Solving the Correlation Cluster LP in Sublinear Time

Nairen Cao* Vincent Cohen-Addad† Euiwoong Lee‡ Shi Li§
David Rasmussen Lolck¶ Alantha Newman|| Mikkel Thorup¶
Lukas Vogl** Shuyi Yan¶ Hanwen Zhang††

October 20, 2025

Abstract

Correlation Clustering is a fundamental and widely-studied problem in unsupervised learning and data mining. The input is a graph and the goal is to construct a clustering minimizing the number of inter-cluster edges plus the number of missing intra-cluster edges.

[CCL+24] introduced the *cluster LP* for Correlation Clustering, which they argued captures the problem much more succinctly than previous linear programming formulations. However, the cluster LP has exponential size, with a variable for every possible set of vertices in the input graph. Nevertheless, [CCL+24] showed how to find a feasible solution for the cluster LP in time $O(n^{\text{poly}(1/\varepsilon)})$ with objective value at most $(1+\varepsilon)$ times the value of an optimal solution for the respective Correlation Clustering instance. Furthermore, they showed how to round a solution to the cluster LP, yielding a $(1.485+\varepsilon)$-approximation algorithm for the Correlation Clustering problem.[1]

The main technical result of this paper is a new approach to find a feasible solution for the cluster LP with objective value at most $(1+\varepsilon)$ of the optimum in time $\widetilde{O}(2^{\text{poly}(1/\varepsilon)}n)$, where n is the number of vertices in the graph. We also show how to implement the rounding within the same time bounds, thus achieving a fast $(1.485+\varepsilon)$-approximation algorithm for the Correlation Clustering problem. This bridges the gap between state-of-the-art methods for approximating Correlation Clustering and the recent focus on fast algorithms.

*New York University, Email: nc1827@nyu.edu

†Google Research. Email: cohenaddad@google.com.

‡University of Michigan. Email: euiwoong@umich.edu. Supported in part by NSF grant CCF-2236669 and Google.

§Nanjing University. Email: shili@nju.edu.cn. Affiliated with the School of Computer Science in Nanjing University, and supported by the State Key Laboratory for Novel Software Technology and the New Cornerstone Science Laboratory.

¶University of Copenhagen (BARC). Emails: dalo@di.ku.dk, mthorup@di.ku.dk, shya@di.ku.dk. Supported by VILLUM Foundation Grant 54451 and Basic Algorithms Research Copenhagen (BARC).

||Université Grenoble Alpes. Email: alantha.newman@grenoble-inp.fr.

**EPFL. Email: lukas.vogl@epfl.ch. Supported by the Swiss National Science Foundation project 200021-184656 "Randomness in Problem Instances and Randomized Algorithms".

††University of Copenhagen. Email: hazh@di.ku.dk. Supported by VILLUM Foundation Grant 54451, Basic Algorithms Research Copenhagen (BARC), and Starting Grant 1054-00032B from the Independent Research Fund Denmark (Sapere Aude).

[1] [CCL+24] claimed that there is a rounding for the cluster LP yielding a 1.437-approximation algorithm for Correlation Clustering. However, the proof has a bug, and the best bound we can currently prove for the cluster LP is 1.485. Proof details will appear shortly in an updated arxiv version of [CCL+24].

Contents

1 Introduction

Correlation Clustering, introduced by Bansal, Blum, and Chawla [BBC04], is a fundamental problem in unsupervised machine learning that neatly captures the inherent tension between grouping similar data elements and separating dissimilar ones. Given a complete graph where each edge receives either a positive or a negative label, representing the pairwise relationship of its endpoints, the objective is to partition the vertices into clusters that minimize the number of "unsatisfied" edges: positive edges across clusters and negative edges within clusters. This framework naturally arises in diverse applications, including clustering ensembles [BGU13], duplicate detection [ARS09], community mining [CSX12], link prediction [YV18] disambiguation tasks [KCMNT08], image segmentation [KYNK14], and automated labeling [AHK+09, CKP08].

Despite its widespread applicability, Correlation Clustering is APX-hard [CGW05], motivating a rich line of research focused on approximation algorithms. Early work by [BBC04] provided an $O(1)$-approximation, which was subsequently improved to a 4-approximation by [CGW05]. The influential Pivot algorithm [ACN08] achieved a 3-approximation, and further refinements using LP-based techniques culminated in a 2.06-approximation [CMSY15], nearly matching the integrality gap of 2.

Recent work has shown how to surpass this barrier. Cohen-Addad, Lee and Newman employed the Sherali-Adams hierarchy to obtain a $(1.994+\varepsilon)$-approximation [CLN22], which was later improved to $(1.73+\varepsilon)$ by Cohen-Addad, Lee, Li and Newman, using a new so-called *preclustering* technique to preprocess the instance [CLLN23]. Most recently, Cao, Cohen-Addad, Lee, Li, Newman and Vogl introduced the cluster LP framework which generalizes all previously known formulations, and showed how to round it to obtain a $(1.485+\varepsilon)$-approximation in polynomial time [CCL+24]. [2] However, the cluster LP (see Section 1.1 for details) has exponential size: It contains a variable for each subset of the vertices indicating whether this subset is a cluster or not. The recent work of [CCL+24] leverages the preclustering method to compute a solution to the cluster LP in time $O(n^{\mathrm{poly}(1/\varepsilon)})$ whose value is within a $(1+\varepsilon)$-factor of the cost an optimal clustering, hence leading to a polynomial-time approximation algorithm.

Correlation Clustering is a versatile, fundamental model for clustering and has thus received a lot of attention from practitioners. Therefore, a large body of work studying Correlation Clustering in popular practical computation models has emerged. Since 2018, researchers have shown how to obtain a $(3+\varepsilon)$-approximation to Correlation Clustering in streaming [BCMT23, MC24], sublinear time [AW22], the Massively Parallel Computation (MPC) model [BCMT22, CHS24, DMM24], or vertex or edge fully dynamic setting [BCC+24, DMM24]. For all but the dynamic setting, a very recent work by Cohen-Addad, Lolck, Pilipczuk, Thorup, Yan and Zhang [CLP+24] has provided a unified approach achieving a $(1.847+\varepsilon)$-approximation to Correlation Clustering in all the above models (1.875 for the MPC model) using a new local search algorithm.

In general, the approximation guarantees obtained in these various restricted settings have remained significantly higher than the best known polynomial time approximation of $(1.485+\varepsilon)$, which employed computationally expensive solutions to Sherali-Adams relaxations of Correlation Clustering to solve the cluster LP.

Thus, it is natural to consider what we can achieve when we are allowed limited computation time. For instance, **How well can Correlation Clustering be approximated in (sub)linear time?** Here, by linear time, we mean time linear in the number of $+$edges, and by sublinear time, we mean time linear in the number of vertices.

[2] See the footnote in the abstract for a remark regarding the 1.485 approximation ratio for cluster LP.

In this work, we answer this question by showing how to solve and round the cluster LP in sublinear time, matching the state-of-the-art approximation achieved in [CCL+24]. This opens up a new path to study Correlation Clustering in other computational models.

Notation. Before we explain our results in more detail, we introduce some basic notation. The input to the correlation clustering problem is a complete graph where each edge is labeled either as a $+$edge or a $-$edge. For a graph G, we often denote its vertex set by $V(G)$ and its edge set by $E(G)$. We let $G = (V, E)$ represent the subgraph induced by the $+$edges (i.e., $E^+ = E(G)$), while the set of $-$edges is given by $E^- = \binom{V(G)}{2} \setminus E(G)$. Let $n = |V|$ and $m = |E|$ represent the number of vertices and $+$edges in G, respectively. Given a graph G and a subset $V' \subseteq V(G)$, $G[V']$ denotes the subgraph of G induced by V'. For technical reasons, we assume E contains all n self-loops $\{uu : u \in V\}$.

For two sets A and B, we use $A \bigoplus B = (A \setminus B) \cup (B \setminus A)$ to denote their symmetric difference. For simplicity, we use the following shorthand. Given a function or vector f and a set S, if f outputs reals, then $f(S) = \sum_{e \in S} f(e)$ or $f(S) = \sum_{e \in S} f_e$.

Finally, we emphasize that we always use OPT to denote an optimal clustering for a given Correlation Clustering instance, and we use cost(OPT) to denote its cost, which we will formally define shortly in Section 1.2.

1.1 Our Results

The **cluster LP** was introduced in [CCL+24]. In this formulation of Correlation Clustering, we have a variable z_S for every non-empty subset $S \subseteq V$, where z_S indicates whether S forms a cluster in the output clustering. Additionally, for every pair of vertices $uv \in \binom{V}{2}$, the variable x_{uv} indicates whether u and v are separated in the clustering.

$$\min \quad \text{obj}(x) := \sum_{uv \in E^+} x_{uv} + \sum_{uv \in E^-} (1 - x_{uv}) \quad \text{s.t.} \qquad \text{(cluster LP)}$$

$$\sum_{S \ni u} z_S = 1 \quad \forall u \in V$$

$$\sum_{S \supseteq \{u,v\}} z_S = 1 - x_{uv} \quad \forall uv \in \binom{V}{2}.$$

With this formal definition of the cluster LP, our main results are stated in the following theorems.

Theorem 1 (Efficient cluster LP)**.** *Let $\varepsilon, \delta > 0$ be small enough constants and let* OPT *be the cost of the optimum solution to the given Correlation Clustering instance. Then there is a small $\Delta = \text{poly}(\varepsilon)$ such that the following statement holds. One can output a solution $(z_S)_{S \subseteq V}$ to the cluster LP with $\text{obj}(x) \leq (1 + \varepsilon)\text{OPT}$ in expectation, described using a list of non-zero coordinates such that each coordinate of z is either 0 or at least Δ. In the various models, the respective procedure has the following attributes.*

- *(Sublinear model) The running time to compute z is $\widetilde{O}(2^{\text{poly}(1/\varepsilon)} n)$.*
- *(MPC model) It takes $2^{\text{poly}(1/\varepsilon)}$ rounds with $O(n^\delta)$ memory per machine and total memory $\tilde{O}(\text{poly}(\frac{1}{\varepsilon})m)$, or takes $\text{poly}(\frac{1}{\varepsilon})$ rounds with $O(n^\delta)$ memory per machine and total memory $\tilde{O}(2^{\text{poly}(1/\varepsilon)} m)$.*

We call the non-zero entries of a solution $(z_S)_{S \subseteq V}$ its *support* and refer to it as $\text{supp}(z)$.

