Communication Complexity of Approximate Matching in Distributed Graphs

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

1 **MADALGO, Aarhus University, Denmark**
huangzf@cse.ust.hk
2,3 **Microsoft Research, Cambridge, UK**
{bozidar,milanv}@microsoft.com
4 **Indiana University Bloomington, USA**
qzhangcs@indiana.edu

Abstract

In this paper we consider the communication complexity of approximation algorithms for maximum matching in a graph in the message-passing model of distributed computation. The input graph consists of n vertices and edges partitioned over a set of k sites. The output is an α-approximate maximum matching in the input graph which has to be reported by one of the sites. We show a lower bound on the communication complexity of $\Omega(\alpha^2 kn)$ and show that it is tight up to poly-logarithmic factors. This lower bound also applies to other combinatorial problems on graphs in the message-passing computation model, including max-flow and graph sparsification.

1998 ACM Subject Classification F.2.3 Tradeoffs between Complexity Measures

Keywords and phrases approximate maximum matching, distributed computation, communication complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.460

1 Introduction

Massive data volumes require scaling out computations using distributed clusters of machines which are nowadays commonly deployed in data centres. The data is typically stored distributively across different machines (we refer to as sites) which are interconnected with a communication network. It is desired to process such distributed data with a limited communication among sites which avoids the communication network becoming a bottleneck. A particular interest has been devoted to data in the form of a graph that arises in many applications including online services, online social networks, biological and other networks. There has been a surge of interest in distributed iterative computations using graph input data and resolving queries in distributed graph databases. In practice, the size of a graph can be as large as in the order of a billion of vertices and a trillion of edges, e.g. semantic web knowledge graphs and online social networks [12]. An important research direction is to design efficient algorithms for processing of large-scale graphs in distributed systems which has been one of the focuses of the theoretical computer science community, e.g. [25, 23, 5, 4].

In this paper we consider the problem of approximate computation of a maximum matching in a graph that is stored edge-partitioned across different sites. There are several performance measures of interest in computations over distributed data including the communication complexity in terms of the number of bits or messages, the time complexity in terms of the number of time units or rounds, and the storage complexity in terms of the number of bits.

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).
Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 460–473
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper we focus on the performance measure of the communication complexity in the number of bits required to approximately compute a maximum matching in a graph. Our main result is a tight lower bound on the communication complexity for computing an approximate maximum matching in a graph.

We consider the distributed computation model known in the literature as the *message-passing* model, see, e.g., [27, 10]. A message-passing model consists of k sites, $P^1, \ldots, P^k$. Each site P^i holds a piece of input data x^i and the sites want to jointly compute a given function $f(x^1, \ldots, x^k)$. 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. Our goal is to minimize 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*. Note that we have essentially replaced the clique communication topology with the star topology, which increases the total communication cost only by a factor of 2, which does not affect the order of the asymptotic communication complexity.

1.1 Our Results and Techniques

We study the approximate maximum matching problem in the message-passing model which we refer to as Distributed Matching Reporting (DMR). Given a set of $k > 1$ sites and an input graph $G = (V, E)$ with $|V| = n$ vertices and the set of edges $E = E^1 \cup E^2 \cup \cdots \cup E^k$ such that the set of edges E^i is assigned to site P^i, at the end of the computation, the coordinator is required to report an α-approximation of the maximum matching in graph G. In this paper show the following main theorem.

▶ **Theorem 1.** *Any approximation algorithm for computing 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, under assumption that $k \leq n$. This communication complexity holds for bipartite graphs.*

It is noteworthy that a simple greedy algorithm solves DMR for $\alpha = 1/2$ with the communication cost of $O(kn \log n)$ bits. This greedy algorithm is based on computing a maximal matching by using a straightforward sequential procedure which we define 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 continues until site is reached P^k, which after computing M^k sends it to the coordinator. The matching M_k is a maximal matching in the 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. This shows that our lower bound it tight up to a $\log n$ factor. In Section 3.4, we show that our lower bound is also tight with respect to the approximation factor α for any $\alpha \leq 1/2$ up to a $\log n$ factor. It was showed by Woodruff and and Zhang [30] that many statistical estimation problems and combinatorial graph 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.

