \\ &\quad + \frac{1}{2} I(A, B; \Pi | W, R, \xi = 0) - 1 \\ &\geq \frac{1}{2} I(A', B'; \Pi | W, R) - 1 \\ &= \Omega(kp(\beta/\delta)^2) \end{aligned}$$

where the first inequality is from (20) and the last equality is by (19).

The proof for the one-sided error case is the same, except that we use the one-sided error lower bound $\Omega(p\beta)$ in Theorem 3 to bound (18). □

3.3 Proof of Theorem 1

Here we provide a proof of Theorem 1. The proof is based on a reduction of DISJ to DMR. We first define the hard input distribution that we use for DMR.

The input graph G is assumed to be a random bipartite graph that consists of $r = n/(2k)$ disjoint, independent and identically distributed random bipartite graphs G^1, G^2, …, G^r. Each bipartite graph $G^j = (U^j, V^j, E^j)$ has the set $U^j = \{u^{j,i} : i \in [k]\}$ of left vertices and the set $V^j = \{v^{j,i} : i \in [k]\}$ of right vertices, both of cardinality k. The sets of edges E^1, E^2, …, E^r are defined by a random variable X that takes values in $\{0, 1\}^{r \times k \times k}$ such that whether or not $(u^{j,i}, v^{j,l})$ is an edge in E^j is indicated by $X_l^{j,i}$.

The distribution of X is defined as follows. Let Y^1, Y^2, …, Y^r be independent and identically distributed random variables with distribution $\mu_k(b)$.[1]. Then, for each $j \in [r]$, conditioned on $Y^j = y^j$, let $X^{j,1}, X^{j,2}, \ldots, X^{j,k}$ be independent and identically distributed random variables with distribution $\mu_k(a|y^j)$, where $\mu_k(a|y^j)$ is the conditional distribution of a given $b = y^j$. Note that for every $j \in [r]$ and $i \in [k]$, $(X^{j,i}, Y^j) \sim \mu_k$.

We will use the following notation:

$$X^i = (X^{1,i}, X^{2,i}, \ldots, X^{r,i}) \text{ for } i \in [k],$$

and

$$X = (X^1, X^2, \ldots, X^k),$$

where each $X^{j,i} \in \{0,1\}^k$, and $X_l^{j,i}$ is the lth bit. In addition, we will also use the following notation:

$$X^{-i} = (X^1, \ldots, X^{i-1}, X^{i+1}, \ldots, X^k), \text{ for } i \in [k]$$

and

$$Y = (Y^1, Y^2, \ldots, Y^r).$$

Note that X is the input to DMR, and Y is *not* part of the input for DMR, but it is used to construct X.

The edge partition of input graph G over k sites p^1, p^2, …, p^k is defined by assigning all edges incident to vertices $u^{1,i}$, $u^{2,i}$,…, $u^{r,i}$ to site p^i, or equivalently p^i gets X^i. See Figure 2 for an illustration.

Input Reduction Let $a \in \{0,1\}^k$ be Alice's input and $b \in \{0,1\}^k$ be Bob's input for DISJ. We will first construct an input of DMR from (a, b), which has the above hard distribution. In this reduction, in each bipartite graph G^j, we carefully embed k instances of DISJ. The output of a DISJ instance determines whether or not a specific edge in the graph exists. This amounts to a total of $kr = n/2$ DISJ instances embedded in graph G. The original input of Alice and Bob is embedded at a random position, and the other $n/2 - 1$ instances are sampled by Alice and Bob using public and private random coins. We then argue that if the original DISJ instance is solved, then with a sufficiently large probability, at least $\Omega(n)$ of the embedded DISJ instances are solved. Intuitively, if a protocol solves an DISJ instance at a random position with high probability, then it should solve many instances at other positions as well, since the input distribution is completely symmetric. We will see that the original DISJ instance can be solved by using any protocol solving DMR, the correctness of which also relies on the symmetric property.