Rounding Algorithms. Given a solution z to the cluster LP, we can round it to a solution for Correlation Clustering efficiently as well. Concretely, we prove the following theorem.

Theorem 2 (Efficient Rounding Algorithm)**.** *Let $\varepsilon > 0$ be a small enough constant and $\Delta = \text{poly}(1/\varepsilon)$. Given a solution to the cluster LP $(z_S)_{S \subseteq V}$ where each coordinate of z is either 0 or at least Δ, one can output a clustering with expected cost at most $(1.485 + \varepsilon)\text{obj}(x)$ in time $\tilde{O}(n/\Delta^2)$.*

Combining Theorem 1 and Theorem 2 gives us our main conclusion for Correlation Clustering.

Corollary 3. *There exists a $(1.485 + \varepsilon)$-approximation algorithm for Correlation Clustering that runs in time $\tilde{O}(2^{\text{poly}(1/\varepsilon)}n)$.*

Remark We emphasize that the sublinear-time algorithm outputs a clustering. However, we do not know the number of disagreements for the clustering.

1.2 Technical Overview

We first discuss the cluster LP in more depth.

Cluster LP. Given a set S, let $E^+(S, V \setminus S) = \{uv \in E^+ \mid u \in S, v \notin S \text{ or } u \notin S, v \in S\}$ denote the set of $+$edges with exactly one endpoint in S, and let $E^-(S) = \{uv \in E^- \mid u, v \in S\}$ denote the set of $-$edges with both endpoints in S. We can rewrite the objective value of cluster LP using z values instead of x values as

$$\text{cost}(z) := \sum_{S \subseteq V} \text{cost}(S) \cdot z_S,$$

where $\text{cost}(S)$ is the cost contribution of the cluster S and

$$\text{cost}(S) := \frac{1}{2} \cdot |E^+(S, V \setminus S)| + |E^-(S)|. \tag{1}$$

The cluster LP is equivalent to minimizing $\text{cost}(z)$, subject to the following constraints.

$$\begin{aligned} \min \quad & \text{cost}(z) \quad \text{s.t.} && \text{(cluster LP)} \\ \sum_{S \ni u} z_S &= 1 \quad \forall u \in V \\ z_S &\geq 0 \quad \forall S \subseteq V, S \neq \emptyset. \end{aligned}$$

Difficulty of Solving cluster LP. A natural approach to solving a linear program with an exponential number of variables is to solve the dual, which has an exponentially many constraints, but polynomially many variables and could potentially be solved via a polynomial-time separation oracle. In the case of the cluster LP, we have the following dual linear program containing a variable for each vertex, which can take on a possibly non-positive value.

$$\begin{aligned} \max & \sum_{u \in V} q_u && \text{(dual cluster LP)} \\ \sum_{u \in S} q_u &\leq \text{cost}(S) \quad \forall S. \end{aligned}$$

In this case, it is not obvious how to find an efficient separation oracle. However, a subroutine used in the multiplicative weights algorithm (discussed next) does yield an approximate separation oracle. Previous approaches to finding a solution for the cluster LP with objective value at most $(1 + \varepsilon)\text{cost}(\text{OPT})$ involved solving a Sherali-Adams relaxation containing $n^{\text{poly}(1/\varepsilon)}$ constraints [CCL+24].

Multiplicative Weights Update Framework. We will use the multiplicative weights update (MWU) framework by Plotkin, Shmoys, and Tardos [PST95] to solve the cluster LP approximately. Actually, we will solve a covering linear program that agrees with the cluster LP on its optimal solutions, but is better adapted for the MWU framework. The framework maintains a weight w_v for each vertex constraint $\sum_{S \ni v} z_S \geq 1$. During each step t, the vertex constraints are collapsed into a single constraint, each scaled according to the normalized vertex weight. The resulting optimization problem is the following.

$$\min \qquad \text{cost}(z) \qquad \text{s.t.} \tag{2}$$

$$\sum_S p(S) z_S \geq 1 \tag{3}$$

$$z_S \geq 0 \qquad \forall S \subseteq V, S \neq \emptyset. \tag{4}$$

Here, $p(S)$ is the normalized weight of vertices in S. The optimal solution to this problem is the set $S \subseteq V$ that has the smallest cost to vertex weight ratio. However, instead of solving the problem optimally we will find a family of several sets that have a small cost to vertex weight ratio compared to the cost of the optimal clustering. To be more precise, we constructing a partial clustering $\mathcal{F}$ that covers at least a constant fraction P of the vertex weight. We set $z_C = \frac{1}{P}$ for each cluster $C \in \mathcal{F}$ as the solution for step t. Then, each vertex weight is scaled depending on the margin by which z violates the covering constraint of that vertex. We can show that a constant number of rounds of this framework is enough to produce an approximately optimal solution to the cluster LP.

Finding the Partial Clustering. We will find a family $\mathcal{F}$ of disjoint clusters where each cluster $C \in \mathcal{F}$ has a small cost to vertex weight ratio. Moreover, this family will cover a constant fraction of the total vertex weight. Given a vertex r and a guess R for the cost of the optimal solution, we can efficiently find a single cluster $C_r \ni r$ with a cost to vertex weight ratio at most R if such a cluster exists. In particular, if the cluster $C \in \text{OPT}$ that contains the vertex r achieves the ratio R, then we will find a cluster C_r that achieves a ratio close to R. There has to exist a cluster $C \in \text{OPT}$ that achieves the ratio R since the normalized vertex weight gives a probability distribution. Thus, we can find a cluster C_r for each vertex $r \in V$ and add the cluster with the best ratio (which is at most R) to the partial clustering $\mathcal{F}$. We then remove the vertices already covered by $\mathcal{F}$ and update all clusters C_r that contain covered/lost vertices. We repeat the process until $\mathcal{F}$ covers a constant fraction of the total vertex weight.

Finding a Small Ratio Cluster. The problem we aim to solve is as follows: given a weight function w for each vertex, we seek a cluster C^* that minimizes the expression $\text{cost}(C^*) - \sum_{v \in C^*} w(v)$, where $\text{cost}(C^*)$ represents the Correlation Clustering cost associated with selecting C^* as a cluster (see (1)). A small ratio cluster will yield a negative value for this expression.

Inspired by the local search algorithm in [CLP+24], we use a similar approach to find such a C^*. Let's focus on how to find C^* to minimize this difference. We start by assuming we know C^*, but we cannot directly query whether a vertex $v \in C^*$ or not. In this case, our key observation is that if C^* is the cluster that minimizes $\text{cost}(C^*) - \sum_{v \in C^*} w(v)$, then according to the optimality of C^*, we have:

1. $\text{Marginal}(C^*, v) - w(v) \leq 0$ for all $v \in C^*$, and
2. $\text{Marginal}(C^*, v) - w(v) \geq 0$ for all $v \notin C^*$,

where $\text{Marginal}(C^*, v) = \text{cost}(C^* \cup \{v\}) - \text{cost}(C^* \setminus \{v\})$ is the marginal cost of adding v to C^* or removing v from C^*. At first glance, it seems we don't make progress, as we don't know how to compute Marginal. The key idea here is that if we can sample some vertices from C^*, we can obtain a good estimate for Marginal. However, if C^* has no structure, we have no way to sample nodes from C^*.

Thanks to the preclustering step [CLLN23], which roughly identifies almost-cliques that should be clustered together and prohibits pairs that are impossible to place in the same cluster, we can make a good guess about the size of C^*. After preclustering, we know a candidate set N_{cand}, and C^* is a subset of N_{cand} with $|C^*| = \Theta(|N_{\text{cand}}|)$. Now we can sample nodes from C^*: if we uniformly sample $\Theta(1)$ nodes from N_{cand}, each sample hits C^* with constant probability, and our sample set has size $\Theta(1)$, hitting $\Theta(1)$ nodes from C^*. By enumerating all possible subsets of the sampled set, we obtain a sample for C^*, and by a standard concentration bound, we have a good estimation for $\text{Marginal}(C^*, v)$.

One last problem remains. Since we no longer obtain $\text{Marginal}(C^*, v)$ exactly, we rely on an estimated $\text{Marginal}(C^*, v)$, meaning we may make some errors in selecting C^*: we might miss some vertices in C^* and add others incorrectly. If we make too many such errors, our final set will differ significantly from C^*. The key observation is that as long as C^* and $\tilde{C}^*$ (the approximated cluster) do not differ too much, our estimate for C^* remains reasonable. To leverage this, we split N_{cand} into many small chunks. The estimation error in each round is bounded by the size of each chunk. After processing one chunk, we update our sample vertices to match the nodes we already found. One might argue that we no longer obtain C^* exactly, which is true due to errors in each chunk. To bound the error affecting $\text{cost}(C^*) - \sum_{v \in C^*} w(v)$, we will use new sample sets based on previous choices. As long as there is some slackness in $\text{cost}(C^*) - \sum_{v \in C^*} w(v)$, we can still obtain a final cluster with a reasonably small ratio.

Achieving Nearly Linear Time. We have to address two problems in order to find the partial clustering $\mathcal{F}$ in nearly linear time. First, we cannot afford to compute all the clusters C_r, one for each vertex $r \in V$. Second, we cannot afford to update each cluster C_r as soon as it loses one vertex $v \in C_r$. To solve the first problem, we will sample a subset of vertices $U \subseteq V$ such that U is not too large and we can compute a cluster C_r for each vertex $r \in U$. Moreover, with high probability, we will hit every cluster $C \in \text{OPT}$ with a vertex, $U \cap C \neq \emptyset$. Because of this, there will still be a cluster among $\{C_r\}_{r \in U}$ that achieves the ratio R. To address the second problem, we will update a cluster C_r only if it lost a constant fraction of its weight. If a cluster C_r did not lose this constant fraction, the ratio did not change by too much. We show that a cluster C_r has to lose at least a constant fraction of vertices in order to lose a constant fraction of weight. We can charge the cost of updating the cluster C_r to the number of vertices the cluster C_r loses.

