Simultaneously Approximating All Norms for Massively Parallel Correlation Clustering

Nairen Cao
Department of Computer Science and Engineering, New York University, NY, USA

Shi Li
School of Computer Science, Nanjing University, China

Jia Ye
School of Computer Science, Nanjing University, China

Abstract

We revisit the simultaneous approximation model for the correlation clustering problem introduced by Davies, Moseley, and Newman [21]. The objective is to find a clustering that minimizes given norms of the disagreement vector over all vertices.

We present an efficient algorithm that produces a clustering that is simultaneously a 63.3-approximation for all monotone symmetric norms. This significantly improves upon the previous approximation ratio of 6348 due to Davies, Moseley, and Newman [21], which works only for ℓ_p-norms.

To achieve this result, we first reduce the problem to approximating all top-k norms simultaneously, using the connection between monotone symmetric norms and top-k norms established by Chakrabarty and Swamy [11]. Then we develop a novel procedure that constructs a 12.66-approximate fractional clustering for all top-k norms. Our 63.3-approximation ratio is obtained by combining this with the 5-approximate rounding algorithm by Kalhan, Makarychev, and Zhou [26].

We then demonstrate that with a loss of ϵ in the approximation ratio, the algorithm can be adapted to run in nearly linear time and in the MPC (massively parallel computation) model with poly-logarithmic number of rounds.

By allowing a further trade-off in the approximation ratio to $(359 + \epsilon)$, the number of MPC rounds can be reduced to a constant.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Massively parallel algorithms

Keywords and phrases Correlation Clustering, All-Norms, Approximation Algorithm, Massively Parallel Algorithm

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.40

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2410.09321`

Funding *Nairen Cao*: Supported by NSF grant CCF-2008422.
Shi Li: Supported by the State Key Laboratory for Novel Software Technology, and the New Cornerstone Science Laboratory.
Jia Ye: Supported by the State Key Laboratory for Novel Software Technology, and the New Cornerstone Science Laboratory.

1 Introduction

Clustering is a classic problem in unsupervised machine learning. It aims to classify a given set of data elements based on their similarities, with the goal of maximizing the similarity between elements within the same class and minimizing the similarity between elements in different classes. Among the various graph clustering problems, correlation clustering stands out as a classic model. Initially proposed by Bansal, Blum, and Chawla [4], the model has

numerous practical applications, including automated labelling [10, 1], community detection and mining [15, 31, 30], and disambiguation task [25], among others.

The input of the standard correlation clustering problem is a complete graph over a set V of n vertices, where edges are partitioned into the set E^+ of $+$edges and the set E^- of $-$edges. The output of the problem is a clustering (or a partition) $\mathcal{C}$ of V that minimizes the number of edges in disagreement: an edge $uv \in \binom{V}{2}$ is in disagreement if $uv \in E^+$ but u and v are in different clusters in $\mathcal{C}$, or $uv \in E^-$ but u and v are in a same cluster in $\mathcal{C}$. Throughout the paper, we shall use a graph $G = (V, E)$ to denote a correlation clustering instance, with E being E^+ and $\binom{V}{2} \setminus E$ being E^-.

This problem is known to be APX-Hard [13]. There has been a long stream of $O(1)$-approximation algorithms for the problem [4, 13, 2, 14, 19, 18, 8], with the current best approximation ratio being 1.437 [8]. In the same paper, the authors presented an improved hardness of 24/23 for the problem, which also made the constant explicit.

Besides the standard setting, other objectives have been studied recently, with the goal of minimizing some norm of the *disagreement vector* of the clustering $\mathcal{C}$ over vertices. For a clustering $\mathcal{C}$ of V, the disagreement vector of $\mathcal{C}$ is defined as $\text{cost}_{\mathcal{C}} \in \mathbb{Z}^n_{\geq 0}$, where $\text{cost}_{\mathcal{C}}(u)$ for every $u \in V$ is the number of edges incident to u that are in disagreement with respect to $\mathcal{C}$. Given some norm $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ [1], the goal of the problem is to minimize $f(\text{cost}_{\mathcal{C}})$. Notice that the standard correlation clustering problem corresponds to the case where f is the ℓ_1 norm.

Puleo and Milenkovic [29] initiated the study of correlation clustering with the goal of minimizing the ℓ_p norm of the disagreement vector, where $p \in [1, \infty]$. They proved that the problem is NP-hard for the ℓ_∞-norm objective. Given a fixed $p \in [1, \infty]$, for the ℓ_p-norms objective, they gave a 48-approximation algorithm. The approximation ratio was subsequently improved by Charikar, Gupta, and Schwartz [12] to 7 for the ℓ_∞-norm, and by Kalhan, Makarychev and Zhou [26] to 5 for the ℓ_p-norm with any fixed $p \in [1, \infty]$. Very recently, Heidrich, Irmai, and Andres [24] improved the approximate ratio to 4 for the ℓ_∞-norm.

Davies, Moseley and Newman [21] introduced the concept of simultaneous approximation for all ℓ_p-norms. They developed an efficient algorithm that outputs a single clustering $\mathcal{C}$, which is simultaneously an $O(1)$-approximation for the ℓ_p norm for all $p \in [1, \infty]$. This is rather surprising, as it was not known a priori whether such a clustering $\mathcal{C}$ even exists. To achieve the goal, they first construct a fractional clustering x that is simultaneously an $O(1)$-approximation for all ℓ_p norms and then use the 5-approximate rounding algorithm of Kalhan, Makarychev, and Zhou [26] to round x into an integral clustering $\mathcal{C}$. Crucially, the algorithm of [26] guarantees a per-vertex 5-approximation, meaning that $\text{cost}_{\mathcal{C}}(u)$ is at most 5 times the fractional number of edges in disagreement incident to u, for every $u \in V$. This strong property is used to obtain the final simultaneous $O(1)$-approximation in [21].

In light of the growing networks, it is imperative to develop efficient parallel algorithms. This urgency is particularly pronounced in machine learning and data mining applications, where timely and efficient processing is essential for extracting meaningful insights from vast datasets. Many works in the literature aim to design efficient parallel algorithms [5, 16, 28, 23, 6, 17, 3, 7, 9]. The MPC model, as a theoretical abstraction of several real-world parallel models such as MapReduce [22], is a prevalent methodology employed in these works.

[1] This means f satisfies $f(\alpha x) = \alpha f(x)$ for every real $\alpha \geq 0$ and $x \in \mathbb{R}^n_{\geq 0}$, and $f(x + y) \leq f(x) + f(y)$ for every $x, y \in \mathbb{R}^n_{\geq 0}$

1.1 Our results

In this paper, we revisit and generalize the simultaneous approximation model for the correlation clustering that was introduced by [21]. Instead of considering only ℓ_p norms, we consider all *monotone symmetric norms*. We say a norm $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is monotone if for every $x, y \in \mathbb{R}^n_{\geq 0}$ with $x \leq y$, we have $f(x) \leq f(y)$. We say f is symmetric if $f(x) = f(x')$ for every $x, x' \in \mathbb{R}^n_{\geq 0}$ such that x' is a permutation of x. Such norms were considered in [11] in the context of load balancing and clustering. Our first result is that there exists simultaneous $O(1)$-approximation for all monotone symmetric norms for correlation clustering and it can be constructed in polynomial time.

▶ **Definition 1.** *Given a correlation clustering instance $G = (V, E)$ and $\alpha \geq 1$, we say a clustering $\mathcal{C}$ over V is simultaneously α-approximate, or a simultaneous α-approximation, for a family F of norms, if we have $f(\text{cost}_{\mathcal{C}}) \leq \alpha \cdot f(\text{cost}_{\text{OPT}_f})$ for every $f \in F$, where OPT_f is the optimum clustering for G under norm f.*

▶ **Theorem 2.** *Given a correlation clustering instance $G = (V, E)$, in polynomial time we can construct a simultaneous 63.3-approximate clustering $\mathcal{C}$ for the family of monotone symmetric norms.*

Next, we are concerned with the running time of the algorithm and its implementation under the MPC model. To state the result, we need a formal description of the MPC model.

The MPC model. In the MPC model, data is distributed across a set of machines, and computation proceeds in synchronous rounds. During each round, each machine first receives messages from other machines, then performs computations based on this information and its own allocated memory, and finally sends messages to other machines to be received at the start of the next round. Each machine has limited local memory, restricting the total number of messages it can receive or send in a round. The efficiency of the algorithm is measured by the number of rounds, the memory used by each machine, the total memory used by all machines, and the running time over all machines, also known as the total work.

In this paper, we consider the MPC model in the *strictly sublinear regime*: Each machine has $O(n^\delta)$ local memory, where n is the input size and $\delta > 0$ is a constant that can be made arbitrarily small. Under this model, we assume the input received by each machine has size $O(n^\delta)$.

We then describe the correlation clustering problem under the MPC model in the strictly sublinear regime. We use $n = |V|$ and $m = |E|$ to denote the number of vertices and edges respectively in the input graph $G = (V, E)$. The edges E are distributed across the machines, where each machine has $O(n^\delta)$ memory for a constant $\delta > 0$ which can be made arbitrarily small. At the end of the algorithm, each machine needs to store in its local memory the IDs of the clusters for all the vertices incident to its assigned edges.

Our main result regarding MPC algorithm is given as follows,

▶ **Theorem 3.** *Let $\epsilon \in (0, 1)$. There exists a randomized MPC algorithm in the strictly sublinear regime that, given a correlation clustering instance $G = (V, E)$, in $O(\log^3 n)$ rounds outputs a simultaneous $(63.3 + O(\epsilon))$ clustering for G for all monotone symmetric norms. This algorithm succeeds with high probability. It uses $\tilde{O}(m/\epsilon^6)$ total memory and $\tilde{O}(m/\epsilon^6)$ total work.*[2]

[2] As usual, we use $\tilde{O}(\cdot)$ to hide a poly-logarithmic factor in the input size.

In particular, the algorithm can be converted into a nearly linear time algorithm that with high probability outputs a $(63.3 + O(\epsilon))$-simultaneous approximation for all monotone symmetric norms.

Along the way, we develop an MPC rounding algorithm with a per-vertex $(5 + 55\epsilon)$ approximation guarantee, based on the sequential algorithm due to [26]. Given its potential independent interest, we state it here for future references.

