# Randomized Composable Coresets for Matching and Vertex Cover

Sepehr Assadi
Department of Computer and Information Sciences
University of Pennsylvania
Philadelphia, PA, USA

Sanjeev Khanna
Department of Computer and Information Sciences
University of Pennsylvania
Philadelphia, PA, USA

## ABSTRACT

A common approach for designing scalable algorithms for massive data sets is to distribute the computation across, say $k$, machines and process the data using limited communication between them. A particularly appealing framework here is the simultaneous communication model whereby each machine constructs a small representative summary of its own data and one obtains an approximate/exact solution from the union of the representative summaries. If the representative summaries needed for a problem are small, then this results in a *communication-efficient* and *round-optimal* (requiring essentially no interaction between the machines) protocol. Some well-known examples of techniques for creating summaries include sampling, linear sketching, and composable coresets. These techniques have been successfully used to design communication efficient solutions for many fundamental graph problems. However, two prominent problems are notably absent from the list of successes, namely, the *maximum matching* problem and the *minimum vertex cover* problem. Indeed, it was shown recently that for both these problems, even achieving a modest approximation factor of polylog($n$) requires using representative summaries of size $\widetilde{\Omega}(n^2)$ i.e. essentially no better summary exists than each machine simply sending its entire input graph.

The main insight of our work is that the intractability of matching and vertex cover in the simultaneous communication model is inherently connected to an *adversarial* partitioning of the underlying graph across machines. We show that when the underlying graph is randomly partitioned across machines, both these problems admit *randomized composable coresets* of size $\widetilde{O}(n)$ that yield an $\widetilde{O}(1)$-approximate solution[1]. In other words, a small *subgraph* of the input graph at each machine can be identified as its representative summary and the final answer then is obtained by simply running any maximum matching or minimum vertex cover algorithm on these combined subgraphs. This results in an $\widetilde{O}(1)$-approximation *simultaneous* protocol for these problems with $\widetilde{O}(nk)$ total communication when the input is randomly partitioned across $k$ machines. We also prove our results are optimal in a very strong sense: we not only rule out existence of smaller randomized composable coresets for these problems but in fact show that our $\widetilde{O}(nk)$ bound for total communication is optimal for *any* simultaneous communication protocol (i.e. not only for randomized coresets) for these two problems. Finally, by a standard application of composable coresets, our results also imply MapReduce algorithms with the same approximation guarantee in one or two rounds of communication, improving the previous best known round complexity for these problems.

## 1 INTRODUCTION

Recent years have witnessed tremendous algorithmic advances for efficient processing of massive data sets. A common approach for designing scalable algorithms for massive data sets is to distribute the computation across machines that are interconnected via a communication network. These machines can then jointly compute a function on the union of their inputs by exchanging messages. Two main measures of efficiency in this setting are the *communication cost* and the *round complexity*; we shall formally define these terms in details later in the paper but for the purpose of this section, communication cost measures the total number of bits exchanged by all machines and round complexity measures the number of rounds of interaction between them.

An important and widely studied framework here is the *simultaneous* communication model whereby each machine constructs a small representative summary of its own data and one obtains a solution for the desired problem from the union of the representative summary of combined pieces. The appeal of this framework lies in the simple fact that the *simultaneous protocols* are inherently *round-optimal*; they perform in only one round of interaction. The only measure that remains to be optimized is the communication cost – this is now determined by the size of the summary created by each machine. An understanding of the communication cost for a problem in the simultaneous model turns out to have value in other models of computation as well. For instance, a lower bound on the maximum communication needed by any machine implies a matching lower bound on the space complexity of the same problem in dynamic streams [7, 45].

Two particularly successful techniques for designing small summaries for simultaneous protocols are *linear sketches* and *composable coresets*. Linear sketching technique corresponds to taking a linear projection of the input data as its representative summary. The "linearity" of the sketches is then used to obtain a sketch of the combined pieces from which the final solution can be extracted. There has been a considerable amount of work in designing linear sketches

[1] Here and throughout the paper, we use $\widetilde{O}(\cdot)$ notation to suppress polylog($n$) factors, where $n$ is the number of vertices in the graph.

Supported in part by National Science Foundation grants CCF-1552909, CCF-1617851, and IIS-1447470.
A full version of this paper is available as an online preprint at [9].

*SPAA '17, July 24-26, 2017, Washington DC, USA*

ACM ISBN 978-1-4503-4593-4/17/07…$15.00
https://doi.org/10.1145/3087556.3087581

for graph problems in recent years [5, 6, 11, 17, 18, 20, 38, 39, 48]. Coresets are subgraphs (in general, subsets of the input) that suitably preserve properties of a given graph, and they are said to be composable if the union of coresets for a collection of graphs yields a coreset for the union of the graphs. Composable coresets have also been studied extensively recently [12, 13, 15, 34, 50, 51], and indeed several graph problems admit natural composable coresets; for instance, connectivity, cut sparsifiers, and spanners (see [47], Section 2.2; the "merge and reduce" approach). Successful applications of these two techniques has yielded $\widetilde{O}(n)$ size summaries for many graph problems (see further related work in Section 1.3). However, two prominent problems are notably absent from the list of successes, namely, the *maximum matching* problem and the *minimum vertex cover* problem. Indeed, it was shown recently [11] that both matching and vertex cover require summaries of size $n^{2-o(1)}$ for even computing a polylog$(n)$-approximate solution[2].

This state-of-affairs is the starting point for our work, namely, intractability of matching and vertex cover in the simultaneous communication model. Our main insight is that a natural *data oblivious partitioning scheme* completely alters this landscape: both problems admit $\widetilde{O}(1)$-approximate composable coresets of size $\widetilde{O}(n)$ provided the edges of the graph are randomly partitioned across the machines. The idea that random partitioning of data can help in distributed computation was nicely illustrated in the recent work of [50] on maximizing submodular functions. Our work can be seen as the first illustration of this idea in the domain of graph algorithms. The applicability of this idea to graph theoretic problems has been cast as an open problem in [50].

*Randomized Composable Coresets.* We follow the notation of [50] with a slight modification to adapt to our application in graphs. Let $E$ be an edge-set of a graph $G(V,E)$; we say that a partition $\left\{E^{(1)},\ldots,E^{(k)}\right\}$ of the edges $E$ is a *random $k$-partitioning* iff the sets are constructed by assigning each edge in $E$ independently to a set $E^{(i)}$ chosen uniformly at random. A random partitioning of the edges naturally defines partitioning the graph $G(V,E)$ into $k$ graphs $G^{(1)},\ldots,G^{(k)}$ whereby $G^{(i)} := G(V,E^{(i)})$ for any $i \in [k]$, and hence we use random partitioning for both the edge-set and the input graph interchangeably.

Definition (Randomized Composable Coresets [50]). *For a graph-theoretic problem $P$, consider an algorithm* ALG *that given any graph $G(V,E)$, outputs a subgraph* ALG$(G) \subseteq G$ *with at most $s$ edges. Let $G^{(1)},\ldots,G^{(k)}$ be a* random $k$-partitioning *of a graph $G$. We say that* ALG *outputs an $\alpha$-*approximation randomized composable core-set *of* size *$s$ for $P$ if $P$ $\left(\text{ALG}(G^{(1)}) \cup \ldots \cup \text{ALG}(G^{(k)})\right)$ is an $\alpha$-approximation for $P(G)$ w.h.p., where the probability is taken over the random choice of the $k$-partitioning. For brevity, we use randomized coresets to refer to randomized composable coresets.*

We further augment this definition by allowing the coresets to also contain a *fixed solution* to be *directly* added to the final solution of the composed coresets. In this case, size of the coreset is measured both in the number of edges in the output subgraph plus the number of vertices and edges picked by the fixed solution (this is mostly relevant for our coreset for the vertex cover problem).

## 1.1 Our Results

We show existence of randomized composable coresets for matching and vertex cover.

Result 1. *There exist randomized coresets of size $\widetilde{O}(n)$ that w.h.p. give an $O(1)$-approximation for maximum matching, and an $O(\log n)$-approximation for minimum vertex cover.*

Result 1 is formalized in Section 3. In contrast to Result 1, when the graph is *adversarially* partitioned, the results of [11] show that the best approximation ratio conceivable for these problems in $\widetilde{O}(n)$ space is only $\Theta(n^{1/3})$. We further remark that Result 1 can also be extended to the weighted version of the problems. Using the Crouch-Stubbs technique [22] one can extend our result to achieve a coreset for weighted matching (with a factor 2 loss in approximation and extra $O(\log n)$ term in the space). Similar ideas of "grouping by weight" of edges can also be used to extend our coreset for weighted vertex cover with an $O(\log n)$ factor loss in approximation and space.