Alice and Bob construct an input X for DMR as follows:

1. Alice and Bob use public coins to sample an index I from a uniform distribution on $[k]$. Alice constructs the input X^I for site p^I, and Bob constructs input X^{-I} for other sites (see Figure 3).
2. Alice and Bob use public coins to sample an index J from a uniform distribution on $[r]$.
3. G^J is sampled as follows: Alice sets $X^{J,I} = a$, and Bob sets $Y^J = b$. Bob privately samples
$$(X^{J,1}, \ldots, X^{J,I-1}, X^{J,I+1}, \ldots, X^{J,k}) \sim \mu_k(a|Y^J)^{k-1}.$$
4. For each $j \in [r] \setminus \{J\}$, G^j is sampled as follows:
   (a) Alice and Bob use public coins to sample $W^j = (W_1^j, W_2^j, \ldots, W_k^j) \sim \tau_p^k$.
   (b) Alice and Bob privately sample $X^{j,I}$ and Y^j from $\nu_k(a|W^j)$ and $\nu_k(b|W^j)$, respectively. Bob privately and independently samples
   $$(X^{j,1}, \ldots, X^{j,I-1}, X^{j,I+1}, \ldots, X^{j,k}) \sim \mu_k(a|Y^j)^{k-1}.$$
   (c) Alice privately draws an independent sample d from a uniform distribution on $[k]$, and resets $X_d^{j,I}$ to 0 or 1 with equal probability. As a result, $(X^{j,I}, Y^j) \sim \mu_k$. For each $i \in [k] \setminus \{I\}$, Bob privately draws a sample d from a uniform distribution on $[k]$ and resets $X_d^{j,i}$ to a sample from $\tau_{1/2}$.

Note that the input X^I of site p^I is determined by the public coins, Alice's input a and her private coins. The inputs X^{-I} are determined by the public coins, Bob's input b and his private coins.

Let ϕ denote the distribution of X when (a, b) is chosen according to the distribution μ_k.

Let α be the approximation ratio parameter. We set $p = \alpha/30 \leq 1/30$ in the definition of μ_k.

Given a private randomized protocol $\mathcal{P}'$ for DMR that achieves an α-approximation with the error probability at most 1/4 under ϕ, we construct a public coin protocol $\mathcal{P}$ for DISJ that has a one-sided error probability of at most $1 - \Theta(\alpha)$ as follows.

Protocol $\mathcal{P}$

1. Given input $(A, B) \sim \mu_k$, Alice and Bob construct an input $X \sim \phi$ for DMR as described by the input reduction above. Let $Y = (Y^1, Y^2, \ldots, Y^r)$ be the samples used for the construction of X. Let I, J be the two indices sampled by Alice and Bob in the reduction procedure.
2. With Alice simulating site $p^{\ I}$ and Bob simulating other sites and the coordinator, they run $\mathcal{P}'$ on the input defined by X. Any communication between site p^I and the coordinator will be exchanged between Alice and Bob. For any communication among

[1] $\mu_k(b)$ is the marginal distribution of b of the joint distribution μ_k (see section 3.2 for the definition)

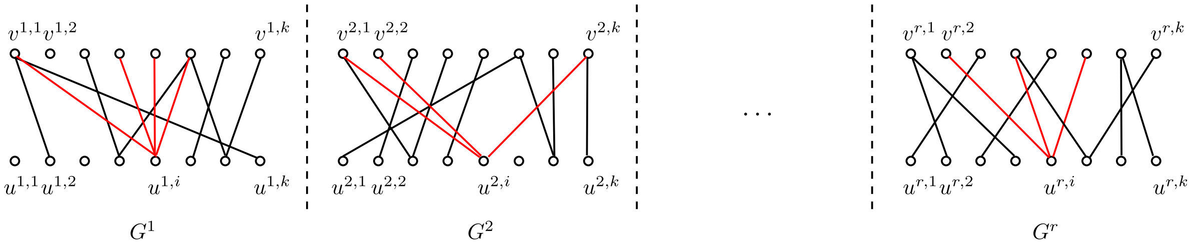

Fig. 2 Construction of input graph G and its partitioning over sites: G is a composition of bipartite graphs $G^1, G^2, \ldots, G^r$, each having k vertices on each side of the bipartition; each site $i \in [k]$ is assigned edges incident to vertices $u^{1,i}, u^{2,i}, \ldots, u^{r,i}$; the neighbourhood set of vertex $u^{j,i}$ is determined by $X^{j,i} \in \{0,1\}^k$.

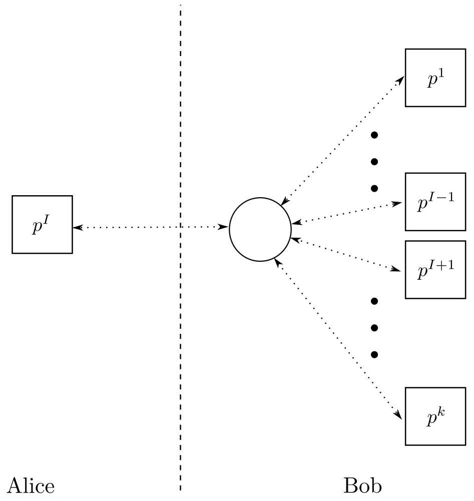

Fig. 3 Alice simulates site p^I and Bob simulates the rest of the system.

other sites and the coordinator, Bob just simulates it without any actual communication. At the end, the coordinator, that is Bob, obtains a matching M.