▶ **Theorem 4.** *Let $\epsilon \in (0, 1)$ be a constant. Given a graph $G = (V, E)$ and a set of LP values $(x_{uv})_{u,v \in V}$ satisfying the approximate triangle inequality, that is, for any $u, v, w \in V$, we have $x_{uv} + x_{uw} + \epsilon \geq x_{vw}$. Let $y_u = \sum_{uv \in E} x_{uv} + \sum_{uv \in \binom{V}{2} \setminus E} (1 - x_{uv})$ be the LP disagreement for node u. There exists an MPC algorithm that computes a clustering $\mathcal{C}$ such that for any node u, we have*

$$\mathrm{cost}_{\mathcal{C}}(u) \leq (5 + 55\epsilon) y_u.$$

This algorithm always succeeds but terminates in $O(\log^3 n/\epsilon)$ rounds with high probability and requires $O(n^\delta)$ memory per machine. Moreover, let $K = E \cup \{uv \in \binom{V}{2} \setminus E \mid x_{uv} < 1\}$ be the set of $+$edges and $-$edges whose LP value is less than 1. The algorithm uses a total memory of $O(|K| \log n)$ and a total work of $O(|K| \log^3 n/\epsilon)$.

The $O(\log^3 n)$ round in the above theorem might not be desirable for many applications. Our next result shows that we can reduce the number of rounds to $O(1)$, albeit with a worse $O(1)$ approximation ratio:

▶ **Theorem 5.** *Let $\epsilon \in (0, 1)$ be a constant. There exists a randomized MPC algorithm in the strictly sublinear regime that, given a correlation clustering instance $G = (V, E)$, in $\mathbf{O(1)}$ rounds outputs a clustering that is simultaneously a $(359+\epsilon)$-approximation, for all monotone symmetric norms. This algorithm succeeds with high probability, and uses a total memory of $\tilde{O}(m/\epsilon^2)$ and a total work of $\tilde{O}(m/\epsilon^2)$.*

Overall, relative to [21], our algorithms demonstrate the following improvements.

1. We generalize the family of norms for the simultaneous approximation from ℓ_p norms to all monotone symmetric norms.
2. We obtain a simpler construction, which leads to a much smaller approximation ratio. Using a result from [11], to simultaneously approximate all monotone symmetric norms, it suffices to approximate all top-k norms: the top-k norm of a non-negative vector is the sum of its largest k coordinates. Though being more general mathematically, the top-k norms are more convenient to deal with compared to ℓ_p norms.
3. We can make our algorithm run in nearly linear time. This is the first nearly-linear time simultaneous $O(1)$-approximation algorithm for the problem, even when we restrict to ℓ_p norms. In contrast, the algorithm of [21] runs in nearly linear time only when the graph G has $O(1)$ maximum degree.
4. We can make our algorithm run in the MPC model with $O(1)$ rounds. Our work is the first to consider the problem in the MPC model.

1.2 Overview of Techniques

We then discuss our techniques for each of our main results.

Polynomial Time Construction of Simultaneous $O(1)$-Approximation for All Symmetric Norms. By [11], we can reduce the problem of approximating all monotone symmetric norms to approximating all top-k norms. We then construct a fractional solution x, which is a metric over V with range $[0,1]$, such that the fractional disagreement vector for x has top-k norm at most $12.66 \cdot \mathrm{opt}_k$ for any $k \in [n]$, where opt_k is the cost of the optimum clustering under the top-k norm. Then, we can use the 5-approximate rounding algorithm of KMZ [26], to obtain a simultaneous 63.3-approximation for all top-k norms. The KMZ rounding algorithm has two crucial properties that we need: it does not depend on k and it achieves a per-vertex guarantee.

We elaborate more on how to construct the metric $x : \binom{V}{2} \to [0,1]$. A natural idea to assign the LP values, that was used by [21], is to set x_{uv} based on the intersection of the neighborhood between u and v. Intuitively, the more common neighbors two nodes share, the closer they should be. A straightforward approach to implementing this idea is to set $x_{uv} = 1 - \frac{|N(u) \cap N(v)|}{\max(d(u),d(v))}$, where $N(u)$ denotes the neighboring nodes of u in G and $d(u) = |N(u)|$ denotes the degree of u; it is convenient to assume $u \in N(u)$. This approach works for the top-1 norm (i.e., the ℓ_∞ norm) as discussed in [20], but fails for the top-n norm (i.e., the ℓ_1 norm). Consider a star graph, where the optimal clustering under the top-n norm has a cost of $n-2$. This approach will assign $x_{uv} = 1 - \frac{1}{2} = 1/2$ for all $-$edges, leading to an LP cost of $\Theta(n^2)$ and a gap of $\Omega(n)$. [21] addressed the issue by rounding up LP values to 1 for $-$edge, if for a given node, its total $-$edges LP disagreement is larger than the number of its $+$edges. After the transformation, the triangle inequalities are only satisfied approximately, but this can be handled with $O(1)$ loss in the approximation ratio.

We address this issue using a different approach, that is inspired by the pre-clustering technique in [17]. We first preprocess the graph G by removing edges $uv \in E$ for which $|N(u) \cap N(v)|$ is small compared to $\max\{d(u), d(v)\}$. Let the resulting graph be H. We then set our LP values as $x_{uv} = 1 - \frac{|N_H(u) \cap N_H(v)|}{\max\{d(u),d(v)\}}$ if $u \neq v$, where $N_H(u)$ is the set of neighbors of u in H. We show that this solution is a 12.66-approximation for all top-k norms simultaneously. When compared to [21], in addition to the improved approximation ratio, we obtain a considerably simpler analysis.

Implementation of Algorithm in Nearly-Linear Time and in MPC Model. We then proceed to discuss our techniques to improve the running time of the algorithm to nearly-linear. The algorithm contains two parts: the construction of the fractional solution x and the rounding procedure. We discuss the two procedures separately.

Constructing x in nearly linear time poses several challenges. First, the construction of the subgraph H requires us to identify edges $uv \in E$ with small $|N(u) \cap N(v)|$. Second, we can not explicitly assign x values to all $-$edges. Finally, to compute x_{uv}, we need to compute $|N_H(u) \cap N_H(v)|$.

The first and third challenges can be addressed through sampling, with an $O(\log n)$ factor loss in the running time. To avoid considering too many $-$edges, we only consider $-$edges with length at most $1-\epsilon$. Consequently, we only need to consider $-$edges whose other endpoints share at least an ϵ fraction of neighbors with u. Given that each neighbor of u in H has degree similar to u, we demonstrate that there will be at most $O(d(u)/\epsilon)$ $-$edges to consider for each node u. Overall, there will be $\tilde{O}(m \cdot \mathrm{poly}(1/\epsilon))$ $-$edges for which we need to explicitly assign x values. Moreover, the nearly-linear time algorithm for the construction of x can be naturally implemented in the MPC model, with $O(1)$ number of rounds.

Then we proceed to the rounding algorithm for x. We are explicitly given the x values for $+$edges, and for nearly-linear number of $-$edges. For other $-$edges, their x values are 1.

The KMZ algorithm works as follows: in each round, the algorithm selects a node u as the cluster center and then includes a ball with some radius, meaning the algorithm includes all nodes v such that $x_{uv} \leq$ radius into the cluster, removes the clustered nodes, and repeats the process on the remaining nodes. The rounding algorithm can be easily implemented in nearly-linear time using a priority queue structure. This leads to a nearly-linear time simultaneous $O(1)$-approximation for correlation clustering for all monotone symmetric norms.

The challenge to implement the algorithm in MPC model is the sequential nature of the algorithm. [26] observes that in each round, if we select the nodes that maximize $L(u) = \sum_{x_{uv} \leq r}(r - x_{uv})$ as cluster center, we can effectively bound each node's algorithmic cost, where r is the final ratio. However, choosing a node that maximizes some target inherently makes the process sequential. Our key observation is that, instead of selecting the node that maximizes $L(u)$, we can allow some approximation. This strategy still permits achieving a reasonable approximate ratio with an additional $1+\epsilon$ overhead while allowing the selection of multiple nodes as cluster centers, thereby parallelizing the rounding process. In each round, there might be several candidate cluster centers with conflicts. To resolve these conflicts, we employ the classical Luby's algorithm [27, 16] to find a maximal independent set, ensuring that none of the cluster centers have conflicts.

Organization. We give some preliminary remarks in Section 2. In Section 3, we describe our simultaneous $O(1)$-approximation algorithm for correlation clustering for all top-k norms. The reduction from any monotone symmetric norm to top-k norms is deferred to Appendix A. Combining the results leads to a simultaneous $O(1)$-approximation algorithm for all monotone symmetric norms. Then in Appendix B and C, we show how we can run the algorithm in the MPC model with nearly linear work. In particular, Appendix B and C discuss how to solve the LP and round the LP solution in the MPC model, respectively. The constant round MPC algorithm is described in Appendix D. Theorem 2, 3, 4 and 5 are proved in Section 3, Appendix C, C and D respectively.

2 Preliminaries

The input to correlation clustering is a complete graph whose edges are partitioned into +edges and −edges. We shall use the graph $G = (V, E)$ of +edges to denote an instance. Let $n = |V|$ and $m = |E|$. For simplicity, we assume E contains all the n self-loops $uu, u \in V$. So, E is the set of +edges, and $\binom{V}{2} \setminus E$ is the set of −edges. The graph G is fixed in most part of the paper.

For any graph $H = (V_H, E_H)$, and any vertex $v \in V_H$, let $N_H(u) = \{v \in V_H \mid uv \in E_H\}$. For any vertex $u \in V_H$ and any subset $S \subseteq V_H$, we define $d_H(u, S) = \sum_{v \in S} \mathbb{1}(uv \in E_H)$ as the number of edges between u and S. We simply use $d_H(u)$ for $d_H(u, V_H)$. When the graph H is the input graph G, we omit the subscript. So we use $N(u)$ for $N_G(u)$ and $d(u)$ for $d_G(u)$. Notice that $u \in N(u)$ and $d(u) = |N(u)| \geq 1$ for every $u \in U$. For the input graph $G = (V, E)$ and any two vertex $u, v \in V$, we define $M_{uv} = \max\{d(u), d(v)\}$ as the maximum degree of u and v for simplicity, as this notion will be frequently used. For any two sets X and Y, we denote their symmetric difference by $X \Delta Y$. Algorithms are parameterized by constants $\beta (0 < \beta < 1), \lambda (0 < \lambda < 1)$ that will be determined later.