The $\widetilde{O}(n)$ space bound achieved by our coresets above is considered a "sweet spot" for graph streaming algorithms [28, 52] as many fundamental problems are provably intractable in $o(n)$ space (sometimes not enough to even store the answer) while admit efficient solutions in $\widetilde{O}(n)$ space. However, in the simultaneous model, these considerations imply only that the total size of all $k$ coresets must be $\Omega(n)$, leaving open the possibility that coreset output by each machine may be as small as $\widetilde{O}(n/k)$ in size (similar in spirit to coresets of [50]). Our next result rules out this possibility and proves the optimality of our coresets size.

Result 2. *Any $\alpha$-approximation randomized coreset for the maximum matching problem must have size $\Omega(n/\alpha^2)$, and any $\alpha$-approximation randomized coreset for the vertex cover problem must have size $\Omega(n/\alpha)$.*

Result 2 is formalized in Section 4. We now elaborate on some applications of our results.

*Distributed Computation.* We use the following distributed computation model in this paper, referred to as the *coordinator model* (see [59]). The input is distributed across $k$ machines. There is also an additional party called the *coordinator* who receives no input. The machines are allowed to only communicate with the coordinator, not with each other. A protocol in this model is called a *simultaneous* protocol iff the machines simultaneously send a message to the coordinator and the coordinator then outputs the answer with no further interaction. *Communication cost* of a protocol in this model is the total number of bits communicated by all parties.

Result 1 can also be used to design simultaneous protocols for matching and vertex cover with $\widetilde{O}(nk)$ total communication and the same approximation guarantee stated in Result 1 in the case the input is partitioned randomly across $k$ machines. Indeed, each machine only needs to compute a coreset of its input, sends it to the coordinator, and coordinator computes an exact maximum matching or a 2-approximate minimum vertex cover on the union

[2] The authors in [11] only showed the inapproximability result for the matching problem. However, a simple modification of their result proves an identical lower bound for the vertex cover problem as well.

of the coresets. We further prove that the communication cost of theses protocols are essentially optimal.

> Result 3. *Any $\alpha$-approximation simultaneous protocol for the maximum matching problem, resp. the vertex cover problem, requires total communication of $\Omega(nk/\alpha^2)$ bits, resp. $\Omega(nk/\alpha)$ bits,* even *when the input is* partitioned randomly *across the machines.*

Proof of Result 3 is deferred to the full version of the paper [9]. We point out that Result 3 is in fact a strengthening of Result 2; it rules out *any* representative summary (not necessarily a randomized coreset) of size $o(n/\alpha^2)$ (resp. $o(n/\alpha)$) that can be used for $\alpha$-approximation of matching (resp. vertex cover) when the input is partitioned randomly.

For the matching problem, it was shown previously in [33] that when the input is adversarially partitioned in the coordinator model, any protocol (not necessarily simultaneous) requires $\Omega(nk/\alpha^2)$ bits of communication to achieve an $\alpha$-approximation of the maximum matching. Result 3 extends this to the case of *randomly partitioned* inputs albeit only for simultaneous protocols.

*MapReduce Framework.* We show how to use our randomized coresets to obtain improved MapReduce algorithms for matching and vertex cover in the MapReduce computation model formally introduced in [40, 44]. Let $k = \sqrt{n}$ be the number of machines, each with a memory of $\widetilde{O}(n\sqrt{n})$; we show that *two* rounds of MapReduce suffice to obtain an $O(1)$-approximation for matching and $O(\log n)$-approximation for vertex cover. In the first round, each machine randomly partitions the edges assigned to it across the $k$ machines; this results in a random $k$-partitioning of the graph across the machines. In the second round, each machine sends a randomized composable coreset of its input to a designated central machine $M$; as there are $k = \sqrt{n}$ machines and each machine is sending $\widetilde{O}(n)$ size coreset, the input received by $M$ is of size $\widetilde{O}(n\sqrt{n})$ and hence can be stored entirely on that machine. Finally, $M$ computes the answer by combining the coresets (similar to the case in the coordinator model). Note that if the input was distributed randomly in the first place, we could have implemented this algorithm in only one round of MapReduce (see [50] for details on when this assumption applies).

Our MapReduce algorithm outperforms the previous algorithms of [44] for matching and vertex cover in terms of the number of rounds it uses, albeit with a larger approximation guarantee. In particular, [44] achieved a 2-approximation to both matching and vertex cover in 6 rounds of MapReduce when using similar space as ours on each machine (the number of rounds of this algorithm is always at least 3 even if we allow $\widetilde{O}(n^{5/3})$ space per each machine). The improvement on the number of rounds is significant in this context; the transition between different rounds in a MapReduce computation is usually the dominant cost of the computation [44] and hence, minimizing the number of rounds is an important goal in the MapReduce framework.

## 1.2 Our Techniques

*Randomized Coreset for Matching.* Greedy and Local search algorithms are the typical choices for composable coresets (see, e.g., [34, 50]). It is then natural to consider the greedy algorithm for the maximum matching problem as a randomized coreset: the one that computes a *maximal matching*. However, one can easily show that this choice of coreset performs poorly in general; there are simple instances in which choosing arbitrary maximal matching in the graph $G^{(i)}$ results only in an $\Omega(k)$-approximation.

Somewhat surprisingly, we show that a simple change in strategy results in an efficient randomized coreset: *any maximum matching* of the graph $G^{(i)}$ can be used as an $O(1)$-approximate randomized coreset for the maximum matching problem. Unlike the previous work in [34, 50] that relied on analyzing a specific algorithm (or a specific family of algorithms) for constructing a coreset, we prove this result by exploiting structural properties of the maximum matching (i.e., the optimal solution) directly, independent of the algorithm that computes it. As a consequence, our coreset construction requires no prior coordination (such as consistent tie-breaking rules used in [50]) between the machines and in fact each machine can use a different algorithm for computing the maximum matching required by the coreset.

*Randomized Coreset for Vertex Cover.* In the light of our coreset for the matching problem, one might wonder whether a minimum vertex cover of a graph can also be used as its randomized coreset. However, it is easy to show that the answer is negative here – there are simple instances (e.g., a star on $k$ vertices) on which this leads to an $\Omega(k)$ approximation ratio. Indeed, the *feasibility constraint* in the vertex cover problem depends heavily on the input graph as a whole and not only the coreset computed by each machine, unlike the case for matching and in fact most problems that admit a composable coreset [13, 34, 50]. This suggests the necessity of using edges in the coreset to *certify* the feasibility of the answer. On the other hand, only sending edges seems too restrictive: a vertex of degree $n-1$ can safely be assumed to be in an optimal vertex cover, but to certify this, one needs to essentially communicate $\Omega(n)$ edges. This naturally motivates a slightly more general notion of coresets – the coreset contains both subsets of vertices (to be always included in the final vertex cover) and edges (to guide the choice of additional vertices in the vertex cover).

To obtain a randomized coreset for vertex cover, we employ an iterative "peeling" process where we remove the vertices with the highest residual degree in each iteration (and add them to the final vertex cover) and continue until the residual graph is sufficiently sparse, in which case we can return this subgraph as the coreset. The process itself is a modification of the algorithm by Parnas and Ron [57]; we point out that other modifications of this algorithm has also been used previously for matching and vertex cover [16, 36, 56].

However, to employ this algorithm as a coreset we need to argue that the set of vertices peeled across different machines is not too large as these vertices are added directly to the final vertex cover. The intuition behind this is that random partitioning of edges in the graph should result in vertices to have essentially the same degree across the machines and hence each machine should peel the same set of vertices in each iteration. But this intuition runs into a technical difficulty: the peeling process is quite sensitive to the exact degree of vertices and even slight changes in degree results in moving vertices between different iterations that potentially leads to a cascading effect. To address this, we design a *hypothetical* peeling process (which is aware of the actual minimum vertex cover in $G$) and show that the our actual peeling process is in fact "sandwiched" between two application of this peeling process with different degree threshold for peeling vertices. We then use this to argue that the set of all vertices peeled across the machines are always contained in the solution of the hypothetical peeling process which in turn can be shown to be a relatively small set.