Achieving Sublinear Time. To achieve a sublinear-time algorithm, we must accelerate the process of finding a small-ratio cluster. The bottleneck is the estimation of the cost for a given cluster. In general, this estimation is impossible in the sublinear-time model. Our key insight is that if the cost is small, the cluster should have been split at the beginning. Thus, in the remaining graph, only clusters with large costs persist, allowing us to estimate the cost efficiently.

MPC Algorithm. To parallelize the algorithm, the main bottleneck is that we need to find a set of small-ratio clusters instead of identifying them one by one. We can select multiple nodes and run the small-ratio cluster-finding algorithm simultaneously. However, their can-

didate sets may overlap, so we remove nodes that appear in multiple candidate sets. As long as we use a sufficiently small constant probability to select nodes, we ensure a large enough candidate set, allowing each round to produce a sufficiently large set of small-ratio clusters.

2 Preclustering

We first introduce some more necessary notation. Define $d^+(v)$ as the degree of vertex v with respect to $+$edges in G. For any subset $C \subseteq V$, let $d^+(v, C)$ and $d^-(v, C)$ denote the number of $+$edges and $-$edges from v to C, respectively, connecting v to vertices in C. When the context is clear, we omit the $+$ symbol, so $d(v)$ and $d(v, C)$ mean $d^+(v)$ and $d^+(v, C)$, respectively.

2.1 Computational Models.

We consider two computational models in this paper: the sublinear model and the MPC model.

The Sublinear model. In the sublinear model, the algorithm can query the following information in $O(1)$ time:

- *Degree queries*: What is the degree of vertex v?
- *Neighbor queries*: What is the i-th neighbor of $v \in V$ for $i \leq d^+(v)$?
- *Edge queries*: Is $uv \in E^+$?

One can think of this model as storing an adjacency list and an adjacency matrix of $+$edges for G. The matrix allows us to check whether $uv \in E^+$ in $O(1)$ time.

The MPC model. In the MPC model, the set of edges 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.

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 $O(n^\delta)$.

2.2 Preclustering

Our results crucially depend on the usage of the following *preclustering* subroutine. Intuitively, preclustering is a preprocessing step that will identify almost-cliques that should be together and prohibit all pairs that are impossible to place in the same cluster. There will be some remaining pairs for which we cannot decide whether or not they should be clustered together or split apart. A major advantage of this procedure is that the number of uncertain pairs can be bounded by the optimal solution. More precisely, we use the definition of a preclustered instance from [CLLN23], which is also applied in [CLP+24].

Definition 4 (Preclustering)**.** *Given a Correlation Clustering instance $(V, E^+ \uplus E^-)$, a* preclustered instance *is defined by a pair $(\mathcal{K}, E_{adm})$. Here, $\mathcal{K}$ is a family of disjoint subsets of V (not necessarily a partition) Each set $K \in \mathcal{K}$ has $|K| \geq 2$ and is called a* (non-singleton) atom*. We use $V_{\mathcal{K}} := \bigcup_{K \in \mathcal{K}} K$ to denote the set of all vertices in non-singleton atoms. A vertex in $V \setminus \mathcal{K}$ is called a* singleton atom*.*

$E_{\text{adm}} \subseteq \binom{V}{2}$ is the set of admissible *edges (sometimes called* admissible pairs*), which are defined to be pairs of vertices $(u, v) \in E_{\text{adm}}$ with at least one of u and v not in $V_{\mathcal{K}}$. A pair of vertices in a same $K \in \mathcal{K}$ is called an* atomic *edge. A pair that is neither an atomic nor an admissible edge is called a* non-admissible *edge.*

For a vertex u, let $d_{\text{adm}}(u)$ denote the number of vertices $v \in V$ such that (u, v) is an admissible pair. Note that admissible, atomic, and non-admissible edges can each be either $+$edges or $-$edges. Let $N_{\text{adm}}(v)$ to be the set of vertices u such that $(u, v) \in E_{\text{adm}}$. We use $K(v)$ to represent the set $K \in \mathcal{K}$ that contains v; to make the notation more general, when v is a singleton atom, $K(v) = \{v\}$.

We also need the definition of ε-similar preclustered instance and ε-similar cluster to bound the value of $d_{\text{adm}}(v)$ for each node $v \in V$.

Definition 5 (ε-similar Preclustering)**.** *Given a Correlation Clustering instance $(V, E^+ \uplus E^-)$, let $(\mathcal{K}, E_{\text{adm}})$ be a* preclustered instance *for G, and let $\varepsilon > 0$ be some parameter. We say $(\mathcal{K}, E_{\text{adm}})$ is an* ε-similar preclustering *if*

- *for any $v \in V$, we have $d_{\text{adm}}(v) \leq 2\varepsilon^{-3} d(v)$,*
- *for any $uv \in E_{adm}$, we have $d(u) \leq 2\varepsilon^{-1} d(v)$ and the number of common neighbors that are degree similar to u and v is at least $\varepsilon \cdot \min\{d(u), d(v)\}$, where a pair $w\tilde{w}$ is degree similar if $\varepsilon d(w) \leq d(\tilde{w}) \leq d(w)/\varepsilon$,*
- *for every atom $K \in \mathcal{K}$, for every vertex $v \in K$, we have that v is adjacent to at least a $(1 - O(\varepsilon))$-fraction of the vertices in K and has at most $O(\varepsilon|K|)$ neighbors not in K.*

Definition 6 (ε-large Cluster)**.** *Given a preclustered instance $(\mathcal{K}, E_{\text{adm}})$ for some Correlation Clustering instance $(V, E^+ \uplus E^-)$, a set $C \subset V$ is called* ε-large *with respect to $(\mathcal{K}, E_{\text{adm}})$ if*

- *C does not break any atomic edge,*
- *C does not contain any non-admissible edge, and*
- *for any vertex v in C, if $|C| > 1$, then $|C| \geq \varepsilon d(v)$.*

Moreover, a clustering scheme $\mathcal{C}$ is called ε-large with respect to $(\mathcal{K}, E_{adm})$ if all clusters $C \in \mathcal{C}$ are ε-large clusters with respect to $(\mathcal{K}, E_{\text{adm}})$.

The next theorem is stated in [CLP+24]. The construction follows the framework of Assadi and Wang [AW22, CLM+21, CLLN23].

Theorem 7 (Preclustering Procedures [CLP+24])**.** *For a Correlation Clustering instance $(V, E^+ \uplus E^-)$ with optimal value* cost(OPT) *(which is not known to us), and for any sufficiently small $\varepsilon > 0$, there are algorithms that produce an ε-similar preclustered instance $(\mathcal{K}, E_{adm})$ that admits an ε-large clustering scheme $\mathcal{C}^*_{(\mathcal{K}, E_{adm})}$ such that*

1. $\text{cost}(\mathcal{C}^*_{(\mathcal{K}, E_{adm})}) \leq (1 + \varepsilon)\text{cost}(\text{OPT})$,
2. *and* $|E_{adm}| \leq O\big(\varepsilon^{-12}\text{cost}(\text{OPT})\big)$.

n the various models, the respective procedure has the following attributes.

- *(MPC model) The algorithm takes $O(1)$ rounds succeeds with probability at least $1 - 1/n$ and requires $O(n^\delta)$ memory per machine. Moreover, the algorithm uses a total memory of $O(m \log n)$, where m is the number of the $+$ edges.*
- *(Sublinear model)In time $O(n \log n)$ one can compute the partition $\mathcal{K}$ and a data structure such that with success probability at least $1 - 1/n^2$, for any pair of vertices, the data structure can answer in $O(\log n)$ time whether the pair is in E_{adm} or not. Moreover, the data structure can list all vertices admissible to v in $O(d(v) \log^2 n)$ time.*

Assumptions. We will use OPT to denote an optimal clustering for the respective Correlation Clustering input instance. We assume that OPT is an ε-large clustering and satisfies $|E_{\text{adm}}| \leq O\left(\varepsilon^{-12}\text{cost}(\text{OPT})\right)$ because of Theorem 7.

3 Multiplicative Weights Update Framework

In this section, we prove the following: there exists an MWU algorithm that, in a constant number of rounds, finds a solution to the cluster LP that is feasible and has a cost of at most $(1 + O(\varepsilon))\text{cost}(\text{OPT})$.

The framework is provided in Algorithm 1. To solve cluster LP, we first run the preclustering subroutine, which provides some structure to the graph. Then, for each non-singleton atom of the preclustering, if the cost of isolating this atom is zero, it means that this atom is a clique with no outgoing edges, and we will simply make it a cluster. Next, we preprocess cluster LP by adding a fixed value to the objective and reformulating it as a covering LP, so that our final LP has certain desirable properties. We will discuss the details of covering cluster LP (Line 3) and how to convert our solution back (Line 5) in Subsection 3.1. We use the MWU algorithm to solve the covering cluster LP. The description of the MWU algorithm (Lines 3-4) is given in Subsection 3.2.

3.1 Converting cluster LP to covering cluster LP

In order to run a multiplicative weights algorithm to solve the cluster LP, we first transform it into a covering LP. We make two key changes to the cluster LP. First, for each set $S \subseteq V$, we will add $\sum_{v \in S} d_{\text{cross}}(v)$ to the cost function.

$$\text{cover}(S) := \text{cost}(S) + d_{\text{cross}}(S),$$

where $d_{\text{cross}}(v)$ is twice the cost attributable to vertex v if we choose to make $K(v)$ a cluster. If v is a singleton atom, then $d_{\text{cross}}(v) = d(v) - 1$ (because of the self-loop vv). If v is an atom, then

$$d_{\text{cross}}(v) = \frac{2 \cdot \text{cost}(K(v))}{|K(v)|} = \frac{1}{|K(v)|} \sum_{u \in K(v)} \left(d(u) + |K(v)| - 2d(u, K(v))\right). \tag{5}$$

The new objective function will be

$$\text{cover}(z) := \sum_{S \subseteq V} \text{cover}(S) \cdot z_S.$$

Secondly, instead of an equality constraint, we relax the constraint to an inequality. Using the new objective function, we can rewrite the cluster LP as a covering cluster LP.