Our lower bound is also of wider applicability to other combinatorial problems on graphs. Since a bipartite maximum matching problem can be find by solving a *max-flow* problem, our lower bound also holds for approximate computation of a max-flow problem. Our lower bound also implies a lower bound for *graph sparsification* problem, see the definition of graph sparsification, e.g., in [5]. This is because in our lower bound construction (see Section 3), the bipartite graph under consideration contains many cuts of size 1 which have to be included in a sparsifier. By our construction these edges form a good approximate maximum matching. In Ahn, Guha, and McGregor [5], it is shown that there is a sketch-based $O(1)$-approximate graph sparsification algorithm with the sketch size of $\tilde{O}(n)$, which directly translates to an approximation algorithm of $\tilde{O}(kn)$ communication in our model. Thus, our lower bound is tight up to a polylogarithmic factor.

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 $q = n/(2k)$ 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). 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. We show that due to symmetry, the answer of DISJ can be recovered from a reported matching, and then we use information complexity to establish the direct-sum theorem. For this purpose, we also need to give a new definition of the information cost of a protocol in the message-passing model. We believe that our techniques could prove useful in establishing communication complexity for other graph problems in the message-passing model. One 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 hard instances would be like for the matching problem, i.e., each of the k sites holds roughly n/k vertices and the neighborhood of each vertex defines an independent instance of a two-party communication problem.

1.2 Related Work

The approximate maximum matching problem has been studied extensively in the literature in various settings. In this section we only review the results obtained in some most related models, namely the streaming computation model [6], the MapReduce model [19, 14], and the traditional distributed model of computation (which is different from ours, see discussions below). In the streaming computation model, the maximum matching problem was presented as one of the open problems by McGregor [1] and a number of results have been established, e.g., by McGregor [26], Epstein et al. [13], Ahn and Guha [2, 3], Ahn, Guha and McGregor [4], Zelke [32], Konard, Magniez and Mathieu [21], Kapralov [17], Kapralov, Khanna and Sudan [18]. Much of the previous work was devoted to the semi-streaming model that allows for $\tilde{O}(n)$ space, and these algorithms can be directly used to obtain an $\tilde{O}(kn)$ communication cost for $O(1)$-approximate matching in the message-passing model. The maximum matching problem was also studied in the MapReduce model, e.g., by Lattanzi et.al. [23]. Under certain assumptions, they obtain a 1/2-approximation algorithm in $O(1)$ rounds and $\tilde{O}(m)$ communication bits where m is the number of edges in the graph. In the context of traditional distributed computation models, Lotker et al [25, 24] 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 computation model is different from the message-passing computation model considered in this paper. In their model the input graph and the communication topology are the same while in the message-passing model considered here 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. Lotker et al. [24] (built on Wattenhofer and Wattenhofer [28], Lotker et al. [25]) showed existence of $(1-\epsilon)$-approximation algorithms for the maximum matching problem with $O(\log n)$ rounds. This implies the communication cost of $\tilde{O}(m)$ bits.

The message-passing computation model has recently attracted quite some attention by the research community, e.g. Phillips, Verbin and Zhang [27], Woodruff and Zhang [29], Braverman et al [10], Woodruff and Zhang [30], Klauck et al [20], and Woodruff and Zhang [31]. A wide set of statistical and graph problems has been shown to be hard in the sense of requiring $\Omega(kn)$ bits of communication, including the graph-connectivity problem [27, 30], exact computation of the number of distinct elements [30], k-party set-disjointness [10], and some were even showed to be hard for random order inputs [20]. A similar but different input distribution from ours was used in [10] to show an $\Omega(kn)$ communication lower bound for the k-party set-disjointness problem. The work presented in this paper was obtained independently and concurrently with [10] with the first version of the paper made online as a technical report [15] in April 2013. Similar distributions were also used previously in [27, 29] which appears to be natural because of the nature of the message-passing model. There may exist a reduction between the k-party set-disjointness studied in [10] and DMR but this is not clear unless one would establish a rigorous proof of this claim. Our proof is different from that in [10]: we use a reduction of the k-party DMR problem to a 2-party set-disjointness problem using symmetrisation, while [10] use a coordinative-wise direct-sum theorem to reduce the k-party set-disjointness problem to a k-party 1-bit problem.