*Lower Bounds for Randomized Coresets.* Our lower bound results for randomized coresets for matching are based on the following simple distribution: the input graph consists of union of two bipartite graphs, one of which is a random $k$-regular graph $G_1$ with $n/2\alpha$ vertices on each side while the other graph $G_2$ is a perfect matching of size $n - n/2\alpha$. Thus the input graph almost certainly contains a matching of size $n - o(n)$ and any $\alpha$-approximate solution must collect $\Omega(n/\alpha)$ edges from $G_2$ overall i.e. $\Omega(n/\alpha k)$ edges from $G_2$ from each machine on average. After random partitioning, the input given to each machine is essentially a matching of size $n/2\alpha$ from $G_1$ and a matching of size roughly $n/k$ from $G_2$. The local information at each machine is not sufficient to differentiate between edges of $G_1$ and $G_2$, and thus any coreset that aims to include $\Omega(n/\alpha k)$ edges from $G_2$, can not reduce the input size by more than a factor of $\alpha$. Somewhat similar ideas can also be shown to work for the vertex cover problem.

*Communication Complexity Lower Bounds.* We briefly highlight the ideas used in obtaining the lower bounds described in Result 3. We will focus on the vertex cover problem to describe our techniques. Our lower bound result is based on analyzing (a variant of) the following distribution: the input graph $G(L, R, E)$ consists of a bipartite graph $G_1$ plus a single edge $e^\star$. $G_1$ is a graph on $n/2\alpha$ vertices $L_1 \subseteq L$, each connected to $k$ random neighbors in $R$, and $e^\star$ is an edge chosen uniformly at random between $L \setminus L_1$ and $R$. This way $G$ admits a minimum vertex cover of size at most $n/2\alpha + 1$. However, when this graph is randomly partitioned, the input to each machine is essentially a matching of size $n/2\alpha$ chosen from the graph $G_1$ with possibly one more edge $e^\star$ (in exactly one machine chosen uniformly at random). The local information at the machine receiving the edge $e^\star$ is not sufficient to differentiate between the edges of $G_1$ and $e^\star$ and thus if the message sent by this machine is much smaller than its input size (i.e., $o(n/\alpha)$ bits), it most likely does not "convey enough information" to the coordinator about the identity of $e^\star$. This in turn forces the coordinator to use more than $n/2$ vertices in order to cover $e^\star$, resulting in an approximation factor larger than $\alpha$.

Making this intuition precise is complicated by the fact that the input across the players are highly correlated, and hence the message sent by one player, can also reveal extra information about the input of another (e.g. a relatively small communication from the players is enough for the coordinator to know the identity of entire $L_1$). To overcome this, we show that by conditioning on proper parts of the input, we can limit the correlation in the input of players and then use the *symmetrization* technique of [59] to reduce the simultaneous $k$-player vertex cover problem to a one-way two-player problem named the *hidden vertex problem* (HVP). Loosely speaking, in HVP, Alice and Bob are given two sets $S, T \subseteq [n]$, each of size $n/\alpha$, with the promise that $|S \setminus T| = 1$ and their goal is to find a set $C$ of size $o(n)$ which contains the single element in $S \setminus T$. We prove a lower bound of $\Omega(n/\alpha)$ bits for this problem using a subtle reduction from the well-known set disjointness problem. In this reduction, Alice and Bob use the protocol for HVP on "non-legal" instances (i.e., the ones for which HVP is not well-defined) to reduce the original disjointness instance between sets $A, B$ on a universe $[N]$ to a lopsided disjointness instance $(A, B')$ whereby $|B'| = o(N)$, and then solve this new instance in $o(N)$ communication (using the Håstad-Wigderson protocol [32]), contradicting the $\Omega(N)$ lower bound on the communication complexity of disjointness.

The lower bound for the matching problem is also proven along similar lines (over the hard distribution mentioned earlier for this problem) using a careful combinatorial argument instead of the reduction from the disjointness problem.

## 1.3 Further Related Work

Maximum matching and minimum vertex cover are among the most studied problems in the context of massive graphs including, in dynamic graphs [14, 53, 56, 60], sub-linear algorithms [31, 54, 55, 57, 62], streaming algorithms [3–6, 10, 11, 20–22, 24–30, 35, 36, 41, 42, 46, 47, 49, 58], MapReduce computation [5, 44], and different distributed computation models [8, 23, 30, 33]. Most relevant to our work are the linear sketches of [20] for computing an *exact* minimum vertex cover or maximum matching in $O(\text{opt}^2)$ space (opt is the size of the solution), and linear sketches of [11, 20] for $\alpha$-approximating maximum matching in $\widetilde{O}(n^2/\alpha^3)$ space. These results are proven to be tight by [21], and [11], respectively. Finally, [11] also studied the simultaneous communication complexity of bipartite matching in the vertex-partition model and proved that obtaining better than an $O(\sqrt{k})$-approximation in this model requires strictly more than $\widetilde{O}(n)$ communication from each player.

Coresets, composable coresets, and randomized composable coresets are respectively introduced in [2], [34], and [50]. Composable coresets have been used previously in the context of nearest neighbor search [1], diversity maximization [34], clustering [13, 15], and submodular maximization [12, 34, 50]. Moreover, while not particularly termed a composable coreset, the "merge and reduce" technique in the graph streaming literature (see [47], Section 2.2) is identical to composable coresets. Similar ideas as randomized coreset for optimization problems has also been used in random arrival streams [36, 42]. Moreover, communication complexity lower bounds have also been studied previously under the random partitioning of the input [19, 37].

# 2 PRELIMINARIES

*Notation.* For any integer $m$, $[m] := \{1, \ldots, m\}$. Let $G(V, E)$ be a graph; $\mathsf{MM}(G)$ denotes the maximum matching size in $G$ and $\mathsf{VC}(G)$ denotes the minimum vertex cover size. We assume that these quantities are $\omega(k \log n)^3$. For a set $S \subseteq V$ and $v \in V$, $N_S(v) \subseteq S$ denotes the neighbors of $v$ in the set $S$. For an edge set $E' \subseteq E$, we use $V(E')$ to refer to vertices incident on $E'$.

*Communication Complexity.* We prove our lower bounds for distributed protocols using the framework of communication complexity, and in particular in the *multi-party simultaneous communication model* and the *two-player one-way communication model* (see, e.g., [43]).

[3] Otherwise, we can use the algorithm of [20] to obtain *exact* coresets of size $\widetilde{O}(k^2)$ as mentioned in Section 1.3.

Formally, in the multi-party simultaneous communication model, the input is partitioned across $k$ players $P^{(1)}, \ldots, P^{(k)}$. All players have access to an infinite shared string of random bits, referred to as *public randomness* (or *public coins*). The goal is for the players to compute a specific function of the input by simultaneously sending a message to a central party called the coordinator (or the referee). The coordinator then needs to output the answer using the messages received by the players. We refer to the case when the input is partitioned randomly as the *random partition* model.

In the two-player one-way communication model, the input is partitioned across two players, namely Alice and Bob. The players again have access to public randomness, and the goal is for Alice to send a single message to Bob, so that Bob can compute a function of the joint input. The *communication cost* of a protocol in both models is the total length of the messages sent by the players.

# 3 RANDOMIZED CORESETS FOR MATCHING AND VERTEX COVER

We present our randomized composable coresets for matching and vertex cover in this section, formalizing Result 1.

## 3.1 A Randomized Coreset for Matching

The following theorem formalizes Result 1 for matching.

Theorem 1. *Any* maximum matching *of a graph $G(V, E)$ is an $O(1)$-approximation randomized composable coreset of size $O(n)$ for the maximum matching problem.*

We remark that our main interest in Theorem 1 is to achieve *some* constant approximation factor for randomized composable coresets of the matching problem and as such we did not optimize the constant in the approximation ratio. Nevertheless, our result already shows that the approximation ratio of this coreset is *at most* 9 (with more care, we can reduce this factor down to 8; however, as this is not the main contribution of this paper, we omit the details).

Let $G(V, E)$ be any graph and $G^{(1)}, \ldots, G^{(k)}$ be a random $k$-partitioning of $G$. To prove Theorem 1, we describe a simple process for combining the maximum matchings (i.e., the coresets) of $G^{(i)}$'s, and prove that this process results in a constant factor approximation of the maximum matching of $G$; this process is only required for the analysis, i.e., to show that there exists a large matching in the union of coresets; in principle, any (approximation) algorithm for computing a maximum matching can be applied to obtain a large matching from the coresets.