A norm on n-dimensional non-negative vectors is a function $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ satisfying $f(\alpha x) = \alpha f(x)$ for every real $\alpha \geq 0$ and $x \in \mathbb{R}^n_{\geq 0}$, and $f(x + y) \leq f(x) + f(y)$ for every $x, y \in \mathbb{R}^n_{\geq 0}$. We say a norm $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is monotone if for every $x, y \in \mathbb{R}^n_{\geq 0}$ with $x \leq y$,

we have $f(x) \le f(y)$. We say f is symmetric if $f(x) = f(x')$ for every $x, x' \in \mathbb{R}^n_{\ge 0}$ such that x' is a permutation of x. We say f is the top-k norm for an integer $k \in [n]$ if $f(x)$ is equal to the sum of the k largest coordinates of x. Chakrabarty and Swamy [11] showed that any monotone and symmetric norm can be written as the maximum of many ordered norms. This leads to the following lemma which reduces the monotone-symmetric norms to top-k norms. For completeness, we defer its proof to Appendix A.

▶ **Lemma 6.** *For any integer $k \in [n]$, if an algorithm returns a single clustering $\mathcal{C}_{ALG}$ that is simultaneously a ρ-approximation for all top-k norm objectives, then $\mathcal{C}_{ALG}$ is a ρ-approximation for any monotone and symmetric norm $f : \mathbb{R}^n_{\ge 0} \to \mathbb{R}_+$.*

For a fixed clustering $\mathcal{C}$, we already defined the disagreement vector of $\mathcal{C}$ as $\text{cost}_{\mathcal{C}} \in \mathbb{Z}^n_{\ge 0}$, with $\text{cost}_{\mathcal{C}}(u)$ for every $u \in V$ being the number of edges incident to u that are in disagreement w.r.t $\mathcal{C}$. Given an integer k, and a clustering $\mathcal{C}$, we denote the top-k value by $\text{cost}^k_{\mathcal{C}} = \max_{T \subseteq V, |T|=k} \sum_{u \in T} \text{cost}_{\mathcal{C}}(u)$. Similarly, for any fractional vector $(x_{uv})_{u,v \in V}$, we denote $\text{cost}_x(u) = \sum_{uv \in E} x_{uv} + \sum_{uv \in \binom{V}{2} \setminus E} (1 - x_{uv})$ as the disagreement for u with respect to x. The top-k value of x is defined as $\text{cost}^k_x = \max_{T \subseteq V, |T|=k} \sum_{u \in T} \text{cost}_x(u)$.

We will use the following theorem from [26]:

▶ **Theorem 7.** *Let $G = (V, E)$ be a correlation clustering instance, and $x \in [0,1]^{\binom{V}{2}}$ be a metric over V with range $[0, 1]$. There is a polynomial time algorithm that, given G and x, outputs a clustering $\mathcal{C}$ of V such that $\text{cost}_{\mathcal{C}}(u) \le 5 \cdot \text{cost}_x(u)$ for every $u \in V$.*

We will use the following well-known concentration inequalities.

▶ **Theorem 8** (Chernoff Bound). *Let $X_1, X_2, ..., X_k$ be independent random variables taking values in $\{0, 1\}$. Let $X = \sum_i X_i$ be the sum of these k random variables. Then the following inequalities hold:*
(a) *For any $\epsilon \in (0, 1)$, if $E[X] \le U$, then $\Pr[X \ge (1 + \epsilon)U] \le \exp(-\epsilon^2 U/3)$.*
(b) *For any $\epsilon \in (0, 1)$, if $E[X] \ge U$, then $\Pr[X \le (1 - \epsilon)U] \le \exp(-\epsilon^2 U/2)$.*

3 Simultaneous $O(1)$-Approximation for Top-k Norms

In this section, we describe our simultaneous 63.3-approximation for correlation clustering for all top-k norms. The algorithm described in this section runs in polynomial time. It first constructs an LP solution to the top-k linear program using a combinatorial procedure. Crucially, the construction does not depend on the value of k. We show that the solution has cost 12.66 times the optimum cost under the top-k norm, for any integer $k \ge 1$. Then we use the rounding algorithm of [26] to round the LP solution to an integral one. As it gives a vertex-by-vertex 5-approximation guarantee, this leads to a 63.3-approximation for the top-k norm for any k.

The LP for minimizing the top-k norm of the clustering is given in LP (1).

$$\begin{aligned}
\min \quad & \text{cost}^k_x && && \text{(1a)}\\
\text{s.t.} \quad & x_{uv} + x_{uw} \ge x_{vw}, && \forall u, v, w \in V, && \text{(1b)}\\
& x_{uv} \in [0, 1], && \forall u, v \in V, && \text{(1c)}\\
& x_{uu} = 0, && \forall u \in V. && \text{(1d)}
\end{aligned}$$

In the correspondent integer program, x_{uv} for every $u, v \in V$ indicates if uv is separated or not. We view uv as an unordered pair and thus x_{uv} and x_{vu} are the same variable. So

Algorithm 1 Construction of norm-oblivious solution x to metric LP.

Input: Graph $G = (V, E)$
Output: $(x_{uv})_{u,v \in V}$
1: **function** AllNormCC($G = (V, E)$)
2: let $E_H = \{uv \in E : |N(u) \Delta N(v)| \leq \beta \cdot M_{uv}\}$ and $H = (V, E_H)$
3: let $x_{uv} \leftarrow 1 - \frac{|N_H(u) \cap N_H(v)|}{\max(d(u), d(v))}$ for every $uv \in \binom{V}{2}$ and $x_{uu} = 0$ for every $u \in V$
4: **end function**

$(x_{uv})_{u,v \in V}$ is a metric with distances in $\{0, 1\}$, which is relaxed to $[0, 1]$ in the linear program. This is captured by constraints (1b), (1c) and (1d). Notice that $\text{cost}_x(u)$ for any $u \in V$ is a linear function of the x variables. The top-k norm of the fractional clustering is defined by $\text{cost}_x^k = \max_{S \subseteq V : |S| = k} \sum_{u \in S} \text{cost}_x(u)$. This could be captured by introducing a variable z and constraints $z \geq \sum_{u \in S} \text{cost}_x(u)$ for any $S \subseteq V$ of size k, and setting z to be the objective to minimize. For simplicity, we use the form as described. Despite having an exponential number of constraints, the LP can be solved efficiently as there is a simple separation oracle. Moreover, we use a combinatorial algorithm to construct a solution x, and thus the algorithm does not solve the LP.

3.1 Algorithm

The algorithm for constructing the LP solution x is given in Algorithm 1. It depends on the parameter $\beta \in (0, 1)$, whose value will be specified later. During the process, we construct a subgraph H by removing any edge $uv \in E$ where u and v have significantly different neighbors. We then set x_{uv} as $1 - \frac{|N_H(u) \cap N_H(v)|}{M_{uv}}$ if $u \neq v$ and $x_{uv} = 0$ otherwise. Recall that $M_{uv} = \max\{d(u), d(v)\}$ is the maximum degree for any nodes u and v in graph G. Intuitively, we treat +edges as indicators of whether two nodes belong to the same cluster. The first step is to remove edges that should not be in the same cluster. The second step ensures that the more common neighbors two nodes have, the closer their distance should be.

In the remaining part of this section, we will show

▶ **Lemma 9.** *Let k be any integer in $[n]$. Algorithm 1 outputs a feasible solution $(x_{uv})_{u,v \in V}$ for (1) such that for any k, we have*

$$\text{cost}_x^k \leq 12.66 \cdot \text{opt}^k,$$

where opt^k is the cost of the optimum solution under the top-k norm.

Proof of Theorem 2. Once we obtain Lemma 9, [26] provides a rounding algorithm for any feasible solution. Assume the final clustering is $\mathcal{C}_{\text{KMZ}}$. [26] ensures that for any node u, we have $\text{cost}_{\mathcal{C}_{\text{KMZ}}}(u) \leq 5\text{cost}_x(u)$, meaning the disagreement for $\mathcal{C}_{\text{KMZ}}$ is at most 5 times the disagreement in the LP solution. We can apply the KMZ rounding algorithm to x_{uv} as output by Lemma 9. For any integer $k \in [n]$, let U' be the set of k vertices with the largest disagreement values with respect to $\mathcal{C}_{\text{KMZ}}$. Then we have:

$$\text{cost}_{\mathcal{C}_{\text{KMZ}}}^k \leq 5 \cdot \sum_{u \in U'} \text{cost}_x(u) \leq 5 \cdot \text{cost}_x^k \leq 5 \cdot 12.66 \cdot \text{opt}^k = 63.3 \cdot \text{opt}^k.$$

By Lemma 6, we know that $\mathcal{C}_{\text{KMZ}}$ is a simultaneous 63.3-approximate clustering for all monotone symmetric norms. ◀

We will first show that our x is feasible in Section 3.2, then we will bound the approximate ratio in Section 3.3.

3.2 The validity of x to LP (1)

To show that x is a valid solution to LP (1), it suffices to prove that it is a metric over V with range $[0, 1]$. Moreover, (1c) and (1d) hold trivially. Therefore, it remains to show that the triangle inequality (i.e., constraint (1b)) is satisfied:

▶ **Lemma 10.** *For any $u, v, w \in V$, we have $x_{uv} + x_{uw} \geq x_{vw}$.*

Proof. We can assume u, v, w are distinct since otherwise the inequality holds trivially. We assume that $d(v) \geq d(w)$ wlog.

$$\begin{aligned}
& x_{uv} + x_{uw} - x_{vw} \\
= \; & \Big(1 - \frac{|N_H(u) \cap N_H(v)|}{M_{uv}}\Big) + \Big(1 - \frac{|N_H(u) \cap N_H(w)|}{M_{uw}}\Big) - \Big(1 - \frac{|N_H(v) \cap N_H(w)|}{d(v)}\Big) \\
= \; & 1 + \frac{|N_H(v) \cap N_H(w)|}{d(v)} - \frac{|N_H(u) \cap N_H(v)|}{M_{uv}} - \frac{|N_H(u) \cap N_H(w)|}{M_{uw}} \\
\geq \; & 1 + \frac{|N_H(v) \cap N_H(w)|}{M_{uv}} - \frac{|N_H(u) \cap N_H(v)|}{M_{uv}} - \frac{|N_H(u) \cap N_H(w)|}{M_{uw}} \\
\geq \; & 1 - \frac{|N_H(u) \setminus N_H(w)|}{M_{uv}} - \frac{|N_H(u) \cap N_H(w)|}{M_{uw}} \\
\geq \; & 1 - \frac{|N_H(u) \setminus N_H(w)| + |N_H(u) \cap N_H(w)|}{d(u)} \quad = \quad 1 - \frac{|N_H(u)|}{d(u)} \quad \geq \quad 0.
\end{aligned}$$

The first inequality used that $d(v) \leq M_{uv}$, the second one follows from $|N_H(u) \cap N_H(v)| - |N_H(v) \cap N_H(w)| \leq |(N_H(u) \cap N_H(v)) \setminus (N_H(v) \cap N_H(w))| = |(N_H(v) \cap (N_H(u) \setminus N_H(w))| \leq |N_H(u) \setminus N_H(w)|$, and the third one used $d(u) \leq M_{uv}$ and $d(u) \leq M_{uw}$. ◀

3.3 Bounding the Top-k Norm Cost of x

In this section, we compare the top-k norm cost of x to opt^k.

