Streaming and Communication Complexity of Load-Balancing via Matching Contractors

Sepehr Assadi[*]
University of Waterloo

Aaron Bernstein[†]
New York University

Zachary Langley
Rutgers University

Lap Chi Lau[‡]
University of Waterloo

Robert Wang
University of Waterloo

Abstract

In the load-balancing problem, we have an n-vertex bipartite graph $G = (L, R, E)$ between a set of clients and servers. The goal is to find an assignment of all clients to the servers, while minimizing the maximum load on each server, where load of a server is the number of clients assigned to it. Motivated by understanding the streaming complexity of this problem, we study load-balancing in the one-way (two-party) communication model: the edges of the input graph are partitioned between Alice and Bob, and Alice needs to send a short message to Bob for him to output a solution of the entire graph.

We show that settling the one-way communication complexity of load-balancing is equivalent to a natural sparsification problem for load-balancing, which can alternatively be interpreted as sparsification for vertex-expansion. We then prove a *dual* interpretation of this sparsifier, showing that the minimum density of a sparsifier is effectively the same as the maximum density one can achieve for an extremal graph family that is new to this paper, called *Matching-Contractors*; these graphs are intimately connected to the well-known Ruzsa-Szemerédi graphs and generalize them in certain aspects. Our chain of equivalences thus shows that the one-way communication complexity of load-balancing can be reduced to a purely graph theoretic question: what is the maximum density of a Matching-Contractor on n vertices?

As our final result, we present a novel combinatorial construction of some-what dense Matching-Contractors, which implies a strong one-way communication lower bound for load-balancing: any one-way protocol (even randomized) with $\widetilde{O}(n)$ communication cannot achieve a better than $n^{\frac{1}{4}-o(1)}$-approximation. Previously, no non-trivial lower bounds were known for protocols with even $O(n \log n)$ bits of communication (a better-than 2-approximation lower bound is trivial). Our result also implies the first non-trivial lower bounds for semi-streaming load-balancing in the edge-arrival model, ruling out $n^{\frac{1}{4}-o(1)}$-approximation in a single-pass.

[*] (sepehr@assadi.info) Supported in part by a Sloan Research Fellowship, an NSERC Discovery Grant, a University of Waterloo startup grant, and a Faculty of Math Research Chair grant.

[†] (bernstei@gmail.com) Supported in part by a Sloan Research Fellowship, a Google Research Fellowship, NSF Grant 1942010, and a Charles S. Baylis endowment from NYU.

[‡] (lapchi@uwaterloo.ca) Supported by an NSERC Discovery Grant.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Contents

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

1 Introduction

We study the **load-balancing** problem. Given a bipartite graph $G = (L, R, E)$, an *assignment* maps each vertex in L to one of its neighbors in R. The load of vertex in R is the number of vertices in L assigned to it. The goal is to find an assignment that minimizes the maximum load. We often refer to vertices in L as *clients* and the ones in R as *servers*.

Load-balancing has a rich history under different names. It is studied in the scheduling literature as job scheduling with restricted assignment [Hor73, BCS74, LST90, LL04, HLLT06, JR17, JR20], in the distributed computing literature as backup placement problem [CHSW12, HKPSR18, OBL18, BO20, ABL20, ALPZ21], and in graph algorithms as the semi-matching problem [HLLT06, KR13a, FLN14]. From an optimization perspective, it serves as a canonical example of a mixed packing-covering problem, with packing constraints on the servers and covering constraints on the clients.

This work focuses on the load-balancing problem in the semi-streaming model [FKM$^+$05], where the edges of the input graph G arrive one-by-one in a stream, and the algorithm is allowed to process these edges using $O(n \cdot \text{polylog}(n))$ memory, where n is the total number of vertices. In addition, the algorithm is limited to a single pass (or a few passes) over the stream. These constraints capture several challenges of processing massive graphs, including I/O-efficiency and efficiently monitoring evolving graphs. As such, the semi-streaming model has been at the forefront of the research on massive graphs in recent years.

Load-balancing can be seen as a natural and useful problem between two of the most well-studied families of problems in the semi-streaming model: matching problems (see e.g. [FKM$^+$05, GKK12, AG11, AKLY16, FMU22, A24]) and coverage problems (see e.g. [DIMV14, AKL16, MV17, KKA23]). Yet, in sharp contrast to these problems, our understanding of semi-streaming load-balancing is quite limited. Over a decade ago, Konrad and Rosén [KR13a] initiated the study of load-balancing in this model, presenting a simple $O(\sqrt{n})$-approximation algorithm in a single-pass and an $O(\log n)$-approximation algorithm in $O(\log n)$ passes. The latter algorithm was only recently improved to an $O(1)$-approximation in $O(\log n)$ passes [ABL23]. At this point, there is no evidence why even a 2-approximation cannot be achieved in a single pass![1]

1.1 Our Contributions

A main consequence of this work is that Konrad and Rosén's single-pass $O(\sqrt{n})$-approximation algorithm cannot be significantly improved.

Result 1. *There is no semi-streaming algorithm that obtains a $n^{\frac{1}{4}-o(1)}$-approximation to the load-balancing problem with success probability at least $\frac{2}{3}$.*

This result stems from a series of reductions and equivalences that we describe below. Ultimately, we reduce the problem to a question in extremal graph theory and present a novel construction to establish the streaming complexity lower bound.

One-Way Communication Complexity and Load-Balancing Sparsifiers Inspired by previous work on matchings and coverage problems [GKK12, KR13a, AKL16, FNSZ20], we study the *one-way communication complexity* of the load balancing problem. In this communication model, the input graph is edge-partitioned between Alice and Bob. Alice sends a single $\widetilde{O}(n)$-bit message to Bob, and Bob then outputs an approximately optimal solution to the entire graph. Semi-streaming algorithms imply one-way communication protocols (but not vice versa), so a lower bound on the one-way communication complexity translates to a lower bound on the semi-streaming complexity.

Our first equivalence shows that the one-way communication complexity of load-balancing is nearly equal to the minimum density of **load-balancing sparsifiers**, which were called *semi-matching skeletons* in [KR13a]. For a graph $G = (L, R, E)$ and approximation ratio $\alpha \geqslant 1$, we define an α-approximate load-balancing sparsifier of G to be any spanning subgraph H of G that can preserve the value of optimal load-balancing for *every subset* of clients in L up to a factor of α (see Definition 3.1 for the formal definition).

[1] Obtaining a better-than-two approximation for load-balancing requires finding a perfect matching when it exists. As such, one can borrow existing lower bounds for the latter problem to obtain that $\Omega(\log n)$ passes are required for load-balancing in this case.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

The original motivation of this definition in [KR13a] is that a good load-balancing sparsifier implies a good one-way communication protocol. The protocol is the natural one: Alice sends Bob an α-approximate sparsifier H_A of her graph G_A, and Bob outputs the optimal assignment $\mathcal{A}$ of the graph $H_A \cup G_B$. It is not hard to check that $\mathcal{A}$ is an $(\alpha+1)$-approximation to the optimal assignment in $G_A \cup G_B$ (see Proposition 3.4).

Our contribution is proving the other direction of the equivalence, showing that load-balancing sparsifiers are the correct combinatorial objects for understanding the one-way communication complexity.

Result 2. *For any $\alpha \geqslant 1$ and large $n \geqslant 1$, the communication cost of the best one-way protocol for α-approximate load-balancing, possibly with randomization and success probability $\frac{2}{3}$, is equal, up to $\operatorname{poly}\log(n)$ factors, to the smallest T such that any n-vertex graph contains a $\Theta(\alpha)$-approximate load-balancing sparsifier with at most T edges.*

Load-Balancing Sparsifiers and Matching Contractors Our key conceptual contribution is the identification of a natural and interesting dual object for load-balancing sparsifiers. We say a bipartite graph $G = (L, R, E)$ is an **$\boldsymbol{\alpha}$-Matching-Contractor** if and only if its edges can be partitioned into a set of matchings $M_1, M_2, \ldots, M_k$ with the following property: vertex-set of each matching M_i in L is "heavily contracting" if we do not use edges of M_i itself; more formally, the neighbor-set of $L(M_i)$ in $G \setminus M_i$ has size $|M_i|/\alpha$ only. Matching-Contractors can be roughly thought of as a more stringent version of *Ruzsa-Szemerédi graphs* [RS78]: see Remark 3.11 for a more detailed comparison. Our second equivalence is between the sparsity of a load-balancing sparsifier and the density of a Matching-Contractor.

Result 3. *For any $\alpha \geqslant 1$ and large $n \geqslant 1$, the minimum T such that every graph n-vertex graph G contains an α-approximate load-balancing sparsifier with at most T edges, is equal, up to $\operatorname{poly}\log(n)$ factors, to the largest density of a $\Theta(\alpha)$-Matching-Contractor with n vertices.*

Construction of Dense Matching-Contractors Result 2 and Result 3 reduce the one-way communication complexity of load balancing to a new question in extremal graph theory: what is the maximum density of a Matching-Contractor on n vertices? Our key technical contribution is a novel construction of somewhat-dense Matching-Contractors.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Result 4. *For any sufficiently small $\varepsilon > 0$ and large integer $n \geqslant 1$, there exists an $\left(n^{\frac{1}{4}-O(\varepsilon)}\right)$-Matching-Contractor with $n^{1+\Omega(\varepsilon^2)}$ edges.*

Combining our results together implies that no $\widetilde{O}(n)$-communication protocol can achieve $n^{\frac{1}{4}-\Omega(1)}$-approximation to load-balancing. This in turn implies Result 1 that the best approximation ratio achievable by single-pass semi-streaming algorithms is only $n^{\frac{1}{4}-o(1)}$, significantly improving the prior 2-approximation lower bounds. Our lower bound also comes quite close to the $n^{\frac{1}{3}}$ one-way communication complexity upper bound of [KR13a], and by the machinery developed in this paper, closing this gap now amounts to pinning down the maximum density of Matching-Contractors.

1.2 Previous Work Konrad and Rosén [KR13a] made the first step in understanding the one-way communication complexity of load-balancing. They defined *load-balancing sparsifiers* (called *semi-matching skeletons* in [KR13a]) and showed that the existence of sparse load-balancing sparsifiers implies good communication protocols. They then use this connection to obtain an $n^{\frac{1}{3}}$-approximate one-way protocol to load-balancing. Their result leaves two significant gaps. Firstly, while they showed a one-way relation between load-balancing sparsifiers and communication protocols, they did not prove an equivalence between the two, hence leaving it uncertain whether this is the "right" notion of sparsification. Secondly, their lower bounds for both sparsifiers and communication protocols are quite limited. They show that a $n^{\frac{1}{c+1}}$-approximate one-way protocol requires cn bits, but this has no implication for protocols that communicate $O(n\log(n))$ bits, and hence no implication for semi-streaming in general.

Our work is heavily inspired by the pioneering work of [GKK12] for the closely related maximum matching problem. [GKK12] initiated a systematic study of semi-streaming matchings through one-way communication complexity. The two main discoveries of [GKK12] are as follows.

- They introduced *matching covers* as a natural notion of sparsifiers for matchings, and proved an equivalence between matching covers and one-way communication complexity of matchings. Matching covers have since found far reaching implications in streaming (e.g., in [Kap13, AB19, Ber20, ABKL23]) and beyond (e.g., in dynamic graph algorithms [ABKL23, BG24, AK24]).
- They formulated the natural "dual" connection between matching covers and Ruzsa-Szemerédi graphs [RS78, FLN$^+$02, AMS12], an extremal graph family with many *large* edge-disjoint *induced* matchings. They thus reduced the one-way communication problem to a problem in extremal graph theory: do there exist dense Ruzsa-Szemerédi graphs? Extending known constructions of Ruzsa-Szemerédi graphs in [FLN$^+$02] for monotonicity testing lower bound, they proved the first non-trivial lower bound for approximation of matchings in the semi-streaming model. Remarkably, after [GKK12] first established this connection, Ruzsa-Szemerédi graphs have become a central tool for proving semi-streaming lower bounds for matchings and beyond (e.g., in [Kap13, AKLY16, AKL17, AR20, Kap21, CKP$^+$21, AS23, AKNS24]).

1.3 Roadmap and Technical Overview In this work, we follow a similar chain of reduction and equivalences as in [GKK12] to relate the one-way communication complexity of load-balancing to the density of Matching-Contractors. In the following, we provide an outline of this paper and discuss the high-level ideas in each step.

In Section 2 we introduce basic notation and preliminaries. In Section 3, we define the two crucial objects of this paper: load-balancing sparsifiers (introduced in [KR13a]) and Matching-Contractors (introduced in this work). We should think of load-balancing sparsifiers as the analog of matching covers in [GKK12], and Matching-Contractors as the analog of Ruzsa-Szemerédi graphs. In Section 3.1, we provide some intuition on the equivalence of load-balancing sparsifiers and one-way communication complexity; the proof is in Section 6 using information theory and Matching-Contractors. We show in Section 3.2 that load-balancing sparsifiers are equivalent to a version of *vertex expansion sparsifiers* in bipartite graphs, suggesting that it is a natural object of its own interest. We also formulate an equivalent operational definition that is simpler to work with. In Section 3.3, we discuss the relation between Matching-Contractors and Ruzsa-Szemerédi graphs.

