

# Min-Max Correlation Clustering via Neighborhood Similarity

**Nairen Cao**

New York University, NY, USA

**Steven Roche**

Boston College, Chestnut Hill, MA, USA

**Hsin-Hao Su**

Boston College, Chestnut Hill, MA, USA

---

## Abstract

We present an efficient algorithm for the min-max correlation clustering problem. The input is a complete graph where edges are labeled as either positive (+) or negative (−), and the objective is to find a clustering that minimizes the  $\ell_\infty$ -norm of the disagreement vector over all vertices.

We address this problem with an efficient  $(3 + \epsilon)$ -approximation algorithm that runs in nearly linear time,  $\tilde{O}(|E^+|)$ , where  $|E^+|$  denotes the number of positive edges. This improves upon the previous best-known approximation guarantee of 4 by Heidrich, Irmay, and Andres [37], whose algorithm runs in  $O(|V|^2 + |V|D^2)$  time, where  $|V|$  is the number of nodes and  $D$  is the maximum degree in the graph  $(V, E^+)$ .

Furthermore, we extend our algorithm to the massively parallel computation (MPC) model and the semi-streaming model. In the MPC model, our algorithm runs on machines with memory sublinear in the number of nodes and takes  $O(1)$  rounds. In the streaming model, our algorithm requires only  $\tilde{O}(|V|)$  space, where  $|V|$  is the number of vertices in the graph.

Our algorithms are purely combinatorial. They are based on a novel structural observation about the optimal min-max instance, which enables the construction of a  $(3 + \epsilon)$ -approximation algorithm using  $O(|E^+|)$  neighborhood similarity queries. By leveraging random projection, we further show these queries can be computed in nearly linear time.

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

**Keywords and phrases** Min Max Correlation Clustering, Approximate algorithms

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2025.41

**Related Version** *Full Version*: <https://arxiv.org/abs/2502.12519> [17]

**Funding** Supported by NSF CCF-2008422.

## 1 Introduction

In the correlation clustering problem, we are given a complete graph where each edge is labeled as either “+” or “−”. A “+” edge indicates that the two vertices are *similar*, while a “−” edge indicates they are *dissimilar*. For any partition of the graph, an edge is considered to be in disagreement if it is a negative edge and its endpoints belong to the same cluster, or if it is a positive edge and its endpoints belong to different clusters. It is typical to assume that the input graph is  $G^+ = (V, E^+)$ , where  $E^+$  is the set of positive edges, and to obtain time bounds in terms of properties of  $G^+$ , such as  $m = |E^+|$  and  $D$ , the maximum degree of  $G^+$ . This is because, in many practical applications, the number of positive edges is often much smaller than the number of negative edges [23].



© Nairen Cao, Steven Roche, and Hsin-Hao Su;  
licensed under Creative Commons License CC-BY 4.0

33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 41; pp. 41:1–41:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Given a clustering (partition), the disagreement  $\rho(v)$  for any node  $v$  is defined as the number of edges incident to  $v$  that are in disagreement with respect to the clustering. The goal of the correlation clustering problem is to find a clustering that minimizes an objective function capturing the disagreement of edges.

Puleo and Milenkovic [41] introduced the objective of minimizing the  $\ell_p$ -norm of the disagreements over the vertices, which is defined as  $\left(\sum_{v \in V} \rho(v)^p\right)^{1/p}$ . This objective generalizes the correlation clustering problem proposed by Bansal, Blum, and Chawla [8], which corresponds to the case  $p = 1$ , where the goal is to minimize the total number of disagreements. Significant progress has been made on the  $\ell_1$ -norm objective [21, 3, 22, 27, 26], Cao, Cohen-Addad, Lee, Li, Newman, and Vogl [14] present a 1.437-approximate algorithm for  $\ell_1$ -norm objective. For the case  $p = \infty$ , it corresponds to the min-max correlation clustering problem, where the goal is to minimize the maximum disagreements over all vertices. While the  $p = 1$  case captures scenarios in which minimizing overall disagreements is desired, the  $p = \infty$  case caters to situations where the quality of every individual needs to be ensured – for example, in community detection problems where no “antagonists” are allowed, with an antagonist referring to an individual whose properties are inconsistent with those of too many other members.

Compared to the  $p = 1$  case where significant progress has been made, the  $p = \infty$  case remains relatively unexplored. Puleo and Milenkovic [41] proposed an algorithm that achieves a 48-approximation ratio. Their approach uses the standard metric linear programming formulation followed by a rounding algorithm. Later, Charikar, Gupta, and Schwartz [20] improved this result to a 7-approximation using the same framework, and Kalhan, Makarychev, and Zhou [39] further reduced it to a 5-approximation. Davies, Moseley, and Newman [29] designed a combinatorial algorithm that achieves a 40-approximation ratio with a runtime of  $O(n^2 \log n)$ , where  $n = |V|$  is the number of vertices in the graph. The best known approximation ratio to date is 4, achieved by Heidrich, Imai, and Andres [37], who also used a combinatorial approach with a runtime of  $O(n^2 + nD^2)$ , where  $D$  is the maximum degree of the  $G^+$ .

Concerning efficiency, the ultimate goal of an efficient algorithm is to achieve a running time that is nearly linear in  $m = |E^+|$ . However, none of the aforementioned algorithms has yet achieved such a goal. Recently, Cao, Li, and Ye [16] proposed an nearly linear-time algorithm that achieves a 63.3-approximation ratio. While their algorithm is fast, the approximation ratio is far from optimal. This naturally leads to the question:

Can we design a nearly linear algorithm for min-max correlation clustering with a small approximation ratio?

We give a nearly linear algorithm for the problem that achieves a  $(3 + \epsilon)$ -approximation. Moreover, it can be made exactly 3-factor approximation with additional running time.

► **Theorem 1.** *Let  $G = (V, E^+)$  be a min-max correlation clustering instance,  $\epsilon > 0$  be a small constant, and  $\text{OPT}$  be the value of the optimal solution. There exist:*

1. *A randomized  $O(m \log^2 n / \epsilon^2)$ -time algorithm that outputs a clustering  $\mathcal{C}$  with  $\text{obj}(\mathcal{C}) \leq (3 + \epsilon) \cdot \text{OPT}$  w.h.p.<sup>1</sup>*
2. *A deterministic  $O(mD \log n)$ -time sequential algorithm that outputs a clustering  $\mathcal{C}$  with  $\text{obj}(\mathcal{C}) \leq 3 \cdot \text{OPT}$ , where  $D$  is the maximum degree of  $G^+ = (V, E^+)$ .*

<sup>1</sup> With high probability, which refers to with probability at least  $1 - 1/n^c$  for a sufficiently large constant  $c$ .

## 1.1 Semi-Streaming and MPC Settings

For large graphs, it can be infeasible to store the graph  $G$  entirely on a single machine and then compute the solution. A substantial body of research has focused on massively parallel computation (MPC) algorithms for the  $p = 1$  case [10, 40, 31, 11, 25, 7, 12, 15, 28]. Very recently, Cao, Cohen-Addad, Lee, Li, Lolck, Newman, Thorup, Vogl, Yan, and Zhang [13] gave an  $O(1)$ -round MPC algorithm that achieves a  $(1.437 + \epsilon)$ -approximation ratio.

Moreover, several constant-round semi-streaming algorithms with  $n$  polylog  $n$  space complexity have been developed [23, 2, 25, 9]. The first single-pass semi-streaming algorithm was given by Ahn, Cormode, Guha, McGregor, and Wirth [2], and it was later improved by multiple works [7, 9, 18, 28]. Very recently, a breakthrough result of Assadi, Khanna, and Putterman [5] showed that it is possible to obtain a  $(\alpha_{\text{best}} + o(1))$ -approximation in a single pass using  $n$  polylog  $n$  space, where  $\alpha_{\text{best}}$  denotes the best approximation ratio achievable by polynomial-time sequential algorithms. Currently, it is known that  $\alpha_{\text{best}} \leq 1.437$  [14].

In contrast, less work in such directions have been done for the  $p = \infty$  setting. For the MPC setting, the only known work is by Cao, Ye, and Li [16], who proposed an algorithm that achieves a 63.3-approximation ratio in  $O(\log^3 n)$  rounds and another algorithm that achieves a 360-approximation ratio in  $O(1)$  rounds. For the semi-streaming setting, to our knowledge, we are not aware of any existing work on semi-streaming algorithms. Most of the aforementioned streaming algorithms for  $p = 1$  were specifically tailored for this case, and the sparsification technique of [5] does not seem to generalize obviously to other values of  $p$ .

In addition to our sequential algorithm, we give a constant-round MPC algorithm and a single-pass semi-streaming algorithm that achieve a  $(3 + \epsilon)$ -approximation for the problem:

► **Theorem 2.** *Let  $G = (V, E^+)$  be a min-max correlation clustering instance,  $\epsilon > 0$  be a small constant, and  $\text{OPT}$  be the value of the optimal solution. In the following models, there exist randomized algorithms that output a clustering  $\mathcal{C}$  with  $\text{obj}(\mathcal{C}) \leq (3 + \epsilon) \cdot \text{OPT}$  w.h.p.:*

1. (MPC model) *An  $O(1)$ -round algorithm using  $O(n^\delta)$  memory per machine and total memory  $O(m \log n / \epsilon^2)$  for any constant  $0 < \delta < 1$ .*
2. (Semi-streaming model) *A single-pass streaming algorithm that uses  $O(n \log n / \epsilon^2)$  space.*

## 1.2 Technical Overview

**Better Approximation.** Our first main technical contribution is a newly achieved approximation factor of 3. Given a guess for the optimal objective value  $\phi$ , if  $\text{OPT} \leq \phi$ , [37] observed that if the neighborhoods of  $u$  and  $v$  share at least  $2\phi$  elements, then they must belong to the same cluster in the optimal solution. Similarly, if neighborhoods of  $u$  and  $v$  differ by more than  $2\phi$  elements, then they must belong to different clusters in the optimal solution.