Consider the following greedy process for computing an approximate matching in $G(V, E)$:

```
GreedyMatch($G$):
  (1) Let $M^{(0)} := \emptyset$. For $i = 1$ to $k$:
  (2) Let $M^{(i)}$ be a maximal matching obtained by adding to
      $M^{(i-1)}$ the edges in an arbitrary maximum matching of
      $G^{(i)}$ that do not violate the matching property.
  (3) return $M := M^{(k)}$.
```

Lemma 3.1. GreedyMatch *is an $O(1)$-approximation algorithm for the maximum matching problem w.h.p (over the randomness of the edge partitioning).*

Before proving Lemma 3.1, we show that Theorem 1 easily follows from this lemma.

Proof of Theorem 1. Let ALG be any algorithm that given a graph $G(V, E)$, ALG$(G)$ outputs an arbitrary maximum matching of $G$. It is immediate to see that to implement GreedyMatch, we only need to compute a maximal matching on the output of ALG on each graph $G^{(i)}$ where $G^{(i)}$'s form a random $k$-partitioning of $G$. Consequently, since GreedyMatch outputs an $O(1)$-approximate matching (by Lemma 3.1), the graph $H := G^{(1)} \cup \ldots \cup G^{(k)}$ should contain an $O(1)$-approximate matching as well. We emphasize here that the use of GreedyMatch for finding a large matching in $H$ is *only* for the purpose of analysis. □

In the rest of this section, we prove Lemma 3.1. Recall that MM$(G)$ denotes the maximum matching size in the input graph $G$. Let $c > 0$ be a small constant to be determined later. To prove Lemma 3.1, we will show that $\left|M^{(k)}\right| \geq c \cdot \text{MM}(G)$ w.h.p, where $M^{(k)}$ is the output of GreedyMatch. Notice that the matchings $M^{(i)}$ (for $i \in [k]$) constructed by GreedyMatch are random variables depending on the random $k$-partitioning.

Our general approach for the proof of Lemma 3.1 is as follows. Suppose at the beginning of the $i$-th step of GreedyMatch, the matching $M^{(i-1)}$ is of size $o(\text{MM}(G))$. It is easy to see that in this case, there is a matching of size $\Omega(\text{MM}(G))$ in $G$ that is entirely incident on vertices of $G$ that are not matched by $M^{(i-1)}$. We can further show that in fact $\Omega(\text{MM}(G)/k)$ edges of this matching are appearing in $G^{(i)}$, *even* when we condition on the assignment of the edges in the first $(i-1)$ graphs. The next step is then to argue that the existence of these edges forces *any* maximum matching of $G^{(i)}$ to match $\Omega(\text{MM}(G)/k)$ edges in $G^{(i)}$ between the vertices that are not matched by $M^{(i-1)}$; these edges can always be added to the matching $M^{(i-1)}$ to form $M^{(i)}$. This ensures that while the maximal matching in GreedyMatch is of size $o(\text{MM}(G))$, we can increase its size by $\Omega(\text{MM}(G)/k)$ edges in each of the first $k/3$ steps, hence obtaining a matching of size $\Omega(\text{MM}(G))$ at the end. The following key lemma formalizes this argument.

Lemma 3.2. *For any $i \in [k/3]$, if $\left|M^{(i-1)}\right| \leq c \cdot \text{MM}(G)$, then, w.p. $1 - O(1/n)$,*

$$\left|M^{(i)}\right| \geq \left|M^{(i-1)}\right| + \left(\frac{1 - 6c - o(1)}{k}\right) \cdot \text{MM}(G)$$

To continue we define some notation. Let $M^{\star}$ be an arbitrary maximum matching of $G$. For any $i \in [k]$, we define $M^{\star<i}$ as the part of $M^{\star}$ assigned to the first $i-1$ graphs in the random $k$-partitioning, i.e., the graphs $G^{(1)}, \ldots, G^{(i-1)}$. We have the following simple concentration result (the proof is deferred to the full version [9]).

Claim 3.3. *W.p. $1 - O(1/n)$, for any $i \in [k]$,*

$$\left|M^{\star<i}\right| \leq \left(\frac{i - 1 + o(i)}{k}\right) \cdot \text{MM}(G).$$

We now prove Lemma 3.2.

Proof of Lemma 3.2. Fix an $i \in [k/3]$ and the set of edges for $E^{(1)}, \ldots, E^{(i-1)}$; this also fixes the matching $M^{(i-1)}$ while the set of edges in $E^{(i)}, \ldots, E^{(k)}$ together with the matching $M^{(i)}$ are still random variables. We further assume that after fixing the edges in $E^{(1)}, \ldots, E^{(i-1)}$, $\left|M^{\star<i}\right| \leq \frac{i-1+o(i)}{k} \cdot \mathsf{MM}(G)$ which happens w.p. $1 - O(1/n)$ by Claim 3.3.

We first define some notation. Let $V_{\mathsf{old}}$ be the set of vertices incident on $M^{(i-1)}$ and $V_{\mathsf{new}}$ be the remaining vertices. Let $E^{\geq i}$ be the set of edges in $E \setminus \left(E^{(1)} \cup \ldots \cup E^{(i-1)}\right)$. We partition $E^{\geq i}$ into two parts: $(i)$ $E_{\mathsf{old}}$: the set of edges with *at least one endpoint* in $V_{\mathsf{old}}$, and $(ii)$ $E_{\mathsf{new}}$: the set of edges *incident entirely* on $V_{\mathsf{new}}$. Our goal is to show that w.h.p. *any* maximum matching of $G^{(i)}$ matches $\Omega(\mathsf{MM}(G)/k)$ vertices in $V_{\mathsf{new}}$ to each other by using the edges in $E_{\mathsf{new}}$; the lemma then follows easily from this.

Notice that the edges in the graph $G^{(i)}$ are chosen by independently assigning each edge in $E^{\geq i}$ to $G^{(i)}$ w.p. $1/(k-i+1)$[4]. This independence allows us to treat the edges in $E_{\mathsf{old}}$ and $E_{\mathsf{new}}$ separately; we can fix the set of sampled edges of $G^{(i)}$ in $E_{\mathsf{old}}$ denoted by $E^i_{\mathsf{old}}$ without changing the distribution of edges in $G^{(i)}$ chosen from $E_{\mathsf{new}}$. Let $\mu_{\mathsf{old}} := \mathsf{MM}(G(V, E^i_{\mathsf{old}}))$, i.e., the maximum number of edges that can be matched in $G^{(i)}$ using only the edges in $E^i_{\mathsf{old}}$. In the following, we show that w.h.p., there exists a matching of size $\mu_{\mathsf{old}} + \Omega(\mathsf{MM}(G)/k)$ in $G^{(i)}$; by the definition of $\mu_{\mathsf{old}}$, this implies that *any* maximum matching of $G^{(i)}$ has to use at least $\Omega(\mathsf{MM}(G)/k)$ edges in $E_{\mathsf{new}}$, proving the lemma.

Let $M_{\mathsf{old}}$ be any arbitrary maximum matching of size $\mu_{\mathsf{old}}$ in $G(V, E^i_{\mathsf{old}})$. Let $V_{\mathsf{new}}(M_{\mathsf{old}})$ be the set of vertices in $V_{\mathsf{new}}$ that are incident on $M_{\mathsf{old}}$. We show that there is a large matching in $G(V, E_{\mathsf{new}})$ that avoids $V_{\mathsf{new}}(M_{\mathsf{old}})$.

Claim 3.4. *There exists a matching in $G(V, E_{\mathsf{new}})$ of size*

$$\left(\frac{k-i+1-o(i)}{k} - 4c\right) \cdot \mathsf{MM}(G)$$

*that avoids the vertices of $V_{\mathsf{new}}(M_{\mathsf{old}})$.*

Proof. We first bound the size of $V_{\mathsf{new}}(M_{\mathsf{old}})$. Since any edge in $M_{\mathsf{old}}$ has at least one endpoint in $V_{\mathsf{old}}$, we have $|V_{\mathsf{new}}(M_{\mathsf{old}})| \leq |M_{\mathsf{old}}| \leq |V_{\mathsf{old}}|$. By the assertion of the lemma, $\left|M^{(i-1)}\right| < c \cdot \mathsf{MM}(G)$, and hence $|V_{\mathsf{new}}(M_{\mathsf{old}})| \leq |V_{\mathsf{old}}| < 2c \cdot \mathsf{MM}(G)$.