In Section 4, we prove Result 3 about the equivalence between the existence of sparse load-balancing sparsifiers and the non-existence of dense Matching-Contractors. This section encapsulates the main

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

conceptual contribution of this paper, that Matching-Contractors is the dual object of load-balancing sparsifiers in a precise sense. To prove this, we formulate a linear programming relaxation for finding a load-balancing sparsifier of a bipartite graph, and prove that it is an $O(\log n)$-approximation. Then we construct the dual linear program and prove that one can construct a Matching-Contractor from a solution to the dual LP. This equivalence is the analog of the equivalence between matching covers and Ruzsa-Szemerédi graphs in [GKK12], and their proof is also based on linear programming duality. For our proof, both steps are done by a randomized rounding argument using some problem-specific insights.

In Section 5, we show an explicit construction of Matching-Contractors (Result 4), which is the main technical challenge in this paper. One reason is that the existence of dense Matching-Contractors is counter-intuitive: For the complete bipartite graphs, one can prove that there are load-balancing sparsifiers with $O(n \log n)$ edges by random sampling. These are known as magical graphs in [HLW06] and have applications in error-correcting codes and super-concentrators. Given this and the recent successes of sparsification in various settings, our initial effort was to prove that sparse load-balancing sparsifiers always exist (and hence dense Matching-Contractors do not). Another challenge is that unlike for Ruzsa-Szemerédi graphs, there were no known constructions of Matching-Contractors, and hence no clear starting point in terms of what tools to use. As we discuss in Section 3.3, Matching-Contractor is an even more stringent version of Ruzsa-Szemerédi graph, which is itself notoriously difficult to construct [FLN$^+$02, AMS12]. Our construction idea is to view each vertex as a string, and use the block structures of the vertices and a set family of small pairwise intersection to argue about the contraction property. Even though the final construction and the analysis are short and elementary, we see this as the key technical innovation in this paper.

Finally, in Section 6, we show the equivalence between the communication complexity of load-balancing and the existence of dense Matching-Contractors. Combining with the equivalence in Result 3, this completes Result 2. This equivalence is the analog of the equivalence between the communication complexity of matchings and the existence of dense Ruzsa-Szemerédi graphs in [GKK12]. Our proof uses basic information theoretic arguments and a simple modification of Matching-Contractors to establish the lower bound.

Overall, we find it interesting to have natural analogs of matching covers and Ruzsa-Szemerédi graphs in load-balancing sparsifiers and Matching-Contractors. Given the large impact of [GKK12], we hope that these new objects and equivalences can play a further role in understanding load-balancing in streaming and other models. One promising direction is to establish lower bound on multi-pass semi-streaming algorithms for the load balancing problem. We believe they also have the potential to shed light on other graph problems, such as vertex expansion and matching conductance as discussed in Section 3.2.

2 Preliminaries

Notation. Given two functions f, g, we use $f \lesssim g$ to denote the existence of a positive constant $c > 0$, such that $f \leqslant c \cdot g$ always holds. We use $f \asymp g$ to denote $f \lesssim g$ and $g \lesssim f$.

Graphs. We use $G = (L, R, E)$ to denote a bipartite graph with sides L and R. For any vertex v, we define $N_G(v)$ to be the set of neighbors of v in G, $E_G(v)$ to be the set of incident edges, $N_G(S) := \bigcup_{v \in S} N(v)$ and $E_G(S) := \bigcup_{v \in S} E(v)$. Given subsets $X \subseteq L$ and $Y \subseteq R$, we define $E_G(X, Y) := \{(u, v) \in E \mid u \in X \wedge v \in Y\}$ and we define $G[X \cup Y]$ to be the induced graph $(X \cup Y, E_G(X, Y))$. When clear from the context, we may drop the subscript G in these notation.

We define a matching $M \subseteq E$ to be a set of disjoint edges. We let $L(M)$ denote the left endpoints of edges in M, and $R(M)$ the right endpoints. For any graph G, we let $\mu(G)$ denote the size of the maximum matching in G. Finally, we say that a set $X \subseteq V$ is **matchable** if there exists a matching for which every vertex in X is matched.

Load Balancing. In the context of load balancing, we will often refer to vertices in L as *clients* and vertices in R as *servers*. An assignment $\mathcal{A}$ assigns every client to some server: formally, $\mathcal{A}$ is a function $\mathcal{A} : L \to R$ such that for any client $c \in L$, we have $\mathcal{A}(c) \in N(c)$. For any server $s \in R$, we define $\mathcal{A}^{-1}(s) := \{c \in L \mid \mathcal{A}(c) = s\}$, and we define the load of a server to be $\textsc{load}_{\mathcal{A}}(s) := |\mathcal{A}^{-1}(s)|$.

We define $\textsc{MaxLoad}(A) := \max_{s \in R} \textsc{load}(s)$. The goal of the load-balancing problem is to find an assignment $\mathcal{A}$ that minimizes $\textsc{MaxLoad}(\mathcal{A})$. To this end, we define $\text{OPT}\textsc{load}(G)$ to be the load of the optimal assignment on G. We say that assignment $\mathcal{A}$ for G is α-approximate if $\textsc{MaxLoad}(\mathcal{A}) \leqslant$

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

$\alpha \cdot \text{OPT}\textsc{Load}(G)$.

We use the following generalization of Hall's Theorem [Hal87], proved in [KR13b, Lemma 4].

Proposition 2.1 ([KR13b])**.** *For any bipartite graph $G = (L, R, E)$,*

$$\text{OPT}\textsc{Load}(G) = \max_{\emptyset \neq X \subseteq L} \left\lceil \frac{|X|}{|N(X)|} \right\rceil.$$

2.1 One-Way Communication Complexity We work with the standard one-way two-player communication complexity model of Yao [Yao79] (see the excellent textbooks by [KN97] and [RY20] for the background on this model). Specifically, we are interested in the following problem.

Problem 1. An n-vertex bipartite graph $G = (C, S, E)$ with bipartition C of *clients* and S of *servers* is edge-partitioned (arbitrarily) between two players: Alice receives $E_A \subseteq E$ and Bob receives $E_B \subseteq E$, where $E_A \cup E_B = E$ and $E_A \cap E_B = \emptyset$.

The goal is for Alice to send a single message to Bob which is only a function of her input $G_A := (C, S, E_A)$, and Bob, given this message and his input $G_B := (C, S, E_B)$ should output a solution to the load-balancing problem on the entire graph G. In a randomized protocol, we assume Alice and Bob also have access to a *shared* source of randomness, commonly referred to as **public randomness**. In that case, the message of Alice and the output of Bob can also additionally depend on this public randomness.

For any $n \geqslant 1$ and approximation ratio $\alpha \geqslant 1$, we use $\mathsf{LoadBal}(n, \alpha)$ to denote this problem on n-vertex graphs wherein the goal is to obtain (at least) an α-approximate solution.

We refer to the algorithm that decide the messages of Alice and the output of Bob in Problem 1 as a **protocol** π. The main measure of interest for us is the **communication cost** of a protocol π, denoted by $\|\pi\|$, and defined as the worst-case length of the message Alice sends to Bob (without loss of generality, via a padding argument, we assume length of all the messages communicated in the protocol is the same). Finally, we define the **(randomized) communication complexity** of $\mathsf{LoadBal}(n, \alpha)$ as the minimum communication cost of any randomized protocol that solves this problem with probability of success at least $\frac{2}{3}$, denoted by $\vec{R}(\mathsf{LoadBal}(n, \alpha))$.

3 Load-Balancing Sparsifiers

One of the key contributions of our paper is showing that the one-way communication problem of Section 2.1 is nearly equivalent to a notion of load-balancing sparsifiers, first introduced by Konrad and Rosén [KR13a]. Given a graph $G = (L, R, E)$, a load-balancing sparsifier is a subgraph $H = (L, R, E_H)$ that approximately preserves all the load-balancing properties of G.

Definition 3.1 ([KR13a])**.** Given $G = (L, R, E)$ and $\alpha \geqslant 1$, we say that subgraph $H = (L, R, E_H)$ is an **α-approximate load-balancing sparsifier** of G iff for every set $C \subseteq L$,

$$\text{OPT}\textsc{Load}(H[C \cup R]) \leqslant \alpha \cdot \text{OPT}\textsc{Load}(G[C \cup R]).$$

When the context is clear, we sometimes refer to H as simply a α-sparsifier of G.

Just like with other sparsification problems, the natural question is whether every graph G contain an α-approximate load-balancing sparsifier with few edges.

Definition 3.2. Define:

- $\textsc{sparsifier}(G, \alpha)$ to be the minimum possible number of edges in an α-approximate load-balancing sparsifier of G;
- $\textsc{sparsifier}(n, \alpha)$ to be the maximum $\textsc{sparsifier}(G, \alpha)$ over all bipartite graphs $G = (L, R, E)$ with $|L| = n$.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Note that $\textsc{sparsifier}(n, \alpha)$ and $\textsc{sparsifier}(G, \alpha)$ are monotonically decreasing as α increases.

Konrad and Rosén [KR13a], who referred to these sparsifiers as *semi-matching skeletons*, presented the following nontrivial upper bound.

Proposition 3.3 ([KR13a])**.** *Every graph G with n clients contains a $n^{\frac{1}{3}}$-sparsifier with at most $2n$ edges, i.e.*

$$\textsc{sparsifier}(n, n^{\frac{1}{3}}) \leqslant 2n.$$

They also showed lower bounds of (roughly) the form $\textsc{sparsifier}(n, n^{\frac{1}{c+1}}) \geqslant cn$ for $c \geqslant 1$. These lower bounds however have no implications for sparsifiers with $O(n \log n)$ edges. In particular, the possibility that every graph G contains an $O(1)$-sparsifier with $O(n \log n)$ edges was not ruled out. Note that when G is a complete bipartite graph, one can indeed construct a $O(1)$-sparsifier with $O(n \log n)$ edges by random sampling (see e.g. the section about magical graphs in [HLW06]).

3.1 Equivalence Between Sparsifier and One-Way Communication Complexity

Konrad and Rosén [KR13a] already showed this equivalence in one direction. We include a proof in Appendix A.1 for completeness.

Proposition 3.4 ([KR13a])**.** *Fix $n, \alpha \geqslant 1$, and suppose $\textsc{sparsifier}(n, \alpha) = T$. Then, there exists a deterministic protocol π for $\mathsf{LoadBal}(n, \alpha + 1)$ with communication cost $\|\pi\| = O(T \log(n))$ bits.*

Combining Proposition 3.4 with their sparsifiers in Proposition 3.3 gives the following result.

Corollary 3.5 ([KR13a])**.** *There exists a deterministic protocol π for $\mathsf{LoadBal}(n, n^{\frac{1}{3}} + 1)$ with communication cost $\|\pi\| = O(n^{\frac{1}{3}} \log(n))$ bits.*

We show that this equivalence also goes in the other direction, even for randomized protocols.

Theorem 1. *Suppose there exists a (randomized) communication protocol π for $\mathsf{LoadBal}(n, \alpha)$ with communication cost $\|\pi\| = C$ and probability of success at least $\frac{2}{3}$. Then,*

$$\textsc{sparsifier}(n, 8\alpha) \lesssim C \cdot \log^2(n).$$

Intuition for Theorem 1: The full proof will be presented in Section 6.4 as it requires the new notion of Matching-Contractors in Section 3.3, as well as the chain of equivalences worked out in Section 4.

For the sake of intuition, let us make the (small) assumptions that $|L| = |R| = n$ and that the protocol π is deterministic, as well as the (large) assumption that π takes the following form: Alice's message is limited to some subgraph H_A of G_A. Clearly H_A has $O(C)$ edges. We argue that H_A must be an α-sparsifier of G_A. Suppose for contradiction that H_A is not a α-sparsifier. Then there must exist some set $X \subseteq L$ such that

$$\text{OPT}\textsc{Load}(H_A[X \cup R]) > \alpha \cdot \text{OPT}\textsc{Load}(G_A[X \cup R]) \tag{3.1}$$

Now, say that Bob's graph G_B contains a complete graph from $L - X$ to R, but no edges incident to X. The key observation is that

$$\text{OPT}\textsc{Load}(H_A \cup G_B) = \text{OPT}\textsc{Load}(H_A[X \cup R]). \tag{3.2}$$

We now justify both directions of Equation 3.2. It is easy to see that $\text{OPT}\textsc{Load}(H_A \cup G_B) \geqslant \text{OPT}\textsc{Load}(H_A[X \cup R])$, since G_B contains no edges incident to X To see that $\text{OPT}\textsc{Load}(H_A \cup G_B) \leqslant \text{OPT}\textsc{Load}(H_A[X \cup R])$, consider the optimal assignment $\mathcal{A}$ in $H_A[X \cup R])$. Let $S_{\mathcal{A}}$ be all servers that have load at least 1 in $\mathcal{A}$; note that $|S_{\mathcal{A}}| \leqslant |X|$, which implies $|L \setminus X| \leqslant |R \setminus S_{\mathcal{A}}|$ (because we assumed $|L| = |R|$), so by construction of G_B, the set $L \setminus X$ is matchable in $G_B[(L \setminus X) \cup (R \setminus S_{\mathcal{A}})]$. Now consider the following assignment $\mathcal{A}'$ in $H_A \cup G_B$: $\mathcal{A}'$ is the same as $\mathcal{A}$ on the set X and assigns every vertex in $L \setminus X$ according to the matching M. It is easy to check that $\textsc{load}_{\mathcal{A}'}(s) = \textsc{load}_{\mathcal{A}}(s)$ for $s \in S_{\mathcal{A}}$ and $\textsc{load}_{\mathcal{A}'}(s) = 1$ for $s \in R \setminus S_{\mathcal{A}}$, so $\textsc{MaxLoad}(\mathcal{A}') = \textsc{MaxLoad}(\mathcal{A})$, which completes the proof of Equation 3.2

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Combining Equations 3.1 and 3.2 we have:

$$\begin{aligned}\textsc{OPTLoad}(H_A \cup G_B) &= \textsc{OPTLoad}(H_A[X \cup R]) \\ &> \alpha \cdot \textsc{OPTLoad}(G_A[X \cup R]) = \alpha \cdot \textsc{OPTLoad}(G_A \cup G_B),\end{aligned}$$

which contradicts the assumption that π is an α-approximate one-way communication protocol.