Furthermore, they observed that these properties can be used to determine the clusters for vertices of degrees at least  $4\phi$ . Specifically, if  $d(x) \geq 4\phi$ , then for every other vertex  $y$ , either  $|N[x] \cap N[y]| \geq 2\phi$  or  $|N[x] \Delta N[y]| \geq 2\phi$ . Here,  $N[x] = N(x) \cup \{x\}$  represents the closed neighborhood of vertex  $x$ . The remaining vertices can then be placed in singleton clusters, as the disagreements per vertex will be upper bounded by their degrees,  $4\phi$ . Therefore, a clustering of disagreements upper bounded by  $4\phi$  can be constructed, resulting in a 4-approximation algorithm.

To achieve a 3-approximation, we first observe that if two vertices  $x$  and  $y$  have degrees greater than  $3\phi$ , then it is also the case either  $|N[x] \cap N[y]| \geq 2\phi$  or  $|N[x] \Delta N[y]| \geq 2\phi$  holds. In other words, whether  $x$  and  $y$  belong to the same cluster is uniquely determined in the optimal solution. Therefore, the clustering induced on the high-degree vertices (vertices with  $d > 3\phi$ ) is uniquely determined.

Now the question lies in the placement of the low-degree vertices, that is, vertices with degrees upper-bounded by  $3\phi$ . It is unclear whether they should be placed in singleton clusters, as it is possible that they need to be included in the same cluster with certain high-degree vertices. Otherwise, the disagreements associated with the high-degree vertices could become too large.

We show that the low-degree vertices can be placed in the high-degree clusters to achieve a maximum disagreements of  $3\phi$ , provided  $OPT \leq \phi$ . A key structural result we show is that if a low-degree vertex  $w$  belongs to some cluster  $C$  in an optimal solution, then no vertex  $v$  outside  $C$  can have similar neighborhood with  $w$ , i.e.,  $|N[v] \Delta N[w]| \leq 2\phi$ . Using this structural result, we show the following algorithm constructs a clustering with maximum disagreement upper bounded by  $3\phi$  (presented slightly differently here than in the main body for the sake of intuition):

1. Form clusters for high-degree vertices based on whether  $|N[u] \Delta N[v]| \leq 2\phi$  for all high-degree vertices  $u, v$  (abort if any inconsistency occurs).
2. Choose an arbitrary vertex  $u$  in each cluster and have it send proposal messages to low-degree neighboring vertices whose neighborhoods are similar to  $u$ .
3. For each low-degree vertex that receives at least one proposal, pick one arbitrary proposal and join the cluster containing the vertex that sent it.
4. Place all low-degree vertices that do not receive a proposal into singleton clusters.

**Efficient Implementations.** The remaining question is how such an algorithm can be implemented efficiently, particularly in time (and total memory) nearly linear in  $|E^+|$ . A main technical challenge lies in Step 1. To implement it within the aforementioned time bound, we can only afford to conduct similarity tests (i.e. to test whether  $|N[u] \Delta N[v]| \leq 2\phi$ ) for  $O(|E^+|)$  times. However, it is possible for two vertices that are endpoints of a negative edge to have similar neighborhoods and thus need to be placed in the same cluster to achieve a good clustering.

Using the structural result, we further show that two high-degree vertices  $u$  and  $v$  are in the same cluster in the optimal solution if and only if there are at least  $\phi + 1$  disjoint paths of length 2 connecting  $u$  and  $v$  in  $E_{\text{sim}}$ , where  $E_{\text{sim}} \subseteq E^+$  consists of all the edges in  $E^+$  whose endpoints have similar neighborhoods. This property enables us to develop efficient algorithms for the sublinear MPC model and the sequential model.

The remaining question lies in how to find  $E_{\text{sim}}$  efficiently. To this end, for each vertex  $u$ , we treat its neighborhood set  $N[u]$  as a point in an  $n$ -dimensional space. Then, we apply the (discrete) random projection technique [1] developed for the Johnson-Lindenstrauss transform [38] to reduce the dimension to  $O(\log n / \epsilon^2)$  while preserving the  $\ell_2$  distance (up to a  $(1 \pm \epsilon)$  factor) between the points. For 0/1 vectors, the square of the  $\ell_2$  distance is exactly the symmetric difference. Since the dimension is  $O(\log n / \epsilon^2)$ , it takes  $O(\log n / \epsilon^2)$  time to compute the difference. To our knowledge, this is the first time that random projection techniques have been applied to computing efficient solutions for correlation clustering and problems alike. This may be of independent interest, as neighborhood similarity is known to be used in various tools such as *almost-clique decompositions* [36, 19, 33, 34, 32, 4, 30, 6, 24, 7].

**Single-Pass Semi-Streaming.** While the above techniques are sufficient for getting our MPC and sequential algorithms, the single-pass semi-streaming algorithm introduces additional technical difficulties. The main difficulty for a single-pass semi-streaming to work here lies in Step 2, where an arbitrarily chosen high-degree vertex in each cluster proposes to its neighbors who have similar neighborhoods. For convenience, we call the chosen high-degree vertices *pivots* here.

To be able to do this, we need to memorize the neighbors of all the chosen vertices using  $n \text{polylog}(n)$  space. Suppose that  $\text{OPT} \leq \phi$ , it can be shown that each cluster in the optimal clustering containing at least one low-degree vertex has size of  $\Theta(\phi)$ . As a result, any vertex from these clusters has degree at most  $O(\phi)$ . Since we pick a pivot per cluster,  $O(\phi \cdot (n/\phi))$  is the space we need to store the neighbors of the pivots.

However, we do not know beforehand how the clusters of high-degree vertices look like, so the pivots cannot be chosen at the beginning of the stream. Without knowing what the pivots are beforehand, it is difficult to store their neighbors in the same pass.

As a result, we sample each vertex (both high-degree and low-degree vertices) independently with probability  $O(\log n/d(v))$  so that w.h.p. each cluster in the optimal clustering has  $O(\log n)$  sampled vertices. Then we store the neighbors of all the sampled vertices. This poses another problem: low-degree vertices may be chosen as pivots. However, a low-degree vertex is exempted from our structural result – using it as a pivot may steal vertices from other clusters in an optimal clustering.

To resolve this, we do the following. For each sampled vertex  $y$ , we first try to recover the high-degree portion  $L$  of the cluster containing  $y$  in the optimal solution. We construct a candidate set  $\text{Cand}(L)$  that contains vertices that would not be added to other clusters. When using  $y$  as a pivot, we restrict it to consider only the intersection with the candidate set,  $N[y] \cap \text{Cand}(L)$  to ensure that it does not steal vertices from other clusters.

Roughly speaking, the candidate set  $\text{Cand}(L)$  contains all the low-degree vertices that have similar neighborhood with every vertex in  $L$  but have different neighborhood with every vertex in any other cluster  $L'$  (as a result,  $\text{Cand}(L) \cap \text{Cand}(L') = \emptyset$ ). We show such a modification does not affect the approximation ratio. Furthermore, since the candidate sets are defined based on similarity of neighborhoods, they can be constructed by the aforementioned dimension reduction technique, which takes  $O(n \log n/\epsilon^2)$  space.

## 2 Preliminaries

► **Definition 3.** Given  $u \in G^+ = (V, E^+)$ , let  $N(u)$  denote the neighbors of  $u \in G^+$ . Define  $N[u] = N(u) \cup \{u\}$ .  $d(u) = |N(u)|$  is the number of neighbors of  $u \in G^+$ . We use  $N_H(u)$ ,  $N_H[u]$  and  $d_H(u)$  to denote the corresponding quantities in graph  $H$  when  $H \neq G^+$ .

► **Definition 4.** Let  $A$  and  $B$  be sets. The symmetric difference between  $A$  and  $B$ ,  $A \Delta B$ , is defined as  $A \Delta B = (A \setminus B) \cup (B \setminus A)$ .

In general, we say  $A$  and  $B$  are similar if  $|A \Delta B|$  is small.

► **Lemma 5** (Triangle Inequality [35]). Let  $A, B, C$  be sets. Then:  $|A \Delta C| \leq |A \Delta B| + |B \Delta C|$ .

► **Definition 6.** Given any clustering  $\mathcal{C}$  and any vertex  $u$ ,  $\mathcal{C}_u$  is defined to be the cluster of  $\mathcal{C}$  containing  $u$ .

► **Definition 7.** Given clustering  $\mathcal{C}$ , define  $\rho_{\mathcal{C}}(x)$  as the number of disagreements incident to vertex  $x$ . More precisely:

$$\rho_{\mathcal{C}}(x) = |\mathcal{C}_x \setminus N[x]| + |N[x] \setminus \mathcal{C}_x| = |N[x] \Delta \mathcal{C}_x| = |N(x) \Delta \mathcal{C}_x| - 1$$

► **Definition 8.** Given a clustering  $\mathcal{C}$ , the objective value of  $\mathcal{C}$ ,  $\text{obj}(\mathcal{C}) = \max_u \rho_{\mathcal{C}}(u)$ , is defined as the maximum incident disagreements over every vertex.

**The MPC Model.** In the MPC model, computation proceeds in synchronous parallel *rounds* across multiple machines. Each machine has memory  $S$ . At the beginning of the computation, data is arbitrarily partitioned across the machines. During each round, machines process data locally, exchange messages with other machines, and send or receive messages of total size  $S$ . The efficiency of an algorithm in this model is measured by the number of rounds required for the algorithm to terminate and the size  $S$  of the memory available to each machine.