Algorithm 1 Solving cluster LP

Input: Graph G
Output: a feasible solution to cluster LP satisfying Theorem 1
Find a preclustering G via Theorem 7; let $(\mathcal{K}, E_{\text{adm}})$ be the preclustered instance.
$V' \leftarrow V$
for all atoms $K \in \mathcal{K}$ such that if $d_{\text{cross}}(K) = 0$ **do**
  make K a cluster and $V' \leftarrow V' \setminus K$
end for
Solve covering cluster LP using MWU Algorithm 2 on $G[V']$ to within $(1 + O(\varepsilon^{13}))$ of the optimum solution.
Let $\{z_S\}_S$ be the solution.
Compute a solution $\{\tilde{z}_S\}_S$ to the cluster LP using Lemma 11.
return $\{\tilde{z}_S\}_S$.

$$\min \quad \text{cover}(z) \quad \text{s.t.} \qquad \text{(covering cluster LP)}$$

$$\sum_{S \ni u} z_S \geq 1 \qquad \forall u \in V \tag{6}$$

$$z_S \geq 0, \qquad \forall S \subseteq V S \neq \emptyset. \tag{7}$$

Note that the cross degrees $d_{\text{cross}}(v)$ can be computed in time $\widetilde{O}(m)$ by accessing each edge and checking if it belongs to the same atom. This is sufficient if we aim for nearly linear time.

One may ask whether this modification changes the solution in terms of the approximation algorithm. Let E_{cross} be the set of disagreement edges if we isolate all singleton and non-singleton atoms as the final clusters (i.e., $2|E_{\text{cross}}| = \sum_{v \in V} d_{\text{cross}}(v)$). If E_{cross} is unbounded, we are unable to obtain any guarantee for cluster LP when solving covering cluster LP approximately. Fortunately, we have the following lemma to help us bound the increase in the objective when adding d_{cross} to the cost.

Lemma 8. *For any non-singleton atom $K \in \mathcal{K}$,*

$$d_{\text{cross}}(K) = 2 \cdot \text{cost}(K) = \Omega(\varepsilon^3 d_{\text{adm}}(K)).$$

Moreover,

$$d_{\text{cross}}(V) = \sum_{v \in V} d_{\text{cross}}(v) = O\left(\frac{1}{\varepsilon^{12}}\right) \text{cost}(\text{OPT}).$$

Proof. We first show the upper bound on $\sum_{v \in V} d_{\text{cross}}(v)$. Each edge counted by $\sum_{v \in V} d_{\text{cross}}(v)$ is either admissible or contributes to the cost of OPT. To finish the upper bound, remember that $|E_{\text{adm}}| = O(\varepsilon^{-12}\text{cost}(\text{OPT}))$ by Theorem 7.

Then, we will prove the lower bound on $d_{\text{cross}}(K)$. Consider a vertex $v \in K$ and an admissible edge (v, u). If uv is a $+$edge, then $d_{\text{cross}}(K)$ already covers it, so we only need to consider $-$edges.

Let $A(v)$ be the set of all vertices that are connected to v by an admissible $-$edge. Let u be one such neighbor in $A(v)$. By Definition 5, v and u are degree similar and must share at least $\varepsilon \cdot \min\{d(v), d(u)\} \geq \varepsilon^2 d(v)$ $+$neighbors, which are degree similar to both u and v. Recall that a pair $w\tilde{w}$ is degree similar if $\varepsilon d(w) \leq d(\tilde{w}) \leq d(w)/\varepsilon$. We will distinguish between two cases.

Let $A_1(v) \subseteq A(v)$ be the set of vertices $u \in A(v)$ such that at least half of the degree-similar $+$neighbors of u and v are not in K. Note that $v \in K$, has at most $d(v) - d(v,K)$ degree similar $+$neighbors outside K. Each $u \in A_1(v)$ has to be adjacent to at least $\frac{\varepsilon^2}{2}d(v)$ of these vertices outside of K which have degree at most $d(v)/\varepsilon$. Therefore, we have at most

$$\frac{(d(v) - d(v,K)) \cdot d(v)/\varepsilon}{\varepsilon^2 d(v)/2} \leq 2\varepsilon^{-3}(d(v) - d(v,K))$$

vertices in $A_1(v)$. Thus, we can bound $\sum_{v\in K} |A_1(v)|$ by $2\varepsilon^{-3}\sum_{v\in K}(d(v) - d(v,K)) \leq 4\varepsilon^{-3}\text{cost}(K) \leq 4\varepsilon^{-3}d_{\text{cross}}(K)$.

Let $A_2(v) \subseteq A(v)$ be the set of vertices $u \in A(v)$ such that that at least half of the degree-similar $+$neighbors of u and v are in K. Each $u \in A_2(v)$ must connect to at least $\frac{\varepsilon^2}{2}d(v)$ vertices in K, so we have at most

$$\frac{d_{\text{cross}}(K)}{\varepsilon^2 d(v)/2} \leq \frac{4d_{\text{cross}}(K)}{\varepsilon^2|K|}$$

vertices that are in $A_2(v)$. Thus, we can bound $\sum_{v\in K} |A_1(v)|$ by $|K| \cdot \frac{4d_{\text{cross}}(K)}{\varepsilon^2|K|} = 4\varepsilon^{-2}d_{\text{cross}}(K)$.

Combining the two cases, we find that the $-$admissible neighbors of K are at most $\sum_{v\in K} |A(v)| = O(\varepsilon^{-3}d_{\text{cross}}(K))$. □

We say that a set of vertices $S \subseteq V$ does not split atoms if $K(v) \subseteq S$ for all $v \in S$. Adding d_{cross} is useful for us because the new objective function $\text{cover}(\cdot)$ remains monotone as long as the involved sets do not split atoms.

Lemma 9. *Let $U, W \subseteq V$ such that neither U nor W splits an atom. If $U \subset W$ then,* $\text{cover}(U) < \text{cover}(W)$.

Proof. We need to show:

$$\text{cost}(U) + d_{\text{cross}}(U) < \text{cost}(W) + d_{\text{cross}}(W) = \text{cost}(W) + d_{\text{cross}}(U) + d_{\text{cross}}(W \setminus U). \tag{8}$$

So we want to show:

$$\text{cost}(U) < \text{cost}(W) + d_{\text{cross}}(W \setminus U), \tag{9}$$

which is true since the $-$edges in U also belong to W and $+$edges contributing to $\text{cost}(U)$ but not $\text{cost}(W)$ are at most those in $E^+(U, W \setminus U)$, whose contribution to $\text{cost}(U)$ is $|E^+(U, W \setminus U)|/2 < d_{\text{cross}}(W \setminus U)$. □

Using the above observation, we can show that an optimal solution of the covering cluster LP is feasible for the cluster LP as long as it does not split atoms.

Lemma 10. *Let z be an optimal solution to covering cluster LP where each set S in the support of z does not split atoms. Then z satisfies all constraints with equality.*

Proof. Assume for contradiction that there exists an atom $K(v)$ that is not tight, $\sum_{S\supseteq K(v)} z_S > 1$. Let $W \subseteq V$ be such that $z_W > 0$. Let $U = W \setminus K(v)$. We modify z by decreasing z_W and increasing z_U. Let $a = (\sum_{S:K(v)\subseteq S} z_S) - 1$, and let $b = \min\{z_W, a\}$. We define a new vector $\widetilde{z}$. Let $\widetilde{z}_U = z_U + b$ and let $\widetilde{z}_W = z_W - b$. For all other S such that $S \neq W, U$, let $\widetilde{z}_S = z_S$.

We obtain a contradiction by showing that $\widetilde{z}$ remains feasible but has a decreased objective value. The only constraints affected by the change are those corresponding to vertices in W. For a vertex $u \in U = W \setminus K(v)$,

$$\sum_{S\ni u} \widetilde{z}_S = \widetilde{z}_W + \widetilde{z}_U + \sum_{\substack{S\ni u \\ S\neq W,U}} z_S = (z_W - b) + (b + z_U) + \sum_{\substack{S\ni u \\ S\neq W,U}} z_S = \sum_{S\ni u} z_S \geq 1.$$

For the atom $K(v)$, we have

$$\sum_{S \supseteq K(v)} \widetilde{z}_S = \widetilde{z}_W + \sum_{\substack{S \supseteq K(v) \\ S \neq W}} z_S = z_W - b + \sum_{\substack{S \ni v \\ S \neq W}} z_S \geq 1.$$

The last inequality holds since $b \leq a$. To finish the proof, we have to show that the objective value has decreased. In other words, we want to show the following quantity is positive.

$$\text{cover}(W) \cdot (z_W - \widetilde{z}_W) + \text{cover}(U) \cdot (z_U - \widetilde{z}_U) = \text{cover}(W) \cdot b + \text{cover}(U) \cdot (-b).$$

To finish the proof, remember that $\text{cover}(\cdot)$ is monotone and $U \subseteq W$ does not split atoms. The lemma follows from Lemma 9. □

We can show a more general version of Lemma 10. In particular, we can transform a suitable solution of the covering cluster LP into a solution to the cluster LP that satisfies the additional condition in Theorem 1, which stipulates that all values in the support are lower bounded by a small constant.

Lemma 11. *Let $\varepsilon > 0$ be a small enough constant, $\gamma = O(\varepsilon^{13})$ and $c = \lceil \frac{1}{\gamma} \rceil$. Given a constant $T_{\text{MW}} \in \mathbb{N}$ and a solution z to the covering cluster LP where,*

- *$z_S \geq \frac{1}{T_{\text{MW}}}$ for each $S \in \text{supp}(z)$,*
- *all $S \in \text{supp}(z)$ do not split atoms,*
- *for each vertex v the number of sets $S \in \text{supp}(z)$ with $v \in S$ is at most T_{MW}.*

We can find a solution $\widetilde{z}$ to the cluster LP where $\widetilde{z}_S \geq \frac{1}{cT_{\text{MW}}}$ for all $S \in \text{supp}(z)$ and $\text{cover}(\widetilde{z}) \leq (1+\gamma)\text{cover}(z)$ and the solution $\widetilde{z}$ can be found in time $O(n)$.