3.2 Equivalent Notions of Sparsification Although our primary motivation for studying load-balancing sparsifiers is to understand the streaming/one-way-communication complexity of load balancing, we believe that they are a very natural combinatorial object with connections to other problems.

Recall that the **vertex expansion** of a subset X is defined as $\psi(X) := |N(X)|/|X|$ (see e.g. [HLW06]). From the characterization of $\textsc{OPTLoad}$ in Proposition 2.1, one can see that load-balancing sparsifiers are closely related to vertex-expansion sparsifiers.

Observation 3.6 (Connection with Vertex Expansion)**.** *Let $G = (L, R, E)$ be a bipartite graph and let $H = (L, R, E_H)$ be a subgraph of G. If, for every subset $X \subseteq L$,*

$$\psi_H(X) = \frac{|N_H(X)|}{|X|} \geqslant \frac{2}{\alpha} \cdot \min\left\{\frac{|N_G(X)|}{|X|}, 1\right\} = \frac{2}{\alpha} \cdot \min\{\psi_G(X), 1\},$$

then, H is an α-load-balancing sparsifier of G. On the other hand, if H is an α-load-balancing sparsifier of G, then $\psi_H(X) \geqslant \frac{1}{2\alpha} \cdot \min\{\psi_G(X), 1\}$ for every subset $X \subseteq L$.

Note that we cannot replace the right hand side by the simpler expression $\Omega(\frac{1}{\alpha}) \cdot \psi_G(X)$, as otherwise any α-sparsifier of the complete bipartite graph $K_{n,n}$ must have at least $\Omega(\frac{n^2}{\alpha})$ edges (because the constraints on the singletons require every vertex on the left to have degree at least $\Omega(\frac{n}{\alpha})$), while one can prove that there is an α-load-balancing sparsifier of $K_{n,n}$ with $O(n \log n)$ edges by random sampling.

Remark 3.7 (Connection with Matching Conductance)**.** The $\min\{\psi(X), 1\}$ term in Observation 3.6 is a natural quantity that is closely related to the notion of **matching conductance** defined in [OZ22], which was used in analyzing the fastest mixing time of a graph. Let $\nu(G)$ be the size of a maximum matching in G. The matching conductance of a set X is defined as $\gamma(X) = \nu(E(X, \overline{X}))/|X|$. For a bipartite graph $G = (L, R, E)$, by Hall's theorem, it can be checked that

$$\min_{X \subseteq L} \gamma(X) := \min_{X \subseteq L} \min\{\psi(X), 1\}.$$

Operational Definition of Sparsifiers: We will not use vertex expansion and matching conductance in this paper, and so we do not provide details of the connections discussed above. Instead, we will use the following characterization in our proofs, as it provides the easiest way of verifying that a graph $H \subset G$ is indeed a load-balancing sparsifier.

Lemma 3.8. *Let $G = (L, R, E)$ be a bipartite graph, and $H = (L, R, E_H)$ be a subgraph of G. Then, for any $\alpha \geqslant 1$, the following statements are equivalent.*

1. *Load-Balancing: For every $X \subseteq L$, $\textsc{OPTLoad}(H[X \cup R]) \leqslant \alpha \cdot \textsc{OPTLoad}(G[X \cup R])$.*
2. *Operational Definition: For every $X \subseteq L$ that is matchable in G, $N_H(X) \geqslant \frac{1}{\alpha} \cdot |X|$.*

Proof. It is easy to see that **(1)** → **(2)**. Let H be an α-load-balancing sparsifier of G. Consider any matchable set $X \subseteq L$. As X is matchable, we have $\textsc{OPTLoad}(G[X \cup R]) = 1$, and so $\textsc{OPTLoad}(H[X \cup R]) \leqslant \alpha$ as H is an α-sparsifier. It follows from Proposition 2.1 that $|N_H(X)| \geqslant \frac{1}{\alpha} |X|$.

The other direction is more useful. For any $X \subseteq L$ with $\textsc{OPTLoad}(G[X \cup R]) = d$, we need to prove that $\textsc{OPTLoad}(H[X \cup R]) \leqslant \alpha d$. By Proposition 2.1, this is equivalent to proving that $|N_H(U)| \geqslant \frac{1}{\alpha d} |U|$ for any $U \subseteq X$. Let $\mathcal{A}$ be the optimal assignment in $G[U \cup R]$. Clearly,

$$\textsc{MaxLoad}(\mathcal{A}) = \textsc{OPTLoad}(G[U \cup R]) \leqslant \textsc{OPTLoad}(G[X \cup R]) \leqslant d.$$

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Let $R_{\mathcal{A}} := \{s \in R \mid \text{LOAD}_{\mathcal{A}}(s) \geqslant 1\}$ and note that $|R_{\mathcal{A}}| \geqslant |U| / \text{MAXLOAD}(\mathcal{A}) \geqslant |U| / d$. Construct a set of clients $C \subseteq U$ as follows: for each server $s \in R_{\mathcal{A}}$, add *exactly one* client from $\mathcal{A}^{-1}(s)$ to C. Then $|C| = |R_{\mathcal{A}}| \geqslant |U| / d$ and C is matchable in G. By the assumed property (2) of the lemma, it follows that

$$|N_H(U)| \geqslant |N_H(C)| \geqslant \frac{1}{\alpha} |C| \geqslant \frac{1}{\alpha d} |U|,$$

concluding the proof. ∎

3.3 Matching-Contractors The key conceptual contribution of this paper is showing that the *non-existence* of a load-balancing sparsifier is nearly equivalent to the *existence* of an extremal combinatorial object that we call a Matching-Contractor, defined formally as follows.

Definition 3.9 (Matching-Contractor)**.** For any $\alpha \geqslant 1$, we say a bipartite graph $G = (L, R, E)$ is an **α-Matching-Contractor** iff the edge-set E can be partitioned into matchings $M_1 \dots M_k$ such that for each M_i, $L(M_i)$ has at most $|L(M_i)| / \alpha$ neighbors in $G \setminus M_i$; i.e., $\left|N_{G \setminus M_i}(L(M_i))\right| \leqslant |M_i| / \alpha$. Note that M_i can be of different sizes and a Matching-Contractor contains $\sum_i |M_i|$ edges.

It is not difficult to see that a Matching-Contractor cannot be sparsified to preserve the load-balancing properties: if we remove many edges of a matching M_i then $|N_H(L(M_i))| \ll |N_G(L(M_i))|$; see Lemma 4.1 for a proof. So, if there exists a dense Matching-Contractor graph, then this provides a lower bound on the size of a load-balancing sparsifier.

Definition 3.10. Define:

- $\mathsf{MC}(n, \alpha)$ as the largest possible number of edges in any α-Matching-Contractor $G = (L, R, E)$ with $|L| = n$ (notice that there is no requirement on the size of R);
- $\mathsf{MC}(G, \alpha)$ for any given graph $G = (L, R, E)$, as the largest number of edges in any α-Matching-Contractor $H = (L, R, E_H)$ such that H is a subgraph of G.

Note that both $\mathsf{MC}(n, \alpha)$ and $\mathsf{MC}(G, \alpha)$ are monotonically *decreasing* as α increases.

We will prove in Section 4 that $\mathsf{MC}(n, \alpha) \approx \text{SPARSIFIER}(n, \Theta(\alpha))$ up to an $O(\log n)$ factor. The proof uses randomized rounding and linear programming duality in the same spirit of [GKK12] (for "matching sparsifiers" and Ruzsa-Szemerédi graphs), although the details use some problem-specific insights. This shows in a precise sense that a Matching-Contractor is the dual object of a load-balancing sparsifier.

Remark 3.11 (Matching-Contractors and Ruzsa-Szemerédi Graphs)**.** Matching-Contractors are closely related to Ruzsa-Szemerédi graphs in the following sense. An (r, t)-RS graph is any graph whose edges can be partitioned into t induced matchings of size r [RS78]. We can turn a Matching-Contractor $G = (L, R, E)$ with k specified matchings $M_1, \dots, M_k$ into an RS graphs as follows: for each matching M_i of G, remove at most $|M_i| / \alpha$ edges to turn it into an induced matching M'_i (remove the edges of M_i incident on vertices of R that are neighbors to $L(M_i)$ in $G \setminus M_i$). Then, round down size of each matching to a power of two by removing at most half of its remaining edges. Finally, among these $\Theta(\log n)$ different classes of matchings (according to their size), pick the one that contains the largest number of edges overall. This way, we obtain an (r, t)-RS graph G' with

$$\text{density} = \frac{|E|}{\Theta(\log n)}, \quad t \leqslant k, \quad \text{and} \quad r \geqslant \frac{|E|}{k \log n}.$$

However, note that the guarantee of Matching-Contractor is a lot stricter which leads to a "stronger" type of RS graph G': each matching M'_i is not only induced, which means $L(M'_i)$ avoids $R(M'_i)$ in $G' \setminus M'_i$, but in fact $L(M'_i)$ avoids almost the entirety of R, except for $\leqslant |M'_i| / \Theta(\alpha)$ vertices. On the other hand, we should note that unlike RS graphs wherein one explicitly fixes size of the matchings to be some parameter r (sometime as large as $\Theta(n)$ even), in Matching-Contractors, there is no explicit bound on the size of the matchings and they can even be of different size.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

4 Relating Matching-Contractors to Load-Balancing Sparsifiers

The key result of this section is a near-equivalence between $\textsc{sparsifier}(n, \alpha)$ and $\mathsf{MC}(n, \alpha)$.

Theorem 2. *For any integer $n \geqslant 1$ and $\alpha \geqslant 2$,*

$$\mathsf{MC}(n, 2\alpha) \lesssim \textsc{sparsifier}(n, \alpha) \lesssim \mathsf{MC}\big(n, \frac{\alpha}{2}\big) \cdot \ln(n).$$

Moreover, for any bipartite graph G, $\textsc{sparsifier}(G, \alpha) \lesssim \mathsf{MC}(G, \frac{\alpha}{2}) \cdot \ln(n)$.

The direction $\mathsf{MC}(n, 2\alpha) \lesssim \textsc{sparsifier}(n, \alpha)$ is easy and gives a good intuition of the definition of a Matching-Contractor. It shows that the vertex expansion of a Matching-Contractor is very brittle under edge removal.

Lemma 4.1 (Easy Direction)**.** $\mathsf{MC}(n, 2\alpha) \lesssim \textsc{sparsifier}(n, \alpha)$ *for any positive integer n and $\alpha \geqslant 2$.*

Proof. Let $G = (L, R, E)$ be an extremal 2α-Matching-Contractor that contains $\mathsf{MC}(n, 2\alpha)$ edges. We will show that any α-approximate load-balancing sparsifier of G must contain at least half of the edges of G; the lemma then follows as

$$\frac{1}{2} \cdot \mathsf{MC}(n, 2\alpha) = \frac{1}{2} \cdot |E(G)| \leqslant \textsc{sparsifier}(G, \alpha) \leqslant \textsc{sparsifier}(n, \alpha).$$

Let $\{M_i\}_{i=1}^{t}$ be matchings into which $E(G)$ is partitioned according to Definition 3.9. We argue that any α-sparsifier H of G must contain at least half of the edges of *every* M_i, and this will complete the proof of the lemma.

Consider any matching M_i and let L_i be the left endpoints and R_i be the right endpoints. Assume, for contradiction, that $|M_i \cap E(H)| < \frac{1}{2}|M_i|$, and let $L_i' \subseteq L_i$ contain the vertices in L_i whose matching edge from M_i is *not* present in H. By our assumption, we have $|L_i'| > |L_i|/2$. Since G is an 2α-Matching-Contractor, it follows that

$$|N_H(L_i')| \leqslant \left|N_{G\setminus M_i}(L_i)\right| \leqslant \frac{|L_i|}{2\alpha} < \frac{|L_i'|}{\alpha}.$$

This contradicts with H being an α-approximate load-balancing sparsifier of G, as L_i' is matchable in G (see Property (2) of Lemma 3.8). ∎

The rest of this section is to prove the other direction of Theorem 2. To do so, we will introduce an LP-relaxation of $\textsc{sparsifier}(G, \alpha)$, and use its primal to relate to $\textsc{sparsifier}(G, \alpha)$ and use its dual to relate to $\mathsf{MC}(G, \frac{\alpha}{2})$.

4.1 Linear Programming Relaxation for Load-Balancing Sparsification The following is the primal LP that captures the problem of finding an α-approximate load-balancing sparsifier of G with the minimum number of edges.