In this paper, we focus on the most practical and challenging regime, also known as the *strictly sublinear regime*, where each machine has  $S = O(n^\delta)$  local memory. Here,  $n$  represents the number of vertices, and  $0 < \delta < 1$  is an arbitrary constant. Under this assumption, the input assigned to each machine and the messages exchanged during any round are of size  $O(n^\delta)$ .

**The Semi-Streaming Model.** In the semi-streaming model, the input is a stream of edges in  $E^+$ . We are allowed to use  $n \text{ polylog } n$  space, where the space complexity is the number of words used and a word consists of  $O(\log n)$  bits. A solution is expected to be output at the end of the stream.

### 3 Algorithm

► **Definition 9.** For any  $0 \leq \eta < 1$ , an  $\eta$ -similarity query  $\Delta_\eta(u, v, t)$  returns:

$$\Delta_\eta(u, v, t) = \begin{cases} 0, & \text{if } |N[u]\Delta N[v]| > (1 + \eta)t. \\ 0 \text{ or } 1, & \text{if } t < |N[u]\Delta N[v]| \leq (1 + \eta)t. \\ 1, & \text{if } |N[u]\Delta N[v]| \leq t. \end{cases}$$

► **Definition 10.** (Shorthand for  $\eta$ -similarity query) For readability, we use  $|N[u]\Delta N[v]| \leq_\eta t$  to denote that an  $\eta$ -similarity query  $\Delta_\eta(u, v, t)$  has been conducted and returned 1. We use  $|N[u]\Delta N[v]| \not\leq_\eta t$  to denote that the query  $\Delta_\eta(u, v, t)$  returned 0.

The parameter  $\eta$  was introduced to accommodate the error induced by an approximate test of whether  $|N[u]\Delta N[v]| \leq 2\phi$ . For the sake of simplicity, we urge the readers to assume  $\eta = 0$  when reading for the first time. When  $\eta = 0$ ,  $|N[u]\Delta N[v]| \leq_\eta 2\phi$  if and only if  $|N[u]\Delta N[v]| \leq 2\phi$ .

► **Definition 11.** Define  $V_{\text{low}} \leftarrow \{w \in V \mid d(w) \leq (3 + \eta)\phi\}$  and  $V_{\text{high}} \leftarrow V \setminus V_{\text{low}}$ , where we call the vertices in  $V_{\text{low}}$  and  $V_{\text{high}}$  as low-degree and high-degree vertices, respectively.

**Algorithm Description.** Algorithm 1 takes two parameters  $\phi$  and  $0 \leq \eta < 1$ , where  $\phi$  is a guess on the upper bound of OPT and  $\eta$  is an error control parameter. The goal of the algorithm is to output a solution of value  $(3 + \eta)\phi$  provided  $\text{OPT} \leq \phi$ . Note that we use  $\eta$  instead of  $\epsilon$  here to indicate that it can be set to zero (at the cost of computing the more expensive exact neighborhood similarity).

The algorithm works as follows: first, we form a clustering of high-degree vertices  $\mathcal{L}$  based on the similarity between the neighborhood of vertices. If two high-degree vertices  $u$  and  $v$  have similar neighborhood (i.e.  $|N[u]\Delta N[v]| \leq_\eta 2\phi$ ) then they will be placed in the same cluster.

Once the high-degree clusters are formed, we go through each cluster  $L_i \in \mathcal{L}$ . For each  $L_i$ , we pick an arbitrary pivot  $u_i \in L_i$ . For each neighbor  $w$  of  $u_i$  that remains unclustered (i.e. in  $V_i$ ), we include it in  $R(u_i)$  if  $w$  and  $u_i$  have similar neighborhoods (i.e.  $|N[w]\Delta N[u_i]| \leq_\eta 2\phi$ ). Then we set our cluster  $C_i$  to be  $L_i \cup R(u_i)$ . Then we update the unclustered vertices to be  $V_{i+1} = V_i \setminus R(u_i)$ .

■ **Algorithm 1** CLUSTERPHI( $G^+ = (V, E^+), \phi, \eta$ ).

**Input:** A graph  $G^+$  and parameters  $\phi$  and  $0 \leq \eta < 1$ .

**Output:** A clustering  $\mathcal{C}$  with  $\text{obj}(\mathcal{C}) \leq (3 + \eta)\phi$  or “OPT  $> \phi$ ”

---

```

1: function CLUSTERPHI( $G^+ = (V, E^+), \phi, \eta$ )
2:    $V_{\text{low}} \leftarrow \{w \mid d(w) \leq (3 + \eta)\phi\}, V_{\text{high}} \leftarrow V \setminus V_{\text{low}}$ 
3:   for  $u, v \in V_{\text{high}}$  do  $\triangleright uv$  not necessarily in  $E^+$ .
4:     if  $|N[u]\Delta N[v]| \leq_{\eta} 2\phi$  then
5:        $E' \leftarrow E' \cup \{uv\}$ .
6:     end if
7:   end for
8:    $V_1 = V_{\text{low}}$ 
9:   Let  $\mathcal{L} = \{L_i\}_{i=1}^{|\mathcal{L}|}$  be the partition formed by the connected components in  $(V_{\text{high}}, E')$ .
10:  for  $i$  from 1 to  $|\mathcal{L}|$  do
11:    Choose any node  $u_i \in L_i$ .
12:    Compute  $R(u_i) = \{w \in V_i \cap N(u_i) \mid |N[w]\Delta N[u_i]| \leq_{\eta/2} 2\phi\}$ 
13:     $C_i \leftarrow L_i \cup R(u_i)$ 
14:     $V_{i+1} \leftarrow V_i \setminus R(u_i)$ 
15:  end for
16:  Let  $\mathcal{C} = \{C_i\}_{i=1}^{|\mathcal{L}|} \cup \bigcup_{v \in V_{|\mathcal{L}|+1}} \{\{v\}\}$ 
17:  if for some  $i$  there exists  $u \in C_i$  such that  $\rho_{\mathcal{C}}(u) > (3 + \eta)\phi$  then
18:    return “OPT  $> \phi$ ”
19:  else
20:    return  $\mathcal{C}$ .
21:  end if
22: end function

```

---

Once we went through every  $L_i$ , there might be still some unclustered low-degree vertices (i.e. those in  $V_{|\mathcal{L}|+1}$ ). For each such vertex, we put it in a singleton cluster. Then, we check if the clustering we have obtained has an objective value at most  $(3 + \eta)\phi$  or not. If it does, then we are done. If not, we conclude that OPT  $> \phi$ , so we would need to set our guess of  $\phi$  larger.

► **Theorem 12.** Suppose that OPT  $\leq \phi$ , Algorithm 1 outputs a clustering  $\mathcal{C}$  with  $\text{obj}(\mathcal{C}) \leq (3 + \eta)\phi$ .

**Proof.** Let  $\mathcal{C}^*$  be an optimal solution so  $\text{obj}(\mathcal{C}^*) \leq \phi$ . In the next subsections, we show the following:

1. (High-Degree Nodes Clustering). For any  $u \in V_{\text{high}}$ ,  $\mathcal{L}_u \cap V_{\text{high}} = \mathcal{C}_u^* \cap V_{\text{high}}$ .
2. (No Stealing on Low-Degree Nodes). For each  $i \in [1, |\mathcal{L}|]$ , let  $C_i^*$  be the cluster in  $\mathcal{C}^*$  such that  $L_i \cap V_{\text{high}} = C_i^* \cap V_{\text{high}}$ . We have  $C_i^* \cap V_i = C_i^* \cap V_{\text{low}}$ . That is, those low-degree vertices in  $C_i^*$  are not taken by other clusters  $C_j$  for  $j < i$ .
3. (Low-Degree Nodes Inclusion). For any  $i \in [1, |\mathcal{L}|]$ ,  $C_i^* \cap N(u_i) \subseteq C_i$ .
4. (Closeness). For any  $i \in [1, |\mathcal{L}|]$ ,  $|N[u_i]\Delta C_i| \leq \phi$  and  $|C_i^* \Delta C_i| \leq \phi$ .

Once we have shown the above, we can see that  $\text{obj}(\mathcal{C}) \leq (3 + \eta)\phi$  as follows. Consider a component  $C_i$  of  $\mathcal{C}$ . If  $C_i$  is a singleton  $\{v\}$  with  $d(v) \leq (3 + \eta)\phi$ , then obviously,  $\rho_{\mathcal{C}}(v) \leq (3 + \eta)\phi$ . Otherwise,  $C_i$  contains some vertex  $x$  with  $d(x) > (3 + \eta)\phi$ . By Item 1, there must exist  $C_i^*$  such that  $C_i \cap V_{\text{high}} = C_i^* \cap V_{\text{high}}$ .