Proof. If z is a solution to the covering cluster LP, we can assume that $z_S \leq 1$ for all $S \in V$. Otherwise, we can set $z_S = 1$ only decreasing the objective. We start by rounding each coordinate z_S to a multiple of $\frac{1}{cT_{\text{MW}}}$. In particular, set $z_S^{(1)}$ to $\frac{k}{cT_{\text{MW}}}$ where $k \in \mathbb{N}$ such that $\frac{k-1}{cT_{\text{MW}}} \leq z_S \leq \frac{k}{cT_{\text{MW}}}$. Since we do not decrease the value of any coordinate, $z^{(1)}$ remains feasible. Furthermore, the objective increases by at most a γ factor. Indeed, for any set $S \subseteq V$,

$$z_S^{(1)} - z_S \leq \frac{1}{cT_{\text{MW}}} \leq \frac{\gamma}{T_{\text{MW}}} \leq \gamma z_S.$$

The second inequality holds since $c \geq \frac{1}{\gamma}$ and the last inequality follows from $z_S \geq \frac{1}{T_{\text{MW}}}$. Next, scale the variable $z^{(1)}$ and the constraints by cT_{MW}. Note that by the above, we have that $z_S^{(1)} = \frac{k_S^{(1)}}{cT_{\text{MW}}}$ for some $k_S^{(1)} \in \mathbb{N}$. After scaling we have variables $\{k_S^{(1)}\}_{S \subseteq V}$ and constraints, $\sum_{S \ni v} k_S^{(1)} \geq cT_{\text{MW}}$. We can transform $k^{(1)}$ to a scaled up solution of the cluster LP similar to the proof of Lemma 10. Pick an atom $K(v)$ that is not tight, $\sum_{S \supseteq K(v)} k_S^{(1)} > cT_{\text{MW}}$. Let W be a set with $k_W^{(1)} > 0$ and $K(v) \subseteq W$. Let $U = W \setminus \{K(v)\}$. We modify $k^{(1)}$ by decreasing $k_W^{(1)}$ and increasing $k_U^{(1)}$. Let $a = (\sum_{S : K(v) \subseteq S} k_S^{(1)}) - cT_{\text{MW}}$, and let $b = \min\{k_W^{(1)}, a\}$. We define a new vector $k^{(2)}$. Let $k_U^{(2)} = b + k_U^{(1)}$ and let $k_W^{(2)} = k_W^{(1)} - b$. For all other S such that $S \neq W, U$, let $k_S^{(2)} = k_S^{(1)}$. We will show that $k^{(2)}$ remains feasible while the objective value decreases. The only constraints affected by the change are those corresponding to sets containing vertices in W. For a vertex $u \in U = W \setminus \{K(v)\}$,

$$\sum_{S \ni u} k_S^{(2)} = k_W^{(2)} + k_U^{(2)} + \sum_{\substack{S \ni u \\ S \neq W, U}} k_S^{(2)} = (k_W^{(1)} - b) + (b + k_U^{(1)}) + \sum_{\substack{S \ni u \\ S \neq W, U}} k_S^{(1)} = \sum_{S \ni u} k_S^{(1)} \geq cT_{\text{MW}}.$$

For the atom $K(v)$, we have

$$\sum_{S \supseteq K(v)} k_S^{(2)} = k_W^{(2)} + \sum_{\substack{S \supseteq K(v) \\ S \neq W}} k_S^{(2)} = k_W^{(1)} - b + \sum_{\substack{S \ni K(v) \\ S \neq W}} k_S^{(1)} \geq cT_{\mathrm{MW}}.$$

The last inequality holds since $b \leq a$. To finish the proof, we have to show that the objective value has decreased. In other words, we want to show the following quantity is positive.

$$\mathrm{cover}(W) \cdot (k_W^{(1)} - k_W^{(2)}) + \mathrm{cover}(U) \cdot (k_U^{(1)} - k_U^{(2)}) = \mathrm{cover}(W) \cdot b + \mathrm{cover}(U) \cdot (-b).$$

Observe that $U \subset W$ does not split atoms. Thus, the objective decreases by Lemma 9. We can repeat this process until $z^{(2)} = k^{(2)}/(cT_{\mathrm{MW}})$ is feasible for the cluster LP. This process terminates after $O(n)$ iterations, which follows from the fact that in each iteration $\sum_{v \in V} \sum_{S \ni v} k_S^{(2)}$ decreases by at least one. To see this, observe that

$$\sum_{v \in V} \sum_{S \ni v} k_S^{(2)} = \sum_S \sum_{v \in S} k_S^{(2)} = \sum_S |S| \, k_S^{(2)}.$$

This last sum decreases by at least one, since we have "shifted" weight from W to a smaller set U. Moreover, by the third property of z and the assumption that $z_S \leq 1$ for all $S \subseteq V$, we have that $\sum_{S \ni v} k_S^{(2)} \leq cT_{\mathrm{MW}}^2 = O(1)$. □

Throughout the following sections, we will fix the parameter $\gamma = O(\varepsilon^{13})$ since we want to obtain a $(1+\gamma)$-approximate algorithm for covering cluster LP. Assume z is a $(1+\gamma)$-approximately feasible solution for covering cluster LP, let $\tilde{z}$ be the solution we apply Lemma 11 to obtain a feasible solution for cluster LP. Observe that we have $\mathrm{cover}(\tilde{z}) = \mathrm{cost}(\tilde{z}) + d_{\mathrm{cross}}(V)$. Thus,

$$\begin{aligned}
\mathrm{cost}(\tilde{z}) &= \mathrm{cover}(\tilde{z}) - d_{\mathrm{cross}}(V) \leq \mathrm{cover}(z) - d_{\mathrm{cross}}(V) \\
&\leq (1+\gamma)\mathrm{cover}(\mathrm{OPT}) - d_{\mathrm{cross}}(V) \\
&\leq (1+\gamma)\mathrm{cost}(\mathrm{OPT}) + (1+\gamma) d_{\mathrm{cross}}(V) - d_{\mathrm{cross}}(V) \\
&\leq (1+2\varepsilon)\mathrm{cost}(\mathrm{OPT}).
\end{aligned}$$

The last inequality holds because $d_{\mathrm{cross}}(V) = O(\varepsilon^{-12}\mathrm{cost}(\mathrm{OPT}))$. In the following section, we present the MWU algorithm to solve covering cluster LP within a $(1+O(\gamma))$-approximate ratio.

3.2 The MWU Algorithm

In this section, we show that a constant number of rounds of the MWU Algorithm suffices to compute a $(1+O(\gamma))$-approximate solution to the covering cluster LP. The MWU algorithm is given in Algorithm 2.

Overview of Algorithm 2. Algorithm 2 solves the covering cluster LP within a $(1+\gamma)$ factor of the optimum. Algorithm 2 maintains a weight w_v for each vertex. During each step $t = 1, \ldots T_{\mathrm{MW}}$, we scale the vertex constraints by the normalized vertex weights and aggregate the scaled constraints into a single constraint. We can find a solution to the now simplified optimization problem by Lemma 15, which we will prove in Section 4. The weights are then updated depending on the margin by which we violate or satisfy the corresponding vertex constraint. The final solution for the covering cluster LP is obtained by taking the average of the solutions to the simplified optimization problems solved at each round.

Algorithm 2 MWU algorithm for the covering cluster LP

1: Initialize the weights $w_v^{(1)} = d_{\text{cross}}(v)$ for each vertex $v \in V$.
2: **for** $t = 1, \ldots, T_{\text{MW}}$ **do**
3: Normalize the weights $p^{(t)} = \frac{w^{(t)}}{\sum_v w_v^{(t)}}$.
4: Aggregate all constraints into single constraint: $\sum_S p^{(t)}(S) \cdot z_S = \sum_v p_v^{(t)} \left(\sum_{S: v \in S} z_S\right) \geq 1$.
5: Find the point $z^{(t)} \in [0, 1/\gamma]^{2^V}$ from Lemma 15 that,

- satisfies the single constraint $\sum_S p^{(t)}(S) \cdot z_S^{(t)} \geq 1$,
- has objective value $\text{cover}(z^{(t)}) \leq (1 + 5\gamma)\,\text{cover}(\text{OPT})$,
- does not split atoms (i.e., if $z_S^{(t)} > 0$ then $K(v) \subseteq S$ for all vertices $v \in S$), and
- has disjoint support (i.e., if $z_S^{(t)}, z_T^{(t)} > 0$ for two distinct $S, T \subseteq V$, then $S \cap T = \emptyset$).

6: The cost of a constraint corresponds to the margin by which it is satisfied or violated, $m_v^{(t)} = \sum_{S: v \in S} z_S^{(t)} - 1$.
7: Update the weights $w_v^{(t+1)} = w_v^{(t)} e^{-\gamma^3 m_v^{(t)}}$.
8: **end for**
9: Let $\hat{z}$ be the average $\frac{1}{T_{\text{MW}}} \sum_{t=1}^{T_{\text{MW}}} z^{(t)}$.
10: **for** each v with $\sum_{S \supseteq K(v)} \hat{z}_S \leq 1 - 2\gamma$ **do**
11: Set the atom entry $\hat{z}_{K(v)}$ to 1.
12: **end for**
13: $z^\star \leftarrow \frac{\hat{z}}{1-2\gamma}$.
14: **return** $z^\star$.

In order to ensure feasibility, we need to increase the values of sets corresponding to atoms containing insufficiently covered vertices. Finally, when all elements are almost covered (to an extent $1 - 2\gamma$), we can scale up the value by a small amount to ensure sufficient coverage of all vertices. The key step in the analysis is showing that there were few insufficiently covered vertices and we do not need to increase the values of too many sets corresponding to their atoms.

Lemma 12. *Let $\varepsilon > 0$ be a small enough constant and $\gamma = O(\varepsilon^{13})$. After $T_{\text{MW}} = \frac{\log(1/\gamma)}{\gamma^4}$ rounds, Algorithm 2 returns a solution z to the covering cluster LP with objective* $\text{cover}(z) \leq (1 + O(\gamma))\,\text{cover}(\text{OPT})$.