Definition 4.2 (Primal LP)**.** The Primal LP is defined for a bipartite graph $G = (L, R, E)$ with $|L| = n$ and a parameter $\alpha \geqslant 1$. There is a variable x_e for every edge $e \in E$. Define a pair of sets $X \subseteq L, Y \subseteq R$ to be *contracting* if X is matchable and $|Y| \leqslant \frac{1}{2\alpha} \cdot |X|$. The LP will have a constraint for every contracting pair (X, Y). For input G and α, the primal LP is defined as

$$\begin{aligned} \mathsf{LP}(G, \alpha) \;:=\; & \text{minimize} && \sum_{e \in E} x_e \\ & \text{subject to} && \sum_{e \in E(X, R\setminus Y)} x_e \geqslant \frac{|X|}{2} && \text{for all contracting pair } X, Y \\ & && x_e \geqslant 0 && \text{for all edge } e \in E. \end{aligned}$$

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

For intuition about the primal LP, think of x_e as representing whether edge e is included in the sparsifier H. We want H to satisfy Property (2) of Lemma 3.8, since this is equivalent to H being a load-balancing sparsifier. For any matchable set $X \subseteq L$, this property requires that X should *not* contract to some small set Y. In other words, for every small set Y, there should be many edges from X to $R \setminus Y$, which is precisely the main constraint of the LP.

The following lemma shows that $\mathsf{LP}(G, \alpha)$ is a $O(\log n)$-approximation of $\textsc{sparsifier}(G, \alpha)$. The proof is by a standard randomized rounding argument.

Lemma 4.3 (LP and $\textsc{sparsifier}$)**.** *For a bipartite graph $G = (L, R, E)$ with $|L| = n$ and $|R| \leqslant n^2$,*

$$\mathsf{LP}(G, \alpha) \leqslant \textsc{sparsifier}(G, \alpha) \leqslant 20 \cdot \ln(n) \cdot \mathsf{LP}\big(G, \frac{\alpha}{2}\big).$$

Proof. We first show that $\mathsf{LP}(G, \alpha) \leqslant \textsc{sparsifier}(G, \alpha)$, which says that $\mathsf{LP}(G, \alpha)$ is a relaxation of $\textsc{sparsifier}(G, \alpha)$. Let H be an α-sparsifier of G with $|E(H)| = \textsc{sparsifier}(G, \alpha)$. We will show that the following is a feasible solution to the LP: $x_e = 1$ if $e \in H$ and $x_e = 0$ otherwise. Consider any contracting pair of sets (X, Y), i.e. X is matchable and $|Y| \leqslant \frac{1}{2\alpha}|X|$. Define

$$X' := \{v \in X \mid N_H(v) \subseteq Y\}.$$

We must have $|X'| \leqslant \frac{1}{2}|X|$, otherwise $|Y| < \frac{1}{\alpha}|X'|$ and X' would violate Property (2) of Lemma 3.8 for an α-approximate load-balancing sparsifier. Thus, in H, at least $|X| - |X'| \geqslant |X|/2$ vertices of X have a neighbor in $R \setminus Y$; so, the LP constraint for the contracting pair (X, Y) is satisfied.

We next prove that $\textsc{sparsifier}(G, \alpha) \leqslant 20 \cdot \ln(n) \cdot \mathsf{LP}(G, \frac{\alpha}{2})$, by showing that a fractional solution to $\mathsf{LP}(G, \frac{\alpha}{2})$ can be rounded to an integral solution to $\textsc{sparsifier}(G, \alpha)$ with at most $20 \cdot \ln(n) \cdot \mathsf{LP}(G, \frac{\alpha}{2})$ edges. Given a feasible solution x_e to the $\mathsf{LP}(G, \frac{\alpha}{2})$, we construct an α-approximate load-balancing sparsifier H by a simple randomized rounding procedure as follows. Define $p = 10 \ln(n)$. For every $e \in G$, add e to H with probability $\min\{p \cdot x_e, 1\}$. Note that H has at most $p \sum_e x_e = p \cdot \mathsf{LP}(G, \frac{\alpha}{2})$ edges in expectation. So, by Markov's inequality,

$$\Pr\Big[|E(H)| \leqslant 20 \cdot \ln(n) \cdot \mathsf{LP}\big(G, \frac{\alpha}{2}\big)\Big] \geqslant \frac{1}{2}.$$

We complete the proof by arguing that, with high probability, H satisfies Property (2) of Lemma 3.8, which would imply that an α-sparsifier with the desired number of edges exists by the probabilistic method. We need to show that, with high probability, for any matchable $X \subseteq L$ and $Y \subseteq R$ with $|Y| < \frac{1}{\alpha} \cdot |X|$, we have $E_H(X, R \setminus Y) \neq \emptyset$. Consider any such sets X, Y and denote $k := |X|$. As we are considering $\mathsf{LP}(G, \frac{\alpha}{2})$, the pair (X, Y) is a contracting pair. So, by the LP constraint, we have $\sum_{e \in E(X, R \setminus Y)} x_e \geqslant k/2$. If any edge e in $E(X, R \setminus Y)$ has $p \cdot x_e \geqslant 1$, then this edge e will be added to H with probability 1 and we are done. Henceforth we assume that $px_e < 1$ for all $e \in E(X, R \setminus Y)$. Since edge e is sampled independently with probability px_e, it follows that

$$\Pr\big[E_H(X, R \setminus Y) = \emptyset\big] = \prod_{e \in E(X, R \setminus Y)} (1 - px_e) \leqslant \exp\Bigg(-\sum_{e \in E(X, R \setminus Y)} px_e\Bigg) \leqslant \exp\Big(-\frac{pk}{2}\Big) = n^{-5k}.$$

Fixing any particular $k = |X|$, the number of such set-pairs (X, Y) is clearly at most $n^k \cdot |R|^k \leqslant n^{3k}$, using the assumption of the lemma that $|R| \leqslant n^2$. Therefore, by a union bound, we have $E_H(X, R \setminus Y) \neq \emptyset$ for $|X| = k$ with probability at least $1 - n^{-2k} \geqslant 1 - n^{-2}$. Applying another union bound over all possible k yields probability at least $1 - \frac{1}{n}$ for all such set-pairs (X, Y). ∎

Our next goal is to relate $\mathsf{LP}(G, \alpha)$ to $\mathsf{MC}(G, \alpha)$ (recall Definition 3.10), for which we consider the dual of the above linear program.

Definition 4.4 (Dual LP)**.** The dual LP is again defined for a bipartite graph $G = (L, R, E)$ with $|L| = n$ and a parameter $\alpha \geqslant 1$. There is a variable $y_{X,Y}$ for every contracting set-pair (X, Y), and a constraint for every edge $e \in E$.

$$\mathsf{LP}(G, \alpha) \;:=\; \text{minimize} \quad \frac{1}{2} \sum_{\text{contracting } (X,Y)} |X| \cdot y_{X,Y}$$

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

$$
\begin{aligned}
\text{subject to} \quad & \sum_{\substack{\text{contracting } (X,Y),\\ e\in E_G(X,R\setminus Y)}} y_{X,Y} \leqslant 1 && \text{for all } e \in E \\
& y_{X,Y} \geqslant 0 && \text{for all contracting pair } (X,Y)
\end{aligned}
$$

It is straightforward to check that the LP in Definition 4.4 is indeed the dual program of the LP in Definition 4.2, and so by the strong LP duality theorem they have the same objective value. The following lemma shows that one can construct a Matching-Contractor from the dual LP solution.

Lemma 4.5 (LP and Matching-Contractor)**.** *For any $G = (L, R, E)$ with $|L| = n$ and $\alpha \geqslant 2$,*

$$\mathsf{LP}(G, \alpha) \leqslant 20 \cdot \mathsf{MC}(G, \alpha).$$

The proof is again by a randomized rounding procedure, but the details are more involved, and we dedicate the next subsection to it.

We end this subsection by showing that the hard direction of Theorem 2 follows immediately from Lemma 4.3 and Lemma 4.5.

Proof of Theorem 2 assuming Lemma 4.5. One direction is already proved in Lemma 4.1. For the other direction, we first apply Claim A.1 to reduce the problem to a bipartite graph $G = (L, R, E)$ with $|L| = n$ and $|R| \leqslant n^2$; this claim is quite trivial, so we defer the formal statement and proof to Appendix A.2 in the appendix. Then, Lemma 4.3 and Lemma 4.5 imply that

$$\textsc{sparsifier}(G, \alpha) \leqslant 20 \cdot \ln(n) \cdot \mathsf{LP}\big(G, \frac{\alpha}{2}\big) \leqslant 400 \cdot \ln(n) \cdot \mathsf{MC}(G, \frac{\alpha}{2}),$$

which also implies that $\textsc{sparsifier}(n, \alpha) \leqslant 400 \cdot \ln(n) \cdot \mathsf{MC}(n, \alpha/2)$. ∎

4.2 Constructing Matching-Contractor from Dual Solution

The goal of this subsection is to prove Lemma 4.5. We first provide some intuitions of the proof in Section 4.2.1 by considering the ideal case when all dual variables $y_{X,Y}$ are integral. We then present the general construction in Section 4.2.2 and the analysis in Section 4.2.3.

4.2.1 Proof Ideas

We start with a simple definition and a simple observation.

Definition 4.6. Given a pair of sets (X, Y) with $X \subseteq L$ and $Y \subseteq R$, we say that an edge (u, v) **deviates from** (X, Y) if $u \in X$ and $v \in R \setminus Y$.

To illustrate some proof ideas, we consider the ideal case when all dual variables $y_{X,Y}$ are integral. By the following observation, each $y_{X,Y}$ is either 0 or 1.

Observation 4.7. *Every dual variable $y_{X,Y}$ is at most 1.*

Proof. Suppose for contradiction that $y_{X,Y} > 1$. By the definition of contracting pair (X, Y), X is matchable by some matching M and X is larger than Y. Thus, there must be some edge $(u, v) \in M$ that deviates from (X, Y), but then the dual constraint of (u, v) is violated. ∎

Intuition: Construction from Integral Dual Solution. We create a Matching-Contractor from a feasible dual $\{0, 1\}$-solution as follows. Let $\mathcal{P}$ contain all contracting pairs (X, Y) for which $y_{X,Y} = 1$. For every $(X, Y) \in \mathcal{P}$, let $M_{X,Y}$ be some matching from X to R, which exists by the definition of contracting pairs (see Definition 4.2)[2]. Then, we remove from $M_{X,Y}$ all edges that are incident to Y, and let $M'_{X,Y}$ denote the remaining matching. Let H be the union of all the $M'_{X,Y}$. Clearly, H is a subgraph of G, and that is the complete construction in this simpler setting.

We claim that H is an α-Matching-Contractor with $\gtrsim \mathsf{LP}(G, \alpha)$ edges. First, we lower bound the number of edges in H. Since $\alpha \geqslant 2$ by the assumption of Lemma 4.5 and $Y \leqslant \frac{1}{2\alpha} |X|$ by the definition of a contracting

[2] To avoid confusion, note that Y does not correspond to the right endpoints of $M_{X,Y}$; instead, Y relates the matching to the dual variable $y_{X,Y}$.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

pair, it follows that $\left|M'_{X,Y}\right| = |X| - |Y| \geqslant \frac{3}{4}|X|$. In the full proof in Section 4.2.3, we will show that the matchings $M'_{X,Y}$ are all edge-disjoint. Therefore,

$$\mathsf{LP}(G,\alpha) = \frac{1}{2}\sum_{(X,Y)\in\mathcal{P}}|X| \leqslant \frac{2}{3}\sum_{(X,Y)\in\mathcal{P}}\left|M'_{X,Y}\right| = \frac{2}{3}\cdot|E(H)|,$$

where the first equality is by the observation that $y_{X,Y}\in\{0,1\}$, and the last equality is by the fact that the matchings $M'_{X,Y}$ are edge-disjoint.

It remains to argue that H satisfies the properties of an α-Matching-Contractor in Definition 3.9. Consider some matching $M := M'_{X,Y}$ in H. Let $X' \subseteq X$ be the left endpoints $L(M)$. Recall that $|X'| \geqslant \frac{3}{4}|X|$. Suppose for contradiction that X' has more than $\frac{1}{\alpha}|X'|$ neighbors in $H \setminus M$. Then

$$\left|N_{H\setminus M}(X')\right| > \frac{1}{\alpha}|X'| \geqslant \frac{3}{4\alpha}|X|.$$

Since $|Y| \leqslant \frac{1}{2\alpha}|X|$, there must be some edge (v,z) in $H\setminus M$ such that $v\in X'\subseteq X$ and $z\notin Y$, and so (v,z) deviates from (X,Y). The edge (v,z) must come from some other matching $M'_{P,Q}$ of H. By our construction of matchings, we must have $y_{P,Q}=1$, and also none of the edges of $M'_{P,Q}$ are incident to Q. This implies that the edge (v,z) also deviates from (P,Q) with $v\in P$ and $z\notin Q$. So, the edge (v,z) deviates from both (X,Y) and (P,Q) with $y_{X,Y}=y_{P,Q}=1$, but this means that the dual constraint for (v,z) is violated, arriving at our contradiction. We conclude that H is an α-Matching-Contractor.