## 41:8 Min-Max Correlation Clustering via Neighborhood Similarity

Let  $v$  be any vertex in  $C_i$ . We will show that  $\rho_C(v) \leq 3\phi$ . Suppose that  $v \in C_i^* \cap C_i$ , we have:

$$\begin{aligned}\rho_C(v) &= |N[v]\Delta C_i| \leq |N[v]\Delta C_i^*| + |C_i^* \Delta C_i| && \text{by Lemma 5} \\ &= \rho_{C^*}(v) + |C_i^* \Delta C_i| \leq \phi + \phi = 2\phi\end{aligned}$$

Otherwise, if  $v \notin C_i^* \cap C_i$  then it must be the case that  $v \in C_i \setminus C_i^*$ . In such a case,  $v$  must be a vertex in  $R(u_i)$  added to  $C_i$  in Line 12 to Line 13. This implies  $|N[v]\Delta N[u_i]| \leq_{\eta/2} 2\phi$  and thus  $|N[v]\Delta N[u_i]| \leq (1 + \eta/2) \cdot 2\phi$ . Therefore:

$$\begin{aligned}\rho_C(v) &= |N[v]\Delta C_i| \leq |N[v]\Delta N[u_i]| + |N[u_i]\Delta C_i| && \text{by Lemma 5} \\ &\leq (1 + \eta/2) \cdot 2\phi + \phi = (3 + \eta)\phi\end{aligned} \quad \blacktriangleleft$$

► **Remark 13.** To get a  $(3 + \epsilon)$ -approximation algorithm,  $O(\log n)$  calls of Algorithm 1 are sufficient, as we can perform a binary search on  $\phi$  in the range of  $[0, n]$  with  $\eta = \epsilon$  and take the solution output by the algorithm with the smallest  $\phi$ . For a 3-approximation algorithm, it can also be achieved within  $O(\log n)$  calls of Algorithm 1 by setting  $\eta = 0$ .

We prove each of the four items in the proof in the following subsections. Note that the no-stealing property of low-degree vertices (Theorem 17 in Section 3.2) is the main structural result, which not only leads to a guarantee on the approximation ratio, but it is also crucial in the design of efficient algorithms in the subsequent sections.

### 3.1 High-Degree Nodes Clustering

Recall that  $\phi$  is our guess of the upper bound of the optimal solution. In this subsection, we show if  $\phi$  is indeed such an upper bound, then there is a unique way to form clustering on high-degree nodes in the optimal solution. Our algorithm clusters the high-degree nodes using the unique way. The following two lemmas are observed by [37].

► **Lemma 14.** Suppose that  $|N[x] \cap N[y]| > t$ . If there is a clustering  $C'$  such that  $x$  and  $y$  are in different clusters, then  $\text{obj}(C') > t/2$ .

**Proof.**

$$t < |N[x] \cap N[y]| \leq |N[x] \cap N[y] \setminus C'_y| + |N[x] \cap N[y] \setminus C'_x| \leq \rho_{C'}(y) + \rho_{C'}(x)$$

Therefore,  $\text{obj}(C') \geq \max(\rho_{C'}(x), \rho_{C'}(y)) \geq (\rho_{C'}(x) + \rho_{C'}(y))/2 > t/2$ . ◀

► **Lemma 15.** Suppose that  $|N[x]\Delta N[y]| > t$ . If there is a clustering  $C'$  such that  $x$  and  $y$  are in the same clusters, then  $\text{obj}(C') > t/2$ .

**Proof.** Let  $C$  be the cluster containing  $x$  and  $y$ . By Lemma 5, we have:

$$t < |N[x]\Delta N[y]| \leq |N[x]\Delta C| + |N[y]\Delta C| = \rho_{C'}(x) + \rho_{C'}(y)$$

Therefore,  $\text{obj}(C') \geq \max(\rho_{C'}(x), \rho_{C'}(y)) \geq (\rho_{C'}(x) + \rho_{C'}(y))/2 > t/2$ . ◀

► **Lemma 16.** Let  $C^*$  be a clustering with  $\text{obj}(C^*) \leq \phi$ . For any  $u \in V_{\text{high}}$ ,  $\mathcal{L}_u \cap V_{\text{high}} = C_u^* \cap V_{\text{high}}$ .



**Proof.** Suppose on the contrary that  $\mathcal{L}_u \cap V_{\text{high}} \neq \mathcal{C}_u^* \cap V_{\text{high}}$ . Then there exists some node  $x$  such that  $x \in (\mathcal{L}_u \cap V_{\text{high}}) \Delta (\mathcal{C}_u^* \cap V_{\text{high}})$ .

Suppose that there exists a node  $x$  such that  $x \in \mathcal{L}_u \cap V_{\text{high}}$  but  $x \notin \mathcal{C}_u^* \cap V_{\text{high}}$ . Then there must exist  $y \in \mathcal{C}_u^*$  such that  $(x, y) \in E'$ . This implies  $|N[x] \Delta N[y]| \leq_\eta 2\phi$  and so  $|N[x] \Delta N[y]| \leq (1 + \eta)2\phi$ . Therefore,

$$\begin{aligned} |N[x] \cap N[y]| &= (d(x) + 1 + d(y) + 1 - |N[x] \Delta N[y]|)/2 \\ &> ((3 + \eta)\phi + 1 + (3 + \eta)\phi + 1 - (1 + \eta)2\phi)/2 \\ &= (6\phi + 2 - 2\phi)/2 = 2\phi + 1 \end{aligned}$$

By Lemma 14,  $\text{obj}(\mathcal{C}^*) > \phi$ , a contradiction.

Otherwise, it must be the case that  $x \in \mathcal{C}_u^* \cap V_{\text{high}}$  but  $x \notin \mathcal{L}_u \cap V_{\text{high}}$ . In this case, there exists  $y \in \mathcal{C}_u^*$  such that  $(x, y) \notin E'$ . This implies that  $|N[x] \Delta N[y]| \not\leq_\eta 2\phi$ , which in turns implies  $|N[x] \Delta N[y]| > 2\phi$ . By Lemma 15,  $\text{obj}(\mathcal{C}^*) > \phi$ , a contradiction.  $\blacktriangleleft$

### 3.2 No Stealing on Low-Degree Nodes

In the algorithm, low-degree nodes (i.e. nodes with degree at most  $(3 + \eta)\phi$ ) are added to the clusters formed by high-degree nodes iteratively. In this subsection, we show that if a low-degree node  $y$  belongs to  $\mathcal{C}_i^*$  for some  $i$ , then it will not be included in  $\mathcal{C}_j$  for  $j < i$ . This implies that  $y \in V_i$ , the candidate set of vertices to be added  $\mathcal{C}_i$ . Then in the next subsection, we show that it will be added to  $\mathcal{C}_i$ .

► **Theorem 17.** *Let  $\mathcal{C}^*$  be a clustering with  $\text{obj}(\mathcal{C}^*) \leq \phi$ . Let  $u$  and  $v$  be vertices of degree greater than  $(3 + \eta)\phi$  where  $\mathcal{C}_u^* \neq \mathcal{C}_v^*$ , and  $w$  be a vertex with  $d(w) \leq (3 + \eta)\phi$ . If  $w \in \mathcal{C}_u^*$  then  $|N[v] \Delta N[w]| > (1 + \eta)2\phi$  and so  $|N[v] \Delta N[w]| \not\leq_\eta 2\phi$ .*

A representative case that illustrates the intuition of why the theorem holds is when  $|N[u] \cap N[v] \cap N[w]| \leq \phi$ , i.e. the three sets have small intersections. Since they have small intersection, together with the fact that  $u$  and  $v$  have degrees greater than  $(3 + \eta)\phi$ , it cannot be the case that both the symmetric differences  $N[u] \Delta N[w]$  and  $N[v] \Delta N[w]$  are small, as illustrated in Figure 1. We will refer the case that  $|N[u] \cap N[v] \cap N[w]| \leq \phi$  as the *easy case*.

When the three sets have a large intersection, with a more sophisticated argument on the relations among  $N[u]$ ,  $N[v]$ , and  $N[w]$ , it can also be shown that  $N[w]$  and  $N[u]$  will not intersect a lot. We will refer to the case that  $|N[u] \cap N[v] \cap N[w]| > \phi$  as the *hard case*.

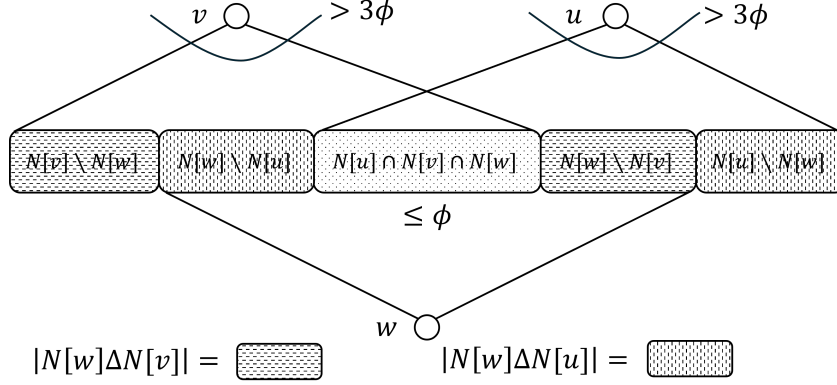
#### 3.2.1 The easy case

We will first show the theorem for the easy case as follows.

► **Lemma 18.** *Let  $\mathcal{C}^*$  be a clustering with  $\text{obj}(\mathcal{C}^*) \leq \phi$ . Let  $u$  and  $v$  be vertices of degree greater than  $(3 + \eta)\phi$  where  $\mathcal{C}_u^* \neq \mathcal{C}_v^*$ , and  $w$  be a vertex with  $d(w) \leq (3 + \eta)\phi$ . If  $w \in \mathcal{C}_u^*$  and  $|N[v] \cap N[u] \cap N[w]| \leq \phi$ , then  $|N[v] \Delta N[w]| > (1 + \eta)2\phi$  and so  $|N[v] \Delta N[w]| \not\leq_\eta 2\phi$ .*

**Proof.** We will show that  $|N[v] \Delta N[w]| + |N[u] \Delta N[w]|$  is greater than  $(1 + \eta)4\phi$ . Figure 1 gives a high level illustration of why this should be true. To show this formally, first note that:

$$\begin{aligned} &|N[v] \cap N[w] \cap N[u]| + |N[w] \setminus N[u]| + |N[v] \setminus N[w]| \\ &\geq |N[v] \cap N[w] \cap N[u]| + |(N[v] \cap N[w]) \setminus N[u]| + |N[v] \setminus N[w]| \\ &= |N[v] \cap N[w]| + |N[v] \setminus N[w]| = |N[v]| \end{aligned}$$



■ **Figure 1** A pictorial illustration of the proof of Lemma 18 when  $\eta = 0$ . It shows that if  $N[u] \cap N[v] \cap N[w]$  is small, then either  $N[w] \Delta N[v]$  or  $N[w] \Delta N[u]$  has to be large, since  $|N[u]|$  and  $|N[v]|$  are more than  $3\phi$ .