3. Bob outputs 1 if, and only if, for some $l \in [k]$, $(u^{J,I}, v^{J,l})$ is an edge in M such that $Y_l^J \equiv B_l = 1$, and 0, otherwise.

Correctness Suppose that $\text{DISJ}(A, B) = 0$, i.e., $A_l = 0$ or $B_l = 0$ for all $l \in [k]$. Then, for each $l \in [k]$, we must either have $Y_l^J \equiv B_l = 0$ or $X_l^{J,I} \equiv A_l = 0$, but $X_l^{J,I} = 0$ means that $(u^{J,I}, v^{J,l})$ is not an edge in M. Thus, $\mathcal{P}$ will always answer correctly when $\text{DISJ}(A, B) = 0$, i.e., it has a one-sided error.

Now suppose that $A_l = B_l = 1$ for some $l \in [k]$. Note that there is at most one such l according to our construction, which we denote by L. The output of $\mathcal{P}$ is correct if $(u^{J,I}, v^{J,L})$ is an edge in M. We next bound the probability of this event.

For each G^j, for $z \in \{0,1\}$, we let

$$U_z^j = \{u^{j,i} \in U^j : \text{DISJ}(X^{j,i}, Y^j) = z\},$$

$$V_z^j = \{v^{j,i} \in V^j : Y_i^j = z\}$$

and

$$U_z = \cup_{j \in [r]} U_z^j \text{ and } V_z = \cup_{j \in [r]} V_z^j.$$

Intuitively, the edges between vertices $U_0 \cup U_1$ and V_0 can be regarded as *noisy edges* because the total number of such edges is large, but the maximum matching they can form is small (Lemma 4 below). On the other hand, the edges between vertices U_1 and V_1 can be regarded as *important edges* because a maximum matching they can form is large though the total number of such edges is small. Note that there is no edge between vertices U_0 and V_1. See Figure 4 for an illustration.

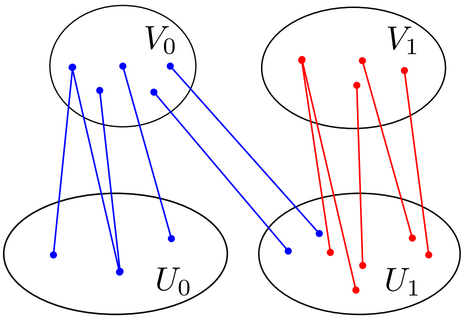

Fig. 4 Edges between $U_0 \cup U_1$ and V_0 are noisy edges. Edges between U_1 and V_1 are important edges. There are no edges between U_0 and V_1.

To find a good matching we must choose many edges from the set of important edges. A key property is that all important edges are statistically identical, that is, each important edge is equally likely to be the edge $(u^{J,I}, v^{J,L})$. Thus, $(u^{J,I}, v^{J,L})$ will be included in the matching returned by $\mathcal{P}'$ with a large enough probability. Using this, we can answer whether $X^{J,I}$ and Y^J intersect or not, thus, solving the original DISJ problem.

Recall that we set $p = \alpha/30 \leq 1/30$ and $\delta = (1 - p)^2/2$. Thus, $9/20 < \delta < 1/2$. In the following, we assume $\alpha \geq c/\sqrt{n}$ for some constant, since otherwise the

$\Omega(\alpha^2 kn)$ lower bound will be dominated by the trivial lower bound of k.[2]

Lemma 4 *With probability at least* $1-1/100$,

$$|V_0| \leq 2pn.$$

Proof Note that each vertex in $\cup_{j\in[r]}V^j$ is included in V_0 independently with probability $p(2-p)$. Hence, $\mathbf{E}[|V_0|] = p(2-p)n/2$, and by the Hoeffding's inequality, we have

$$\begin{aligned}\mathbf{Pr}[|V_0| \geq 2pn] &\leq \mathbf{Pr}[|V_0| - \mathbf{E}[|V_0|] \geq pn] \\ &\leq e^{-2p^2 n} \\ &\leq 1/100.\end{aligned}$$

□

Notice that Lemma 4 implies that with probability at least $1-1/100$, the size of a maximum matching formed by edges between vertices V_0 and $U_0 \cup U_1$ is smaller than or equal to $2pn$.