Moreover, by the assumption that $\left|M^{\star<i}\right| \leq \frac{i-1+o(i)}{k} \cdot \mathsf{MM}(G)$, there is a matching $M$ of size $\frac{k-i+1-o(i)}{k} \cdot \mathsf{MM}(G)$ in the graph $G(V, E^{\geq i})$. By removing the edges in $M$ that are either incident on $V_{\mathsf{old}}$ or $V_{\mathsf{new}}(M_{\mathsf{old}})$, at most $4c \cdot \mathsf{MM}(G)$ edges are removed from $M$. Now the remaining matching is entirely contained in $E_{\mathsf{new}}$ and also avoids $V_{\mathsf{new}}(M_{\mathsf{old}})$, hence proving the claim. □

We are now ready to finalize the proof. Let $M_{\mathsf{new}}$ be the matching guaranteed by Claim 3.4. Each edge in this matching is chosen in $G^{(i)}$ w.p. $1/(k-i+1)$ independent of the other edges; hence, by Chernoff bound (and the assumption that $\mathsf{MM}(G) = \omega(k \log n)$), there is a matching of size

$$(1-o(1)) \cdot \left(\frac{1}{k} - \frac{o(i)}{k(k-i+1)} - \frac{4c}{k-i+1}\right) \cdot \mathsf{MM}(G)$$

$$\geq \left(\frac{1-6c-o(1)}{k}\right) \cdot \mathsf{MM}(G) \quad (i \leq k/3)$$

in the edges of $M_{\mathsf{new}}$ that appear in $G^{(i)}$. This matching can be directly added to the matching $M_{\mathsf{old}}$, implying the existence of a matching of size $\mu_{\mathsf{old}} + \left(\frac{1-6c-o(1)}{k}\right) \cdot \mathsf{MM}(G)$ in $G^{(i)}$. As argued before, this ensures that any maximum matching of $G^{(i)}$ contains at least $\left(\frac{1-6c-o(1)}{k}\right) \cdot \mathsf{MM}(G)$ edges in $E_{\mathsf{new}}$. These edges can always be added to $M^{(i-1)}$ to form $M^{(i)}$, hence proving the lemma. □

Proof of Lemma 3.1. Recall that $M := M^{(k)}$ is the output matching of GreedyMatch. For the first $k/3$ steps of GreedyMatch, if at any step we obtained a matching of size $c \cdot \mathsf{MM}(G)$, then we are already done. Otherwise, at each step, by Lemma 3.2, w.p. $1 - O(1/n)$, we increase the size of the maximal matching by $\left(\frac{1-6c-o(1)}{k}\right) \cdot \mathsf{MM}(G)$ edges; consequently, by taking a union bound on the $k/3$ steps, w.p. $1 - o(1)$, the size of the maximal matching would be $\left(\frac{1-6c-o(1)}{3}\right) \cdot \mathsf{MM}(G)$. By picking $c = 1/9$, we ensure that in either case, the matching computed by GreedyMatch is of size at least $\mathsf{MM}(G)/9 - o(\mathsf{MM}(G))$, proving the lemma. □

## 3.2 A Randomized Coreset for Vertex Cover

The following theorem formalizes Result 1 for vertex cover.

Theorem 2. *There exists an $O(\log n)$-approximation randomized composable coreset of size $O(n \log n)$ for the vertex cover problem.*

Let $G(V, E)$ be a graph and $G^{(1)}, \ldots, G^{(k)}$ be a random $k$-partitioning of $G$; we propose the following coreset for computing an approximate vertex cover of $G$. This coreset construction is a modification of the algorithm for vertex cover first proposed by [57].

VC-Coreset($G^{(i)}$). An algorithm for computing a composable coreset of each $G^{(i)}$.

1. Let $\Delta$ be the smallest integer such that $n/(k \cdot 2^{\Delta}) \leq 4 \log n$ and define $G^{(i)}_1 := G^{(i)}$.
2. For $j = 1$ to $\Delta - 1$, let:
$$V^{(i)}_j := \left\{\text{vertices of degree} \geq n/(k \cdot 2^{j+1}) \text{ in } G^{(i)}_j\right\}$$
$$G^{(i)}_{j+1} := G^{(i)}_j \setminus V^{(i)}_j.$$
3. Return $V^{(i)}_{\mathsf{cs}} := \bigcup_{j=1}^{\Delta-1} V^{(i)}_j$ as a *fixed solution* plus the graph $G^{(i)}_{\Delta}$ as the coreset.

In VC-Coreset we allow the coreset to, in addition to returning a subgraph, identify a set of vertices (i.e., $V^{(i)}_{\mathsf{cs}}$) to be added directly to the final vertex cover. In other words, to compute a vertex cover of the graph $G$, we compute a vertex cover of the graph $\bigcup_{i=1}^{k} G^{(i)}_{\Delta}$ and return it together with the vertices $\bigcup_{i=1}^{k} V^{(i)}_{\mathsf{cs}}$. It is easy to see that this set of vertices indeed forms a vertex cover of $G$: any edge in $G$ that belongs to $G^{(i)}$ is either incident on some $V^{(i)}_j$, and hence

[4] This is true even when we condition on the size of $\left|M^{\star<i}\right|$ since this event does not depend on the choice of edges in $E^{\geq i}$.

is covered by $V_j^{(i)}$, or is present in $G_\Delta^{(i)}$, and hence is covered by the vertex cover of $G_\Delta^{(i)}$.

In the rest of this section, we bound the approximation ratio of this coreset. To do this, we need to prove that $\left|\bigcup_{i=1}^{k} V_{\mathrm{cs}}^{(i)}\right| = O(\log n)\cdot \mathrm{VC}(G)$. The bound on the ratio then follows as the vertex cover of $\bigcup_{i=1}^{k} G_\Delta^{(i)}$ can be computed to within a factor of 2.

It is easy to prove (and follows from [57]) that the set of vertices $V_{\mathrm{cs}}^{(i)}$ is of size $O(\log n)\cdot \mathrm{VC}(G)$; however, using this fact directly to bound the size of $\bigcup_{i=1}^{k} V_{\mathrm{cs}}^{(i)}$ only implies an approximation ratio of $O(k\log n)$ which is far worse than our goal of achieving an $O(\log n)$-approximation. In order to obtain the $O(\log n)$ bound, we need to argue that not only each set $V_{\mathrm{cs}}^{(i)}$ is relatively small, but also that these sets are all intersecting in many vertices. In order to do so, we introduce a hypothetical algorithm (similar to VC-Coreset) on the graph $G$ and argue that the set $V_{\mathrm{cs}}^{(i)}$ output by VC-Coreset$(G^{(i)})$ is, with high probability, a subset of the output of this hypothetical algorithm. This allows us to then bound the size of the union of the sets $V_{\mathrm{cs}}^{(i)}$ for $i \in [k]$.

Let $O^\star$ denote the set of vertices in an arbitrary optimum vertex cover of $G$ and $\overline{O^\star} := V \setminus O^\star$. Consider the following process on the original graph $G$ (defined only for analysis):

```
(1) Let $G_1$ be the bipartite graph obtained from $G$ by removing
    edges between vertices in $O^\star$.
(2) For $j = 1$ to $t := \lceil \log n \rceil$, let:
        $O_j := \left\{\text{vertices in } O^\star \text{ of degree} \geq n/2^j \text{ in } G_j\right\}$
        $\overline{O}_j := \left\{\text{vertices in } \overline{O^\star} \text{ of degree} \geq n/2^{j+2} \text{ in } G_j\right\}$
        $G_{j+1} := G_j \setminus (O_j \cup \overline{O}_j)$.
```

We first prove that the sets $O_j$'s and $\overline{O}_j$'s in this process form an $O(\log n)$ approximation of the minimum vertex cover of $G$ and then show that VC-Coreset$(G^{(i)})$ (for any $i \in [k]$) is *mimicking* this hypothetical process in a sense that the set $V_{\mathrm{cs}}^{(i)}$ is essentially *contained* in the union of the sets $O_j$'s and $\overline{O}_j$'s.