Re-arranging, we have:

$$\begin{aligned} |N[w] \setminus N[u]| + |N[v] \setminus N[w]| &\geq |N[v]| - |N[v] \cap N[w] \cap N[u]| \\ &> (3 + \eta)\phi - \phi = (2 + \eta)\phi \end{aligned} \quad (1)$$

By the same reasoning, we have

$$|N[w] \setminus N[v]| + |N[u] \setminus N[w]| > (2 + \eta)\phi \quad (2)$$

Now consider the following:

$$\begin{aligned} &|N[v] \Delta N[w]| + |N[u] \Delta N[w]| \\ &= |N[v] \setminus N[w]| + |N[w] \setminus N[v]| + |N[u] \setminus N[w]| + |N[w] \setminus N[u]| \\ &> (2 + \eta)\phi + (2 + \eta)\phi \quad \text{by (1) and (2)} \\ &= (4 + 2\eta)\phi \end{aligned}$$

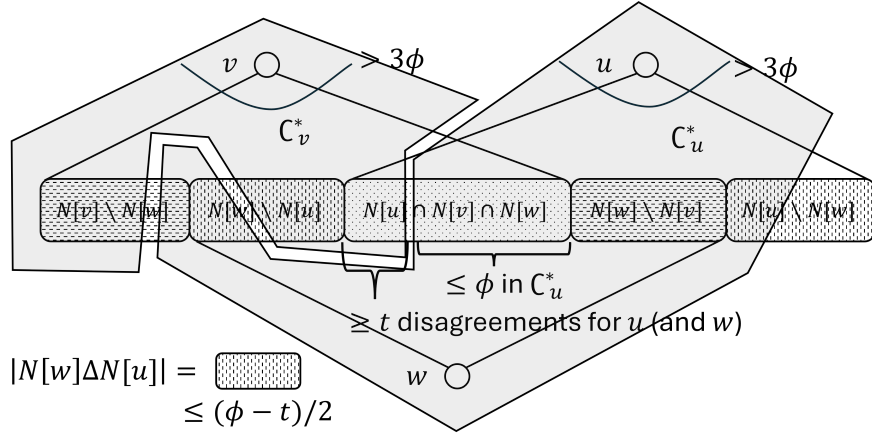
Therefore, assume to the contrary that  $|N[v] \Delta N[w]| \leq_\eta 2\phi$  and so  $|N[v] \Delta N[w]| \leq (1 + \eta)2\phi$ . Then, it must be the case that  $|N[u] \Delta N[w]| > (4 + 2\eta)\phi - |N[v] \Delta N[w]| \geq 2\phi$ . By the fact that  $w \in \mathcal{C}_v^*$  and Lemma 15,  $\text{obj}(\mathcal{C}^*) > \phi$ , a contradiction. ◀

### 3.2.2 The hard case

Now we consider the case where  $|N[w] \cap N[u] \cap N[v]| > \phi$ . To illustrate the idea of proof, suppose that  $|N[w] \cap N[u] \cap N[v]| = \phi + t$  for some  $t > 0$ . If we follow the same argument as Lemma 18, we would only be able to show that  $|N[w] \Delta N[v]| + |N[w] \Delta N[u]| > (4 + 2\eta)\phi - 2t$  as opposed to  $(4 + 2\eta)\phi$ .

However, if this is the case, we can show that  $|N[w] \Delta N[u]| \leq 2(\phi - t)$ , as stated in Lemma 19. This would make  $|N[w] \Delta N[v]| > (1 + \eta)2\phi$ .

The high-level idea of the proof of Lemma 19 is illustrated in Figure 2. First, we observe that at most  $\phi$  vertices in  $|N[u] \cap N[v] \cap N[w]|$  can be contained in  $\mathcal{C}_u^*$ ; otherwise the vertex  $v$  would have disagreements greater than  $\phi$ . This implies the contribution of disagreement from  $N[u] \cap N[w]$  to  $u$  (and to  $w$ ) is already at least  $t$ . If the symmetric difference  $|N[u] \Delta N[w]|$  is too large, i.e. larger than  $2(\phi - t)$ , then it would force the disagreements of either  $u$  or  $w$  to be too large.



■ **Figure 2** A pictorial illustration of the proof of Lemma 19 when  $\eta = 0$ . This means that  $N[u] \cap N[v] \cap N[w]$  must contribute at least  $t$  disagreements to both  $u$  and  $v$ . Additionally, each element in  $|N[u] \Delta N[w]|$  contributes a disagreement to at least one of  $u$  or  $w$ , so  $|N[u] \Delta N[w]|$  cannot exceed  $(\phi - t)/2$ ; otherwise, the disagreement of  $u$  or  $w$  would exceed  $\phi$ .

► **Lemma 19.** *Let  $C^*$  be a clustering with  $\text{obj}(C^*) \leq \phi$ . Let  $u$  and  $v$  be vertices of degree greater than  $(3 + \eta)\phi$  where  $C_u^* \neq C_v^*$ , and  $w \in C_u^*$  be a vertex with  $d(w) \leq (3 + \eta)\phi$ . Suppose that  $|N[w] \cap N[v] \cap N[u]| = \phi + t$  for some  $t > 0$ . Then,  $|N[w] \Delta N[u]| \leq 2(\phi - t)$ .*

**Proof.** Let  $S = N[w] \cap N[v] \cap N[u]$ . First we claim that at most  $\phi$  vertices in  $S$  can be in  $C_u^*$ . This in turns implies that vertices in  $S$  contributes at least  $t$  disagreements to vertex  $u$  and vertex  $w$ . To see why this holds, suppose to the contrary that  $|S \cap C_u^*| > \phi$ . Since  $S \subseteq N[v]$  and  $C_v^* \neq C_u^*$ , it must be the case that  $\rho_{C^*}(v) \geq |N[v] \setminus C_v^*| \geq |S \setminus C_v^*| \geq |S \cap C_u^*| > \phi$ , a contradiction. Therefore, we conclude that:

$$|S \setminus C_u^*| \geq t \quad (3)$$

Now note that the disagreements of  $u$  and  $w$  are at least:

$$\begin{aligned} \rho_{C^*}(u) &\geq |S \setminus C_u^*| + |(N[w] \setminus N[u]) \cap C_u^*| + |(N[u] \setminus N[w]) \setminus C_u^*| \\ \rho_{C^*}(w) &\geq |S \setminus C_u^*| + |(N[u] \setminus N[w]) \cap C_u^*| + |(N[w] \setminus N[u]) \setminus C_u^*| \end{aligned}$$

Using the fact  $\rho_{C^*}(u), \rho_{C^*}(w) \leq \phi$  and summing up the above two inequalities, we have:

$$\begin{aligned} 2\phi &\geq \rho_{C^*}(u) + \rho_{C^*}(w) \\ &\geq 2|S \setminus C_u^*| + |(N[w] \setminus N[u]) \cap C_u^*| + |(N[u] \setminus N[w]) \setminus C_u^*| + \\ &\quad |(N[u] \setminus N[w]) \cap C_u^*| + |(N[w] \setminus N[u]) \setminus C_u^*| \\ &= 2|S \setminus C_u^*| + |(N[w] \Delta N[u]) \cap C_u^*| + |(N[u] \Delta N[w]) \setminus C_u^*| \\ &= 2|S \setminus C_u^*| + |(N[w] \Delta N[u])| \\ &\geq 2t + |(N[w] \Delta N[u])| \end{aligned}$$

Therefore,  $|N[w] \Delta N[v]| \leq 2(\phi - t)$ . ◀

► **Lemma 20.** *Let  $C^*$  be a clustering with  $\text{obj}(C^*) \leq \phi$ . Let  $u$  and  $v$  be vertices of degree greater than  $(3 + \eta)\phi$  where  $C_u^* \neq C_v^*$ , and  $w$  be a vertex with  $\deg(w) \leq (3 + \eta)\phi$ . If  $w \in C_u^*$  and  $|N[v] \cap N[u] \cap N[w]| > \phi$ , then  $|N[v] \Delta N[w]| > (1 + \eta)2\phi$  and so  $|N[v] \Delta N[w]| \not\leq_\eta 2\phi$ .*

**Proof.** By (1) in the proof of Lemma 18, we have:

$$\begin{aligned} |N[w] \setminus N[u]| + |N[v] \setminus N[w]| &\geq |N[v]| - |N[v] \cap N[w] \cap N[u]| \\ &> (3 + \eta)\phi - (\phi + t) = (2 + \eta)\phi - t \\ |N[w] \setminus N[v]| + |N[u] \setminus N[w]| &\geq |N[u]| - |N[v] \cap N[w] \cap N[u]| \\ &> (3 + \eta)\phi - (\phi + t) = (2 + \eta)\phi - t \end{aligned}$$

Therefore,

$$\begin{aligned} |N[v] \Delta N[w]| + |N[u] \Delta N[w]| &= |N[v] \setminus N[w]| + |N[w] \setminus N[v]| \\ &\quad + |N[u] \setminus N[w]| + |N[w] \setminus N[u]| \\ &> (4 + 2\eta)\phi - 2t \end{aligned}$$

By Lemma 19, we have  $|N[u] \Delta N[w]| \geq 2(\phi - t)$ , therefore:

$$|N[v] \Delta N[w]| > (4 + 2\eta)\phi - 2t - 2(\phi - t) = (1 + \eta)2\phi$$

Hence,  $|N[v] \Delta N[w]| \not\leq_{\eta} 2\phi$ . ◀

Lemma 18 and Lemma 20 complete the proof of Theorem 17. Theorem 17 immediately implies the following:

► **Corollary 21.** *Let  $C_i^*$  be the cluster in  $\mathcal{C}^*$  such that  $L_i \cap V_{\text{high}} = C_i^* \cap V_{\text{high}}$ . For any  $i$ ,  $C_i^* \cap V_i = C_i^* \cap V_{\text{low}}$ .*

### 3.3 Low-Degree Nodes Inclusion

Here, we show that all the vertices in  $C_i^*$  have similar neighborhoods with  $u_i$ . As a result, if they are neighbors of  $u_i$ , they will be added to  $C_i$ .