Lemma 5 *With probability at least* $1-1/100$*, the size of a maximum matching in* G *is at least* $n/10$.

Proof Consider the size of a matching in G^j for an arbitrary $j \in [r]$. For each $i \in [k]$, let L^i be the index $l \in [k]$ such that $X_l^{j,i} = Y_l^j = 1$ if such an l exists (note that by our construction at most one such index exists), and let L^i be defined as NULL, otherwise.

We use a greedy algorithm to construct a matching between vertices U^j and V^j. For $i \in [k]$, we connect $u^{j,i}$ and v^{j,L^i} if L^i is not NULL and v^{j,L^i} is not connected to some $u^{j,i'}$ for $i' < i$. The size of such constructed matching is equal to the number of distinct elements in $\{L^1, L^2, \ldots, L^k\}$, which we denote by R. We next establish the following claim:

$$\mathbf{Pr}[R \geq k/4] \geq 1 - O(1/k). \tag{21}$$

By our construction, we have

$$\mathbf{E}[|U_1^j|] = \delta k \text{ and } \mathbf{E}[|V_1^j|] = (1-p)^2 k.$$

By the Hoeffding's inequality, with probability $1-e^{-\Omega(k)}$,

$$|V_1^j| \geq \frac{9}{10}\mathbf{E}[|V_1^j|] \geq \frac{4}{5}k$$

and

$$|U_1^j| \geq \frac{9}{10}\mathbf{E}[|U_1^j|] \geq \frac{2}{5}k.$$

It follows that with probability $1-e^{-\Omega(k)}$, it holds that R is at least of value R', where R' is as defined as follows.

Consider a balls-into-bins process with s balls and t bins. Throw each ball to a bin sampled uniformly at random from the set of all bins. Let Z be the number of non-empty bins at the end of this process. Then, it is straightforward to observe that the expected number of non-empty bins is

$$\mathbf{E}[Z] = t\left(1 - \left(1-\frac{1}{t}\right)^s\right) \geq t\left(1 - e^{-s/t}\right).$$

By Lemma 1 in [18], for $100 \leq s \leq t/2$, the variance of the number of non-empty bins satisfies[3]

$$\mathbf{Var}[Z] \leq 5\frac{s^2}{t}$$

Let R' be the number of non-empty bins in the balls-into-bins process with $s = 2k/5$ balls and $t = 4k/5$ bins. Then, we have

$$\mathbf{E}[R'] \geq \frac{4}{5}k\left(1 - 1/\sqrt{e}\right)$$

and

$$\mathbf{Var}[R'] \leq 5\frac{(2k/5)^2}{4k/5} = k.$$

By the Chebyshev's inequality,

$$\mathbf{Pr}[R' < \mathbf{E}[R'] - k/20] \leq \frac{\mathbf{Var}[R']}{(k/20)^2} < \frac{400}{k}.$$

Hence, with probability $1-O(1/k)$, $R \geq R' \geq k/4$, which proves the claim in (21).

It follows that for each G^j, we can find a matching in G^j of size at least $k/4$ with probability $1-O(1/k)$. If $r = n/(2k) = o(k)$, then by the union bound, it holds that with probability at least $1-1/100$, the size of a maximum matching in G is at least $n/4$. Otherwise, let $R^1, R^2, \ldots, R^r$ be the sizes of matchings that are independently computed using the greedy matching algorithm described above for respective input graphs $G^1, G^2, \ldots, G^r$. Let $Z_j = 1$ if $R^j \geq k/4$, and $Z_j = 0$, otherwise. Since $R^j \geq kZ_j/4$ for all $j \in [r]$ and $\mathbf{E}[Z_j] = 1 - O(1/k)$, by the Hoeffding's inequality, we have

$$\begin{aligned}\mathbf{Pr}\left[\sum_{j=1}^r R^j < \frac{n}{10}\right] &\leq \mathbf{Pr}\left[\sum_{j=1}^r Z_j < \frac{2n}{5k}\right] \\ &= \mathbf{Pr}\left[\sum_{j=1}^r Z_j < \frac{4r}{5}\right] \quad \text{as } r = \tfrac{n}{2k}\end{aligned}$$

[2] Since none of the sites can see messages sent by other sites to the coordinator (unless this is communicated by the coordinator), each site needs to communicate with the coordinator at least once to determine the status of the protocol.