Lemma 3.5. $\left|\bigcup_{j=1}^{t} O_j \cup \overline{O}_j\right| = O(\log n)\cdot \mathrm{VC}(G)$.

Proof. Fix any $j \in [t]$; we prove that $\overline{O}_j \leq 8\cdot\mathrm{VC}(G)$. The lemma follows from this since there are at most $O(\log n)$ different sets $\overline{O}_j$ and the union of the sets $O_j$'s is a subset of $O^\star$ (with size $\mathrm{VC}(G)$).

Consider the graph $G_j$. The maximum degree in this graph is at most $n/2^{j-1}$ by the definition of the process. Since all the edges in the graph are incident on at least one vertex of $O^\star$, there can be at most $\left|O^\star\right|\cdot n/2^{j-1}$ edges between the remaining vertices in $O^\star$ and $\overline{O^\star}$ in $G_j$. Moreover, any vertex in $\overline{O}_j$ has degree at least $n/2^{j+2}$ by definition and hence there can be at most $\left(\left|O^\star\right|\cdot n/2^{j-1}\right) / \left(n/2^{j+2}\right) \leq 8\left|O^\star\right| = 8\cdot\mathrm{VC}(G)$ vertices in $\overline{O}_j$, proving the claim. □

We now prove the main relation between the sets $O_j$'s and $\overline{O}_j$'s defined above and the intermediate sets $V_j^{(i)}$'s computed by VC-Coreset$(G^{(i)})$. The following lemma is the heart of the proof.

Lemma 3.6. *Fix an* $i \in [k]$, *and let* $A_j = V_j^{(i)} \cap O^\star$ *and* $B_j = V_j^{(i)} \cap \overline{O^\star}$. *With probability* $1 - O(1/n)$, *for any* $t \in [\Delta]$*:*

*(1)* $\bigcup_{j=1}^{t} A_j \supseteq \bigcup_{j=1}^{t} O_j$.
*(2)* $\bigcup_{j=1}^{t} B_j \subseteq \bigcup_{j=1}^{t} \overline{O}_j$.

Proof. To simplify the notation, for any $t \in [\Delta]$, we let $A_{<t} = \bigcup_{j=1}^{t-1} A_j$ and $A_{\geq t} = \bigcup_{j=t}^{\Delta} A_j$ (and similarly for $B_j$'s, $O_j$'s, and $\overline{O}_j$'s). We also use $N_S(v)$ to denote the neighbor-set of the vertex $v$ in the set $S \subseteq V$.

Note that the vertex-sets of the graphs $G$ and $G^{(i)}$ are the same and we can "project" the sets $O_j$'s and $\overline{O}_j$'s on graph $G^{(i)}$ as well. In other words, we can say a vertex $v$ in $G^{(i)}$ belongs to $O_j$ iff $v \in O_j$ in the original graph $G$. In the following claim, we crucially use the fact that the graph $G^{(i)}$ is obtained from $G$ by sampling each edge w.p. $1/k$ to prove that the degree of vertices across different sets $O_j$'s (and $\overline{O}_j$'s) in $G^{(i)}$ are essentially the same as in $G$ (up to the scaling factor of $1/k$).

Claim 3.7. *For any* $j \in [\Delta]$*:*

- *For any vertex* $v \in O_j$, $\left|N_{\overline{O}_{\geq j}}(v)\right| \geq n/(k\cdot 2^{j+1})$ *in the graph* $G^{(i)}$ *w.p.* $1 - O(1/n^2)$.
- *For any vertex* $v \in \overline{O}_{\geq j+1}$, $\left|N_{O_{\geq j}}(v)\right| < n/(k\cdot 2^{j+1})$ *in the graph* $G^{(i)}$ *w.p.* $1 - O(1/n^2)$.

We defer the proof of Claim 3.7 to the full version of the paper [9]. By a union bound on the $n$ vertices in $G$, the statements in Claim 3.7 hold for all vertices of $G$ w.p. $1 - O(1/n)$; in the following we condition on this event. We now prove Lemma 3.6 by induction.

Let $v$ be a vertex that belongs to $O_1$; we prove that $v$ belongs to the set $V_1^{(i)}$ of VC-Coreset, i.e., $v \in A_1$. By Claim 3.7 (for $j = 1$), the degree of $v$ in $G_1^{(i)}$ is at least $n/4k$. Note that in $G_1^{(i)}$, $v$ may also have edges to other vertices in $O^\star$ but this can only increase the degree of $v$. This implies that $v$ also belongs to $A_1$ by the threshold chosen in VC-Coreset. Similarly, let $u$ be a vertex in $\overline{O}_{\geq 2}$ (i.e., *not* in $\overline{O}_1$); we show that $u$ is not chosen in $V_1^{(i)}$, implying that $B_1$ can only contain vertices in $\overline{O}_1$. By Claim 3.7, degree of $u$ in $G_1^{(i)}$ is less than $n/4k$. This implies that $u$ does not belong to $B_1$. In summary, we have $O_1 \subseteq A_1$ and $B_1 \subseteq \overline{O}_1$.

Now consider some $t > 1$ and let $v$ be a vertex in $O_t$. By induction, $B_{<t} \subseteq \overline{O}_{<t}$. This implies that the degree of $v$ to $B_{\geq t}$ is at least as large as its degree to $O_{\geq t}$. Consequently, by Claim 3.7 (for $j = t$), degree of $v$ in the graph $G_t^{(i)}$ is at least $n/(k\cdot 2^{t+1})$ and hence $v$ also belongs to $A_t$. Similarly, fix a vertex $u$ in $\overline{O}_{\geq t+1}$. By induction, $A_{<t} \supseteq O_{<t}$ and hence the degree of $u$ to $A_{\geq t}$ is at most as large as its degree to $O_{\geq t}$; note that since $O^\star$ is a vertex cover, $u$ does not have any other edge in $G_t^{(i)}$ except for the ones to $A_{\geq t}$. We can now argue as before that $u$ does not belong to $B_t$. □

Proof of Theorem 2. The bound on the coreset size follows immediately from the fact that the graph $G_\Delta^{(i)}$ contains at most $O(n\log n)$ edges and size of $V_{\mathrm{cs}}^{(i)}$ is at most $n$. As argued before, to prove the bound on the approximation ratio, we only need to show that $\bigcup_{i=1}^{k} V_{\mathrm{cs}}^{(i)}$ is of size $O(\log n)\cdot\mathrm{VC}(G)$. Let $A^{(i)} = V_{\mathrm{cs}}^{(i)} \cap O^\star$

and $B^{(i)} = V_{\text{cs}}^{(i)} \cap \overline{O^\star}$; clearly, each $A^{(i)} \subseteq O^\star$ and moreover, by Lemma 3.6 (for $t = \Delta$), each $B^{(i)} \subseteq \cup_{j=1}^{\Delta} \overline{O}_j$. Consequently, $\left|\bigcup_{i=1}^{k} V_{\text{cs}}^{(i)}\right| \leq \left|O^\star\right| + \left|\bigcup_{j=1}^{\Delta} \overline{O}_j\right| \leq O(\log n) \cdot \mathsf{VC}(G)$, where the last inequality is by Lemma 3.5. □

# 4 LOWER BOUNDS

We formalize Result 2 in this section. As argued earlier, Result 2 is a special case of Result 3 and hence follows directly from that result; however, as the proof of Result 3 is rather technical and complicated, we instead provide a self-contained proof of Result 2 that is easier to present and conveys some of the main ideas behind Result 3, and postpone the proof of Result 3 to the full version [9].

## 4.1 A Lower Bound for Randomized Composable Coresets of Matching

The following theorem formalizes Result 2 for matching.

Theorem 3. *For any $k = o(n/\log n)$ and $\alpha = o(\min\{n/k, k\})$, any $\alpha$-approximation randomized composable coreset of the maximum matching problem is of size $\Omega(n/\alpha^2)$.*

By Yao's minimax principle [61], to prove the lower bound in Theorem 3, it suffices to analyze the performance of deterministic algorithms over a fixed (hard) distribution. We propose the following distribution for this task. For simplicity of exposition, in the following, we prove a lower bound for $(\alpha/4)$-approximation algorithms; a straightforward scaling of the parameters proves the lower bound for $\alpha$-approximation.