2 Preliminaries

Conventions. Let $[n] = \{1, 2, \ldots, n\}$. All logarithms are with base of 2. We 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. We ofter refer to a player as a *site* which is suitable in the coordinator model under consideration.

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

Communication Complexity. In the two party communication complexity, 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 we have k parties and each has a piece of input, and they want to compute some function together by exchanging messages. For more information about the communication complexity we refer readers to [22].

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 protocol. It was extensively studied in the last decade, e.g., [11, 7, 8, 29, 9]. There are several definitions of information complexity. In this paper, we will follow the definition used in [7]. 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 $IC_\mu(\Pi) = I(XY; \Pi \mid R)$, 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 $IC_{\mu,\delta}(f) = \min_{\delta\text{-error } \Pi} IC_\mu(\Pi)$.

Information Complexity in the Coordinator Model. We can indeed extend the above definition of information complexity to k-party coordinator 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 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 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 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 coordinator model as $IC_{\mu,\delta}(f) = \min_{\delta\text{-error } \Pi} \sum_{i=1}^k I(X^1, \cdots, X^k; \Pi^i)$.

▶ **Theorem 2.** *$R_\delta(f) \geq IC_{\mu,\delta}(f)$ 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) $\mathsf{E}[|\Pi|] = \sum_{i=1}^k \mathsf{E}[|\Pi^i|] \geq \sum_{i=1}^k H(\Pi^i) \geq IC_{\mu,\delta}(\Pi)$. The theorem then follows since the worst-case cost is at least the average. ◀

▶ **Lemma 3.** *If Y is independent of the random coins used by the protocol Π, then $IC_{\mu,\delta}(f) \geq \min_\Pi \sum_{i=1}^k I(X^i, Y; \Pi^i)$.*

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

3 The Complexity of DMR

In this section we first prove the lower bound in Theorem 1 and then establish its tightness.

An outline of the proof of the lower bound is given as follows. The lower bound is established by constructing a hard distribution on the set of bipartite graphs $G = (U, V, E)$ with $|U| = |V| = n/2$. For the purpose of this outline, we consider the special case in which the number of sites is such that $k = n/2$. Each site is assigned one node in U together with all its adjacent edges. A natural idea 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 create many *noisy* edges: we randomly pick a small set of nodes $V_0 \subset V$ of size roughly $\alpha n/10$ and connect each node in U to each node in V_0 randomly with a constant probability. There are $\Theta(\alpha n^2)$ such edges and the size of the matching formed by these edges is at most $\alpha n/10 \approx \alpha/2 \cdot \mathrm{OPT}$ where OPT is the size of the optimal solution. We next create a set of *important* edges between U and $V \setminus V_0$ such that each node in U is adjacent to at most one random node in $V \setminus V_0$. These edges are important in the sense that although there are only $\Theta(|U|) = \Theta(n)$ of such edges, the size of the matching they can form is $\Theta(\mathrm{OPT})$. Therefore, to compute a matching of size at least $\alpha \cdot \mathrm{OPT}$, it is necessary to find and include $\Theta(\alpha \cdot \mathrm{OPT}) = \Theta(\alpha n)$ 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 many DISJ instances. The concrete implementation of this intuition is via an embedding argument. In the general case, we create $n/(2k)$ copies of the random bipartite graph, each of size $2k$. Each site gets $n/(2k)$ nodes. We then prove a direct-sum theorem using information complexity.