[3] The constants used here are slightly different from [18].

$$\leq \mathbf{Pr}\left[\sum_{j=1}^{r} Z_j < \mathbf{E}[\sum_{j=1}^{r} Z_j] - \frac{r}{10}\right]$$
$$\leq e^{-\frac{r^2/100}{r}} = e^{-\Omega(r)}$$

Hence, the size of a maximum matching in G is at least $n/10$ with probability at least $1 - e^{-\Omega(r)} \geq 1 - 1/100$. □

If $\mathcal{P}'$ is an α-approximation algorithm with error probability at most $1/4$, then by Lemma 4 and 5, with probability at least $3/4 - 1/100 \geq 2/3$, $\mathcal{P}'$ will output a matching M that contains at least $\alpha n/10 - 2pn$ important edges, and we denote this event by $\mathcal{F}$. We know that there are at most $n/2$ important edges and edge $(u^{J,I}, v^{J,L})$ is one of them. We say that (i,j,l) is important for G, if $(u^{j,i}, v^{j,l})$ is an important edge in G. Given an input G, the algorithm cannot distinguish between any two important edges. We can apply the principle of deferred decisions to decide the value of (I,J) after the matching has already been computed, which means, conditioned on $\mathcal{F}$, the probability that $(u^{J,I}, v^{J,L}) \in M$ is at least $(\alpha n/10 - 2pn)/(n/2) = \alpha/15$, as $p = \alpha/30$. Since $\mathcal{F}$ happens with probability at least $2/3$, we have

$$\mathbf{Pr}[(u^{J,I}, v^{J,L}) \in M] \geq \alpha/30.$$

To sum up, we have shown that protocol $\mathcal{P}$ solves DISJ correctly with one-sided error of at most $1-\alpha/30$.

Information cost We analyze the information cost of DMR. Let $\Pi = \Pi^1 \circ \Pi^2 \circ \cdots \circ \Pi^k$ be any protocol for DMR having error probability $1/4$ with respect to input distribution ϕ. We will show below that the information cost of Π is lower bounded by $\Omega(\alpha^2 kn)$ and thus $IC_{\phi,1/4}(\text{DMR}) \geq \Omega(\alpha^2 kn)$.

Let $W^{-J} = (W^1, \ldots, W^{J-1}, W^{J+1}, \ldots, W^r)$, and $W = (W^1, W^2, \ldots, W^r)$. Let $W_{A,B} \sim \tau_p^k$ denote the random variable used to sample (A,B) from μ_k. Recall that in our input reduction I, J, W^{-J} are the only public coins used by Alice and Bob. That means, $\mathcal{P}$ is a public coin protocol whose public coins are I, J, W^{-J} (since $\mathcal{P}'$ only uses private randomness).

Recall Y^1, Y^2, ..., Y^r are auxiliary random vectors used for (privately) sample X, which are independent and identically distributed random variables with distribution $\mu_k(b)$, and let $Y = \{Y^1, \ldots, Y^r\}$. We have the following:

$$\frac{2}{n}\sum_{i=1}^{k} I(X^1, X^2, \cdots, X^k; \Pi^i)$$
$$\geq \frac{1}{rk}\sum_{i=1}^{k} I(X^i, Y; \Pi^i) \tag{22}$$
$$\geq \frac{1}{rk}\sum_{i=1}^{k} I(X^i, Y; \Pi^i | W) \tag{23}$$
$$\geq \frac{1}{rk}\sum_{i=1}^{k}\sum_{j=1}^{r} I(X^{j,i}, Y^j; \Pi^i | W^{-j}, W^j) \tag{24}$$
$$= \frac{1}{rk}\sum_{i=1}^{k}\sum_{j=1}^{r} I(A, B; \Pi^i | I = i, J = j, W^{-j}, W_{A,B}) \tag{25}$$
$$= I(A, B; \Pi^I | I, J, W^{-J}, W_{A,B})$$
$$= I(A, B; \mathcal{P} | I, J, W^{-J}, W_{A,B})$$
$$\geq I(A, B; \Pi^* | W_{A,B}, R) \tag{26}$$
$$= \Omega(\alpha^2 k), \tag{27}$$

where (22) is by Lemma 3, (23) is by the data processing inequality, (24) is by the super-additivity property, (25) holds because the distribution of W^j is the same as that of $W_{A,B}$, and the conditional distribution of $(X^{j,i}, Y^j, \Pi^i)$ given W^{-j}, W^j is the same as the conditional distribution of (A, B, Π^i) given $I = i$, $J = j$, W^{-j}, $W_{A,B}$, in (26), Π^* is the best protocol for DISJ with one-sided error probability at most $1 - \alpha/30$ and R is the public randomness used in Π^*, and (27) holds by Theorem 4 where recall that we have set $p = \alpha/30$.