► **Lemma 22.** *Let  $C_i^*$  be the cluster in  $\mathcal{C}^*$  such that  $L_i \cap V_{\text{high}} = C_i^* \cap V_{\text{high}}$ . For any  $i$ ,  $C_i^* \cap N(u_i) \subseteq C_i$ .*

**Proof.** If  $w \in C_i^* \cap V_{\text{high}}$ , then  $w \in C_i$  because  $C_i = L_i \cup R(u_i)$  and  $L_i \cap V_{\text{high}} = C_i^* \cap V_{\text{high}}$  by Lemma 16. Now it remains to consider the case  $w \in C_i^* \cap V_{\text{low}} \cap N(u_i)$ . It suffices for us to show that  $w \in R(u_i)$ , where  $R(u_i) = \{w \in V_i \cap N(u_i) \mid |N[w] \Delta N[u_i]| \leq_{\eta} 2\phi\}$ , as  $R(u_i)$  will be added to  $C_i$ .

By Corollary 21, we have  $w \in C_i^* \cap V_i \cap N(u_i)$ , which means  $w$  is not going to be stolen by other clusters. To show that  $w \in C_i$ , now it suffices to show that  $|N[w] \Delta N[u_i]| \leq_{\eta} 2\phi$  always holds. This is indeed true, because:

$$\begin{aligned} |N[w] \Delta N[u_i]| &\leq |N[w] \Delta C_i^*| + |C_i^* \Delta N[u_i]| && \text{by Lemma 5} \\ &\leq \phi + \phi \leq 2\phi && u_i, w \in C_i^*, \rho_{\mathcal{C}^*}(w), \rho_{\mathcal{C}^*}(u_i) \leq \phi \quad \blacktriangleleft \end{aligned}$$

### 3.4 Closeness

In the following, we will show that the cluster we constructed  $C_i$  will be similar to  $C_i^*$  and  $N[u_i]$ . Intuitively, this holds because the low-degree part of  $C_i$  will be sandwiched between the low-degree part of  $C_i^* \cap N[u_i]$  and  $N[u_i]$ , i.e.  $C_i^* \cap N[u_i] \cap V_{\text{low}} \subseteq C_i \cap V_{\text{low}} \subseteq N[u_i] \cap V_{\text{low}}$ . Also, we know that  $N[u_i]$  and  $C_i^*$  are close (i.e.  $|N[u_i] \Delta C_i^*| \leq \phi$ ), so somehow  $C_i$  cannot be too far away from  $C_i^*$  and  $N[u_i]$ . Note that the high-degree parts of  $C_i$  and  $C_i^*$  coincide.

► **Lemma 23.** *Let  $C_i^*$  be the cluster in  $\mathcal{C}^*$  such that  $L_i \cap V_{\text{high}} = C_i^* \cap V_{\text{high}}$ . For any  $i$ ,  $|N[u_i] \Delta C_i| \leq \phi$  and  $|C_i^* \Delta C_i| \leq \phi$ .*

**Proof.** We first show that  $|N[u_i] \Delta C_i| \leq \phi$ .

$$\begin{aligned} |V_{\text{low}} \cap (N[u_i] \Delta C_i)| &= |V_{\text{low}} \cap (N[u_i] \setminus C_i)| && (V_{\text{low}} \cap C_i) \subseteq N[u_i] \\ &\leq |V_{\text{low}} \cap (N[u_i] \setminus C_i^*)| && \text{by Lemma 22} \\ &\leq |V_{\text{low}} \cap (N[u_i] \Delta C_i^*)| \end{aligned} \quad (4)$$

Moreover, since  $C_i^* \cap V_{\text{high}} = C_i \cap V_{\text{high}}$ , it must be the case that

$$|(N[u_i] \Delta C_i) \cap V_{\text{high}}| = |(N[u_i] \Delta C_i^*) \cap V_{\text{high}}| \quad (5)$$

Therefore,

$$\begin{aligned} |N[u_i] \Delta C_i| &= |(N[u_i] \Delta C_i) \cap V_{\text{low}}| + |(N[u_i] \Delta C_i) \cap V_{\text{high}}| \\ &\leq |(N[u_i] \Delta C_i^*) \cap V_{\text{low}}| + |(N[u_i] \Delta C_i^*) \cap V_{\text{high}}| && \text{by (4) and (5)} \\ &= |N[u_i] \Delta C_i^*| \leq \phi && u_i \in C_i^* \text{ and } \rho_{C^*}(u_i) \leq \phi \end{aligned}$$

Now we show that  $|C_i^* \Delta C_i| \leq \phi$ .

$$\begin{aligned} |C_i^* \Delta C_i| &= |V_{\text{low}} \cap (C_i^* \Delta C_i)| && C_i^* \cap V_{\text{high}} = C_i \cap V_{\text{high}} \\ &= |V_{\text{low}} \cap (C_i^* \setminus C_i)| + |V_{\text{low}} \cap (C_i \setminus C_i^*)| \\ &= |V_{\text{low}} \cap (C_i^* \cap N[u_i] \setminus C_i)| + |V_{\text{low}} \cap (C_i^* \setminus N[u_i] \setminus C_i)| \\ &\quad + |V_{\text{low}} \cap (C_i \setminus C_i^*)| \\ &= |V_{\text{low}} \cap (C_i^* \cap N[u_i] \setminus C_i)| + |V_{\text{low}} \cap (C_i^* \setminus N[u_i])| \\ &\quad + |V_{\text{low}} \cap (C_i \setminus C_i^*)| && V_{\text{low}} \cap C_i \subseteq V_{\text{low}} \cap N[u_i] \\ &= |V_{\text{low}} \cap (C_i^* \setminus N[u_i])| + |V_{\text{low}} \cap (C_i \setminus C_i^*)| && C_i^* \cap N[u_i] \subseteq C_i \\ &\leq |V_{\text{low}} \cap (C_i^* \setminus N[u_i])| + |V_{\text{low}} \cap (N[u_i] \setminus C_i^*)| && V_{\text{low}} \cap C_i \subseteq V_{\text{low}} \cap N[u_i] \\ &= |V_{\text{low}} \cap (C_i^* \Delta N[u_i])| \leq |C_i^* \Delta N[u_i]| \leq \phi \end{aligned} \quad \blacktriangleleft$$

## 4 Efficient Algorithms for the Sequential and MPC Models

To make the algorithm efficient in the sequential and MPC settings, we need to address several challenges. First, we need to compute the graph  $E^l$  for high-degree nodes. Second, we need to be able to conduct approximate neighborhood similarity queries efficiently. We will address these issues one by one.

### 4.1 Computing the Clustering for High-Degree Nodes

In this section, we show that although  $E^l$  may be significantly larger than  $E^+$ , it is sufficient to make only  $O(|E^+|) = O(m)$  neighborhood similarity queries to identify the high-degree clusters. Algorithm 2 describes how to form a clustering  $\mathcal{F}$  for high-degree vertices by conducting  $O(m)$  similarity queries.

■ **Algorithm 2** HIGHDEGREECLUSTERING( $G^+ = (V, E^+), \phi, \eta$ ).

**Input:** A graph  $G^+$ , a parameter  $\phi$ , a parameter  $0 \leq \eta < 1$ .

**Output:** A clustering  $\mathcal{F} = \{L_i\}_{i=1}^{|\mathcal{L}|}$  of  $V_{\text{high}}$  such that  $\mathcal{F} = \mathcal{L}$ .