Proof. First, we will prove that z is feasible for the covering cluster LP. Note that at the end of the **for** loop on Line 10, we have that $\sum_{S \ni v} \hat{z}_S \geq 1 - 2\gamma$ for each vertex v. Feasibility follows since we scale by $1/(1 - 2\gamma)$. Next, we will prove the bound on the objective of z. Consider the potential

$$\Phi^{(t)} = \sum_{v \in V} w_v^{(t)}.$$

The starting potential is

$$\Phi^{(1)} = \sum_{v \in V} d_{\text{cross}}(v) = d_{\text{cross}}(V).$$

Because of the way we update the weights, the potential at the end of the execution is

$$\Phi^{(T_{\mathrm{MW}}+1)} = \sum_{v \in V} w_v^{(1)} \cdot \exp\left(-\gamma^3 \sum_{t \le T_{\mathrm{MW}}} m_v^{(t)}\right).$$

Claim 13. *[AHK12] We can relate the potential at the start of the execution and at the end as follows.*

$$\Phi^{(T_{\mathrm{MW}}+1)} \le \Phi^{(1)} \cdot \exp\left(\gamma^4 T_{\mathrm{MW}} - \gamma^3 \sum_{t \le T_{\mathrm{MW}}} \langle p^{(t)}, m^{(t)} \rangle\right).$$

Proof. Again, by our update rule in Line 7,

$$\Phi^{(t+1)} = \sum_{v \in V} w_v^{(t+1)} = \sum_{v \in V} w_v^{(t)} \cdot \exp(-\gamma^3 m_v^{(t)}).$$

Remember that $0 \le z_S^{(t)} \le \frac{1}{\gamma}$ for all $S \subseteq V$. Moreover, the support of $z^{(t)}$ consists of disjoint sets. This implies that $m_v^{(t)} \in [-1, \frac{1}{\gamma} - 1]$ for all vertices v and steps t.

$$\begin{aligned}
\sum_{v \in V} w_v^{(t)} \cdot \exp(-\gamma^3 m_v^{(t)}) &\le \sum_{v \in V} w_v^{(t)} (1 - \gamma^3 m_v^{(t)} + (\gamma^3 m_v^{(t)})^2) \\
&\le \sum_{v \in V} w_v^{(t)} (1 - \gamma^3 m_v^{(t)} + \gamma^4) \\
&= \sum_{v \in V} w_v^{(t)} (1 + \gamma^4) - \sum_{v \in V} w_v^{(t)} \gamma^3 m_v^{(t)} \\
&= \Phi^{(t)} (1 + \gamma^4) - \sum_{v \in V} \Phi^{(t)} p_v^{(t)} \gamma^3 m_v^{(t)} \\
&= \Phi^{(t)} \left(1 + \gamma^4 - \gamma^3 \langle p^{(t)}, m^{(t)} \rangle\right) \\
&\le \Phi^{(t)} \exp\left(\gamma^4 - \gamma^3 \langle p^{(t)}, m^{(t)} \rangle\right).
\end{aligned}$$

The first inequality follows from the fact that $e^x \le (1 + x + x^2)$ for $x \in [-1, 1]$. □

Claim 14. *Let U be the set of vertices that are uncovered by $\hat{z}$ (i.e., $U = \{v \in V \mid \sum_{S \ni v} \hat{z}_S \le 1 - 2\gamma\}$). Then,*

$$\sum_{v \in U} d_{\mathrm{cross}}(v) \le 2\gamma \, d_{\mathrm{cross}}(V).$$

Proof. Remember that

$$\Phi^{(T_{\mathrm{MW}}+1)} = \sum_{v \in V} w_v^{(1)} \cdot \exp(-\gamma^3 \sum_{t \le T_{\mathrm{MW}}} m_v^{(t)}).$$

Fix a vertex $u \in U$. Since u is uncovered by $\hat{z}$, we have

$$T_{\mathrm{MW}} \cdot 2\gamma \le T_{\mathrm{MW}} \cdot \left(1 - \sum_{S \ni u} \hat{z}_S\right) = \sum_{t \le T_{\mathrm{MW}}} \left(1 - \sum_{S \ni u} z_S^{(t)}\right) = -\sum_{t \le T_{\mathrm{MW}}} m_u^{(t)}.$$

Using this, we can lower bound the potential,

$$\Phi^{(T_{\mathrm{MW}}+1)} \ge \sum_{v \in U} w_v^{(1)} \cdot \exp(2\gamma^4 T_{\mathrm{MW}}).$$

Together with the upper bound from Claim 13,

$$\sum_{v\in U} w_v^{(1)} \cdot \exp(2\gamma^4 T_{\mathrm{MW}}) \le \Phi^{(T_{\mathrm{MW}}+1)} \le \Phi^{(1)} \cdot \exp\left(\gamma^4 T_{\mathrm{MW}} - \gamma^3 \sum_{t\le T_{\mathrm{MW}}} \langle p^{(t)}, m^{(t)}\rangle\right).$$

Observe that that $\langle p^{(t)}, m^{(t)}\rangle$ will be non-negative since the points $z^{(t)}$ satisfy the single constraint $\sum_S p^{(t)}(S)\cdot z_S \ge 1$. Indeed,

$$\langle p^{(t)}, m^{(t)}\rangle = \sum_{v\in V} p_v^{(t)} m_v^{(t)} = \sum_{v\in V} p_v^{(t)} \left(\sum_{S\ni v} z_S - 1\right) = \sum_S p^{(t)}(S)\cdot z_S - 1.$$

We can conclude that,

$$\sum_{v\in U} w_v^{(1)} \cdot \exp(2\gamma^4 T_{\mathrm{MW}}) \le \Phi^{(1)} \cdot \exp\left(\gamma^4 T_{\mathrm{MW}} - \gamma^3 \sum_{t\le T_{\mathrm{MW}}} \langle p^{(t)}, m^{(t)}\rangle\right) \le \sum_{v\in V} w_v^{(1)} \cdot \exp(\gamma^4 T_{\mathrm{MW}}).$$

Rearranging,

$$\sum_{v\in U} d_{\mathrm{cross}}(v) = \sum_{v\in U} w_v^{(1)} \le \sum_{v\in V} w_v^{(1)} \cdot \exp((\gamma^4 - 2\gamma^4) T_{\mathrm{MW}}) = \gamma \cdot \sum_{v\in V} w_v^{(1)} = 2\gamma \cdot d_{\mathrm{cross}}(V).$$

The second equality holds by the choice of $T_{\mathrm{MW}} = \frac{\log(1/\gamma)}{\gamma^4}$.

□

Next, we will bound the objective value of $z^\star$. By the second property of Lemma 15, we have $\mathrm{cover}(z^{(t)}) \le (1+5\gamma)\,\mathrm{cover}(\mathrm{OPT})$ for each $t \le T_{\mathrm{MW}}$. Since cover is a linear function, this upper bound also holds for the average $\frac{1}{T_{\mathrm{MW}}}\sum_{t=1}^{T_{\mathrm{MW}}} z^{(t)}$. At the end of Algorithm 2, we cover all uncovered atoms $U = \{v \in V \mid \sum_{S\ni v} \hat{z}_S \le 1 - 2\gamma\}$. When we set a coordinate $\hat{z}_K$ to 1, the cover objective value increases by at most $3/2 \cdot d_{\mathrm{cross}}(K)$. We have $\mathrm{cost}(K) = 1/2 \cdot d_{\mathrm{cross}}(K)$. The additional $d_{\mathrm{cross}}(K)$ is the term added to $\mathrm{cost}(K)$ for the covering cluster LP. Recall that each $z^{(t)}$ does not split atoms. Hence, if $v \in U$ then $K(v) \subseteq U$. By Claim 14, the cost increases by at most,

$$\sum_{v\in U} d_{\mathrm{cross}}(v) \le 2\gamma \cdot d_{\mathrm{cross}}(V) \le 2\gamma \cdot \mathrm{cover}(\mathrm{OPT}).$$

Thus, the objective of z can be bounded by,

$$\mathrm{cover}(z) \le \frac{1+7\gamma}{1-2\gamma}\mathrm{cover}(\mathrm{OPT}) \le (1+10\gamma)\mathrm{cover}(\mathrm{OPT}).$$

□

4 Finding a Partial Clustering with Small Ratio in Polynomial Time

In this section we will show how to find the point $z^{(t)}$ in Line 5 of the MWU Algorithm 2.

Lemma 15. *Given vertex weights $p_v > 0$ for all $v \in V$, we can construct a point $z \in [0, 1/\gamma]^{2^{|V|}}$ that,*

1. *satisfies the single constraint $\sum_S p(S)\cdot z_S \ge 1$,*
2. *has objective $\mathrm{cover}(z) \le (1+5\gamma)\,\mathrm{cover}(\mathrm{OPT})$,*

Algorithm 3 Algorithm to find the family $\mathcal{F}$

Let R be the guess for cover(OPT) such that $R \in [\text{cover}(\text{OPT}), (1+\gamma)\text{cover}(\text{OPT}))$.
$\hat{p} \leftarrow p, \mathcal{F} \leftarrow \emptyset, \hat{V} \leftarrow V$
for all $v \in V$ **do**
  Find a small ratio cluster C_v where $K(v) \subseteq C_v \subseteq N_{\text{adm}}(v)$ with vertex weights $\hat{p} > 0$ and target ratio $(1+3\gamma)R$ (Lemma 18).
end for
while $p(\mathcal{F}) \leq \gamma$ **do**
  Choose C with the smallest ratio $\frac{\text{cover}(C)}{\hat{p}(C)}$ among clusters $\{C_v\}_{v \in \hat{V}}$.
  Add C to $\mathcal{F}$, set $\hat{p}_v$ to 0 for all $v \in C$ and remove vertices in C from $\hat{V}$.
  for all $v \in \hat{V}$ **do**
    {Update C_v if some node in C_v is added to $\mathcal{F}$}
    if $C_v \cap C \neq \emptyset$ **then**
      Find a new small ratio cluster C_v with vertex weights $\hat{p} > 0$ and target ratio $(1+3\gamma)R$ (Lemma 18).
    end if
  end for
end while
return $\mathcal{F}$

3. *does not split atoms (i.e., if $z_S > 0$ then $K(v) \subseteq S$ for all vertices $v \in S$),*

4. *has disjoint support (i.e., if $z_S, z_T > 0$ for two distinct $S, T \subseteq V$, then $S \cap T = \emptyset$).*

Throughout this section, we assume we are given a distribution p over vertices. We have $p_v > 0$ for each vertex $v \in V$ and $\sum_{v \in V} p_v = 1$. For a set $S \subseteq V$ of vertices, we write $p(S) = \sum_{v \in S} p_v$. Similarly, we will abuse notation and write $p(\mathcal{F}) = \sum_{S \in \mathcal{F}} p(S)$ for a family $\mathcal{F} \subseteq 2^V$ of subsets of vertices.