We have thus shown that $IC_{\phi,1/4}(\text{DMR}) \geq \Omega(\alpha^2 kn)$. Since by Theorem 2, $R_{1/4}(\text{DMR}) \geq IC_{\phi,1/4}(\text{DMR})$, it follows that

$$R_{1/4}(\text{DMR}) \geq \Omega(\alpha^2 kn)$$

which proves Theorem 1.

4 Upper Bound

In this section we present an α-approximation algorithm for the distributed matching problem with an upper bound on the communication complexity that matches the lower bound for any $\alpha \leq 1/2$ up to polylogarithmic factors.

We have described a simple algorithm that guarantees an 1/2-approximation for DMR at the communication cost of $O(kn \log n)$ bits in Section 1. This algorithm is a greedy algorithm that computes a maximal matching. The communication cost of the algorithm is $O(\alpha^2 kn \log n)$ bits. If $1/8 < \alpha \leq 1/2$, we simply apply the greedy 1/2-approximation algorithm that has the communication cost of $O(kn \log n)$ bits. Therefore, we assume that $\alpha \leq 1/8$ in the rest of this section. We next present an α-approximation algorithm that uses the greedy maximal matching algorithm as a subroutine.

Algorithm: The algorithm consists of two steps:

1. The coordinator sends a message to each site asking to compute a local maximum matching, and each site then follows up with reporting back to the coordinator the *size* of its local maximum matching. The coordinator sends a message to a site that holds a local maximum matching of maximum size, and this site then responds with sending back to the coordinator at most αn edges from its local maximum matching. Then, the algorithm proceeds to the second step.
2. The coordinator selects each site independently with probability q, where q is set to 8α (recall we assume $\alpha \leq 1/8$), and computes a maximal matching by applying the greedy maximal matching algorithm to the selected sites.

It is readily observed that the expected communication cost of Step 1 is at most $O((k + \alpha n) \log n)$ bits, and that the communication cost of Step 2 is at most $O((k + \alpha^2 kn) \log n)$ bits. We next show correctness of the algorithm.

Correctness of the algorithm. Let X_i be a random variable that indicates whether or not site p^i is selected in Step 2. Note that $\mathbf{E}[X_i] = q$ and $\mathbf{Var}[X_i] = q(1 - q)$. Let M be a maximum matching in G and let m denote its size. Let m_i be the number of edges in M which belong to site p^i. Hence, we have $\sum_{i=1}^{k} m_i = m$ because the edges of G are assumed to be partitioned disjointly over the k sites. We can assume that $m_i \leq \alpha m$ for all $i \in [k]$; otherwise, the coordinator has already gotten an α-approximation from Step 1.

Let Y be the size of the maximal matching that is output of Step 2. Recall that any maximal matching is at least 1/2 of any maximum matching. Thus, we have $Y \geq X/2$, where $X = \sum_{i=1}^{k} m_i X_i$. Note that we have $\mathbf{E}[X] = qm$ and $\mathbf{Var}[X] = q(1-q) \sum_{i=1}^{k} m_i^2$. Under the constraint $m_i \leq \alpha m$ for all $i \in [k]$, we have

$$\sum_{i=1}^{k} m_i^2 \leq \alpha m \sum_{i=1}^{k} m_i = \alpha m^2.$$

Hence, combining with the assumption $q = 8\alpha$, it follows that $\mathbf{Var}[X] \leq 8\alpha^2 m^2$. By Chebyshev's inequality, we have

$$\mathbf{Pr}[|X - qm| \geq 6\alpha m] \leq \frac{8}{36} < \frac{1}{4}.$$

Since $q = 8\alpha$, it follows that $X \geq 2\alpha m$ with probability at least 3/4. Combining with $Y \geq X/2$, we have that $Y \geq \alpha m$ with probability at least 3/4.

We have shown the following theorem.

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

Note that $\Omega(\alpha n)$ is a trivial lower bound, simply because the size of the output could be as large as $\Omega(\alpha n)$. Obviously, $\Omega(k)$ is a lower bound, because the coordinator has to send at least one message to each site. 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.