- 
- 1: **function** HIGHDEGREECLUSTERING( $G^+ = (V, E^+), \phi, \eta$ )
  - 2:   Let  $G_{\text{sim}} = (V, E_{\text{sim}})$ , where  $E_{\text{sim}} = \{uv \in E^+ \mid |N[u] \Delta N[v]| \leq_{\eta} 2\phi\}$ .
  - 3:   Assign each vertex  $V$  with a unique ID,  $\text{ID}(v)$ .
  - 4:   Every vertex  $v \in V$  sets  $\min(v) \leftarrow \min_{x \in N_{G_{\text{sim}}}[v] \cap V_{\text{high}}} \text{ID}(x)$
  - 5:   Every vertex  $v \in V$  sends a token of value  $\min(v)$  to the neighbor of  $v$  in  $G_{\text{sim}}$ .
  - 6:   If a vertex  $v \in V_{\text{high}}$  receives at least  $\phi + 1$  tokens of the same value, set  $\text{cluster}(v)$  to be the minimum such value.
  - 7:   Let  $\mathcal{F}$  be a clustering where each cluster is formed by vertices with the same  $\text{cluster}$  values.
  - 8: **end function**
- 

**Algorithm Description.** In Algorithm 2, we first perform  $O(m)$  neighborhood queries to construct  $G_{\text{sim}} = (V, E_{\text{sim}})$ , where  $E_{\text{sim}} \subseteq E^+$  consists of those endpoints who have similar neighborhoods. Then we assign every vertex  $u$  with a unique identifier,  $\text{ID}(u)$ . By sending the ID of every high-degree vertex to its neighbors in  $G_{\text{sim}}$ , every vertex  $u$  can learn ID of the high-degree vertex with the smallest ID in its closed neighborhood in  $G_{\text{sim}}$  (Line 4) and then set  $\min(u)$  to be such an ID. Now, every vertex  $u$  (including low-degree vertices) sends a token with value  $\min(u)$  to all the neighbors of  $u$  in  $G_{\text{sim}}$ . Then, if a vertex  $u$  received at least  $\phi + 1$  tokens of the same value, it will set its cluster ID,  $\text{cluster}(u)$ , to be the minimum value that occurs at least  $\phi + 1$  times. Finally, we assign all high-degree vertices with the same cluster ID to be in the same cluster.

In Lemma 24, we show that if  $\text{OPT} \leq \phi$ , then the output  $\mathcal{F}$  is exactly  $\mathcal{L}$ . The proof is based on showing that if two high-degree vertices  $u, v$  are in the same cluster in  $\mathcal{C}^*$ , then there will be at least  $\phi + 1$  disjoint paths of length two in  $G_{\text{sim}}$ .

► **Lemma 24.** *efficienthighdeg* Suppose that  $\text{OPT} \leq \phi$ . Algorithm 2 outputs a clustering  $\mathcal{F}$  such that  $\mathcal{F} = \mathcal{L}$ .

**Proof.** Let  $\mathcal{C}^*$  be a clustering with  $\text{obj}(\mathcal{C}^*) \leq \phi$ . Let  $u$  be a vertex in  $V_{\text{high}}$ . Let  $u_*$  be the vertex with the minimum ID in  $\mathcal{L}_u$ . We will show that

1. Every vertex  $x \in \mathcal{L}_u \cap V_{\text{high}}$  will receive at least  $\phi + 1$  tokens of values  $\text{ID}(u_*)$  in Line 6.
2. In addition, if  $x$  received at least  $\phi + 1$  tokens of value  $T$ , then  $T \geq \text{ID}(u_*)$ .

Once these two points are established, every vertex  $x \in \mathcal{L}_u \cap V_{\text{high}}$  will be assigned  $\text{cluster}(x) = \text{ID}(u_*)$ . Consequently, this ensures that  $\mathcal{F}_u = \mathcal{L}_u$ . Now we start with the first point, since  $x, u_* \in \mathcal{C}_u^*$ ,  $|N[u_*] \Delta N[x]| \leq 2\phi$ ; for otherwise  $\text{obj}(\mathcal{C}^*) > \phi$  by Lemma 15. This implies:

$$\begin{aligned} |N[u_*] \cap N[x]| &= (|N[x]| + |N[u_*]| - |N[u_*] \Delta N[x]|)/2 \\ &\geq (3\phi + 2 + 3\phi + 2 - 2\phi)/2 = 2\phi + 2 \end{aligned}$$

Also, since  $u_* \in \mathcal{C}_u^*$  and by definition of  $\rho_{\mathcal{C}^*}(u_*)$ , we have:

$$\phi \geq \rho_{\mathcal{C}^*}(u_*) \geq |N[u_*]| - |(\mathcal{C}_u^* \cap N[u_*])| \quad (6)$$

Therefore,

$$\begin{aligned}
|(\mathcal{C}_u^* \cap N[u_*]) \cap N[x]| &= |N[x] \cap N[u_*]| - |N[x] \cap (N[u_*] \setminus (\mathcal{C}_u^* \cap N[u_*]))| \\
&\geq |N[x] \cap N[u_*]| - |N[u_*] \setminus (\mathcal{C}_u^* \cap N[u_*])| \\
&\geq (2\phi + 2) - \phi && \text{by (6)} \\
&\geq \phi + 2
\end{aligned}$$

Hence:

$$|(\mathcal{C}_u^* \cap N[u_*]) \cap N(x)| \geq |(\mathcal{C}_u^* \cap N[u_*]) \cap N[x]| - 1 \geq \phi + 1$$

Note that all vertices in  $(\mathcal{C}_u^* \cap N[u_*])$  must have their  $\min(\cdot)$  values equal to  $\text{ID}(u_*)$ . This is because if  $v \in (\mathcal{C}_u^* \cap N[u_*])$  and  $v \in V_{\text{high}}$ , then  $(N_{G_{\text{sim}}}(v) \cap V_{\text{high}}) \subseteq \mathcal{C}_u^*$ . If  $v \in (\mathcal{C}_u^* \cap N[u_*])$  and  $v \notin V_{\text{high}}$ , then it must be the case that  $v \in N_{G_{\text{sim}}}(u_*)$ . Moreover,  $N_{G_{\text{sim}}}[v] \cap V_{\text{high}}$  cannot contain any cluster nodes outside  $\mathcal{C}_u^*$  by Theorem 17. Thus,  $\min(v) = \text{ID}(u_*)$ . This implies  $x$  will receive at least  $\phi + 1$  tokens with value  $\text{ID}(u_*)$ .

Next we show that if there is a value  $T$  such that  $x$  receives from at least  $\phi + 1$  tokens, then  $T \geq \text{ID}(u_*)$ . Suppose to the contrary that  $T < \text{ID}(u_*)$ . First note that if  $v \in N(x)$  has  $\min(v) = T$ , then it cannot be the case that  $v \in V_{\text{high}}$ . Otherwise,  $u_*$ ,  $v$ , and a vertex whose ID is  $T$ , would all be in  $\mathcal{C}_u^*$ , which contradict with the fact  $\text{ID}(u_*)$  is the smallest ID in  $\mathcal{L}_u = \mathcal{C}_u^* \cap V_{\text{high}}$ . Therefore, it must be the case that vertices  $v \in N(x)$  with  $\min(v) = T$  are all in  $V_{\text{low}}$ .

Now note that if  $v \in \mathcal{C}_u^*$  then by Theorem 17,  $v$  can only be connected to vertices in  $\mathcal{C}_u^*$  in  $G_{\text{sim}}$ . In this case,  $\min(v) \geq \text{ID}(u_*)$  so  $\min(v) \neq T$  by the assumption of  $\text{ID}(u_*)$ . So the only possibility for  $v$  to have  $\min(v) = T$  is when  $v \notin \mathcal{C}_u^*$ . However, as  $\rho_{\mathcal{C}^*}(x) \leq \phi$ , there are at most  $\phi$  neighbors of  $x$  not in  $\mathcal{C}_x^*$ . This implies  $x$  will receive less than  $\phi + 1$  tokens whose value equals to  $T$ .  $\blacktriangleleft$

## 4.2 Approximate Neighborhood Similarity Testing by Random Projection

We show that for  $0 < \eta < 1$ , w.h.p. the queries  $\Delta_\eta(x, y, 2\phi)$  for all  $xy \in E^+$  can be answered in  $O(m \log n / \eta^2)$  time and thus  $G_{\text{sim}}$  can be constructed in  $O(m \log n / \eta^2)$  time.

► **Definition 25.** Given a vertex  $x$ , let  $\vec{N}[x] \subseteq \{0, 1\}^n$  denote the characteristic vector of  $N[x]$ .

► **Lemma 26.** Given  $\epsilon > 0$ , let  $k = C \cdot (\log n / \epsilon^2)$  for some large enough constant  $C > 0$ . Let  $A$  be a  $k \times n$  matrix where each entry is drawn from  $\{-1, +1\}$  uniformly at random. W.h.p. for every two vertices  $x$  and  $y$ , we have:

$$(1 - \epsilon) \cdot |N[x] \Delta N[y]| \leq \|A \cdot \vec{N}[x] - A \cdot \vec{N}[y]\|_2^2 / k \leq (1 + \epsilon) \cdot |N[x] \Delta N[y]|$$

**Proof.** By the Johnson-Lindenstrauss lemma [1, 38], w.h.p. for every  $x$  and  $y$ , we have:

$$(1 - \epsilon) \cdot \|\vec{N}[x] - \vec{N}[y]\|_2^2 \leq \|A \cdot \vec{N}[x] - A \cdot \vec{N}[y]\|_2^2 / k \leq (1 + \epsilon) \cdot \|\vec{N}[x] - \vec{N}[y]\|_2^2$$

The lemma follows by observing that  $|N[x] \Delta N[y]| = \|\vec{N}[x] - \vec{N}[y]\|_2^2$ .  $\blacktriangleleft$

► **Lemma 27.** Let  $\epsilon = \Theta(\eta)$  be such that  $(1 + \eta) = (1 + \epsilon) / (1 - \epsilon)$ ,  $k = O(\log n / \epsilon^2)$ , and set:

$$\Delta_\eta(x, y, t) = \begin{cases} 0 & \text{if } \|A \cdot \vec{N}[x] - A \cdot \vec{N}[y]\|_2^2 / ((1 + \epsilon)k) > t \\ 1 & \text{otherwise} \end{cases}$$

W.h.p. the above implementation returns a correct answer for  $\Delta_\eta(x, y, t)$ .



**Proof.** It suffices to show that if  $|N[x]\Delta N[y]| \leq t$  then w.h.p. we will set  $\Delta_\eta(x, y, t)$  to be 0 and if  $|N[x]\Delta N[y]| > (1 + \eta)t$ , then w.h.p. we will set  $\Delta_\eta(x, y, t)$  to be 1.