4.2.2 Construction from Fractional Dual Solution In general, we have a *fractional* solution to the Dual LP of value $\mathsf{LP}(G,\alpha)$. We will use randomized rounding to construct a subgraph H of G and argue that H is the desired Matching-Contractor with positive probability. We shall note that unlike the argument for the Primal LP in Lemma 4.3, here, the Dual LP has a very large integrality gap and thus the rounding should be *bicriteria* (this will become more clear shortly).

We sample every contracting pair (X,Y) with probability $\frac{1}{10}\cdot y_{X,Y}$ and let $\mathcal{P}$ be the set of all sampled contracting pairs. For every $(X,Y)\in\mathcal{P}$, let $M_{X,Y}$ be a perfect matching from X to R in G; such a matching must exist by the definition of contracting pairs in Definition 4.2. We will fix in advance a matching for every matchable set X, so that the choice of edges in matching $M_{X,Y}$ does not depend on the dual variables or any of our random choices. As before, for every $M_{X,Y}$, remove all edges in $M_{X,Y}$ that are incident to Y, and let $M'_{X,Y}$ be the remaining matching. In the ideal case above, the union of these $M'_{X,Y}$ is a Matching-Contractor. In the general case, however, this is not necessarily true and we will do the following post-processing step to obtain a Matching-Contractor.

Definition 4.8. We say that an edge $(u,v)\in G$ is **overloaded** if there exist two different set-pairs $(X,Y)\in\mathcal{P}$ and $(P,Q)\in\mathcal{P}$ such that (u,v) deviates from both (P,Q) and (X,Y).

In the postprocessing step, for every $M'_{X,Y}$, we remove all edges in $M'_{X,Y}$ that are overloaded, and let $M''_{X,Y}\subseteq M'_{X,Y}$ be the resulting matching. We say $M''_{X,Y}$ is **good** if $\left|M''_{X,Y}\right|\geqslant\frac{1}{2}|X|$. Our final graph H will consist of the union of all good $M''_{X,Y}$. The construction is summarized in Algorithm 1.

4.2.3 Analysis As in Section 4.2.1, we will lower bound the number of edges in H, and prove that H is an α-Matching-Contractor. The following claim will help us upper bound the number of edges that we remove in the post-processing step.

Claim 4.9. *For an edge (u,v) in a matching $M'_{X,Y}$,*

$$\Pr\left[(u,v)\text{ is overloaded}\right]\leqslant\frac{1}{10}.$$

Proof. Consider all contracting pairs $(P,Q)\neq(X,Y)$ such that (u,v) deviates from (P,Q). The edge (u,v) can be overloaded only if one of these (P,Q) is also sampled into $\mathcal{P}$. Since each (P,Q) is sampled independently with probability $\frac{1}{10}y_{P,Q}$ (and in particular independent from (X,Y)), it follows from the union

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

```
Algorithm 1: Construction of Matching-Contractor from Dual LP Solution
  Input: a solution $\{y_{X,Y}\}_{\text{contracting } (X,Y)}$ to the dual LP with objective value $\mathsf{LP}(G,\alpha)$.
    1. Sample every contracting pair $(X,Y)$ into $\mathcal{P}$ with probability $\frac{1}{10} \cdot y_{X,Y}$.
    2. For every $(X,Y) \in \mathcal{P}$ do:
        (a) Let $M_{X,Y}$ be an arbitrary matching from $X$ to $R$.
        (b) Construct $M'_{X,Y} \subseteq M_{X,Y}$ by removing from $M_{X,Y}$ all edges that are incident to $Y$.
        (c) Construct $M''_{X,Y} \subseteq M'_{X,Y}$ by removing all overloaded edges from $M'_{X,Y}$ as defined in
            Definition 4.8. Label $M''_{X,Y}$ as good if $\left|M''_{X,Y}\right| \geqslant \frac{1}{2}|X|$.
  Output: the graph $H$ that is the union of all the good matchings $M''_{X,Y}$.
```

bound and the dual constraint that

$$\Pr\big[(u,v) \text{ is overloaded }\big] \leqslant \sum_{(P,Q)\neq(X,Y) \ | \ (u,v) \text{ deviates from } (P,Q)} \frac{1}{10} \cdot y_{P,Q} \leqslant \frac{1}{10}. \qquad \blacksquare$$

By the same argument as in Section 4.2.1, each $\left|M'_{X,Y}\right| \geqslant \frac{3}{4}|X|$. So, it follows from Claim 4.9 and Markov's inequality that many $M''_{X,Y}$ are good.

Observation 4.10. *Every $M''_{X,Y}$ is good with probability at least $\frac{1}{2}$.*

This allows us to lower bound the number of edges in H.

Claim 4.11. *With positive probability,*

$$|E(H)| \geqslant \frac{1}{20} \cdot LP(G,\alpha).$$

Proof. First we show that the matchings $M''_{X,Y}$ in H are edge-disjoint. Suppose for contradiction that some (u,v) is in both $M''_{X,Y}$ and $M''_{P,Q}$, where $(X,Y),(P,Q) \in \mathcal{P}$. By Step (2b) of Algorithm 1, it follows that (u,v) deviates from both (X,Y) and (P,Q). But this means that (u,v) is overloaded, which contradicts the removal of overloaded edges in Step (2c).

Consider any contracting pair (X,Y). If the pair is sampled into $\mathcal{P}$, then the matching $M''_{X,Y}$ is good with probability at least $\frac{1}{2}$ by Observation 4.10, in which case $\left|M''_{X,Y}\right| \geqslant \frac{1}{2}|X|$ edges are added to H. Each contracting pair (X,Y) is sampled into $\mathcal{P}$ with probability $\frac{1}{10} \cdot y_{X,Y}$. Thus, in expectation, each contracting pair (X,Y) contributes $\frac{1}{40} \cdot y_{X,Y} \cdot |X|$ to $E(H)$. As the matchings $M''_{X,Y}$ are edge-disjoint,

$$\mathbb{E}\,|E(H)| \geqslant \sum_{\text{contracting } (X,Y)} \frac{1}{40} \cdot y_{X,Y} \cdot |X| = \frac{1}{20} \cdot \mathsf{LP}(G,\alpha).$$

We conclude that there exists such an H that satisfies the statement in the claim. $\blacksquare$

We finish the proof of Lemma 4.5 by showing that H is an α-Matching-Contractor. The argument is similar to that in Section 4.2.1.

Claim 4.12. *H satisfies the properties of an α-Matching-Contractor in Definition 3.9.*

Proof. Consider some matching $M := M''_{X,Y}$ and let $X'' \subseteq X$ be the left endpoints $L(M)$. Since M is good by construction, we have $|X''| \geqslant \frac{1}{2}|X|$. Suppose for contradiction that X'' has more than $\frac{1}{\alpha}|X''|$ neighbors in $H \setminus M$. Then $\left|N_{H\setminus M}(X'')\right| > \frac{1}{\alpha}|X''| \geqslant \frac{1}{2\alpha}|X|$. Since $|Y| \leqslant \frac{1}{2\alpha}|X|$ by the definition of a contracting pair, there must be some edge (v,z) in $H \setminus M$ such that $v \in X' \subseteq X$ and $z \notin Y$, and so (v,z) deviates

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

from (X,Y). Since $(v,z) \in H \setminus M$, it must come from some other matching $M''_{P,Q}$ of H. This implies that $(P,Q) \in \mathcal{P}$. Moreover, (v,z) deviates from (P,Q), by Step (2b) of the construction algorithm. Thus, the edge (v,z) deviates from two different pairs in $\mathcal{P}$ – namely (X,Y) and (P,Q) – which means that $(v,z) \in H$ is overloaded, but this contradicts with the removal of overloaded edges in Step (2c) of the construction algorithm. ■

Lemma 4.5 follows immediately from Claim 4.12 and Claim 4.11.

5 A Construction of Matching-Contractors

In this section, we present a simple construction of somewhat dense Matching-Contractors.

Theorem 3 (Density of Matching-Contractors)**.** *For $\varepsilon \in (0,1)$ a sufficiently small constant and $n \geqslant 1$ a sufficiently large integer, there are $(n^{\frac{1}{4}-O(\varepsilon)})$-Matching-Contractors $G = (L,R,E)$ with $n^{1+\Omega(\varepsilon^2)}$ edges where $|L| =: n$ and $|R| = \sqrt{n}$.*

The bipartite graph $G = (L,R,E)$ that we construct has $|L| = w^{2k}$ vertices on the left and $|R| = w^k$ vertices on the right, for some integers $w, k \geqslant 2$. We think of each vertex in L as a string of length $2k$ where each character is in $\{1,\ldots,w\}$, and similarly each vertex in R as a string of length k where each character is in $\{1,\ldots,w\}$. The general idea is to use the "block structure" of the vertices to argue about the contraction property of the matchings added as we will see.

The edge set in G is simple to describe. We add edges to G in t rounds. In each round, we choose a subset of indices $S \subseteq [2k]$ with $|S| = k$. For each vertex $v \in [w]^{2k}$ in L, we define v_S to be the subsequence of v of length k by restricting v to the indices in S (e.g. if $v = \{2,5,3,8\}$ and $S = \{1,4\}$ then $v_S = \{2,8\}$). Note that $v_S \in [w]^k$ corresponds to the unique vertex in R with the same string in $[w]^k$. In each round, for each vertex $v \in L$, we add the edge (v, v_S) in the graph G where $v_S \in R$. To establish the contraction property, we will choose subsets $S_1, S_2, \ldots, S_t$ where the pairwise intersection size $|S_i \cap S_j|$ is small for $i \neq j$, and run the above process for t rounds.

```
Algorithm 2: Construction of Matching-Contractors
  Input: an integer $w \geqslant 2$, an integer $k \geqslant 2$, and $t$ subsets $S_1,\ldots,S_t \subseteq [2k]$ where $|S_i| = k$ for
  $1 \leqslant i \leqslant t$ and $|S_i \cap S_j| \leqslant \ell$ for $1 \leqslant i \neq j \leqslant t$.
  Initialization: $L = [w]^{2k}$, $R = [w]^k$, and $E = \emptyset$.
  For $i$ from 1 to $t$ do:
      for each vertex $v \in [w]^{2k}$ in $L$, add the edge $(v, v_{S_i})$ to $E$ where $v_{S_i} \in [w]^k$ is in $R$.
  Output: the graph $G = (L,R,E)$.
```

To see that G is a Matching-Contractor, we will partition the edges added in each round into w^k matchings of size w^k as follows, where each matching connects a disjoint subset of L to R. In each round, when we fix a subset $S \subseteq [2k]$ of size k, we also consider the complement $\overline{S} := [2k] \setminus S$ and the subsequence $v_{\overline{S}}$ restricting a string $v \in [w]^{2k}$ to the subset $\overline{S}$. For each string $x \in [w]^k$ of length k, we define

$$L_{S,x} := \left\{ v \in [w]^{2k} \mid v_{\overline{S}} = x \right\}.$$

In words, each group $L_{S,x}$ is the subset of vertices in L where we fix the subsequence in $\overline{S}$ to be x. Then $\{L_{S,x}\}_{x \in [w]^k}$ is a partition of L into w^k groups, with one group for each possible x, and each group has w^k vertices. Note that in Algorithm 2, for each group $L_{S,x}$, we added a perfect matching $M_{S,x}$ from $L_{S,x}$ to R where $M_{S,x} := \{(v, v_S) \mid v \in L_{S,x}\}$ is of size w^k. The edge set added in each round is the union of $M_{S,x}$ over all $x \in [w]^k$, and the edge set in the output is the union of $t \cdot w^k$ matchings such that

$$E = \bigcup_{i:1 \leqslant i \leqslant t} \; \bigcup_{x:x \in [w]^k} M_{S_i,x}. \tag{5.3}$$

The reason that we choose $S_1,\ldots,S_t$ to have pairwise intersection size $|S_i \cap S_j| \leqslant \ell$ for $i \neq j$ is explained in the following lemma.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Lemma 5.1 (Contraction Property)**.** *In the output graph $G = (L, R, E)$ of Algorithm 2, for each S_i and each $x \in [w]^k$, the neighbor set of $L_{S_i,x}$ in $G \setminus M_{S_i,x}$ has size*

$$\left|N_{G\setminus M_{S_i,x}}(L_{S_i,x})\right| \leqslant \sum_{j:1\leqslant j\neq i\leqslant t} w^{|S_j\cap S_i|} \leqslant t\cdot w^{\ell}.$$

Proof. Fix S_i and $x \in [w]^k$. In round $j \neq i$, for each vertex $v \in L$, we add the edge (v, v_{S_j}) to E, where the neighbor of v depend only on the values of v in the indices in S_j. By the definition of $L_{S_i,x}$, every vertex v in $L_{S_i,x}$ has the same values in $v_{\overline{S_i}}$ such that $v_{\overline{S_i}} = x$. In particular, every vertex v in $L_{S_i,x}$ has the same values in $v_{\overline{S_i}\cap S_j}$, and thus the neighbors of $L_{S_i,x}$ in round j is contained in the set $\{v_{S_j} \in [w]^k \mid v_{S_j\cap\overline{S_i}} \text{ is fixed}\}$. This set has size exactly $w^{|S_i\cap S_j|}$ since there are w possible choices for each index in $S_j \cap S_i$ but only one choice for each index in $S_j \cap \overline{S_i}$. Summing over all $j \neq i$ gives the first inequality of the lemma, and the second inequality follows by the assumption that $|S_i \cap S_j| \leqslant \ell$ for $j \neq i$. ∎

To instantiate our construction, we need a large set family with small pairwise intersection. The existence of such a family is a standard result in extremal set theory dating back to the work of Erdős and Rényi in [ER56] (e.g., it follows from Gilbert-Varshamov bound in coding theory). For completeness, we present a short proof in Appendix A.3 using a standard probabilistic argument.