Notations. We fix the integer $k \in [n]$. Let $\mathcal{C}$ be the clustering that minimizes the top-k norm of disagreement vector, but our analysis works for any clustering. For every $v \in V$, let $C(v)$ be the cluster in $\mathcal{C}$ that contains v.

For every $u \in V$, let $\mathrm{cost}^+_{\mathcal{C}}(u), \mathrm{cost}^-_{\mathcal{C}}(u)$ and $\mathrm{cost}_{\mathcal{C}}(u)$ respectively be the number of $+$edges, $-$edges and edges incident to u that are in disagreement in the clustering $\mathcal{C}$. Recall $\mathrm{cost}^k_{\mathcal{C}} = \max_{S \subseteq V : |S| = k} \sum_{u \in S} \mathrm{cost}_{\mathcal{C}}(u)$ is the top-k norm cost of the clustering $\mathcal{C}$, thus we have $\mathrm{opt}^k = \mathrm{cost}^k_{\mathcal{C}}$.

Let U be the set of k vertices u with the largest $\mathrm{cost}_x(u)$ values. So, $\mathrm{cost}^k_x = \sum_{u \in U} \mathrm{cost}_x(u)$. In order to provide a clear demonstration, we divide all of the edges in G into five parts. First, we separate out the parts that are easily constrained by $\mathrm{cost}^k_{\mathcal{C}}$. Let φ^+_1 be the set of $+$edges that are cut in $\mathcal{C}$, and φ^-_1 be the set of $-$edges that are not cut in $\mathcal{C}$. For the remaining $+$edges in E that are not cut in $\mathcal{C}$, it is necessary to utilize the properties of $+$edges in E_H. To this end, let φ^+_2 be the set of $+$edges in $E \setminus E_H$ that are not cut in $\mathcal{C}$, and φ^+_3 be the set of $+$edges in E_H that are not cut in $\mathcal{C}$. Finally, we define φ^-_2 as

the set of $-$edges that are cut in $\mathcal{C}$. Formally, we set

$$\varphi_1^+ := \{uv \mid uv \in E, C(u) \neq C(v)\}, \quad \varphi_2^+ := \{uv \mid uv \in E \setminus E_H, C(u) = C(v)\},$$
$$\varphi_3^+ := \{uv \mid uv \in E_H, C(u) = C(v)\}, \quad \varphi_1^- := \{uv \mid uv \in \binom{V}{2} \setminus E, C(u) = C(v)\},$$
$$\text{and} \quad \varphi_2^- := \{uv \mid uv \in \binom{V}{2} \setminus E, C(u) \neq C(v)\}.$$

For every $(i,j) \in \{(1,+),(2,+),(3,+),(1,-),(2,-)\}$ and $u \in V$, we let $\varphi_i^j(u)$ be the set of pairs in φ_i^j incident to u. Notice that φ_3^+ contains all the self-loops. We use $\phi_i^j(u) = \{v : uv \in \varphi_i^j(u)\}$ to denote the end-vertices of the edges in $\varphi_i^j(u)$ other than u; so $|\phi_i^j(u)| = |\varphi_i^j(u)|$. We let $y_i^j(u)$ denote the cost of edges in $\varphi_i^j(u)$ in the solution x. For every $(i,j) \in \{(1,+),(2,+),(3,+),(1,-),(2,-)\}$, we define $f_i^j = \sum_{u \in U} y_i^j(u)$. Therefore, the top-$k$ norm cost of x is $f_1^+ + f_2^+ + f_3^+ + f_1^- + f_2^-$.

With the notations defined, we can proceed to the analysis. Prior to this, several propositions are presented, which will prove useful in the following analysis. We start with the property of edges in $E \setminus E_H$.

▶ **Lemma 11.** *For every $uv \in \varphi_2^+$, we have $\text{cost}_{\mathcal{C}}(u) + \text{cost}_{\mathcal{C}}(v) \geq \beta \cdot M_{uv}$.*

Proof. Since $C(u) = C(v)$, there are at least $|N(u)\Delta N(v)|$ disagreements incident to u and v. For every $uv \in \varphi_2^+$, we have $\text{cost}_{\mathcal{C}}(u) + \text{cost}_{\mathcal{C}}(v) \geq |N(u)\Delta N(v)| \geq \beta \cdot M_{uv}$. ◀

Then, we show that edges in E_H have similar degrees.

▶ **Lemma 12.** *For every $uv \in E_H$, we have $(1-\beta) \cdot d(v) \leq d(u) \leq \frac{1}{1-\beta} \cdot d(v)$.*

Proof. For every $uv \in E_H$, we have $|N(u)\Delta N(v)| \leq \beta \cdot M_{uv}$. Therefore,

$$\begin{aligned}\min\{d(u), d(v)\} &\geq |N(u) \cap N(v)| = |N(u) \cup N(v)| - |N(u)\Delta N(v)| \\ &\geq (1-\beta) \cdot \max\{d(u), d(v)\},\end{aligned}$$

which gives us $(1-\beta) \cdot d(v) \leq d(u) \leq \frac{1}{1-\beta} \cdot d(v)$ since $\beta \in (0,1)$. ◀

As we are about to analyze the top-k norm objective, we can bound the cost in the solution x using coefficients of $\text{cost}_{\mathcal{C}}$. This key observation can be formally demonstrated as follows:

▷ Claim 13. Let $c \in \mathbb{R}^n_{\geq 0}$ be a vector and $\alpha > 0$ satisfying that $|c|_\infty \leq \alpha$ and $|c|_1 \leq \alpha k$. Then we have

$$\sum_{r \in V} c(r) \cdot \text{cost}_{\mathcal{C}}(r) \leq \alpha \cdot \text{cost}^k_{\mathcal{C}}.$$

Proof. We have $|c/\alpha|_1 \leq k$ and $|c/\alpha|_\infty \leq 1$. It is well known that c/α is a convex combination of $\{0,1\}$-vectors, each of which has at most k 1's. For each $\{0,1\}$-vector d in the combination, we have $\sum_{r \in V} d(r)\text{cost}_{\mathcal{C}}(r) \leq \text{cost}^k_{\mathcal{C}}$. Taking the convex combination of these inequalities gives $\sum_{r \in V} \frac{c(r)}{\alpha} \cdot \text{cost}_{\mathcal{C}}(r) \leq \text{cost}^k_{\mathcal{C}}$, which is equivalent to the inequality in the claim. ◁

From the definitions of f_1^+ and f_1^-, we can see that it can be constrained by $\text{cost}^k_{\mathcal{C}}$.

▷ Claim 14. $f_1^+ + f_1^- \leq \text{cost}^k_{\mathcal{C}}$.

Proof.

$$f_1^+ + f_1^- = \sum_{u\in U}(y_1^+(u) + y_1^-(u)) = \sum_{u\in U, uv\in\varphi_1^+\cup\varphi_1^-} x_{uv}$$
$$\leq \sum_{u\in U}(|\varphi_1^+(u)| + |\varphi_1^-(u)|) \leq \sum_{u\in U}\mathrm{cost}_{\mathcal{C}}(u) \leq \mathrm{cost}_{\mathcal{C}}^k. \qquad \lhd$$

To bound the cost of the remaining +edges, we separately analyze the cost coefficients of vertices in f_2^+ and f_3^+.

▶ **Lemma 15.** *There exists a vector $c_2^+ \in \mathbb{R}^n_{\geq 0}$ with the following properties:*

(a) $f_2^+ \leq \sum_{r\in V} c_2^+(r)\cdot \mathrm{cost}_{\mathcal{C}}(r)$.

(b) $c_2^+(r) \leq \frac{2}{\beta}\cdot\frac{|\varphi_2^+(r)|}{d(r)}$, *for every* $r\in V$.

(c) $|c_2^+|_1 \leq \frac{2}{\beta}\sum_{u\in U}\frac{|\varphi_2^+(u)|}{d(u)}$.

Proof. We bound f_2^+ as follows:

$$f_2^+ \leq \sum_{u\in U, uv\in\varphi_2^+} 1 \leq \sum_{u\in U, uv\in\varphi_2^+}\frac{\mathrm{cost}_{\mathcal{C}}(u)+\mathrm{cost}_{\mathcal{C}}(v)}{\beta\cdot M_{uv}} =: \sum_{r\in V} c_2^+(r)\cdot\mathrm{cost}_{\mathcal{C}}(r).$$

The second inequality used Lemma 11. Therefore, Lemma 15(a) holds.

To show Lemma 15(b), we bound the coefficients for $\mathrm{cost}_{\mathcal{C}}(u)$ and $\mathrm{cost}_{\mathcal{C}}(v)$ respectively. If $u\in U$, the coefficient for $\mathrm{cost}_{\mathcal{C}}(u)$ is $\sum_{uv\in\varphi_2^+}\frac{1}{\beta\cdot M_{uv}} \leq \frac{|\varphi_2^+(u)|}{\beta\cdot d(u)}$; if $u\notin U$, the coefficient is 0. The coefficient for $\mathrm{cost}_{\mathcal{C}}(v)$ is $\sum_{u\in U\cap\phi_2^+(v)}\frac{1}{\beta\cdot M_{uv}} \leq \frac{|\varphi_2^+(v)|}{\beta\cdot d(v)}$. Therefore, $c_2^+(r) \leq \frac{2\cdot|\varphi_2^+(r)|}{\beta\cdot d(r)}$.

To bound $|c_2^+|_1$, we can replace $\mathrm{cost}_{\mathcal{C}}(u)$ and $\mathrm{cost}_{\mathcal{C}}(v)$ with 1.

Then $|c_2^+|_1 = \sum_{u\in U, uv\in\varphi_2^+}\frac{2}{\beta\cdot M_{uv}} \leq \frac{2}{\beta}\sum_{u\in U}\frac{|\varphi_2^+(u)|}{d(u)}$. This proves Lemma 15(c). ◀

Following the analysis of the cost coefficients c_2^+ of f_2^+, we analyze the cost coefficients of f_3^+ in edge set E_H.