**Distribution $\mathcal{D}_{\mathsf{Matching}}$.**

- Let $G(L, R, E)$ (with $|L| = |R| = n$) be constructed as follows:
  (1) Pick $A \subseteq L$ and $B \subseteq R$, each of size $n/\alpha$, uniformly at random.
  (2) Define $E_{AB}$ as a set of edges between $A$ and $B$, chosen by picking each edge in $A \times B$ w.p. $k \cdot \alpha/n$.
  (3) Define $E_{\overline{AB}}$ as a *random* perfect matching between $\overline{A}$ and $\overline{B}$.
  (4) Let $E := E_{AB} \cup E_{\overline{AB}}$.
- Let $E^{(1)}, \ldots, E^{(k)}$ be a *random $k$-partitioning* of $E$ and let the input to player $P^{(i)}$ be the graph $G^{(i)}(L, R, E^{(i)})$.

Let $G$ be a graph sampled from the distribution $\mathcal{D}_{\mathsf{Matching}}$. Notice first that the graph $G$ always has a matching of size at least $n - n/\alpha \geq n/2$, i.e., the matching $E_{\overline{AB}}$. Additionally, it is easy to see that any matching of size more than $2n/\alpha$ in $G$ uses at least $n/\alpha$ edges from $E_{\overline{AB}}$: the edges in $E_{AB}$ can only form a matching of size $n/\alpha$ by construction. This implies that any $(\alpha/4)$-approximate solution requires recovering at least $n/\alpha$ edges from $E_{\overline{AB}}$. In the following, we prove that this is only possible if the coresets of the players are sufficiently large.

For any $i \in [k]$, define the *induced matching* $M^{(i)}$ as the unique matching in $G^{(i)}$ that is incident on *vertices of degree exactly one*, i.e., both end-points of each edge in $M^{(i)}$ have degree one in $G^{(i)}$. We emphasize that the notion of induced matching is with respect to the entire graph and not only with respect to the vertices included in the induced matching. We have the following crucial lemma on the size of $M^{(i)}$. The proof appears in the full version of the paper [9].

Lemma 4.1. *W.p. $1 - O(1/n)$, for all $i \in [k]$, $\left|M^{(i)}\right| = \Theta(n/\alpha)$.*

Proof of Theorem 3. Fix any randomized composable coreset for the matching problem that has size $o(n/\alpha^2)$. We show that such a coreset cannot achieve a better than $(\alpha/4)$-approximation over the distribution $\mathcal{D}_{\mathsf{Matching}}$. As argued earlier, to prove this, we need to show that this coreset only contains $o(n/\alpha)$ edges from $E_{\overline{AB}}$ in expectation.

Fix any player $i \in [k]$, and let $M^{\star(i)}$ be the subset of the matching $E_{\overline{AB}}$ assigned to $P^{(i)}$. It is clear that $M^{\star(i)} \subseteq M^{(i)}$ by the definition of $M^{(i)}$. Moreover, define $X_i$ as the random variable denoting the number of edges from $M^{\star(i)}$ that belong to the coreset sent by player $P^{(i)}$. Notice that $X_i$ is clearly an upper bound on the number of edges of $E_{\overline{AB}}$ that are in the final matching of coordinator and also belong to the input graph of player $P^{(i)}$. In the following, we show that $\mathbb{E}[X_i] = o\left(\frac{n}{k \cdot \alpha}\right)$. Having proved this, we have that the expected size of the output matching by the coordinator is at most $n/\alpha + \sum_{i=1}^{k} \mathbb{E}[X_i] = n/\alpha + o(n/\alpha) < (\alpha/4) \cdot \mathsf{MM}(G)$, a contradiction.

We now prove $\mathbb{E}[X_i] = o\left(\frac{n}{k \cdot \alpha}\right)$. In the following, we condition on the event that $\left|M^{\star(i)}\right| = \Theta(n/k)$ and $\left|M^{(i)}\right| = \Theta(n/\alpha)$; by Chernoff bound (for the first part, since $n/k = \omega(\log n)$) and Lemma 4.1 (for the second part), this event happens with probability $1 - O(1/n)$. As such, this conditioning can only change $\mathbb{E}[X_i]$ by an additive factor of $O(1)$ which we ignore in the following.

A crucial property of the distribution $\mathcal{D}_{\mathsf{Matching}}$ is that the edges in $M^{\star(i)}$ and the remaining edges in $M^{(i)}$ are indistinguishable in $G^{(i)}$. More formally, for any edge $e \in G^{(i)}$,

$$\Pr\left(e \in M^{\star(i)} \mid e \in M^{(i)}\right) = \frac{\left|M^{\star(i)}\right|}{\left|M^{(i)}\right|} = \Theta(\alpha/k)$$

On the other hand, for a fixed input $M^{(i)}$ to player $P^{(i)}$, the computed coreset $C_i$ is always the same (as the coreset is a deterministic function of the player input). Hence,

$$\begin{aligned}\mathbb{E}[X_i] &= \sum_{e \in C_i} \Pr\left(e \in M_i^\star \mid e \in M^{(i)}\right) \\ &= |C_i| \cdot \Theta(\alpha/k) = o(n/\alpha^2) \cdot \Theta(\alpha/k) = o\left(n/(\alpha \cdot k)\right)\end{aligned}$$

where the second last equality is by the assumption that the size of the coreset, i.e., $|C_i|$, is $o(n/\alpha^2)$. This finalizes the proof. □

## 4.2 A Lower Bound for Randomized Composable Coresets of Vertex Cover

The following is a formal statement of Result 2 for vertex cover.

Theorem 4. *For any $k = o(n/\log n)$ and $\alpha = o(\min\{n/k, k\})$, any $\alpha$-approximation randomized composable coreset of the minimum vertex cover problem is of size $\Omega(n/\alpha)$.*

By Yao's minimax principle [61], to prove the lower bound in Theorem 4, it suffices to analyze the performance of deterministic algorithms over a fixed (hard) distribution. We propose the following distribution for this task. For simplicity of exposition, in the following, we prove a lower bound for $(c \cdot \alpha)$-approximation algorithms (for some constant $c > 0$); a straightforward scaling of the parameters proves the lower bound for $\alpha$-approximation.

**Distribution $\mathcal{D}_{\text{VC}}$.**

- Construct $G(L, R, E)$ (with $|L| = |R| = n$) as follows:
  (1) Pick $A \subseteq L$ of size $n/\alpha$ uniformly at random.
  (2) Let $E_A$ be a set of edges chosen by picking each edge in $A \times R$ w.p. $k/2n$.
  (3) Pick a single vertex $v^\star$ uniformly at random from $\overline{A}$ and let $e^\star$ be an edge incident on $v^\star$ chosen uniformly at random.
  (4) Let $E := E_A \cup \left\{e^\star\right\}$.
- Let $E^{(1)}, \ldots, E^{(k)}$ be a *random $k$-partitioning* of $E$ and let the input to player $P^{(i)}$ be the graph $G^{(i)}(L, R, E^{(i)})$.

For any $i \in [k]$, we define $L_i^1$ as the set of vertices in $L$ with degree *exactly one* in $G^{(i)}$. We further define $R_i^1$ as the set of neighbors of vertices in $L_i^1$ (note that vertices in $R_i^1$ do not *not* necessarily have degree exactly one). We have (the proof is deferred to the full version of the paper [9]),

Lemma 4.2. *For any $i \in [k]$, $\left|L_i^1\right| = \Theta(n/\alpha)$ and $\left|R_i^1\right| = \Theta(n/\alpha)$ w.p. $1 - o(1)$.*

Proof of Theorem 4. Let $i$ be the index of the player $P^{(i)}$ that the edge $e^\star$ is given to. We argue that if the coreset sent by player $P^{(i)}$ is of size $o(n/\alpha)$, then the coordinator cannot obtain a vertex cover of size $o(n)$. As the graph $G$ admits a vertex cover of size $(n/\alpha + 1)$ (pick $A$ and $v^\star$), this proves the theorem.