If  $|N[x]\Delta N[y]| \leq t$ , then w.h.p.

$$\|A \cdot \vec{N}[x] - A \cdot \vec{N}[y]\|_2^2 / ((1 + \epsilon)k) \leq |N[x]\Delta N[y]| \leq t$$

Thus,  $\Delta_\eta(x, y, t)$  returns 1.

On the other hand, if  $|N[x]\Delta N[y]| > (1 + \eta)t$  then w.h.p.

$$\|A \cdot \vec{N}[x] - A \cdot \vec{N}[y]\|_2^2 / ((1 + \epsilon)k) \geq \frac{1 - \epsilon}{1 + \epsilon} \cdot (|N[x]\Delta N[y]|) > \frac{1 - \epsilon}{1 + \epsilon} (1 + \eta) \cdot t = t$$

Thus,  $\Delta_\eta(x, y, t)$  returns 0. ◀

We have presented the key ingredients that lead to efficient sequential and MPC algorithms. The details of their implementations are presented in the full version [17]. For the semi-streaming algorithm, different challenges arise. The necessary modifications and details are also included in the full version.

---

## References

- 1 Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003. doi:10.1016/S0022-0000(03)00025-4.
- 2 Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. *Algorithmica*, 83(7):1980–2017, 2021. doi:10.1007/S00453-021-00816-9.
- 3 Nir Ailon, Moses Charikar, and Alanthan Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5):23:1–23:27, 2008. doi:10.1145/1411509.1411513.
- 4 Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for  $(\Delta + 1)$  vertex coloring. In *Proc. 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 767–786, 2019. doi:10.1137/1.9781611975482.48.
- 5 Sepehr Assadi, Sanjeev Khanna, and Aaron Putterman. Correlation clustering and (de)sparsification: Graph sketches can match classical algorithms. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC)*, pages 417–428, 2025. doi:10.1145/3717823.3718194.
- 6 Sepehr Assadi, Pankaj Kumar, and Parth Mittal. Brooks’ theorem in graph streams: A single-pass semi-streaming algorithm for  $\Delta$ -coloring. *TheoretCS*, 2, 2023. doi:10.46298/THEORETICS.23.9.
- 7 Sepehr Assadi and Chen Wang. Sublinear time and space algorithms for correlation clustering via sparse-dense decompositions. In *13th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 215, pages 10:1–10:20, 2022. doi:10.4230/LIPICS.ITCS.2022.10.
- 8 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. doi:10.1023/B:MACH.0000033116.57574.95.
- 9 S. Behnezhad, M. Charikar, W. Ma, and L. Tan. Almost 3-approximate correlation clustering in constant rounds. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 720–731, 2022.
- 10 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.
- 11 Mélanie Cambus, Davin Choo, Havu Miikonen, and Jara Uitto. Massively parallel correlation clustering in bounded arboricity graphs. In Seth Gilbert, editor, *35th International Symposium on Distributed Computing (DISC)*, volume 209 of *LIPICs*, pages 15:1–15:18, 2021. doi:10.4230/LIPICS.DISC.2021.15.

- 12 Mélanie Cambus, Fabian Kuhn, Etna Lindy, Shreyas Pai, and Jara Uitto. A single-pass semi-streaming algorithm for  $(3 + \varepsilon)$ -approximate correlation clustering. In *Proc. 35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024. to appear.
- 13 Nairen Cao, Vincent Cohen-Addad, Euiwoong Lee, Shi Li, David Rasmussen Lolck, Alantha Newman, Mikkel Thorup, Lukas Vogl, Shuyi Yan, and Hanwen Zhang. Solving the correlation cluster lp in sublinear time. In *Proc. 57th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1154–1165, 2025. doi:10.1145/3717823.3718181.
- 14 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 (STOC)*, pages 1605–1616, 2024. doi:10.1145/3618260.3649749.
- 15 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.
- 16 Nairen Cao, Shi Li, and Jia Ye. Simultaneously approximating all norms for massively parallel correlation clustering. In *Proc. ICALP*, 2025. doi:10.4230/LIPIcs.ICALP.2025.40.
- 17 Nairen Cao, Steven Roche, and Hsin-Hao Su. Min-max correlation clustering via neighborhood similarity. *arXiv preprint arXiv:2502.12519*, 2025. doi:10.48550/arXiv.2502.12519.
- 18 Sayak Chakrabarty and Konstantin Makarychev. Single-pass pivot algorithm for correlation clustering. keep it simple! *arXiv preprint arXiv:2305.13560*, 2023. doi:10.48550/arXiv.2305.13560.
- 19 Yi-Jun Chang, Wenzheng Li, and Seth Pettie. An optimal distributed  $(\Delta+1)$ -coloring algorithm? In *Proc. 50th ACM Symposium on Theory of Computing (STOC)*, pages 445–456, 2018. doi:10.1145/3188745.3188964.
- 20 Moses Charikar, Neha Gupta, and Roy Schwartz. Local guarantees in graph cuts and clustering. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 136–147, 2017. doi:10.1007/978-3-319-59250-3\_12.
- 21 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.
- 22 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.
- 23 Flavio Chierichetti, Nilesch Dalvi, and Ravi Kumar. Correlation clustering in mapreduce. In *Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 641–650, 2014. doi:10.1145/2623330.2623743.
- 24 Vincent Cohen-Addad, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub Tarnawski. Correlation clustering in constant many parallel rounds. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 2069–2078, 2021. URL: <http://proceedings.mlr.press/v139/cohen-addad21b.html>.
- 25 Vincent Cohen-Addad, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub Tarnawski. Correlation clustering in constant many parallel rounds. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2069–2078. PMLR, 2021. URL: <http://proceedings.mlr.press/v139/cohen-addad21b.html>.
- 26 Vincent Cohen-Addad, Euiwoong Lee, Shi Li, and Alantha Newman. Handling correlated rounding error via preclustering: A 1.73-approximation for correlation clustering. In *64rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023. to appear. doi:10.1109/FOCS57990.2023.00065.

- 27 Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. Correlation clustering with sherali-adams. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 651–661. IEEE, 2022. doi:10.1109/FOCS54457.2022.00068.
- 28 Vincent Cohen-Addad, David Rasmussen Lolck, Marcin Pilipczuk, Mikkel Thorup, Shuyi Yan, and Hanwen Zhang. Combinatorial correlation clustering. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1617–1628, 2024. doi:10.1145/3618260.3649712.
- 29 Sami Davies, Benjamin Moseley, and Heather Newman. Fast combinatorial algorithms for min max correlation clustering. In *International Conference on Machine Learning*, pages 7205–7230. PMLR, 2023. URL: <https://proceedings.mlr.press/v202/davies23a.html>.
- 30 Manuela Fischer, Magnús M. Halldórsson, and Yannic Maus. Fast distributed brooks’ theorem. In *Proc. ACM-SIAM Symposium on Discrete Algorithms SODA*, pages 2567–2588, 2023. doi:10.1137/1.9781611977554.CH98.
- 31 Manuela Fischer and Andreas Noever. Tight analysis of parallel randomized greedy MIS. *ACM Trans. Algorithms*, 16(1):6:1–6:13, 2020. doi:10.1145/3326165.
- 32 Maxime Flin, Mohsen Ghaffari, Magnús M. Halldórsson, Fabian Kuhn, and Alexandre Nolin. A distributed palette sparsification theorem. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4083–4123. SIAM, 2024. doi:10.1137/1.9781611977912.142.
- 33 Magnús M. Halldórsson, Fabian Kuhn, Yannic Maus, and Tigran Tonoyan. Efficient randomized distributed coloring in CONGEST. In *Proc. 53rd ACM Symposium on Theory of Computing (STOC)*, pages 1180–1193, 2021. doi:10.1145/3406325.3451089.
- 34 Magnús M. Halldórsson, Fabian Kuhn, Alexandre Nolin, and Tigran Tonoyan. Near-optimal distributed degree+1 coloring. In *Proc. 54th ACM Symposium on Theory of Computing (STOC)*, pages 450–463, 2022. doi:10.1145/3519935.3520023.
- 35 Paul R. Halmos. *Naive Set Theory*. Van Nostrand Reinhold, Princeton, NJ, 1960.
- 36 David G. Harris, Johannes Schneider, and Hsin-Hao Su. Distributed  $(\Delta+1)$ -coloring in sublogarithmic rounds. *J. ACM*, 65(4):19:1–19:21, 2018. Preliminary version appeared in *Proc. 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 465–478, 2016. doi:10.1145/3178120.
- 37 Holger SG Heidrich, Jannik Irmay, and Bjoern Andres. A 4-approximation algorithm for min max correlation clustering. In *AISTATS*, pages 1945–1953, 2024.
- 38 William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984. URL: <https://api.semanticscholar.org/CorpusID:117819162>.
- 39 Sanchit Kalhan, Konstantin Makarychev, and Timothy Zhou. Correlation clustering with local objectives. In *Advances in Neural Information Processing Systems (NuerIPS)*, volume 32, 2019.
- 40 Xinghao Pan, Dimitris Papailiopoulos, Samet Oymak, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. Parallel correlation clustering on big graphs. In *Proc. 28th International Conference on Neural Information Processing Systems (NIPS)*, pages 82–90, 2015. URL: <https://proceedings.neurips.cc/paper/2015/hash/b53b3a3d6ab90ce0268229151c9bde11-Abstract.html>.
- 41 Gregory J. Puleo and Olgica Milenkovic. Correlation clustering and biclustering with locally bounded errors. *IEEE Trans. Inf. Theory*, 64(6):4105–4119, 2018. doi:10.1109/TIT.2018.2819696.