▶ **Lemma 16.** *There exists a vector $c_3^+ \in \mathbb{R}^n_{\geq 0}$ with the following properties:*

(a) $f_3^+ \leq \sum_{r\in V} c_3^+(r)\cdot\mathrm{cost}_{\mathcal{C}}(r)$.

(b) $c_3^+(r) \leq 2\left(\frac{|\varphi_2^+(r)|\cdot|\varphi_3^+(r)|}{\beta\cdot d^2(r)} + \frac{|\varphi_2^+(r)|}{\beta\cdot d(r)} + \frac{|\varphi_3^+(r)|}{d(r)}\right)$, *for every* $r\in V$.

(c) $|c_3^+|_1 \leq 2\sum_{u\in U}\left(\frac{|\varphi_2^+(u)|\cdot|\varphi_3^+(u)|}{\beta\cdot d^2(u)} + \frac{|\varphi_3^+(u)|}{\beta\cdot d(u)} + \frac{|\varphi_3^+(u)|}{d(u)}\right)$.

Proof. Fix some $uv\in\varphi_3^+\subseteq E_H$ with $u\neq v$. We let $\tilde{u} = \arg\max_{w\in\{u,v\}} d(w)$, and $\tilde{v}$ be the other vertex in $\{u,v\}$. Notice that $d(\tilde{u}) = M_{uv}$. Then we have

$$x_{uv} = 1 - \frac{|N_H(u)\cap N_H(v)|}{d(\tilde{u})} \tag{2}$$
$$= \frac{d(\tilde{u}) - |N_H(u)\cap N_H(v)|}{d(\tilde{u})} = \frac{|\varphi_1^+(\tilde{u})| + |\varphi_2^+(\tilde{u})| + |\varphi_3^+(\tilde{u})| - |N_H(\tilde{u})\cap N_H(\tilde{v})|}{d(\tilde{u})}$$
$$\leq \frac{\mathrm{cost}_{\mathcal{C}}^+(\tilde{u}) + |\varphi_2^+(\tilde{u})| + |\varphi_2^+(\tilde{v})| + \mathrm{cost}_{\mathcal{C}}^-(\tilde{v})}{d(\tilde{u})} \leq \frac{|\varphi_2^+(u)| + |\varphi_2^+(v)| + \mathrm{cost}_{\mathcal{C}}(u) + \mathrm{cost}_{\mathcal{C}}(v)}{M_{uv}}. \tag{3}$$

We prove the first inequality in sequence (3). Notice $N_H(\tilde{u}) \supseteq \phi_3^+(\tilde{u})$. Therefore,

$$|\varphi_3^+(\tilde{u})| - |N_H(\tilde{u})\cap N_H(\tilde{v})| \leq |\phi_3^+(\tilde{u})| - |\phi_3^+(\tilde{u})\cap N_H(\tilde{v})|$$
$$= |\phi_3^+(\tilde{u})\setminus N_H(\tilde{v})| \leq |\phi_2^+(\tilde{v})\cup\phi_1^-(\tilde{v})|$$
$$= |\varphi_2^+(\tilde{v})\cup\varphi_1^-(\tilde{v})| \leq |\varphi_2^+(\tilde{v})| + \mathrm{cost}_{\mathcal{C}}^-(\tilde{v}).$$

Above, we used that $\phi_3^+(\tilde{u}) \setminus N_H(\tilde{v}) \subseteq C(\tilde{u}) \setminus N_H(\tilde{v}) = C(\tilde{v}) \setminus N_H(\tilde{v}) \subseteq \phi_2^+(\tilde{v}) \cup \phi_1^-(\tilde{v})$. Also (3) holds trivially when $u = v$; this holds for every $uv \in \varphi_3^+$.

Therefore,

$$\begin{aligned}
f_3^+ = \sum_{u\in U} y_3^+(u) &\le \sum_{u\in U, uv\in\varphi_3^+} \frac{|\varphi_2^+(u)| + |\varphi_2^+(v)| + \mathrm{cost}_{\mathcal{C}}(u) + \mathrm{cost}_{\mathcal{C}}(v)}{M_{uv}} \\
&\le \sum_{u\in U, uv\in\varphi_3^+, uw\in\varphi_2^+} \frac{1}{M_{uv}} + \sum_{u\in U, uv\in\varphi_3^+, vw\in\varphi_2^+} \frac{1}{M_{uv}} + \sum_{u\in U, uv\in\varphi_3^+} \frac{\mathrm{cost}_{\mathcal{C}}(u) + \mathrm{cost}_{\mathcal{C}}(v)}{M_{uv}} \\
&\le \sum_{u\in U, uv\in\varphi_3^+, uw\in\varphi_2^+} \frac{\mathrm{cost}_{\mathcal{C}}(u) + \mathrm{cost}_{\mathcal{C}}(w)}{\beta \cdot M_{uv}M_{uw}} + \sum_{u\in U, uv\in\varphi_3^+, vw\in\varphi_2^+} \frac{\mathrm{cost}_{\mathcal{C}}(v) + \mathrm{cost}_{\mathcal{C}}(w)}{\beta \cdot M_{uv}M_{vw}} \\
&\quad + \sum_{u\in U, uv\in\varphi_3^+} \frac{\mathrm{cost}_{\mathcal{C}}(u) + \mathrm{cost}_{\mathcal{C}}(v)}{M_{uv}} =: \sum_{r\in V} c_3^+(r) \cdot \mathrm{cost}_{\mathcal{C}}(r).
\end{aligned}$$

Again, we used Lemma 11 twice to prove the last inequality. Therefore, Lemma 16(a) holds.

To prove Lemma 16(b), we consider the coefficients for $\mathrm{cost}_{\mathcal{C}}(u), \mathrm{cost}_{\mathcal{C}}(v)$ and $\mathrm{cost}_{\mathcal{C}}(w)$ respectively. For a $u \in U$, the coefficient for $\mathrm{cost}_{\mathcal{C}}(u)$ is

$$\sum_{uv\in\varphi_3^+, uw\in\varphi_2^+} \frac{1}{\beta \cdot M_{uv}M_{uw}} + \sum_{uv\in\varphi_3^+} \frac{1}{M_{uv}} \le \frac{|\varphi_3^+(u)| \cdot |\varphi_2^+(u)|}{\beta d^2(u)} + \frac{|\varphi_3^+(u)|}{d(u)}.$$

The coefficient for $\mathrm{cost}_{\mathcal{C}}(v)$ is

$$\sum_{u\in U\cap\phi_3^+(v), vw\in\varphi_2^+} \frac{1}{\beta \cdot M_{uv}M_{vw}} + \sum_{u\in U\cap\phi_3^+(v)} \frac{1}{M_{uv}} \le \frac{|\varphi_3^+(v)| \cdot |\varphi_2^+(v)|}{\beta d^2(v)} + \frac{|\varphi_3^+(v)|}{d(v)}.$$

The coefficient for $\mathrm{cost}_{\mathcal{C}}(w)$ is at most

$$\begin{aligned}
&\sum_{u\in U\cap\phi_2^+(w), uv\in\varphi_3^+} \frac{1}{\beta \cdot M_{uv}M_{uw}} + \sum_{vw\in\varphi_2^+, u\in U\cap\phi_3^+(v)} \frac{1}{\beta \cdot M_{uv}M_{vw}} \\
\le &\sum_{u\in U\cap\phi_2^+(w)} \frac{d(u)}{\beta \cdot d(u)M_{uw}} + \sum_{vw\in\varphi_2^+} \frac{d(v)}{\beta \cdot d(v)M_{vw}} \\
= &\sum_{u\in U\cap\phi_2^+(w)} \frac{1}{\beta \cdot M_{uw}} + \sum_{vw\in\varphi_2^+} \frac{1}{\beta \cdot M_{vw}} \le \frac{|\varphi_2^+(w)|}{\beta \cdot d(w)} + \frac{|\varphi_2^+(w)|}{\beta \cdot d(w)} = \frac{2 \cdot |\varphi_2^+(w)|}{\beta \cdot d(w)}.
\end{aligned}$$

Therefore, for every $r \in V$, we have $c_3^+(r) \le 2\left(\frac{|\varphi_2^+(r)|\cdot|\varphi_3^+(r)|}{\beta\cdot d^2(r)} + \frac{|\varphi_2^+(r)|}{\beta\cdot d(r)} + \frac{|\varphi_3^+(r)|}{d(r)}\right)$.

We then bound $|c_3^+|_1$ as follows:

$$\begin{aligned}
|c_3^+|_1 &= \sum_{u\in U, uv\in\varphi_3^+, uw\in\varphi_2^+} \frac{2}{\beta \cdot M_{uv}M_{uw}} + \sum_{u\in U, uv\in\varphi_3^+, vw\in\varphi_2^+} \frac{2}{\beta \cdot M_{uv}M_{vw}} + \sum_{u\in U, uv\in\varphi_3^+} \frac{2}{M_{uv}} \\
&\le \sum_{u\in U} \left(\frac{2 \cdot |\varphi_2^+(u)| \cdot |\varphi_3^+(u)|}{\beta \cdot d(u) \cdot d(u)} + \frac{2 \cdot |\varphi_3^+(u)|}{\beta \cdot d(u)} + \frac{2 \cdot |\varphi_3^+(u)|}{d(u)} \right) \\
&= 2 \sum_{u\in U} \left(\frac{|\varphi_2^+(u)| \cdot |\varphi_3^+(u)|}{\beta \cdot d^2(u)} + \frac{|\varphi_3^+(u)|}{\beta \cdot d(u)} + \frac{|\varphi_3^+(u)|}{d(u)} \right).
\end{aligned}$$

This finishes the proof of Lemma 16(c) and thus Lemma 16. ◀

With Lemma 15 and Lemma 16, we can then bound $f_2^+ + f_3^+$:

▶ **Lemma 17.** $f_2^+ + f_3^+ \le \frac{4}{\beta} \cdot \mathrm{cost}_{\mathcal{C}}^k$.