Proposition 5.2 (Set Family with Small Pairwise Intersection)**.** *Let $0 < \delta < \frac{1}{2}$ be any constant and let $c_\delta := 2\cdot\delta^\delta\cdot(1-\delta)^{(1-\delta)}$. There exists a set family $\mathcal{F} \subseteq \binom{[2k]}{k}$ of size $\Theta(k^{-\frac{1}{4}}\cdot c_\delta^k)$ such that for all $S \neq S' \in \mathcal{F}$, it holds that $|S \cap S'| \leqslant (1-\delta)k$.*

For two random k-subsets of $[2k]$, the expected size of intersection is $\frac{k}{2}$, and so we cannot hope to have $\delta > \frac{1}{2}$ in the above statement. Moreover, $c_\delta = 1$ at $\delta = \frac{1}{2}$ and is monotonically increasing towards 2 as $\delta \to 0$. As long as δ is bounded away from $1/2$, size of $|\mathcal{F}|$ is exponential in k.

We can now conclude the proof of Theorem 3 by combining Lemma 5.1 and Proposition 5.2.

Proof of Theorem 3. Set $\delta := \frac{1}{2} - 2\varepsilon$. By Proposition 5.2, there is a set family $\mathcal{F}$ with $\Theta(k^{-\frac{1}{4}}\cdot c_\delta^k)$ subsets of size k of $[2k]$ with pairwise intersection size at most $(1-\delta)k = (\frac{1}{2}+2\varepsilon)k$. Use this set family $\mathcal{F}$ as an input to algorithm 2 so that $t = \Theta(k^{-\frac{1}{4}}\cdot c_\delta^k)$ and $\ell = (\frac{1}{2}+2\varepsilon)k$. We will show below that $c_\delta \asymp e^{8\varepsilon^2}$. [3]

The edge set of G is the union of $M_{S_i,x}$ for $S_i \in \mathcal{F}$ and $x \in [w]^k$ as shown in Eq (5.3), where $n := |L| = w^{2k}$ in our construction. By definition, $L_{S_i,x} = L(M_{S_i,x})$. By setting $w = 2$ and $k = \frac{1}{2}\log_2 n$, it follows from Lemma 5.1 that

$$\frac{\left|N_{G\setminus M_{S_i,x}}(L(M_{S_i,x}))\right|}{|M_{S_i,x}|} \leqslant \frac{t\cdot w^l}{w^k} \lesssim \frac{k^{-\frac{1}{4}}\cdot c_\delta^k\cdot w^{(\frac{1}{2}+2\varepsilon)k}}{w^k} \leqslant \frac{e^{8\varepsilon^2 k}}{w^{(\frac{1}{2}-2\varepsilon)k}} = \frac{n^{\Theta(\varepsilon^2)}}{n^{\frac{1}{4}-\varepsilon}} = \frac{1}{n^{\frac{1}{4}-O(\varepsilon)}},$$

and so the bipartite graph G is a $(n^{\frac{1}{4}-O(\varepsilon)})$-Matching-Contractor.

The number of edges in G is $t\cdot n \asymp k^{-\frac{1}{4}}\cdot c_\delta^k\cdot n \asymp \log^{-\frac{1}{4}} n\cdot n^{\Theta(\varepsilon^2)}\cdot n = n^{1+\Theta(\varepsilon^2)}$, by setting $w = 2$ and $k = \frac{1}{2}\log_2 n$ and using that n is sufficiently large.

It remains to show that $c_\delta \asymp e^{8\varepsilon^2}$ for ε a sufficiently small constant.

$$\begin{aligned} c_\delta = c_{\frac{1}{2}-2\varepsilon} &= 2\Big(\frac{1}{2}-2\varepsilon\Big)^{\frac{1}{2}-2\varepsilon}\Big(\frac{1}{2}+2\varepsilon\Big)^{\frac{1}{2}+2\varepsilon} = 2\sqrt{\Big(\frac{1}{2}-2\varepsilon\Big)\Big(\frac{1}{2}+2\varepsilon\Big)}\cdot\left(\frac{\frac{1}{2}+2\varepsilon}{\frac{1}{2}-2\varepsilon}\right)^{2\varepsilon} \\ &= 2\sqrt{\frac{1}{4}-4\varepsilon^2}\cdot\left(\frac{1+4\varepsilon}{1-4\varepsilon}\right)^{2\varepsilon} = \sqrt{1-16\varepsilon^2}\cdot\left(\frac{1+4\varepsilon}{1-4\varepsilon}\right)^{2\varepsilon} \asymp e^{-8\varepsilon^2}\cdot e^{8\varepsilon\cdot 2\varepsilon} = e^{8\varepsilon^2}, \end{aligned}$$

where we used $e^x \asymp 1+x$ when x is sufficiently small. This completes the proof. ∎

[3] The notation $f(n) \asymp g(n)$ means that $f(n)$ and $g(n)$ are asymptotically equal.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

6 Communication Complexity of Load-Balancing

We now prove our lower bound on the randomized communication complexity of the load-balancing problem using the construction of Matching-Contractors developed in Section 5.

Theorem 4. *For any $n \geqslant 1$ and approximation ratio $\alpha \geqslant 1$, any two-player one-way randomized communication protocol for finding an α-approximation to load-balancing with probability of success at least $\frac{2}{3}$ requires $\Omega(\frac{1}{\log n} \cdot \mathsf{MC}(n, \alpha))$ bits of communication. That is,*

$$\vec{R}(\mathsf{LoadBal}(n, \alpha)) \gtrsim \frac{1}{\log n} \cdot \mathsf{MC}(n, 4\alpha).^{4}$$

Combining this with our construction of Matching-Contractors in Theorem 3 on one hand, and the standard reduction from communication to streaming lower bounds on the other hand, gives the following corollaries.

Corollary 6.1. *For any sufficiently large $n \geqslant 1$ and sufficiently small $\varepsilon > 0$,*

$$\vec{R}(\mathsf{LoadBal}(n, n^{\frac{1}{4} - O(\varepsilon)})) = n^{1+\Omega(\varepsilon^2)}.$$

In particular, obtaining any $n^{\frac{1}{4}-o(1)}$-approximation to load-balancing requires strictly more than any $O(n \cdot \text{polylog}\,(n))$ communication.

Corollary 6.2. *There is no semi-streaming algorithm for obtaining a $n^{\frac{1}{4}-o(1)}$-approximation to the load-balancing problem with probability of success at least $\frac{2}{3}$.*

The rest of this section is dedicated primarily to the proof of Theorem 4. We start by defining a simple family of graphs that "encode" different strings inside Matching-Contractors. We then use these graphs to define our hard input distribution. After that, we recall some basic information theory and use them to derive the proof of Theorem 4. We provide short and standard proofs of Corollaries 6.1 and 6.2. Finally, we conclude with the complete proof of Theorem 1 that shows the equivalence between load balancing sparsifiers and one-way communication complexity of load-balancing in Section 6.4.

6.1 Encoding Graphs and the Hard Input Distribution

Let $G_0 := (L_0, R_0, E_0)$ be a (4α)-Matching-Contractorwith

$$|L_0| = n \qquad \text{and} \qquad m_0 \gtrsim \frac{1}{\log n} \cdot \mathsf{MC}(n, 4\alpha)$$

edges and matchings $M_1^0, \ldots, M_k^0$ such that there exists an integer $r_0 \in [n]$ with

$$r_0 \leqslant \left|M_i^0\right| < \frac{4}{3} \cdot r_0 \quad \text{for all } i \in [k].$$

The existence of such a graph follows by grouping matchings of any α-Matching-Contractor with density $\mathsf{MC}(n, 4\alpha)$ based on sizes of matchings relative to powers of $(4/3)$, and picking the group with the largest number of edges.

Our encoding graphs are defined as follows.

Definition 6.3 (Encoding Graphs)**.** Fix a graph $G_0 := (L_0, R_0, E_0)$ as described above. Let $x \in \{0,1\}^{E_0}$ be any string whose entries are indexed by edges in E_0. Define the **encoding graph** $G_x := (L, R, E_x)$ of x inside G_0 as follows:

- $L := L_0$ – we use vertices $u \in L_0$ and $u \in L$ interchangeably.
- $R := R_0 \times \{0, 1\}$ – vertices in R are denoted by v_0 and v_1 for $v \in R_0$;

[4] The constant 4 in the $\mathsf{MC}(n, 4\alpha)$ term in this theorem can be replaced with any other constant strictly larger than two. Since the choice of the constant is immaterial for our purpose, we have not attempted to optimize it.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

- E_x: for every edge $e = (u, v) \in E_0$, there is exactly one of the edges (u, v_0) or (u, v_1) depending on whether $x_e = 0$ or $x_e = 1$, respectively.

Observation 6.4. *For any $x \in \{0,1\}^{E_0}$, the graph G_x is a (2α)-Matching-Contractor with matchings $M_1, \ldots, M_k$, where each M_i is obtained from M_i^0 by mapping the edge $(u, v) \in M_i^0$ to either (u, v_0) or (u, v_1), depending on whichever one exist in G_x.*

Proof. Consider a choice of G_x and one of its designated matchings M_i for $i \in [k]$. By construction,

$$N_{G_x \setminus M_i}(L(M_i)) \subseteq N_{G_0 \setminus M_i^0}(L(M_i^0)) \times \{0, 1\}\,.$$

The proof follows since $|N_{G_0 \setminus M_i^0}(L(M_i^0))| \leqslant \frac{1}{4\alpha} \left|M_i^0\right|$ as G^0 is a (4α)-Matching-Contractor. ∎

Our hard distribution of inputs is defined as follows. We emphasize that the graph G_0 is fixed throughout and both players know this graph.

Input distribution μ.

- **Alice:** Sample $x \in \{0,1\}^{E_0}$ uniformly at random and give the encoding graph G_x to Alice.
- **Bob:** Sample $i \in [k]$ uniformly at random and consider the matching M_i of G_x. Give a perfect matching M from $L \setminus L(M_i)$ to a new set of (server) vertices as the input to Bob.

Given we have fixed the choice of the graph G_0, the following observation is immediate.

Observation 6.5. *In the distribution μ, the input to Alice is uniquely identified by $x \in \{0,1\}^{E_0}$ and the input to Bob is uniquely identified by $i \in [k]$.*

6.2 Analysis of the Input Distribution

Let π be any *deterministic* protocol for $\mathsf{LoadBal}(n, \alpha)$ that succeeds with probability at least $\frac{2}{3}$ on the inputs sampled from the distribution μ.

We use $\pi(x)$ to denote the message of Alice to Bob in the protocol (which by Observation 6.5 is only a function of $x \in \{0,1\}^{E_0}$, hence the notation $\pi(x)$). Similarly, we use $a(\pi(x), i)$ to denote the *assignment* output by Bob, given the message $\pi(x)$ and the index $i \in [k]$ as input (again, using Observation 6.5).

In the following, we use (X, I, Π) to denote, respectively, the *random variable* for the input x of Alice, the input i of Bob, and the message $\pi(x)$ of Alice. We further use X_I to denote the subsequence of X that corresponds to the edges in M_I, namely, the "special" matching corresponding to Bob's input.

Given that protocol π is deterministic, the randomness of all these variables comes solely from the distribution μ of the inputs, namely, the choice of $(X, I) \sim \mu$. We prove that protocol π recovers a large fraction of the values in X_I whenever it outputs a correct answer.

Lemma 6.6. *On any input $(x, i) \sim \mu$ that π outputs a correct answer, at least half the values in x_i are deterministically fixed given only $\pi(x)$ and i, where x_i denotes the subsequence of x corresponding to edges of M_i.*

Proof. On any input (x, i) sampled from μ, there is an L-perfect matching: match $L \setminus L(M_i)$ using the new edges outside G_x given to Bob and use edges of M_i for assigning the vertices in $L(M_i)$ to $R(M_i)$. This means that in any α-approximate solution, load of any vertex in R can be at most α.

Let $A = a(\pi(x), i)$ be the assignment output by Bob. Consider vertices S in $L(M_i)$ that do not use edges of M_i in the assignment A. Since by Observation 6.4, G_x is a (2α)-Matching-Contractor, all these vertices are incident on at most $\frac{1}{2\alpha} \cdot |M_i|$ vertices $T \subseteq R$. Thus, the load of some vertex in T is at least

$$\frac{|S|}{|T|} \geqslant \frac{2\alpha \cdot |S|}{|M_i|}.$$

Combining this with the upper bound of α on the load implies that $|S| \leqslant \frac{1}{2} |M_i|$. This means that whenever Bob's output is correct, it contains at least $\frac{1}{2} |M_i|$ edges from M_i. But each such edge uniquely identifies the

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

corresponding bit in x by the construction of G_x. Thus, whenever the protocol is correct on input (x, i), at least half the values of x_i should be determined, given the assignment $a(\pi(x), i)$ output by Bob. ∎

Using this lemma and the independence of the input to Alice and Bob (by Observation 6.5), we can conclude the proof using a (very) basic information theory argument. We first provide a brief refresher of basic information theory tools that we use in this proof.