The lower bound is established by characterizing the information cost of the DISJ problem for specific input distributions. Before doing this we first characterize the information complexity of a primitive problem AND. We next reduce DISJ to DMR and prove an information cost lower bound for DMR.

3.1 The AND Problem

In the AND problem, 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 ν_1 and μ_1 for (A, B) as follows. Let $p = c \cdot \alpha \in (0, 1/2]$, where c is a constant to be chosen later.

ν_1: 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

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

W here serves as an auxiliary random variable to break the dependence between A and B, since ν_1 is not a product 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 distribution.

μ_1: 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 ν_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 a distribution if it is always correct when $\mathrm{AND}(x, y) = 0$, and is correct with probability at least $1 - \delta$ when $\mathrm{AND}(x, y) = 1$.

▶ **Theorem 4.** *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 $1 - \beta$, then $I(A, B; \Pi \mid W, R) = \Omega(\beta p)$.*

Proof. Our proof follows [7]. To handle a general $p \leq 1/2$, we explore the convexity of mutual information. To extend the result to distributional error, we give a more careful analysis and show that the information cost is high as long as the average error is small. The proof is somewhat technical and is deferred to the full version of the paper. ◀

3.2 The DISJ Problem

In the DISJ problem, 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 $\text{DISJ}(s, t) = \bigvee_{\ell=1}^{k} \text{AND}(s_\ell, t_\ell)$. Let $S = \{S_1, \ldots, S_k\}$ be Alice's input and $T = \{T_1, \ldots, T_k\}$ be Bob's input. We define two input distributions ν_k and μ_k for (S, T) as follows.

ν_k: 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: 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 $1 \leq \ell \leq k$. 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 $\text{DISJ}(S, T) = 1$ 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 $T = t$.

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 $\text{DISJ}(x, y) = 0$, and is correct with probability at least $1 - \delta$ when $\text{DISJ}(x, y) = 1$.

▶ **Theorem 5.** *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 $1 - \gamma$, then $I(S, T; \Pi \mid W, R) = \Omega(\gamma pk)$.*

Proof. The proof is deferred to the full version of the paper. ◀

3.3 Proof of the Main Theorem

To give a proof for Theorem 1, we first reduce DISJ to DMR. Before going to the detailed reduction, we 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; it is used to construct $X^{j,i}$ ($i \in [k], j \in [q]$).

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 constructs the input X^I for the I-th site, and Bob constructs 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. We have included Figure 1 for the illustration purpose.

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 will see the proof that the original DISJ instance can be solved by solving DMR also relies 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 $1-\alpha/10$ 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 set sampled during the construction of $\{X^1, \ldots, X^k\}$. 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 just simulates it without any actual communication. At the end the coordinator (that is, Bob) gets a matching M.