Proof. We define $c(r) = c_2^+(r) + c_3^+(r)$ for every $r \in V$. Then,

$$\begin{aligned} c(r) &\le \frac{2 \cdot |\varphi_2^+(r)|}{\beta \cdot d(r)} + 2\left(\frac{|\varphi_2^+(r)| \cdot |\varphi_3^+(r)|}{\beta \cdot d^2(r)} + \frac{|\varphi_2^+(r)|}{\beta \cdot d(r)} + \frac{|\varphi_3^+(r)|}{d(r)}\right) \\ &= 2\left(\frac{|\varphi_2^+(r)| \cdot |\varphi_3^+(r)|}{\beta \cdot d^2(r)} + \frac{|\varphi_3^+(r)|}{d(r)} + \frac{2 \cdot |\varphi_2^+(r)|}{\beta \cdot d(r)}\right) \\ &\le 2\left(\frac{|\varphi_3^+(r)|}{\beta \cdot d(r)} + \frac{|\varphi_3^+(r)|}{\beta \cdot d(r)} + \frac{2 \cdot |\varphi_2^+(r)|}{\beta \cdot d(r)}\right) \\ &\le \frac{4}{\beta}, \end{aligned}$$

where the first inequality holds because Lemma 15(b) and Lemma 16(b), the second inequality holds because $|\varphi_2^+(r)| \le d(r)$ and $\beta < 1$, the last inequality holds because $|\varphi_2^+(r)| + |\varphi_3^+(r)| \le d(r)$.

Similarly, we have

$$\begin{aligned} |c|_1 &\le \frac{2}{\beta} \sum_{u \in U} \frac{|\varphi_2^+(u)|}{d(u)} + 2 \sum_{u \in U} \left(\frac{|\varphi_2^+(u)| \cdot |\varphi_3^+(u)|}{\beta \cdot d^2(u)} + \frac{|\varphi_3^+(u)|}{\beta \cdot d(u)} + \frac{|\varphi_3^+(u)|}{d(u)}\right) \\ &= 2 \sum_{u \in U} \left(\frac{|\varphi_2^+(u)| \cdot |\varphi_3^+(u)|}{\beta \cdot d^2(u)} + \frac{|\varphi_3^+(u)|}{\beta \cdot d(u)} + \frac{|\varphi_3^+(u)|}{d(u)} + \frac{|\varphi_2^+(u)|}{\beta \cdot d(u)}\right) \\ &\le 2 \sum_{u \in U} \left(\frac{|\varphi_2^+(u)|}{\beta \cdot d(u)} + \frac{|\varphi_3^+(u)|}{\beta \cdot d(u)} + \frac{|\varphi_3^+(u)|}{\beta \cdot d(u)} + \frac{|\varphi_2^+(u)|}{\beta \cdot d(u)}\right) \\ &\le \frac{4}{\beta} \cdot k, \end{aligned}$$

where the first inequality holds because Lemma 15(c) and Lemma 16(c), the second inequality holds because $\beta < 1$ and for any $u \in U$ there is $|\varphi_3^+(u)| \le d(u)$, the last inequality holds because for any $u \in U$ there is $|\varphi_2^+(u)| + |\varphi_3^+(u)| \le d(u)$.

The lemma follows from Claim 13. ◀

For the cost of the remaining $-$edges, i.e. f_2^-, we also analyze the cost coefficients of each vertex.

▶ **Lemma 18.** *There exists a vector $c_2^- \in \mathbb{R}^n_{\ge 0}$ with the following properties:*

(a) $f_2^- \le \sum_{r \in V} c_2^-(r) \cdot \mathrm{cost}_{\mathcal{C}}(r)$.

(b) $c_2^-(r) \le \frac{2}{1-\beta}$, *for every* $r \in V$.

(c) $|c_2^-|_1 \le \frac{2k}{1-\beta}$.

Proof. We have

$$f_2^- = \sum_{u \in U, uv \in \varphi_2^-} (1 - x_{uv}) = \sum_{u \in U, uv \in \varphi_2^-} \frac{|N_H(u) \cap N_H(v)|}{M_{uv}} \le \sum_{\substack{u \in U, uv \in \varphi_2^- \\ w \in N_H(u) \cap N_H(v)}} \frac{1}{M_{uv}}.$$

Given that $uv \in \varphi_2^-$ indicates that $C(u) \neq C(v)$, we distinguish between two cases: $C(w) = C(u)$ and $C(w) \neq C(u)$. For simplicity, in the summations below, we automatically impose the constraints $u \in U, uv \in \varphi_2^-$ and $uw, vw \in E_H$ when the vertices involved in constraints are defined. Notice that we have $d(v) \geq (1-\beta)d(w)$ from Lemma 12.

$$\sum_{u,v,w:C(w)=C(u)} \frac{1}{M_{uv}} \leq \sum_{u,w:C(w)=C(u)} \frac{\mathrm{cost}_{\mathcal{C}}^+(w)}{\max(d(u),(1-\beta)d(w))}. \tag{4}$$

$$\sum_{u,v,w:C(w)\neq C(u)} \frac{1}{M_{uv}} \leq \sum_{u,w:C(w)\neq C(u)} \frac{d(w)}{d(u)} \leq \sum_{u,w:C(w)\neq C(u)} \frac{1}{1-\beta} \leq \sum_u \frac{\mathrm{cost}_{\mathcal{C}}^+(u)}{1-\beta}. \tag{5}$$

Adding (4) and (5), and using that $\mathrm{cost}_{\mathcal{C}}^+(u) \leq \mathrm{cost}_{\mathcal{C}}(u)$ for every $u \in V$, we get

$$f_2^- \leq \sum_{u\in U, w\in N_H(u):C(w)=C(u)} \frac{\mathrm{cost}_{\mathcal{C}}(w)}{\max(d(u),(1-\beta)d(w))} + \sum_{u\in U} \frac{\mathrm{cost}_{\mathcal{C}}(u)}{1-\beta} =: \sum_{r\in V} c_2^-(r)\cdot \mathrm{cost}_{\mathcal{C}}(r).$$

Therefore, Lemma 18(a) holds.

To prove Lemma 18(b), we consider the coefficients for $\mathrm{cost}_{\mathcal{C}}(u)$ and $\mathrm{cost}_{\mathcal{C}}(w)$ respectively. The coefficient for $\mathrm{cost}_{\mathcal{C}}(u)$ is $\frac{1}{1-\beta}$. The coefficient for $\mathrm{cost}_{\mathcal{C}}(w)$ is

$$\sum_{u\in U, w\in N_H(u):C(w)=C(u)} \frac{1}{\max(d(u),(1-\beta)d(w))} \leq \sum_{u\in U, w\in N_H(u):C(w)=C(u)} \frac{1}{(1-\beta)d(w)} \leq \frac{1}{1-\beta}.$$

Therefore, for any $r \in V$, we have $c_2^-(r) \leq \frac{2}{1-\beta}$.

By replacing $\mathrm{cost}_{\mathcal{C}}(u)$ and $\mathrm{cost}_{\mathcal{C}}(w)$ with 1, we then finished the proof of Lemma 18(c) as follows:

$$\begin{aligned}
|c_2^-|_1 &\leq \sum_{u\in U, w\in N_H(u):C(w)=C(u)} \frac{1}{\max(d(u),(1-\beta)d(w))} + \sum_{u\in U} \frac{1}{1-\beta} \\
&\leq \sum_{u\in U, w\in N_H(u):C(w)=C(u)} \frac{1}{d(u)} + \sum_{u\in U} \frac{1}{1-\beta} \\
&\leq \sum_{u\in U} \frac{|N_H(u)|}{d(u)} + \sum_{u\in U} \frac{1}{1-\beta} \leq \sum_{u\in U} 1 + \sum_{u\in U} \frac{1}{1-\beta} \leq \sum_{u\in U} \frac{2}{1-\beta} = \frac{2k}{1-\beta}.
\end{aligned}$$

◀

With Lemma 18, we can then bound f_2^-.

▶ **Lemma 19.** $f_2^- \leq \frac{2}{1-\beta} \cdot \mathrm{cost}_{\mathcal{C}}^k$.

Proof. By Claim 13 and Lemma 18, we have $f_2^- \leq \sum_{r\in V} c_2^-(r)\cdot \mathrm{cost}_{\mathcal{C}}(r) \leq \frac{2}{1-\beta} \cdot \mathrm{cost}_{\mathcal{C}}^k$. ◀

So the final ratio for Algorithm 1 is $1 + \frac{4}{\beta} + \frac{2}{1-\beta}$. Let $\beta = 0.5858$, then the ratio is at most 12.66. This finishes the proof of Lemma 9.

References

1 Rakesh Agrawal, Alan Halverson, Krishnaram Kenthapadi, Nina Mishra, and Panayiotis Tsaparas. Generating labels from clicks. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 172–181, 2009. doi:10.1145/1498759.1498824.

2 Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):1–27, 2008. doi:10.1145/1411509.1411513.

3 Sepehr Assadi and Chen Wang. Sublinear time and space algorithms for correlation clustering via sparse-dense decompositions. In *Proceedings of the 13th Conference on Innovations in Theoretical Computer Science (ITCS)*, volume 215 of *LIPIcs*, pages 10:1–10:20, 2022. doi:10.4230/LIPICS.ITCS.2022.10.

4 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1):89–113, 2004. doi:10.1023/B:MACH.0000033116.57574.95.

5 Guy E Blelloch, Jeremy T Fineman, and Julian Shun. Greedy sequential maximal independent set and matching are parallel on average. In *Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures*, pages 308–317, 2012. doi:10.1145/2312005.2312058.

6 Mélanie Cambus, Davin Choo, Havu Miikonen, and Jara Uitto. Massively parallel correlation clustering in bounded arboricity graphs. In *35th International Symposium on Distributed Computing (DISC)*, volume 209 of *LIPIcs*, pages 15:1–15:18, 2021. doi:10.4230/LIPICS.DISC.2021.15.

7 Mélanie Cambus, Fabian Kuhn, Etna Lindy, Shreyas Pai, and Jara Uitto. A $(3+\varepsilon)$-approximate correlation clustering algorithm in dynamic streams. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2861–2880. SIAM, 2024. doi:10.1137/1.9781611977912.101.

8 Nairen Cao, Vincent Cohen-Addad, Euiwoong Lee, Shi Li, Alantha Newman, and Lukas Vogl. Understanding the cluster linear program for correlation clustering. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1605–1616, 2024. doi:10.1145/3618260.3649749.

9 Nairen Cao, Shang-En Huang, and Hsin-Hao Su. Breaking 3-factor approximation for correlation clustering in polylogarithmic rounds. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4124–4154. SIAM, 2024. doi:10.1137/1.9781611977912.143.

10 Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. A graph-theoretic approach to webpage segmentation. In *Proceedings of the 17th International conference on World Wide Web (WWW)*, pages 377–386, 2008. doi:10.1145/1367497.1367549.

11 Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 126–137, 2019. doi:10.1145/3313276.3316322.

12 Moses Charikar, Neha Gupta, and Roy Schwartz. Local guarantees in graph cuts and clustering. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 136–147, 2017. doi:10.1007/978-3-319-59250-3_12.

13 Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005. doi:10.1016/J.JCSS.2004.10.012.