In order to construct the point z, we will find a partial clustering $\mathcal{F} = \{S_1, S_2, ..., S_l \mid S_i \subseteq V\}$ that achieves a small ratio $\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})}$. Each vertex v is contained in at most one set $S \in \mathcal{F}$. Moreover, $\mathcal{F}$ is a partial clustering, meaning that $\mathcal{F}$ might not cover all the vertices in V. However, we require that $\mathcal{F}$ covers at least a constant fraction of the probability mass of the vertices (i.e., $p(\mathcal{F}) \geq \gamma$). We show that Algorithm 3 will find such a partial clustering.

Description of Algorithm 3. Algorithm 3 relies on Lemma 18, which we will prove in Section 5. Lemma 18 states that given a vertex r and the optimal value $R \approx \text{cover}(\text{OPT})$, we can efficiently find a single cluster $C_r \ni r$ with a small ratio $\frac{\text{cover}(C_r)}{p(C_r)} \leq R$ if there exists such a cluster and in particular, if the cluster $C \in \text{OPT}$ that contains r achieves the ratio R. There has to exist a cluster $C \in \text{OPT}$ that achieves the ratio R since p is a probability distribution. Thus, we can find a cluster C_r for each vertex $r \in V$ and add the cluster with the smallest ratio to the partial clustering $\mathcal{F}$; this ratio will be at most R. If the cluster S we find that way has small vertex probability mass $p(S) < \gamma$, we have to repeat this process. Note that when we remove the vertices in S, we only remove $p(S) < \gamma$ of probability mass. We argue that in this case, there will exist a cluster in OPT with ratio at most $(1+2\gamma)\text{cover}(\text{OPT})$ and we can find a cluster disjoint from S that achieves ratio $(1+2\gamma)\text{cover}(\text{OPT})$. Hence, we can repeat until we cover constant probability mass with $\mathcal{F}$. For now, the algorithm will have a runtime of $\widetilde{O}(n^4)$. In Section 6, we will show how we can modify Algorithm 3 to achieve a nearly linear runtime.

Lemma 16. *Given vertex weights $p_v > 0$ for all $v \in V$, Algorithm 3 finds a family $\mathcal{F} = \{S_1, S_2, ..., S_l \mid S_i \subseteq V\}$ such that,*

1. *for any distinct $S, T \in \mathcal{F}$, $S \cap T = \emptyset$,*
2. $\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})} \leq (1 + 5\gamma)\text{cover}(\text{OPT})$,
3. *$p(\mathcal{F})$ is at least γ,*
4. *no $S \in \mathcal{F}$ splits an atom (i.e., $K(v) \subseteq S$, for all vertices $v \in S$).*

Proof. First, observe that Property 3. holds by the condition of the **while** loop. Property 4. holds since the all small ratio clusters $\{C_v\}_{v \in V}$ do not split atoms by Lemma 18. Next, we observe that Property 1. holds: the family $\mathcal{F}$ consists of disjoint sets. Observe that after we added a cluster $C \subseteq V$ to $\mathcal{F}$ we set the weight $\hat{p}_v$ to 0 for all $v \in C$. Note that if the weight $\hat{p}_v$ is set to 0 it remains 0 throughout the execution of the algorithm. Moreover, when we add a cluster C to $\mathcal{F}$, it contains only vertices such that $\hat{p}_v > 0$. Thus, when we add C to $\mathcal{F}$ we have $\hat{p}_v > 0$ for all $v \in C$ and $\hat{p}_v = 0$ for all $v \in S, S \in \mathcal{F}$.

Now we address Property 2. Based on our assumption, we have R such that $\text{cover}(\text{OPT}) \leq R < (1 + \gamma)\text{cover}(\text{OPT})$. It remains to prove the bound on the ratio $\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})}$. Let $C \in \text{OPT}$ and $v \in C$ such that if K is a non-singleton atom in C then $K = K(v)$. (If C contains no such atom, then v can be any vertex in C.) As long as,

$$\frac{\text{cover}(C) + \gamma^2 d_{\text{adm}}(C)}{\hat{p}(C)} \leq (1 + 3\gamma)R, \tag{10}$$

we maintain the following invariant:

$$\frac{\text{cover}(C_v)}{\hat{p}(C_v)} \leq (1 + 3\gamma)R. \tag{11}$$

Before starting the **while** loop on Line 6, we have by Lemma 18,

$$\frac{\text{cover}(C_v)}{p(C_v)} \leq (1 + 3\gamma)R.$$

since C satisfies the requirements of Lemma 18 for the vertex v and ratio $(1 + 3\gamma)R$ by assumption. Consider one iteration of the **while** loop on Line 6. If we did not remove a vertex from C_v then the invariant remains true. Again, if C_v was updated, we have by Lemma 18,

$$\frac{\text{cover}(C_v)}{\hat{p}(C_v)} \leq (1 + 3\gamma)R.$$

Let $C^\star$ be the cluster in the optimal clustering OPT with the best ratio $\frac{\text{cover}(C^\star) + \gamma^2 d_{\text{adm}}(C^\star)}{\hat{p}(C^\star)}$. By the condition of the **while** loop on Line 6, $\hat{p}(V) = p(V) - p(\mathcal{F}) \geq 1 - \gamma$. Note that,

$$\begin{aligned}(1 + 3\gamma)R &\geq \frac{1 + \gamma}{1 - \gamma}R \geq \frac{(1 + \gamma)\text{cover}(\text{OPT})}{\hat{p}(V)} \geq \frac{\text{cover}(\text{OPT}) + \gamma^2 |E_{\text{adm}}|}{\hat{p}(V)} \\ &= \frac{\sum_{C \in \text{OPT}} (\text{cover}(C) + \gamma^2 d_{\text{adm}}(C))}{\sum_{C \in \text{OPT}} \hat{p}(C)} \geq \frac{\text{cover}(C^\star) + \gamma^2 d_{\text{adm}}(C^\star)}{\hat{p}(C^\star)}.\end{aligned}$$

For the second inequality, we use that γ cover(OPT) $\geq \gamma$ cost(OPT) $\geq \gamma^2 |E_{\text{adm}}|$ because of the preclustering (Theorem 7). The last inequality holds by the definition of $C^\star$. Let $v \in C^\star$ such that if K is a non-singleton atom in $C^\star$ then $K = K(v)$. By (11),

$$\frac{\text{cover}(C_v)}{\hat{p}(C_v)} \leq (1+3\gamma)R.$$

Since we choose C_r as the cluster with the best ratio among clusters $\{C_u\}_{u \in V}$,

$$\frac{\text{cover}(C_r)}{\hat{p}(C_r)} \leq \frac{\text{cover}(C_v)}{\hat{p}(C_v)} \leq (1+3\gamma)R.$$

We can conclude that,

$$\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})} = \frac{\sum_{S \in \mathcal{F}} \text{cover}(S)}{\sum_{S \in \mathcal{F}} p(S)} \leq (1+3\gamma)R \leq (1+5\gamma)\text{cover}(\text{OPT}).$$

□

We can use the family $\mathcal{F}$ to construct the point $z^{(t)}$ that we will use in one iteration of the MWU algorithm. From Lemma 16 we can readily derive Lemma 15.

Lemma 15. *Given vertex weights $p_v > 0$ for all $v \in V$, we can construct a point $z \in [0, 1/\gamma]^{2^{|V|}}$ that,*

1. *satisfies the single constraint $\sum_S p(S) \cdot z_S \geq 1$,*
2. *has objective* cover$(z) \leq (1+5\gamma)$ cover(OPT),
3. *does not split atoms (i.e., if $z_S > 0$ then $K(v) \subseteq S$ for all vertices $v \in S$),*
4. *has disjoint support (i.e., if $z_S, z_T > 0$ for two distinct $S, T \subseteq V$, then $S \cap T = \emptyset$).*

Proof. For each $S \in \mathcal{F}$, set $z_S = \frac{1}{p(\mathcal{F})} \geq 1$. First, we prove properties 3. and 4. Observe that the support of z is equal to $\mathcal{F}$ which contains only disjoint sets. Similarly, z does not split atoms since $\mathcal{F}$ does not. Property 1. is satisfied since,

$$\sum_S p(S) \cdot z_S = \sum_{S \in \mathcal{F}} \frac{p(S)}{p(\mathcal{F})} = 1.$$

To finish the proof, we can bound the objective as follows,

$$\text{cover}(z) = \sum_{S \in \mathcal{F}} \text{cover}(S) z_S = \frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})} \leq (1+5\gamma)\text{cover}(\text{OPT}).$$

□

Lemma 17. *The runtime of Algorithm 3 is $\widetilde{O}(n^4)$.*

Proof. Finding the small ratio cluster C_v for a single vertex v takes time $\widetilde{O}(d^2(v))$. So overall, we need time $\widetilde{O}(n^3)$ to find all $\{C_v\}_{v \in V}$. There are at most n iterations of the **while** loop on Line 6 since we set $\hat{p}_v = 0$ for at least one vertex v. This can happen at most n times. Indeed, once a vertex has weight $\hat{p}_v = 0$, the weight will remain 0. During one iteration of the **while** loop on Line 6, we might have to update all $\{C_v\}_{v \in V}$. Again, this takes time at most $\widetilde{O}(n^3)$. □

5 Finding One Small Ratio Cluster

In this section, we show how to find a small ratio cluster C_r, which is a key subroutine in Algorithm 3. We define the candidate set of r, $N_{\text{cand}}(r)$, to be the vertices that could possibly belong to the small ratio cluster C_r.

$$N_{\text{cand}}(r) = \begin{cases} N_{\text{adm}}(r) \setminus V_{\mathcal{K}} & \text{if } r \text{ is a singleton atom,} \\ K(r) \cup \left(\bigcap_{u \in K(r)} N_{\text{adm}}(u)\right) & \text{if } r \text{ belongs to a non-singleton atom.} \end{cases}$$

Intuitively, if r belongs to some non-singleton atom, then we only need to consider all admissible neighbors plus the nodes that are in the same atom. If r is a singleton atom, we do not need to consider any neighboring non-singleton atoms; we only need to consider admissible neighbors that are singleton atoms. Notice that the definition of N_{cand} is not symmetric (i.e., it might be the case that $u \in N_{\text{cand}}(v)$, but $v \notin N_{\text{cand}}(u)$).

In this section, we want to show the following Lemma.