One can see that the above algorithm needs $O(\alpha k)$ rounds, as we use a naive algorithm to compute a maximal matching among αk sites. If k is large, say, n^β for some constant $\beta \in (0, 1)$, this may not be acceptable. Fortunately, Luby's parallel algorithm [29] can be easily adapted to our model, using only $O(\log n)$ rounds at the cost of increasing the communication by at most a $\log n$ factor.

4.1 Luby's algorithm in the coordinator model

Luby's algorithm [29]: Let $G = (V, E)$ be the input graph, and M be a matching initialized to $\emptyset$. Luby's algorithm for maximal matching is as follows.

1. If E is empty, return M.
2. Randomly assign unique priority π_e to each $e \in E$.
3. Let M' be the set of edges in E with higher priority than all of its neighboring edges. Delete M' and all the neighboring edges of M' from E, and add M' to M. Go to step 1.

It is easy to verify that the output M is a maximal matching. The number of iterations before E becomes empty is at most $O(\log n)$ in expectation [29]. Next we briefly describe how to implement this algorithm in the coordinator model. Let E^i be the edges held by site p^i.

1. For each i, if E^i is empty, p^i halts. Otherwise p^i randomly assigns unique priority π_e to each $e \in E^i$.
2. Let M'^i be the set of edges in E^i with higher priority than all of its neighboring edges in E^i. Then p^i sends M'^i together with their priorities to the coordinator.
3. Coordinator gets $W = M'^1 \cup M^2 \cup \cdots \cup M'^k$. Let M' be the set of edges in W with higher priority than all of its neighboring edges in W. Coordinator adds M' to M and then sends M' to all sites.
4. Each site p^i deletes all neighboring edges of M' from E^i, and goes to step 1.
5. After all the sites halt, the coordinator outputs M.

It is easy to see that the above algorithm simulates the algorithm of Luby. Therefore, the correctness follows from the correctness of Luby's algorithm, and the number of rounds is the same, which is $O(\log n)$. The communication cost in each round is at most $O(kn \log n)$ bits because, in each round, each site sends a matching to the coordinator, and the coordinator sends back another matching. Hence, the total communication cost is $O(kn \log^2 n)$ bits.

5 Conclusion

We have established a tight lower bound on the communication complexity for approximate maximum matching problem in the message-passing model.

An interesting open problem is the complexity of the counting version of the problem, i.e., the communication complexity if we only want to compute an approximation of the *size* of a maximum matching in a graph. Note that our proof of the lower bound relies on the fact that the algorithm has to return a certificate of the matching. Hence, in order to prove a lower bound for the counting version of the problem, one may need to use new ideas and it is also possible that a better upper bound exists. In a recent work [20], the counting version of the matching problem was studied in the random-order streaming model. They proposed an algorithm that uses one pass and poly-logarithmic space, which computes a poly-logarithmic approximation of the size of a maximum matching in the input graph.

A general interesting direction for future research is to investigate the communication complexity for other combinatorial problems on graphs, for example, connected components, minimum spanning tree, vertex cover and dominating set. The techniques used for approximate maximum matching in the present paper could be of use in addressing these other problems.