14 Shuchi Chawla, Konstantin Makarychev, Tselil Schramm, and Grigory Yaroslavtsev. Near optimal LP rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 219–228, 2015. doi:10.1145/2746539.2746604.

15 Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In *Advances in Neural Information Processing Systems (Neurips)*, pages 2204–2212, 2012.

16 Flavio Chierichetti, Nilesh Dalvi, and Ravi Kumar. Correlation clustering in MapReduce. In *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 641–650, 2014. doi:10.1145/2623330.2623743.

17 Vincent Cohen-Addad, Silvio Lattanzi, Slobodan Mitrović, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub Tarnawski. Correlation clustering in constant many parallel rounds. In *International Conference on Machine Learning*, pages 2069–2078. PMLR, 2021. URL: http://proceedings.mlr.press/v139/cohen-addad21b.html.

18 Vincent Cohen-Addad, Euiwoong Lee, Shi Li, and Alantha Newman. Handling correlated rounding error via preclustering: A 1.73-approximation for correlation clustering. In *Proceedings of the 64rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2023.

19 Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. Correlation clustering with Sherali-Adams. In *Proceedings of 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 651–661, 2022. doi:10.1109/FOCS54457.2022.00068.

20 Sami Davies, Benjamin Moseley, and Heather Newman. Fast combinatorial algorithms for min max correlation clustering. *arXiv preprint arXiv:2301.13079*, 2023. doi:10.48550/arXiv.2301.13079.

21 Sami Davies, Benjamin Moseley, and Heather Newman. Simultaneously approximating all lp-norms in correlation clustering. In *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024*, 2024.

22 Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008. doi:10.1145/1327452.1327492.

23 Manuela Fischer and Andreas Noever. Tight analysis of parallel randomized greedy mis. *ACM Transactions on Algorithms (TALG)*, 16(1):1–13, 2019. doi:10.1145/3326165.

24 Holger SG Heidrich, Jannik Irmai, and Bjoern Andres. A 4-approximation algorithm for min max correlation clustering. In *AISTATS*, pages 1945–1953, 2024.

25 Dmitri V. Kalashnikov, Zhaoqi Chen, Sharad Mehrotra, and Rabia Nuray-Turan. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering*, 20(11):1550–1565, 2008. doi:10.1109/TKDE.2008.78.

26 Sanchit Kalhan, Konstantin Makarychev, and Timothy Zhou. Correlation clustering with local objectives. *Advances in Neural Information Processing Systems*, 32, 2019.

27 Michael Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 1–10, 1985. doi:10.1145/22145.22146.

28 Xinghao Pan, Dimitris S. Papailiopoulos, Samet Oymak, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. Parallel correlation clustering on big graphs. In *Advances in Neural Information Processing Systems (Neurips)*, pages 82–90, 2015. URL: https://proceedings.neurips.cc/paper/2015/hash/b53b3a3d6ab90ce0268229151c9bde11-Abstract.html.

29 Gregory Puleo and Olgica Milenkovic. Correlation clustering and biclustering with locally bounded errors. In *International Conference on Machine Learning*, pages 869–877. PMLR, 2016. URL: http://proceedings.mlr.press/v48/puleo16.html.

30 Jessica Shi, Laxman Dhulipala, David Eisenstat, Jakub Łkacki, and Vahab Mirrokni. Scalable community detection via parallel correlation clustering. *arXiv preprint arXiv:2108.01731*, 2021.

31 Nate Veldt, David F. Gleich, and Anthony Wirth. A correlation clustering framework for community detection. In *Proceedings of the 2018 ACM World Wide Web Conference (WWW)*, pages 439–448, 2018. doi:10.1145/3178876.3186110.

A Reduction from All Monotone Symmetric Norms to Top-k Norms

▶ **Definition 20** (Ordered Norms). *For any vector $x \in \mathbb{R}^n_{\geq 0}$, let $x^{\downarrow}$ denote the vector x with its coordinates sorted in non-increasing order. Given weight vector $w \in \mathbb{R}^n_{\geq 0}$ with $w_1 \geq w_2 \geq \cdots \geq w_n$, the w-ordered norm of x is defined as* $\operatorname{order}(w; x) = \sum_{i=1}^{n} w_i x_i^{\downarrow}$.

▶ **Lemma 21** (Lemma 5.2 of [11]). *For any monotone and symmetric norm $f : \mathbb{R}^n \to \mathbb{R}_+$, define the set $\mathbb{B}_+(f) := \{x \in \mathbb{R}^n_+ : f(x) \leq 1\}$, and $W = \{w \in \mathbb{R}^n_+ : w_1 \geq w_2 \geq \cdots \geq w_n, w$ is a subgradient of f at some $x \in \mathbb{B}_+(f)\}$. Then we have $f(x) = \max_{w \in W} \operatorname{order}(w; x)$ for every $x \in \mathbb{R}^n_{\geq 0}$.*

▶ **Lemma 6.** *For any integer $k \in [n]$, if an algorithm returns a single clustering $\mathcal{C}_{ALG}$ that is simultaneously a ρ-approximation for all top-k norm objectives, then $\mathcal{C}_{ALG}$ is a ρ-approximation for any monotone and symmetric norm $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_+$.*

Proof of Lemma 6. For any $w = (w_1, w_2, \ldots, w_n)$ such that $w_1 \geq w_2 \geq \cdots \geq w_n \geq 0$, if we set $w' = (w'_1, w'_2, \ldots, w'_n)$ as

$$w'_i = \begin{cases} w_i - w_{i+1} & i \in [1, n-1] \\ w_n & i = n \end{cases}$$

Let $\text{top}_k(x)$ denote the top-k norm of x. Then we have $\text{order}(w; x) = \sum_{k=1}^n w'_k \cdot \text{top}_k(x)$.

Let $\mathbb{B}_+(f) := \{x \in \mathbb{R}^n_+ : f(x) \leq 1\}$ and $W = \{w \in \mathbb{R}^n_+ : w_1 \geq w_2 \geq \cdots \geq w_n, w \text{ is a subgradient of } f \text{ at some } x \in \mathbb{B}_+(f)\}$. We construct a new set $W' = \{w' | w \in W : w' = (w'_1 = w_1 - w_2, w'_2 = w_2 - w_3, \ldots, w'_{n-1} = w_{n-1} - w_n, w'_n = w_n)\}$. By Lemma 21, we have $f(x) = \max_{w \in W} \text{order}(w; x) = \max_{w' \in W'} \sum_{k=1}^n w'_k \cdot \text{top}_k(x)$.

Let y be the disagreement vector for the given clustering $\mathcal{C}_{ALG}$. For any symmetric monotone norm $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$, define y^*_f to be the disagreement vector for the optimal clustering under the norm f. By the assumption that $\mathcal{C}_{ALG}$ is a simultaneous ρ-approximation for every top-k norm, we have we have $\text{top}_k(y) \leq \rho \cdot \text{top}_k(y^*_{\text{top-k}})$ for every $k \in [n]$, where $y^*_{\text{top-k}}$ is the disagreement vector for the optimal clustering under top-k objective. Now we bound $f(y)$ in terms of $f(y^*_f)$ for any monotone symmetric norm f:

$$\begin{aligned} f(y) &= \max_{w' \in W'} \sum_{k=1}^n w'_k \cdot \text{top}_k(y) \leq \max_{w' \in W'} \sum_{k=1}^n w'_k \cdot \rho \cdot \text{top}_k(y^*_{\text{top-k}}) \\ &= \rho \cdot \max_{w' \in W'} \sum_{k=1}^n w'_k \cdot \text{top}_k(y^*_{\text{top-k}}) \leq \rho \cdot \max_{w' \in W'} \sum_{k=1}^n w'_k \cdot \text{top}_k(y^*_f) = \rho \cdot f(y^*_f). \end{aligned}$$

◀

B Implementing Algorithm 1 in Nearly Linear Time

In this section, we show how to run Algorithm 1 approximately in nearly linear time. Indeed, the algorithm can be implemented in MPC model with $O(1)$ rounds. More precisely, we will show the following theorem:

▶ **Theorem 22.** *Let $\epsilon > 0$ and $\delta > 0$ be small enough constants. Given a graph $G = (V, E)$, there exists an MPC algorithm that computes a solution $\{\tilde{x}_{uv}\}_{u,v \in V}$ such that*

1. *For any integer $k \in [n]$, we have $\text{cost}^k_{\tilde{x}} \leq (12.66 + \epsilon)\text{opt}^k$.*
2. *For any $u, v, w \in V$, we have $\tilde{x}_{uv} + \tilde{x}_{uw} + \epsilon \geq \tilde{x}_{vw}$.*

The algorithm succeeds with probability at least $1 - 1/n$. Moreover, the algorithm runs in $O(1)$ rounds, has a total work of $\tilde{O}(m/\epsilon^6)$, requires $O(n^\delta)$ memory per machine and a $\tilde{O}(m/\epsilon^6)$ total memory.

We give the nearly linear time implementation of Algorithm 1 in Algorithm 2. Line 2 constructs the graph H efficiently. Line 3-6 find the set K of $-$edges we want to assign LP value. For any $-$edge $uv \notin K$, we will simply set its LP value as 1. Last, Line 7 to Line 14 is to set up $\tilde{x}_{uv}$ satisfying the conditions in Theorem 22. The full version of this paper will contain a complete analysis of Algorithm 2 and proof of Theorem 22.

Algorithm 2 Nearly Efficient Algorithm for Algorithm 1.