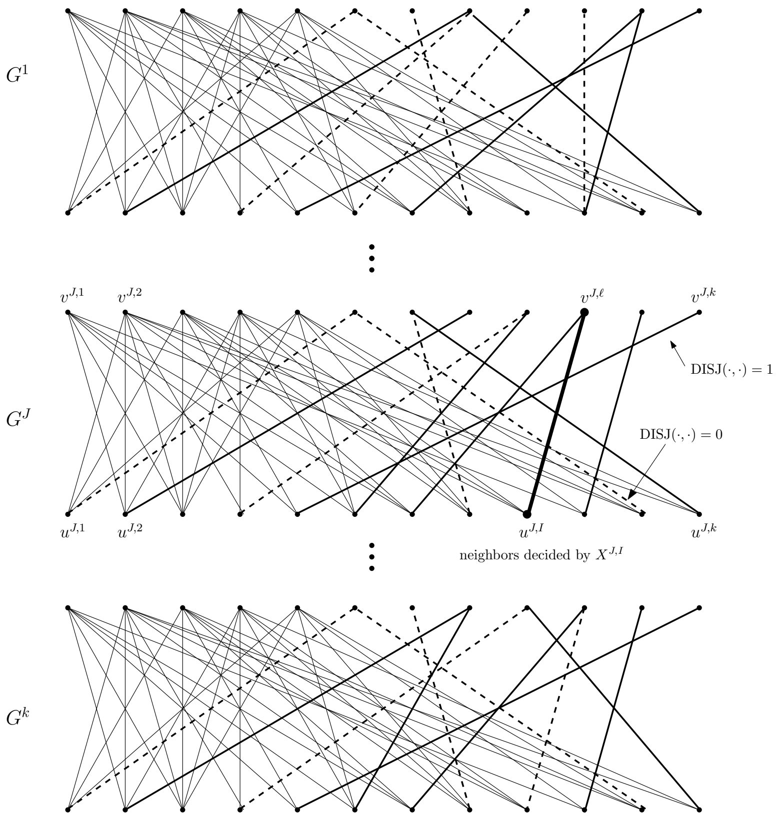

Figure 1 Each edge corresponds to a DISJ instance where a solid edge indicates an instance with output 1 and a dashed edge indicates an instance with output 0. Solid thick edges are the important edges. A good approximate matching has to output many important edges, thus a solid thick edge needs to be in the output matching with sufficiently large probability. The thick edge $(u^{J,I}, v^{J,\ell})$ corresponds to $\text{DISJ}(X^{J,I}, Y^J) = \text{DISJ}(s,t)$, that is to the original 2-party disjointness problem embedded by Alice and Bob. If $\text{DISJ}(s,t) = 1$, then $(u^{J,I}, v^{J,\ell})$ is a solid edge and needs to be included in the output matching with a sufficiently large probability.

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.

Correctness. First, suppose $\text{DISJ}(S,T) = 0$, 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 $\text{DISJ}(S,T) = 0$, 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 $U^j_1 = \{u^{j,i} \mid \text{DISJ}(X^{j,i}, Y^j) = 1\}$ and $U^j_0 = U^j \setminus U^j_1$. Let $V^j_1 = \{v^{j,\ell} \mid Y^j_\ell = 1\}$ and $V^j_0 = V^j \setminus V^j_1$. Let $U_0 = \cup^q_{j=1} U^j_0$, $U_1 = \cup^q_{j=1} U^j_1$, $V_0 = \cup^q_{j=1} V^j_0$ and $V_1 = \cup^q_{j=1} V^j_1$. 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 $|V_0| \leq 2pn$ according to Lemma 6, 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 6.** *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 $\mathsf{Pr}[|V_0| \ge 2pn] \le \mathsf{Pr}[|V_0| - \mathsf{E}[|V_0|] \ge pn] \le e^{-\Omega(p^2 n)}$.

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.

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

Proof. The proof is similar to Lemma 4 in [29]. 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 $\left(1 - e^{-\Omega(k)}\right)$, $|V_1^j| \ge 0.9 \cdot \mathsf{E}[|V_1^j|] = 0.9 \cdot (1 - 2p + p^2)k \ge 0.8k$ (recall $p \le 1/20$) and $|U_1^j| \ge 0.9 \cdot \mathsf{E}[|U_1^j|] \ge 0.4k$. 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 [16], 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

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

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

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 6 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 $1 - \alpha/10$.

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 $1 - \alpha/10$. By Lemma 3, we have $IC_{\phi,\delta}(\text{DMR}) \geq \sum_{i=1}^{k} I(X^i, Y; \Pi^i)$. 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.

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

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 $1 - \alpha/10$ and R is the public randomness used in Π^*. The information is measured according to μ_k.
3. Eq. (4) holds by Theorem 5. 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.

3.4 Tightness of the Lower Bound

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

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 $\alpha' = 8\alpha$, 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 next show correctness of the algorithm.