References

1. Ahn, K.J., Guha, S.: Laminar families and metric embeddings: Non-bipartite maximum matching problem in the semi-streaming model. CoRR **abs/1104.4058** (2011). URL http://arxiv.org/abs/1104.4058
2. Ahn, K.J., Guha, S.: Linear programming in the semi-streaming model with application to the maximum matching problem. Inf. Comput. **222**, 59–79 (2013)
3. Ahn, K.J., Guha, S., McGregor, A.: Analyzing graph structure via linear measurements. In: Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '12, pp. 459–467 (2012)
4. Ahn, K.J., Guha, S., McGregor, A.: Graph sketches: Sparsification, spanners, and subgraphs. In: Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS '12, pp. 5–14 (2012)
5. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. Journal of Computer and System Sciences **58**(1), 137 – 147 (1999)
6. Alon, N., Nisan, N., Raz, R., Weinstein, O.: Welfare maximization with limited interaction. In: Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on, pp. 1499–1512 (2015)
7. Assadi, S., Khanna, S., Li, Y., Yaroslavtsev, G.: Maximum Matchings in Dynamic Graph Streams and the Simultaneous Communication Model, chap. 93, pp. 1345–1364 (2016)
8. Bar-Yossef, Z., Jayram, T., Kumar, R., Sivakumar, D.: Special issue on focs 2002 an information statistics approach to data stream and communication complexity. Journal of Computer and System Sciences **68**(4), 702 – 732 (2004)
9. Barak, B., Braverman, M., Chen, X., Rao, A.: How to compress interactive communication. SIAM Journal on Computing **42**(3), 1327–1363 (2013)
10. Braverman, M.: Interactive information complexity. SIAM Journal on Computing **44**(6), 1698–1739 (2015)
11. Braverman, M., Ellen, F., Oshman, R., Pitassi, T., Vaikuntanathan, V.: A tight bound for set disjointness in the message-passing model. In: Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS '13, pp. 668–677 (2013)
12. Chakrabarti, A.: Informational complexity and the direct sum problem for simultaneous message complexity. In: Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science, FOCS '01, pp. 270– (2001)
13. Cover, T., Thomas, J.: Elements of information theory. Wiley-interscience (2006)
14. Dobzinski, S., Nisan, N., Oren, S.: Economic efficiency requires interaction. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC '14, pp. 233–242. ACM, New York, NY, USA (2014). DOI 10.1145/2591796.2591815. URL http://doi.acm.org/10.1145/2591796.2591815
15. Epstein, L., Levin, A., Mestre, J., Segev, D.: Improved approximation guarantees for weighted matching in the semi-streaming model. SIAM Journal on Discrete Mathematics **25**(3), 1251–1265 (2011)
16. Goodrich, M.T., Sitchinava, N., Zhang, Q.: Sorting, searching, and simulation in the mapreduce framework. Algorithms and Computation **7074 of the series Lecture Notes in Computer Science**, 374–383 (2011)
17. Israeli, A., Itai, A.: A fast and simple randomized parallel algorithm for maximal matching. Information Processing Letters **22**(2), 77–80 (1986)
18. Kane, D.M., Nelson, J., Woodruff, D.P.: An optimal algorithm for the distinct elements problem. In: Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '10, pp. 41–52 (2010)
19. Kapralov, M.: Better bounds for matchings in the streaming model, chap. 121, pp. 1679–1697
20. Kapralov, M., Khanna, S., Sudan, M.: Approximating matching size from random streams, chap. 55, pp. 734–751
21. Karloff, H., Suri, S., Vassilvitskii, S.: A model of computation for mapreduce. In: Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '10, pp. 938–948 (2010)
22. Klauck, H., Nanongkai, D., Pandurangan, G., Robinson, P.: The distributed complexity of large-scale graph processing. CoRR **abs/1311.6209** (2013)
23. Konrad, C.: Maximum Matching in Turnstile Streams, pp. 840–852. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)

24. Konrad, C., Magniez, F., Mathieu, C.: Maximum Matching in Semi-streaming with Few Passes, pp. 231–242. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
25. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press
26. Lattanzi, S., Moseley, B., Suri, S., Vassilvitskii, S.: Filtering: A method for solving graph problems in mapreduce. In: Proceedings of the Twenty-third Annual ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '11, pp. 85–94 (2011)
27. Lotker, Z., Patt-Shamir, B., Pettie, S.: Improved distributed approximate matching. In: Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures, SPAA '08, pp. 129–136 (2008)
28. Lotker, Z., Patt-Shamir, B., Rosen, A.: Distributed approximate matching. In: Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Distributed Computing, PODC '07, pp. 167–174 (2007)
29. Luby, M.: A simple parallel algorithm for the maximal independent set problem. SIAM journal on computing **15**(4), 1036–1053 (1986)
30. McGregor, A.: Finding Graph Matchings in Data Streams, pp. 170–181. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
31. McGregor, A.: Question 16: Graph matchings: Open problems in data streams and related topics. In: Workshop on algorithms for data streams (2006). URL http://www.cse.iitk.ac.in/users/sganguly/data-stream-probs.pdf
32. Phillips, J.M., Verbin, E., Zhang, Q.: Lower bounds for number-in-hand multiparty communication complexity, made easy. SIAM Journal on Computing **45**(1), 174–196 (2016)
33. Wattenhofer, M., Wattenhofer, R.: Distributed Weighted Matching, pp. 335–348. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
34. Woodruff, D.P., Zhang, Q.: Tight bounds for distributed functional monitoring. In: Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing, STOC '12, pp. 941–960 (2012)
35. Woodruff, D.P., Zhang, Q.: An Optimal Lower Bound for Distinct Elements in the Message Passing Model, chap. 54, pp. 718–733 (2014)
36. Woodruff, D.P., Zhang, Q.: When distributed computation is communication expensive. Distributed Computing pp. 1–15 (2014)
37. Zelke, M.: Weighted matching in the semi-streaming model. Algorithmica **62**(1), 1–20 (2012)