Input: Graph $G = (V, E), \epsilon \in (0, 1)$

Output: $\{\tilde{x}_{uv}\}_{u,v \in V}$ such that is $(12.66+26\epsilon)$-approximate and satisfy approximate triangle inequality.

function AllNormCCBySampling($G = (V, E)$)
  Construct subgraph $H = (V, E_H \subseteq E)$
  for every $u \in V$ **do** ▷ Compute set K
    $S_u \leftarrow \emptyset$
    for every $v \in d_H(u)$ **do**: add v to S_u with probability $\min(\frac{8 \log n}{\epsilon d_H(u)}, 1)$
  $K \leftarrow \{uv \notin E : \exists w \in S_u, vw \in E_H \text{ or } \exists w \in S_v, uw \in E_H\}$
  $\tau \leftarrow \frac{400 \log n}{\epsilon^5}$ ▷ Compute the $\tilde{x}_{uv}$ values
  for every $j \in [1, \log n]$ **do**
    $S(j) \leftarrow \emptyset$
    for $v \in V$ **do**: add v to $S(j)$ with probability $\min(\tau/2^j, 1)$
    for every $uv \in E \cup K$ with $M_{uv} \in [2^{j-1}, 2^j)$ **do**: $\tilde{x}_{uv} \leftarrow 1 - \frac{2^j |N_H(u) \cap N_H(v) \cap S(j)|}{\tau \cdot M_{uv}}$
  for every $uv \in E$ **do**: **if** $\tilde{x}_{uv} \leq \epsilon$ **then** $\tilde{x}_{uv} \leftarrow 0$
  for every $uv \in K$ **do**: **if** $\tilde{x}_{uv} \geq 1 - \epsilon$ **then** $\tilde{x}_{uv} \leftarrow 1$
  for every $uv \in \binom{V}{2} \setminus (E \cup K)$ **do**: $\tilde{x}_{uv} \leftarrow 1$
end function

C Rounding

We will present a nearly linear time rounding algorithm. Furthermore, our algorithm only takes $\tilde{O}(1)$ rounds in the MPC model. The purpose of this section is to show

▶ **Theorem 4.** *Let $\epsilon \in (0, 1)$ be a constant. Given a graph $G = (V, E)$ and a set of LP values $(x_{uv})_{u,v \in V}$ satisfying the approximate triangle inequality, that is, for any $u, v, w \in V$, we have $x_{uv} + x_{uw} + \epsilon \geq x_{vw}$. Let $y_u = \sum_{uv \in E} x_{uv} + \sum_{uv \in \binom{V}{2} \setminus E} (1 - x_{uv})$ be the LP disagreement for node u. There exists an MPC algorithm that computes a clustering $\mathcal{C}$ such that for any node u, we have*

$$\text{cost}_{\mathcal{C}}(u) \leq (5 + 55\epsilon) y_u.$$

This algorithm always succeeds but terminates in $O(\log^3 n/\epsilon)$ rounds with high probability and requires $O(n^\delta)$ memory per machine. Moreover, let $K = E \cup \{uv \in \binom{V}{2} \setminus E \mid x_{uv} < 1\}$ be the set of $+$edges and $-$edges whose LP value is less than 1. The algorithm uses a total memory of $O(|K| \log n)$ and a total work of $O(|K| \log^3 n/\epsilon)$.

We emphasize that even if the LP values satisfy the exact triangle inequality, rather than an approximate triangle inequality, the ϵ terms in the approximate ratio will still be present. These ϵ terms arise from two sources: the approximate inequality itself and the inherent characteristics of our MPC algorithm.

Given Theorem 22 and Theorem 4, we are now able to show the main result of this paper.

Proof of Theorem 3. We first run Algorithm 2, which outputs $\tilde{x}_{uv}$ as input to Algorithm 3. Note that by Theorem 22, we know that for any $k \in [1, n]$, we have

$$\text{cost}^k_{\tilde{x}} \leq (12.66 + \epsilon)\text{opt}^k$$

where opt^k is the cost of the optimal correlation clustering solution when using the top-k norm objective. By Theorem 4, we know that the final cluster $\mathcal{C}$ satisfies

$$\text{cost}^k_{\mathcal{C}} \leq (5+55\epsilon)\text{cost}^k_{\tilde{x}} \leq (63.3+O(\epsilon))\text{opt}^k.$$

By Lemma 6, we know that $\mathcal{C}$ is a simultaneous $(63.3+O(\epsilon))$-approximate clustering for all monotone symmetric norms.

For the number of rounds, Algorithm 1 takes $O(1)$ rounds, and Algorithm 3 takes $O(\log^3 n/\epsilon)$ rounds, resulting in a total of $O(\log^3 n/\epsilon)$ rounds.

Algorithm 1 requires a total memory of $\tilde{O}(m/\epsilon^6)$ and a total work of $\tilde{O}(m/\epsilon^6)$ in the MPC model. By analyzing Algorithm 2, we know that $|K| = O(m \log n/\epsilon)$. Therefore, Algorithm 3 requires a total memory of $\tilde{O}(m/\epsilon^2)$ and a total work of $\tilde{O}(m/\epsilon)$ in the MPC model, as established by Theorem 4. In total, it requires a total memory of $\tilde{O}(m/\epsilon^6)$ and a total work of $\tilde{O}(m/\epsilon^6)$ in the MPC model. ◀

Rounding Algorithm

Assume that we are given an instance graph $G = (V, E)$ and a LP solution $(x_{uv})_{u,v\in V}$ such that $x_{uv} + x_{uw} + \epsilon \geq x_{vw}$ for any $u, v, w \in V^3$. Given a subgraph V_t and node w, radius r, define the ball centering at w with radius r as $\text{Ball}_{V_t}(w, r) = \{u \mid u \in V_t, x_{wu} \leq r\}$. Define $L_t(w) = \sum_{u\in\text{Ball}_{V_t}(w,r)}(r - x_{uw})$. Note that $L_t(w) \geq r$ since w itself is in $\text{Ball}_{V_t}(w, r)$.

Algorithm Description. Our algorithm works as follows: in each round, we choose a set of cluster centers and choose the ball with radius $\frac{2}{5}$ as a cluster. More precisely. At step t, V_t is the set of unclustered vertices. We first compute $L_t^{\max}$ to ensure that for each $u \in V_t$, we have $L_t(u) < (1+\epsilon)L_t^{\max}$. For any node u, if $L_t(u) \geq L_t^{\max}$, we will add u to the set of candidate cluster centers M_t (Line 6-10). Then, we compute cluster centers by adding vertices in the M_t set to the S_t set with probability p_t, where the more vertices in M_t are in $\text{Ball}(radius = \frac{2}{5})$ with each other, the smaller the probability (Line 11-14). After that, to avoid conflicts, we remove some cluster centers in S_t if they are too close to each other, and derive the final cluster center set H_t (Line 15). Let $F_t = \text{Ball}_{V_t}(H_t, \frac{2}{5})$ be the nodes clustered at step t. Then we add each $u \in F_t$ to the cluster from H_t with minimum ID. We will remove the clustered nodes and repeat the above process until all vertices are clustered.

Complete analysis of Algorithm 3 and proof of Theorem 4 appear in the full version.

D A Constant Round MPC Algorithm

We will show Theorem 5 in this section. We repeat for convenience,

▶ **Theorem 5.** *Let $\epsilon \in (0, 1)$ be a constant. There exists a randomized MPC algorithm in the strictly sublinear regime that, given a correlation clustering instance $G = (V, E)$, in $\mathbf{O(1)}$ rounds outputs a clustering that is simultaneously a $(359+\epsilon)$-approximation, for all monotone symmetric norms. This algorithm succeeds with high probability, and uses a total memory of $\tilde{O}(m/\epsilon^2)$ and a total work of $\tilde{O}(m/\epsilon^2)$.*

Algorithm

Theorem 3 gives us an $O(\log^3 n)$ rounds MPC algorithm. The bottleneck is the rounding procedure. To achieve a constant rounds MPC algorithm, instead of setting up the LP and rounding, we use the pre-clustering algorithm from [17], which is very useful for ℓ_1-norm correlation clustering. We show that the pre-clustering algorithm can also provide an $O(1)$-approximate ratio for all monotone symmetric norms simultaneously.

Algorithm 3 The Rounding Algorithm.

Input: Graph $G = (V, E)$, LP solution $(x_{uv})_{u,v \in V}$ satisfying $x_{uv} + x_{uw} + \epsilon \geq x_{vw}$ for any $u, v, w \in V^3$
Output: A function of clustering $\mathcal{C}$; i.e., $C(u) = C(v)$ iff u and v belongs to the same cluster.

function ParallelRounding($G = (V, E), (x_{uv})$)
  $V_1 \leftarrow V$.
  $C(v) \leftarrow \emptyset$ for all $v \in V$.
  $t \leftarrow 1$.
  while $V_t \neq \emptyset$ **do**
    $M_t \leftarrow \emptyset, S_t \leftarrow \emptyset$
    $L_t^{\max} = \max\{r(1+\epsilon)^j | r(1+\epsilon)^j \leq \max_{w \in V_t} L_t(w), \text{where } j \text{ is an integer}\}$
    for $u \in V_t$ **do** ▷ Find cluster center candidate
      if $L_t(u) \geq L_t^{\max}$ **then**.
        $M_t \leftarrow M_t \cup \{u\}$
    $\Delta_t \leftarrow \max_{u \in M_t} |\text{Ball}_{V_t}(u, \frac{2}{5}) \cap M_t|$.
    $p_t \leftarrow 1/(2\Delta_t)$.
    for $u \in M_t$ **do**
      Add u to S_t with probability p_t ▷ Choose cluster centers in parallel.
    $H_t = \{u \in S_t \mid \nexists v \in S_t \cap V_t \text{ such that } x_{uv} \leq \frac{2}{5}\}$ ▷ remove cluster centers that conflict.
    $F_t \leftarrow H_t \cup \text{Ball}_{V_t}(H_t, \frac{2}{5})$ ▷ the set of settled vertices in round t.
    for $u \in F_t$ **do**
      Let v be the vertex with minimum ID among $\text{Ball}_{V_t}(u, \frac{2}{5}) \cap H_t$.
      $C(u) \leftarrow v$ ▷ adding u to the cluster of v.
    $V_{t+1} \leftarrow V_t \setminus F_t$, $t \leftarrow t + 1$
  return $\mathcal{C}$
end function

Algorithm Description. The algorithm from [17] is parameterized by β and λ. It has three steps:

1. The first step is the same as the first step of Algorithm 1, where we compute the graph H (Line 2).
2. The algorithm marks a node as light if it loses more than a λ fraction of its neighbors in the first step. Otherwise, it marks the node as heavy. The algorithm removes all edges between two light nodes in H (Line 3 - Line 6).
3. The last step is to output the connected components F of the final graph.

Algorithm 4 Pre-clustering – Algorithm 1 in [17].

Input: Graph $G = (V, E)$
Output: Clustering F.

function PreClustering($G = (V, E)$)
  Let $E_H = \{uv \in E : |N(u) \Delta N(v)| \leq \beta \cdot \max(d(u), d(v))\}$ and $H = (V, E_H)$
  for $v \in V$ **do**
    if $d_H(v) < (1 - \lambda)d(v)$ **then** Mark it as light
    else Mark it as Heavy
  Let $E_{\tilde{G}} = \{uv \in E_H : u \text{ or } v \text{ is heavy}\}$ and $\tilde{G} = (V, E_{\tilde{G}})$
  Compute its connected components on $\tilde{G}$, denoted as F, **return** F.
end function

A detailed analysis of Algorithm 4 along with the formal proof of Theorem 5 is deferred to the full version.