6.2.1 A Quick Refresher on Information Theory For a random variable Y on support Ω with distribution $p(y)$ for $y \in \Omega$, the **entropy** of Y is:

$$\mathbb{H}(Y) := \sum_{y \in \Omega} p(y) \cdot \log \frac{1}{p(y)}.$$

The **conditional entropy** of a random variable Y conditioned on another random variable Z is:

$$\mathbb{H}(Y \mid Z) := \mathop{\mathbb{E}}_{z \sim Z} \left[\mathbb{H}(Y \mid Z = z) \right],$$

where $\mathbb{H}(Y \mid Z = z)$ is the entropy of the random variable distributed as $p(y \mid Z = z)$ for $y \in \Omega$.

We need the following facts about (conditional) entropy (the proofs of the statements in this fact are all standard applications of Jensen's inequality and can be found, e.g., in [?]).

Fact 6.7 (cf. [?]). *For any random variables Y, Z with support Ω:*

1. $0 \leqslant \mathbb{H}(Y) \leqslant \log |\Omega|$*; moreover, the left (respectively, right) inequality is tight if and only if Y is deterministic (respectively, uniformly distributed over Ω);*
2. $\mathbb{H}(Y \mid Z) \leqslant \mathbb{H}(Y)$*: conditioning (on a random variable) can only reduce the entropy;*
3. $\mathbb{H}(Y, Z) = \mathbb{H}(Y) + \mathbb{H}(Z \mid Y)$*: chain rule of entropy (also true for conditional entropy).*

6.2.2 Proof of Theorem 4 By Lemma 6.6, with probability at least $\frac{2}{3}$, Bob, given only Π and $I \in [k]$ can recover at least half the values in X_I. This means there is "considerably less" uncertainty about X_I conditioned on Π and I, than without this conditioning. The following claim formalizes this.

Claim 6.8. $\mathbb{H}(X_I \mid \Pi, I) \leqslant \frac{8}{9} \cdot r_0 + 1.$

Proof. Let $Z \in \{0, 1\}$ be the indicator random variable for the event of Lemma 6.6, namely, $Z = 1$ iff the output by Bob is correct and at least half the indices in X_I are fixed by Π and I. Then

(by chain rule in Fact 6.7-(3))

$$\mathbb{H}(X_I \mid \Pi, I) = \mathbb{H}(X_I, Z \mid \Pi, I) - \mathbb{H}(Z \mid X_I, \Pi, I)$$

(again, by chain rule)

$$= \mathbb{H}(X_I \mid \Pi, I, Z) + \mathbb{H}(Z \mid \Pi, I) - \mathbb{H}(Z \mid X_I, \Pi, I)$$

(as conditional entropy of Z is non-negative and is at most 1 by Fact 6.7-(1))

$$\leqslant \mathbb{H}(X_I \mid \Pi, I, Z) + 1$$

(by the definition of conditional entropy)

$$\begin{aligned} &= \Pr(Z = 1) \cdot \mathbb{H}(X_I \mid \Pi, I, Z = 1) + \Pr(Z = 0) \cdot \mathbb{H}(X_I \mid \Pi, I, Z = 0) + 1 \\ &\leqslant \frac{2}{3} \cdot \frac{1}{2} \cdot \left(\frac{4}{3} \cdot r_0 \right) + \frac{1}{3} \cdot \left(\frac{4}{3} \cdot r_0 \right) + 1 \\ &= \frac{8}{9} \cdot r_0 + 1. \end{aligned}$$

where we used the following in the second to last step: the probability of $Z = 1$ is at least $\frac{2}{3}$ and conditioned on $Z = 1$, Π and I reveal at least half of the indices of X_I by Lemma 6.6, and so the entropy of X_I is at most $\frac{1}{2} \cdot \frac{4}{3} \cdot r_0$ by Fact 6.7-(1) since its unfixed part is a binary string of at most this length; the other term corresponding to $Z = 0$ is bounded by Fact 6.7-(1) by simply using the fact that X_I is a binary string of length at most $\frac{4}{3} \cdot r_0$ to begin with. ∎

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

On the other hand, given that Alice is unaware of the choice of I, the only way for her to reduce the uncertainty about X_I substantially, is to reduce the *overall* uncertainty about her entire input. This in turn requires Alice to communicate a lot. The following claim formalizes this.

Claim 6.9. $\mathbb{H}(X_I \mid \Pi, I) \geqslant r_0 - \frac{1}{k} \cdot \|\pi\|$.

Proof. By the definition of conditional entropy,

$$\mathbb{H}(X_I \mid \Pi, I) = \sum_{i=1}^{k} \Pr(I = i) \cdot \mathbb{H}(X_i \mid \Pi, I = i)$$

(as I is distributed uniformly over $[k]$)

$$= \frac{1}{k} \cdot \sum_{i=1}^{k} \mathbb{H}(X_i \mid \Pi, I = i)$$

(as the joint distribution of $(X_i, \Pi = \Pi(X))$ is independent of the event $I = i$ by Observation 6.5)

$$= \frac{1}{k} \cdot \sum_{i=1}^{k} \mathbb{H}(X_i \mid \Pi)$$

(as conditioning can only reduce the entropy by Fact 6.7-(2))

$$\geqslant \frac{1}{k} \cdot \sum_{i=1}^{k} \mathbb{H}(X_i \mid \Pi, X_1, \ldots, X_{i-1})$$

(by the chain rule of entropy in Fact 6.7-(3))

$$= \frac{1}{k} \cdot \mathbb{H}(X \mid \Pi)$$

(by applying chain rule and non-negativity of entropy)

$$\geqslant \frac{1}{k} \cdot (\mathbb{H}(X) - \mathbb{H}(\Pi))$$

$$\geqslant \frac{1}{k} \cdot (m_0 - \|\pi\|),$$

where the last step holds because the distribution of X is uniform over 2^{m_0} binary strings of length m_0 and the support of messages in π are of size $2^{\|\pi\|}$; hence, in both cases we can apply the inequality of Fact 6.7-(1) (which is tight for $\mathbb{H}(X)$, and provides an upper bound for $\mathbb{H}(\Pi)$). Noting that $m_0 \geqslant k \cdot r_0$ concludes the proof. ∎

We are ready to conclude the proof of Theorem 4.

Proof of Theorem 4. The lower bound holds trivially whenever $\mathsf{MC}(n, 4\alpha) = O(n \log n)$ because even if the entire input is a random perfect matching given to Alice, she needs to communicate $\Omega(n \log n)$ bits to send one edge per each vertex in L to Bob which is needed for any finite approximation. Thus, in the following, we focus on the (only interesting) case when $\mathsf{MC}(n, 4\alpha) = \omega(n \log n)$, which implies that $r_0 = \omega(1)$.

By the easy direction of Yao's minimax principle, to prove the lower bound, we only need to focus on deterministic protocols that succeeds with probability at least $\frac{2}{3}$ on inputs sampled from the distribution μ (this is simply an averaging argument over randomness of the protocol against the input distribution). For any such protocol π, by Claim 6.8 and Claim 6.9,

$$r_0 - \frac{1}{k} \cdot \|\pi\| \leqslant \frac{8}{9} \cdot r_0 + 1,$$

which implies that

$$\|\pi\| \geqslant \frac{1}{9} \cdot (k \cdot r_0) - k \gtrsim \frac{1}{\log n} \cdot \mathsf{MC}(n, 4\alpha),$$

given that $r_0 = \omega(1)$. This concludes the proof. ∎

6.3 Proofs of Corollaries 6.1 and 6.2 We now provide short and standard proofs of Corollaries 6.1 and 6.2 using the results we established already.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Proof of Corollary 6.1. By Theorem 4, (randomized) one-way communication complexity of α-approximation of load-balancing can be lower bounded, up to a $\approx \log n$ term, by the density of $\Theta(\alpha)$-Matching-Contractors. Setting $\alpha = n^{\frac{1}{4}-O(\varepsilon)}$ and using our construction of $(n^{\frac{1}{4}-O(\varepsilon)})$-Matching-Contractors in Theorem 3 with $n^{1+\Omega(\varepsilon^2)}$ edges implies this corollary (note that the extra $\log n$ term is subsumed by the hidden-constant of the Ω-notation in the exponent).

The final part of the corollary holds by taking $\varepsilon \to 0$ in the limit. ∎

Proof of Corollary 6.2. We use the well-established fact that one-way communication complexity lower bounds imply space lower bounds for single-pass streaming algorithms. Given a streaming algorithm A for load-balancing, we obtain a one-way communication protocol as follows: Alice runs A by treating her part of the input as the first part of the stream and then communicates the memory content of the algorithm to Bob, who continue running A on his part of the input as the second part of the stream. This way, at the end, Bob obtains the output of A on the entire input, with communication cost from Alice being at most equal to the worst-case memory size of the algorithm.

The lower bound for streaming algorithms now follows immediately from the above reduction and Corollary 6.1. ∎

6.4 Proof of Theorem 1: Sparsifiers = One-Way Communication

Finally, we provide the proof of our main equivalence result in Theorem 1, restated below.

Theorem (Restatement of Theorem 1)**.** *Suppose there is a (randomized) communication protocol π for $\mathsf{LoadBal}(n, \alpha)$ with communication cost $\|\pi\| \leqslant C$ and probability of success at least $2/3$. Then,*

$$\textsc{sparsifier}(n, 8\alpha) \lesssim C \cdot \log^2(n).$$

Proof. Suppose towards a contradiction that $\textsc{sparsifier}(n, 8\alpha) \geqslant \eta \cdot C \cdot \log^2 n$ for some sufficiently large constant η. Then,

- By Theorem 2, this implies that $\mathsf{MC}(G, 4\alpha) \gtrsim \eta \cdot C \cdot \log(n)$;
- By Theorem 4, this in turn implies that $\vec{R}(\mathsf{LoadBal}(n, \alpha)) \gtrsim \eta \cdot C$.
- By taking η to be a sufficiently large constant, this contradicts the fact that there is a randomized protocol π for $\mathsf{LoadBal}(n, \alpha)$ with $\|\pi\| \leqslant C$.

Thus, our contradicting assumption is false, and the theorem holds. ∎

Acknowledgments

We would like to thank Thatchaphol Saranurak for many fruitful discussions in the early stages of this project.

Part of this work was conducted while the first named author was visiting the Simons Institute for the Theory of Computing as part of the Sublinear Algorithms program.

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

References

[A24] S. Assadi. A simple $(1-\varepsilon)$-approximation semi-streaming algorithm for maximum (weighted) matching. In *Symposium on Simplicity in Algorithms, SOSA 2024*, pages 337–354, 2024. 1

[AB19] S. Assadi and A. Bernstein. Towards a unified theory of sparsification for matching problems. In *2nd Symposium on Simplicity in Algorithms, SOSA 2019, January 8-9, 2019*, volume 69, pages 11:1–11:20, 2019. 3

[ABKL23] S. Assadi, S. Behnezhad, S. Khanna, and H. Li. On regularity lemma and barriers in streaming and dynamic matching. In *STOC '23: 55th Annual ACM SIGACT Symposium on Theory of Computing*, 2023. 3

[ABL20] S. Assadi, A. Bernstein, and Z. Langley. Improved bounds for distributed load balancing. In *34th International Symposium on Distributed Computing, DISC 2020*, volume 179, pages 1:1–1:15, 2020. 1

[ABL23] S. Assadi, A. Bernstein, and Z. Langley. All-norm load balancing in graph streams via the multiplicative weights update method. In *14th Innovations in Theoretical Computer Science Conference, ITCS 2023*, volume 251, pages 7:1–7:24, 2023. 1

[AG11] K. J. Ahn and S. Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011*, pages 526–538, 2011. 1

[AK24] S. Assadi and S. Khanna. Improved bounds for fully dynamic matching via Ordered Ruzsa-Szemerédi graphs. *arXiv preprint arXiv:2406.13573*, 2024. 3

[AKL16] S. Assadi, S. Khanna, and Y. Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 698–711, 2016. 1

[AKL17] 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*, pages 1723–1742, 2017. 3

[AKLY16] 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*, pages 1345–1364, 2016. 1, 3

[AKNS24] S. Assadi, C. Konrad, K. K. Naidu, and J. Sundaresan. $O(\log \log n)$ passes is optimal for semi-streaming maximal independent set. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024*, pages 847–858, 2024. 3

[ALPZ21] S. Ahmadian, A. Liu, B. Peng, and M. Zadimoghaddam. Distributed load balancing: A new framework and improved guarantees. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, volume 185, pages 79:1–79:20, 2021. 1

[AMS12] N. Alon, A. Moitra, and B. Sudakov. Nearly complete graphs decomposable into large induced matchings and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, pages 1079–1090, 2012. 3, 4

[AR20] S. Assadi and R. Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 342–353, 2020. 3

[AS23] S. Assadi and J. Sundaresan. Hidden permutations to the rescue: Multi-pass streaming lower bounds for approximate matchings. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*, pages 909–932, 2023. 3

[BCS74] J. Bruno, E. G. Coffman, Jr., and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Comm. ACM*, 17:382–387, 1974. 1