Let X_i be a 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 belong to 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 $m_i \le \alpha m$ 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 $Y \ge \frac{1}{2} \cdot \sum_{i=1}^{k} m_i X_i$. Let $Y' = \sum_{i=1}^{k} m_i X_i$. So we have $\mathsf{E}[Y'] = \alpha' m$ and $\mathsf{Var}[Y'] = \alpha'(1 - \alpha') \sum_{i=1}^{k} m_i^2 \le \alpha' \cdot \alpha m^2 = 8\alpha^2 m^2$. The inequality holds since we assume that $m_i \le \alpha m$ for all $i \in [k]$. Now, we can apply Chebyshev's inequality to bound the error probability. We have $\mathsf{Pr}[|Y' - \alpha' m| \ge 6\alpha m] \le 8/36 < 1/4$. Therefore, with probability at least $3/4$, it holds $Y \ge 1/2 \cdot Y' \ge 1/2 \cdot 2\alpha m = \alpha m$.

▶ **Theorem 7.** *For every given $\alpha \le 1/2$, there exists a randomized algorithm that computes an α-approximation of the maximum matching in a graph 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 $\Omega(\alpha n)$. Obviously, $\Omega(k)$ is a lower bound, since the coordinator has to talk to each of the sites at least once. 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.

4 Concluding Remarks

In this paper we showed a tight lower bound on the communication complexity for the 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 [18], the counting version of the matching problem was studied in the random-order streaming model. They proposed an algorithm that uses one pass and polylog space, which computes a polylog approximation of the size of the maximum matching. 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 problem in the present paper could be of use here.

Acknowledgements 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 Kook Jin Ahn and Sudipto Guha. Laminar families and metric embeddings: Non-bipartite maximum matching problem in the semi-streaming model. *CoRR*, abs/1104.4058, 2011.

3 Kook Jin Ahn and Sudipto 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 Kook Jin Ahn, Sudipto Guha, and Andrew 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 Kook Jin Ahn, Sudipto Guha, and Andrew 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 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

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 Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *FOCS*, pages 668–677, 2013.

11 Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew 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.

12 Jack Clark. Facebook rides Unicorn to graph search nirvana. The Register, http://www.theregister.co.uk/2013/03/07/facebook_unicorn_helps_graph_search, January 2013.

13 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.

14 Michael T. Goodrich, Nodari Sitchinava, and Qin Zhang. Sorting, searching, and simulation in the mapreduce framework. pages 374–383, 2011.

15 Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. Communication complexity of approximate maximum matching in distributed graph data. no. MSR-TR-2013-35, http://research.microsoft.com/apps/pubs/default.aspx?id=188946, 2013.

16 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proc. ACM Symposium on Principles of Database Systems*, 2010.

17 Michael 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.

18 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751, 2014.

19 Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for mapreduce. pages 938–948, 2010.

20 Hartmut Klauck, Danupon Nanongkai, Gopal Pandurangan, and Peter Robinson. The distributed complexity of large-scale graph processing. *CoRR*, abs/1311.6209, 2013.

21 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *APPROX-RANDOM*, pages 231–242, 2012.

22 E. Kushilevitz and N. Nisan. Communication complexity. 1997.

23 Silvio Lattanzi, Benjamin Moseley, Siddharth Suri, and Sergei 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.

24 Zvi Lotker, Boaz Patt-Shamir, and Seth Pettie. Improved distributed approximate matching. In *SPAA*, pages 129–136, 2008.

25 Zvi Lotker, Boaz Patt-Shamir, and Adi 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.

26 Andrew 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.

27 Jeff M. Phillips, Elad Verbin, and Qin 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 Mirjam Wattenhofer and Roger Wattenhofer. Distributed weighted matching. In *Distributed Computing, 18th International Conference, DISC 2004, Amsterdam, The Netherlands, October 4-7, 2004, Proceedings*, pages 335–348, 2004.

29 David P. Woodruff and Qin 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.

30 David P. Woodruff and Qin Zhang. When distributed computation does not help. *CoRR*, abs/1304.4636, 2013.

31 David P. Woodruff and Qin Zhang. An optimal lower bound for distinct elements in the message passing model. In *SODA*, pages 718–733, 2014.

32 Mariano Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, February 2012.