By Lemma 4.2, the set of vertices in $L$ with degree exactly one in $G^{(i)}$ and the set of their neighbors in $R$, i.e., the sets $L_i^1$ and $R_i^1$, are of size $\Theta(n/\alpha)$ w.p. $1 - o(1)$. In the following, we condition on this event. As the algorithm used by $P^{(i)}$ to create the coreset is deterministic, given a fixed input, it always creates the same coreset. However, a crucial property of the distribution $\mathcal{D}_{\text{VC}}$ is that, conditioned on a fixed assignment to $L_i^1$, the vertex $v^\star$ is chosen uniformly at random from $L_i^1$. This implies that if the coreset of player $P^{(i)}$ contains $o(n/\alpha)$ edges, then w.p. $1 - o(1)$, $e^\star$ is not part of the coreset ($e^\star$ is chosen uniformly at random from the set of all edges incident on $L_i^1$). Similarly, if the coreset fixes $o(n/\alpha)$ vertices to be added to the final solution, w.p. $1 - o(1)$, no end point of $e^\star$ is added to this fixed set ($v^\star$ is chosen uniformly at random from $L_i^1$ of size $\Theta(n/\alpha)$, and the other end point of $e^\star$ is chosen uniformly at random from $R_i^1$ of size $\Theta(n/\alpha)$). Finally, the coresets of other players are all independent of the edge $e^\star$ and hence as long as the total number of fixed vertices sent by the players is $o(n)$, w.p. $1 - o(1)$, no end points of $e^\star$ are present in the fixed solution. Conditioned on these three events, w.p. $1 - o(1)$, the output of the algorithm does not cover the edge $e^\star$ and hence is not a feasible vertex cover.

We remark that this argument holds even if we are allowed to add extra vertices to the final vertex cover (other than the ones fixed by the players or computed as a vertex cover of the edges in the coresets), since conditioned on $e^\star$ not being present in any coreset, the end point of this edge are chosen uniformly at random from all vertices in $L \setminus A$ and $R$ and hence a solution of size $o(n)$ would not contain either of them w.p. $1 - o(1)$. □

## Acknowledgements

The first author would like to thank Sepideh Mahabadi and Ali Vakilian for many helpful discussions in the earlier stages of this work. The authors are grateful to the anonymous reviewers of SPAA 2017 for many insightful comments and suggestions.

## REFERENCES

[1] S. Abbar, S. Amer-Yahia, P. Indyk, S. Mahabadi, and K. R. Varadarajan. Diverse near neighbor problem. In *Symposuim on Computational Geometry 2013, SoCG '13, Rio de Janeiro, Brazil, June 17-20, 2013*, pages 207–214, 2013.

[2] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004.

[3] K. J. Ahn and S. Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput.*, 222:59–79, 2013.

[4] K. J. Ahn and S. Guha. Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 202–211, 2015.

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

[6] K. J. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 5–14, 2012.

[7] Y. Ai, W. Hu, Y. Li, and D. P. Woodruff. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 20:1–20:22, 2016.

[8] N. Alon, N. Nisan, R. Raz, and O. Weinstein. Welfare maximization with limited interaction. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1499–1512, 2015.

[9] S. Assadi and S. Khanna. Randomized composable coresets for matching and vertex cover. *CoRR*, abs/1705.08242, 2017.

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

[11] S. Assadi, S. Khanna, Y. Li, and G. Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364, 2016.

[12] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: massive data summarization on the fly. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 671–680, 2014.

[13] M. Balcan, S. Ehrlich, and Y. Liang. Distributed k-means and k-median clustering on general communication topologies. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 1995–2003, 2013.

[14] S. Baswana, M. Gupta, and S. Sen. Fully dynamic maximal matching in o(log n) update time. *SIAM J. Comput.*, 44(1):88–113, 2015.

[15] M. Bateni, A. Bhaskara, S. Lattanzi, and V. S. Mirrokni. Distributed balanced clustering via mapping coresets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2591–2599, 2014.

[16] S. Bhattacharya, M. Henzinger, and G. F. Italiano. Deterministic fully dynamic data structures for vertex cover and matching. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 785–804, 2015.

[17] S. Bhattacharya, M. Henzinger, D. Nanongkai, and C. E. Tsourakakis. Space- and time-efficient algorithm for maintaining dense subgraphs on one-pass dynamic streams. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 173–182, 2015.

[18] L. Bulteau, V. Froese, K. Kutzkov, and R. Pagh. Triangle counting in dynamic graph streams. *Algorithmica*, 76(1):259–278, 2016.

[19] A. Chakrabarti, G. Cormode, and A. McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 641–650, 2008.

[20] R. Chitnis, G. Cormode, H. Esfandiari, M. Hajiaghayi, A. McGregor, M. Monemizadeh, and S. Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1326–1344, 2016.

[21] R. H. Chitnis, G. Cormode, M. T. Hajiaghayi, and M. Monemizadeh. Parameterized streaming: Maximal matching and vertex cover. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1234–1251, 2015.

[22] M. Crouch and D. S. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pages 96–104, 2014.

[23] S. Dobzinski, N. Nisan, and S. Oren. Economic efficiency requires interaction. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 233–242, 2014.

[24] S. Eggert, L. Kliemann, and A. Srivastav. Bipartite graph matchings in the semi-streaming model. In *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, pages 492–503, 2009.

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

[26] H. Esfandiari, M. Hajiaghayi, and M. Monemizadeh. Finding large matchings in semi-streaming. In *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain.*, pages 608–614, 2016.

[27] H. Esfandiari, M. T. Hajiaghayi, V. Liaghat, M. Monemizadeh, and K. Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1217–1233, 2015.

[28] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005.

[29] A. Goel, M. Kapralov, and S. Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 468–485. SIAM, 2012.

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

[31] A. Hassidim, J. A. Kelner, H. N. Nguyen, and K. Onak. Local graph partitions for approximation and testing. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 22–31, 2009.

[32] J. Håstad and A. Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(1):211–219, 2007.

[33] Z. Huang, B. Radunovic, M. Vojnovic, and Q. Zhang. Communication complexity of approximate matching in distributed graphs. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 460–473, 2015.

[34] P. Indyk, S. Mahabadi, M. Mahdian, and V. S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 100–108, 2014.

[35] M. Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697, 2013.

[36] M. Kapralov, S. Khanna, and M. 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.

[37] M. Kapralov, S. Khanna, and M. Sudan. Streaming lower bounds for approximating MAX-CUT. In *SODA*, 2015.

[38] M. Kapralov, Y. T. Lee, C. Musco, C. Musco, and A. Sidford. Single pass spectral sparsification in dynamic streams. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 561–570, 2014.

[39] M. Kapralov and D. Woodruff. Spanners and sparsifiers in dynamic streams. *PODC*, 2014.

[40] H. J. Karloff, S. Suri, and S. Vassilvitskii. A model of computation for mapreduce. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 938–948, 2010.

[41] C. Konrad. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852, 2015.

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

[43] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.

[44] S. Lattanzi, B. Moseley, S. Suri, and S. Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, pages 85–94, 2011.

[45] Y. Li, H. L. Nguyen, and D. P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 174–183, 2014.

[46] A. McGregor. Finding graph matchings in data streams. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, pages 170–181, 2005.

[47] A. McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014.

[48] A. McGregor, D. Tench, S. Vorotnikova, and H. T. Vu. Densest subgraph in dynamic graph streams. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part II*, pages 472–482, 2015.

[49] A. McGregor and S. Vorotnikova. Planar matching in streams revisited. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, pages 17:1–17:12, 2016.

[50] V. S. Mirrokni and M. Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 153–162, 2015.

[51] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2049–2057, 2013.

[52] S. Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

[53] O. Neiman and S. Solomon. Simple deterministic algorithms for fully dynamic maximal matching. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 745–754, 2013.

[54] H. N. Nguyen and K. Onak. Constant-time approximation algorithms via local improvements. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 327–336, 2008.

[55] K. Onak, D. Ron, M. Rosen, and R. Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1123–1131, 2012.

[56] K. Onak and R. Rubinfeld. Maintaining a large matching and a small vertex cover. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 457–464, 2010.

[57] M. Parnas and D. Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theor. Comput. Sci.*, 381(1-3):183–196, 2007.

[58] A. Paz and G. Schwartzman. A $(2 + \varepsilon)$-approximation for maximum weight matching in the semi-streaming model. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2153–2161, 2017.

[59] J. M. Phillips, E. Verbin, and Q. Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 486–501, 2012.

[60] S. Solomon. Fully dynamic maximal matching in constant update time. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 325–334, 2016.

[61] A. C. Yao. Lower bounds to randomized algorithms for graph properties (extended abstract). In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 393–400, 1987.

[62] Y. Yoshida, M. Yamamoto, and H. Ito. Improved constant-time approximation algorithms for maximum matchings and other optimization problems. *SIAM J. Comput.*, 41(4):1074–1093, 2012.