[Ber20] A. Bernstein. Improved bounds for matching in random-order streams. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, pages 12:1–12:13, 2020. 3

[BG24] S. Behnezhad and A. Ghafari. Fully dynamic matching and Ordered Ruzsa-Szemerédi graphs. *CoRR*, abs/2404.06069. To appear in FOCS 2024, 2024. 3

[BO20] L. Barenboim and G. Oren. Distributed backup placement in one round and its applications to maximum matching and self-stabilization. In *Proc. 3rd Symposium on Simplicity in Algorithms*, pages 99–105, 2020. 1

[CHSW12] A. Czygrinow, M. Hanćkowiak, E. Szymańska, and W. Wawrzyniak. Distributed 2-approximation algorithm for the semi-matching problem. In *Proc. 26th International Symposium on Distributed Computing*, volume 7611, pages 210–222, 2012. 1

[CKP$^+$21] L. Chen, G. Kol, D. Paramonov, R. R. Saxena, Z. Song, and H. Yu. Almost optimal super-constant-pass streaming lower bounds for reachability. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing 2021*, pages 570–583, 2021. 3

[DIMV14] E. D. Demaine, P. Indyk, S. Mahabadi, and A. Vakilian. On streaming and communication complexity of the set cover problem. In *Distributed Computing - 28th International Symposium, DISC 2014*, volume 8784, pages 484–498, 2014. 1

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

[ER56] P. Erdős and A. Rényi. On some combinatorical problems. *Publ. Math. Debrecen*, 4:398–405, 1956. 15

[FKM$^+$05] 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. 1

[FLN$^+$02] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, 2002*, pages 474–483, 2002. 3, 4

[FLN14] J. Fakcharoenphol, B. Laekhanukit, and D. Nanongkai. Faster algorithms for semi-matching problems. *ACM Trans. Algorithms*, 10(3):Art. 14,23, 2014. 1

[FMU22] M. Fischer, S. Mitrovic, and J. Uitto. Deterministic $1+\varepsilon$-approximate maximum matching with $\mathrm{poly}(1/\varepsilon)$ passes in the semi-streaming model and beyond. In *54th Annual ACM SIGACT Symposium on Theory of Computing, 2022*, pages 248–260, 2022. 1

[FNSZ20] M. Feldman, A. Norouzi-Fard, O. Svensson, and R. Zenklusen. The one-way communication complexity of submodular maximization with applications to streaming and robustness. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 1363–1374, 2020. 1

[GKK12] 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 2012*, pages 468–485, 2012. 1, 3, 4, 8

[Hal87] P. Hall. On representatives of subsets. *Classic Papers in Combinatorics*, pages 58–62, 1987. 5

[HKPSR18] M. M. Halldórsson, S. Köhler, B. Patt-Shamir, and D. Rawitz. Distributed backup placement in networks. *Distrib. Comput.*, 31(2):83–98, 2018. 1

[HLLT06] N. J. A. Harvey, R. E. Ladner, L. Lovász, and T. Tamir. Semi-matchings for bipartite graphs and load balancing. *J. Algorithms*, 59(1):53–78, 2006. 1

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–562, 2006. 4, 6, 7

[Hor73] W. A. Horn. Minimizing average flow time with parallel machines. *Oper. Res.*, 21(3), 1973. 1

[JR17] K. Jansen and L. Rohwedder. On the configuration-lp of the restricted assignment problem. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 2670–2678, 2017. 1

[JR20] K. Jansen and L. Rohwedder. A quasi-polynomial approximation for the restricted assignment problem. *SIAM J. Comput.*, 49(6):1083–1108, 2020. 1

[Kap13] 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*, pages 1679–1697, 2013. 3

[Kap21] M. Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1874–1893. SIAM, 2021. 3

[KKA23] S. Khanna, C. Konrad, and C. Alexandru. Set cover in the one-pass edge-arrival streaming model. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2023*, pages 127–139, 2023. 1

[KN97] E. Kushilevitz and N. Nisan. *Communication complexity.* Cambridge University Press, 1997. 5

[KR13a] C. Konrad and A. Rosén. Approximating semi-matchings in streaming and in two-party communication. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013*, volume 7965, pages 637–649, 2013. 1, 2, 3, 5, 6

[KR13b] C. Konrad and A. Rosén. Approximating semi-matchings in streaming and in two-party communication. 2013, 1304.6906. 5

[LL04] Y. Lin and W. Li. Parallel machine scheduling of machine-dependent jobs with unit-length. *European J. Oper. Res.*, 156(1):261–266, 2004. 1

[LST90] J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.*, 46:259–271, 1990. 1

[MV17] A. McGregor and H. T. Vu. Better streaming algorithms for the maximum coverage problem. In *20th International Conference on Database Theory, ICDT 2017*, volume 68, pages 22:1–22:18, 2017. 1

[OBL18] G. Oren, L. Barenboim, and H. Levin. Distributed fault-tolerant backup-placement in overloaded wireless sensor networks. In *Proc. 9th International Conference on Broadband Communications, Networks, and Systems*, pages 212–224, 2018. 1

[OZ22] S. Olesker-Taylor and L. Zanetti. Geometric bounds on the fastest mixing markov chain. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, volume 215, pages 109:1–109:1, 2022. 7

[RS78] I. Z. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1978. 2, 3, 8

[RY20] A. Rao and A. Yehudayoff. *Communication Complexity: and Applications.* Cambridge University Press, 2020. 5

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

[Yao79] A. C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, 1979*, pages 209–213, 1979. 5

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

Appendix

A Deferred Proofs

A.1 Proof of Proposition 3.4

Proof. The protocol is simple: Alice computes an α-approximation load-balancing sparsifier H_A of G_A with at most T edges and sends H_A to Bob. Each edge requires $O(\log(n))$ bits to send, for a total of $O(T\log(n))$ bits.

Define $G := G_A \cup G_B$ and $G' := H_A \cup G_B$. To prove correctness, we need to show that $\mathrm{OPT}\textsc{Load}(G') \leqslant \alpha \cdot \mathrm{OPT}\textsc{Load}(G)$. Let $\mathcal{A}^*$ be the optimal assignment in G. We partition L into sets L_A and L_B: for every $x \in L$, add x to L_A if $(x, \mathcal{A}^*(x)) \in G_A$ and add x to L_B if $(x, \mathcal{A}^*(x)) \in G_B$; if the edge is in both G_A and G_B, then assign x to L_B.

Let $\mathcal{A}^*_A$ denote the assignment $\mathcal{A}^*$ restricted to L_A and define $\mathcal{A}^*_B$ analogously. Note that $\mathcal{A}^*_A$ is an assignment in G_A, $\mathcal{A}^*_B$ is an assignment in G_B, and $\textsc{load}(\mathcal{A}^*_A), \textsc{load}(\mathcal{A}^*_B)$ are both at most $\textsc{load}(\mathcal{A}^*)$. Since $\mathcal{A}^*_B$ is contained in $G_B \subseteq G'$, we can also use in our final assignment for G'. But we must replace $\mathcal{A}^*_A$ with a new assignment that is contained in H_A. To this end, note that by the existence of $\mathcal{A}^*_A$ we have $\mathrm{OPT}\textsc{Load}(G_A[L_A \cup R]) \leqslant \textsc{load}(\mathcal{A}^*_A) \leqslant \textsc{load}(\mathcal{A}^*)$, so by definition of a load-balancing sparsifier, there exists an assignment $\mathcal{A}'_A$ of $H_A[L_A \cup R]$ with $\textsc{load}(\mathcal{A}'_A) \leqslant \alpha \cdot \textsc{load}(\mathcal{A}^*_A) \leqslant \alpha \cdot \textsc{load}(\mathcal{A}^*)$.

We now define an assignment $\mathcal{A}'$ of G' as follows: for $x \in L_A$ we set $\mathcal{A}'(x) = \mathcal{A}'_A(x)$ and for $x \in L_B$ we set $\mathcal{A}'(x) = \mathcal{A}^*_B(x)$. It is easy to see that all edges of A' are contained in G', and that $\textsc{load}(\mathcal{A}') \leqslant \textsc{load}(\mathcal{A}'_A) + \textsc{load}(\mathcal{A}^*_B) \leqslant (\alpha+1)\textsc{load}(\mathcal{A}^*)$, as desired. $\blacksquare$

A.2 Reducing the Number of Servers In order to apply Lemma 4.3 inside Theorem 2, we rely on the following claim, which shows that one can assume w.l.o.g that $|R| \leqslant |L|^2$.

Claim A.1. *Given a bipartite graph $G = (L, R, E)$ with $|L| = n$, there exists a subgraph $G' = (L, R', E')$ of G such that $|R'| \leqslant n^2$ with $\textsc{sparsifier}(G, \alpha) \leqslant \textsc{sparsifier}(G', \alpha)$ and $\mathsf{MC}(G', \alpha) \leqslant \mathsf{MC}(G, \alpha)$ for any $\alpha \geqslant 1$.*

Proof. Define a vertex $v \in L$ to be high-degree if $\deg_G(v) > n$. We define E' as follows: start with $E' = E$, and then for every high-degree vertex $v \in L$, remove an aribtrary set of $\deg_G(V) - n$ edges incident to v. Every $v \in L$ now has $\deg_{E'}(v) \leqslant n$. Let R' contain all vertices in R with at least one incident edge in E'; it is easy to see that $|R'| \leqslant n^2$. We then define $G' := (L, R', E')$

To show that $\textsc{sparsifier}(G, \alpha) \leqslant \textsc{sparsifier}(G', \alpha)$, we argue that any α-sparsifier H' of G' is also an α-sparsifier of G. Using the criterion of load-balancing sparsifiers in Property (2) of Lemma 3.8, it suffices to show that any set $X \subseteq L$ that is matchable in G is also matchable in G'. Let M be the matching from X to R in G; we argue that X is also matchable in G'. For every $u \in X$, if u is not of high-degree in G, then $E_G(u) = E_{G'}(u)$, so we can use the same edge from M. If u is of high degree in G, then since $\deg_{G'}(u) = n$, there must be at least one free vertex in R' that u can be matched to.

The inequality $\mathsf{MC}(G', \alpha) \leqslant \mathsf{MC}(G, \alpha)$ follows from the fact that G' is a subgraph of G. $\blacksquare$

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited

A.3 Proof of Proposition 5.2

Proof of Proposition 5.2. We use the probabilistic method. Suppose we pick t random subsets $S \in \binom{[2k]}{k}$ independently and uniformly at random. For any two random subsets S and S', we would like to bound the probability that their intersection size is equal to ℓ for any $\frac{k}{2} \leqslant \ell \leqslant k$. If S and S' are chosen uniformly and independently, then

$$\Pr[|S \cap S'| = \ell] = \frac{\binom{k}{\ell}\binom{k}{k-\ell}}{\binom{2k}{k}} = \frac{\binom{k}{k-\ell}^2}{\binom{2k}{k}}.$$

To see this, if we fix S, the numerator counts the number of subsets of size k that intersects S exactly ℓ times. This can be done by picking ℓ elements from S and picking $k-\ell$ elements from $\overline{S}$, for both of which there are exactly $\binom{k}{\ell} = \binom{k}{k-\ell}$ ways. By the union bound, the probability that any two sets have intersection size greater than $k-s$ is

$$\Pr[|S \cap S'| \geqslant k-s] \leqslant \sum_{\ell=k-s}^{k} \frac{\binom{k}{k-\ell}^2}{\binom{2k}{k}} = \sum_{\ell=0}^{s} \frac{\binom{k}{\ell}^2}{\binom{2k}{k}} \leqslant s \cdot \frac{\binom{k}{s}^2}{\binom{2k}{k}}$$

Let $s = \delta k$ for some positive constant $\delta < \frac{1}{2}$. Using Stirling's approximation $n! \asymp \sqrt{2\pi n}(n/e)^n$,

$$\begin{aligned}
s \cdot \frac{\binom{k}{s}^2}{\binom{2k}{k}} &\asymp s \cdot \left(\frac{\sqrt{2\pi k}\cdot k^k}{2\pi\sqrt{s(k-s)}s^s(k-s)^{k-s}}\right)^2 \Bigg/ \frac{\sqrt{4\pi k}(2k)^{2k}}{2\pi k \cdot k^{2k}} \\
&\asymp \frac{sk^{1.5}}{s(k-s)} \cdot \frac{k^{4k}}{(2k)^{2k}s^{2s}(k-s)^{2(k-s)}} \\
&= \frac{\sqrt{k}}{(1-\delta)} \cdot \frac{k^{4k}}{(2k)^{2k}(\delta k)^{2\delta k}(k(1-\delta))^{2k(1-\delta)}} \\
&= \frac{\sqrt{k}}{(1-\delta)} \cdot \frac{1}{(4 \cdot \delta^{2\delta} \cdot (1-\delta)^{2(1-\delta)})^k} \\
&\leqslant 2\sqrt{k} \cdot c_\delta^{-2k}
\end{aligned}$$

Finally, by using the union bound over all pairs of random subsets, we see that as long as $t \leqslant \frac{1}{2}k^{-\frac{1}{4}} \cdot c_\delta^k$, then every pair of subsets has intersection size less than $(1-\delta)k$ with positive probability. This implies that there exists a set family $\mathcal{F}$ as claimed in the statement. ∎

Copyright © 2025 by SIAM
Unauthorized reproduction of this article is prohibited