Lemma 18. *Suppose we are given a graph $G = (V, E)$, vertex weights $\hat{p}$, a target ratio R, a vertex r and the set of vertices $N_{\text{adm}}(r)$.*

(i) Assume there exists a cluster C be an ε-large cluster with $K(r) \subseteq C \subseteq N_{\text{cand}}(r)$ such that $\text{cover}(C) + \gamma^2 d_{\text{adm}}(C) \leq R \cdot \hat{p}(C)$.

(ii) Assume that $\text{cover}(\{v\}) > R \cdot \hat{p}(\{v\})$ for all $v \in C$.

Then, with high probability, in time $\widetilde{O}(d^2(r))$, we can find a cluster $C_r \subseteq N_{\text{cand}}(r)$ such that,

$$\text{cover}(C_r) \leq R \cdot \hat{p}(C_r).$$

Moreover, C_r does not split atoms and contains exactly one non-singleton atom $K(r) \subseteq C_r$ iff $K(r)$ is a non-singleton atom.

Lemma 18 may return a cluster C_r that contains some nodes with $\hat{p}_v = 0$. In such cases, we can simply remove these nodes, as the monotonicity of cover ensures that removing nodes with $\hat{p} = 0$ will only decrease the ratio.

The algorithm is parameterized by $\gamma, \eta, \varepsilon$. Recall that $\gamma = O(\varepsilon^{13})$ is a small enough constant and $\eta = \Omega(\gamma^{-2}\varepsilon^{-8}) = \Omega(\varepsilon^{-34})$ is a sufficiently large constant. We also assume that $|N_{\text{cand}}(r)| = \Omega(\eta^2)$, otherwise, we can enumerate all possible subsets of $N_{\text{cand}}(r)$, which takes $O(2^{\text{poly}(1/\varepsilon)})$ time.

5.1 Overview of the Algorithm

In this section, we aim to solve the following optimization problem: Given a preclustered Correlation Clustering instance and vertex weights w, the goal is to find a set T of vertices such that

$$\text{cost}(T) \leq \sum_{v \in T} w(v).$$

Setting $w(v) := R \cdot p(v) - d_{\text{cross}}(v)$, if we can find a set T such that

$$\text{cost}(T) \leq \sum_{v \in T} \left(Rp(v) - d_{\text{cross}}(v)\right) \leq \sum_{v \in T} Rp(v) - \sum_{v \in T} d_{\text{cross}}(v),$$

then T is a small ratio cluster, since

$$\text{cover}(T) = \text{cost}(T) + \sum_{v \in T} d_{\text{cross}}(v) \leq R \sum_{v \in T} p(v) = R \cdot p(T).$$

Define $D(v) := N_{\text{cand}}(v) \setminus K(v)$ for any $v \in V$, and $\Delta(T) := \text{cost}(T) - w(T)$ for any $T \subseteq V$. As is the case of $N_{\text{adm}}(v)$ (i.e., see Definition 5), we can also bound the size of $N_{\text{cand}}(v)$, which we do by the following lemma.

Lemma 19. *For any $r \in V$, if $v \in N_{\text{cand}}(r)$, then $|N_{\text{cand}}(r)| = O(\varepsilon^{-4} d(v))$.*

Proof. We have by definition of N_{cand} that $|N_{\text{cand}}(r)| \leq |K(r)| + |N_{\text{adm}}(r)| = O(\varepsilon^{-3} d(r))$. The last equality is true since the preclustering is ε-similar. Note that for any vertex $v \in N_{\text{cand}}(r)$ the edge (r, v) is admissible. Thus, $d(r) \leq \varepsilon^{-1} d(v)$ and $|N_{\text{cand}}(r)| = O(\varepsilon^{-4} d(v))|$. □

For a set of vertices T and a vertex v, we want to compare the cost of including v in T to the cost of not including v in T, define the marginal value v with respect to T as,

$$\begin{aligned}
\text{Marginal}(T, v) &= \text{cost}(T \cup \{v\}) - \text{cost}(T \setminus \{v\}) \\
&= d^-(v, T) + \frac{1}{2} d^+(v, V \setminus (T \cup \{v\})) - \frac{1}{2} d^+(v, T \setminus \{v\}) \\
&= |T| - d^+(v, T) + \frac{1}{2}(d^+(v) - 1) - d^+(v, T \setminus \{v\}) \\
&= \frac{d^+(v) - 1}{2} + |T| - 2d^+(v, T) + \mathbb{1}(v \in T).
\end{aligned}$$

where $\mathbb{1}$ is the indicator function for the event $v \in T$. Recall that we assume uu is also a $+$edge in G. In the above calculation, the indicator variable corresponding to whether v belongs to T is used to adjust for the fact that the calculation differs slightly depending on whether v is in T or not. Since we do not know whether or not $v \in T$, we need this extra term.

One of the most important properties for function Marginal is the following.

Claim 20. *For any non-empty set T and T' and any vertex v, we have*

$$\text{Marginal}(T, v) - \text{Marginal}(T', v) \leq 2|T \oplus T'|$$

where $\oplus$ denotes the symmetric difference of two sets.

Proof. Using the definition of Marginal, we have

$$\begin{aligned}
\text{Marginal}(T, v) - \text{Marginal}(T', v) &= |T \cup \{v\}| - |T' \cup \{v\}| - 2(d^+(v, T) - d^+(v, T')) \\
&\leq |T \setminus T'| - |T' \setminus T| + 1 - 2(d^+(v, T \setminus T') - d^+(v, T' \setminus T)) \\
&\leq |T \setminus T'| - |T' \setminus T| + 1 + 2d^+(v, T' \setminus T)) \\
&\leq |T \oplus T'| + 1 \leq 2|T \oplus T'|.
\end{aligned}$$

□

Claim 20 implies that we can estimate the value $\text{Marginal}(T, v)$ by the value $\text{Marginal}(T', v)$ if T' is almost the same as T. In each round, we try to decide whether or not a node v should be in T using $\text{Marginal}(T, v)$. However, we do not know T, so we will instead sample a small set S from T and estimate $\text{Marginal}(T, v)$. To do this estimation, given a small sample set S, an integer guess t for the size of T and a vertex v, define

$$\text{EstMarg}(S, t, v) := \frac{d^+(v) - 1}{2} + t - 2\frac{d(v, S)}{|S|} t.$$

We will sample a constant number of nodes, so with constant probability, we can ensure EstMarg is close to Marginal.

Description of Algorithms 4 and 5. Given the definitions of Marginal(T, v) and EstMarg(S, t, v), we can now describe Algorithm 4 and Algorithm 5. Conceptually, we begin by guessing $C^* \subset N_{\text{cand}}(r)$, where C^* is defined to be the cluster for which $\text{cost}(C^*) - w(C^*)$ is minimized. Due to the optimality of C^*, adding a node to C^* or removing a node from C^* will increase this value, which implies that $\text{Marginal}(C^*, v) - w(v) \leq 0$ for all $v \in C^*$ and $\text{Marginal}(C^*, v) - w(v) \geq 0$ for all $v \notin C^*$. Based on this observation, we should add any node v if and only if $\text{Marginal}(C^*, v) - w(v) \leq 0$.

The algorithm does not know C^*, so to compute the value of $\text{Marginal}(C^*, v)$ for any node v, Algorithm 4 attempts to sample enough nodes from C^*. We will later show that $|C^*| = \Omega(\gamma^2 \varepsilon^{-8} |N_{\text{cand}}(r)|)$, ensuring that we always obtain some sampled vertices from C^*.

In this process, instead of sampling once, we sample η different sets A_i. This is because we need new sampled nodes each time we add vertices to our final set $\hat{T}$. The algorithm also tries to guess the size of C^*. Since C^* is an ε-large cluster, we know the size of C^* will be within $[\varepsilon d^+(r), |N_{\text{cand}}(r)|]$. We do not need the exact size, so we will enumerate possible sizes with different granularities, choosing values from the following set:

$$L(r) = \left\{ \left(1 + \frac{1}{\eta}\right)^j \in [\varepsilon d^+(r), \varepsilon^{-4} d^+(r)] \mid j \text{ is an integer} \right\}.$$

Algorithm 4 GenerateClusterBySampling$(G, N_{\text{adm}}(r), w, r, R)$

Input: The graph G, $K(r)$, $N_{\text{adm}}(r)$, $N_{\text{cand}}(r)$, w, r, ratio R.
Output: A small ratio cluster $\hat{T}$ if r satisfies Assumption (i) from Lemma 18.
Repeat the following steps $O(\log n)$ times.
for i from 1 to η **do**
  Uniformly sample $\Theta(\eta^4 \gamma^{-2} \varepsilon^{-8})$ vertices from $N_{\text{cand}}(r)$ with replacement
  Let the sample set be A_i. $\{A_i$ may contain some element multiple times.$\}$
end for
$D(r) \leftarrow N_{\text{cand}}(r) \setminus K(r)$
for every $(S^1, S^2, ..., S^\eta) \subset (A_1, A_2, ..., A_\eta)$ such that $|S^i| \leq \eta$, where $i \in [\eta]$ **do**
  for every $(\tilde{t}_1, \tilde{t}_2, ..., \tilde{t}_\eta) \in (L(r), L(r), ...L(r))$, where $\tilde{t}_j \in L(r)$ for $j \in [\eta]$ **do**
    $T \leftarrow \text{GenerateCluster}(r, D(r), S^1, \ldots, S^\eta, \tilde{t}_1, \ldots, \tilde{t}_\eta)$
    if $\text{cost}(T) \leq w(T)$ **then**
      return T
    end if
  end for
end for
return $\emptyset$

After the sampling step, we proceed to add vertices to $\hat{T}$. We must be careful regarding C^* because, once we add vertices, we may introduce errors. Using the same sampled nodes repeatedly could make the estimation of Marginal inaccurate. To address this, we divide $N_{\text{cand}}(r)$ into η "chunks", each of size $|N_{\text{cand}}(r)|/\eta$. For each chunk, we use the estimated Marginal to decide whether to add a node to $\hat{T}$. Once we finish processing a chunk, we update our sampled set to align with the choices we have already made. This process ensures that we return a good cluster with constant probability. By repeating it $O(\log n)$ times, we obtain our final small ratio cluster.

Algorithm 5 GenerateCluster$(r, D(r), S^1, \ldots, S^\eta, \tilde{t}_1, \ldots, \tilde{t}_\eta)$

1: $T \leftarrow K(r), \hat{T}_1 \leftarrow K(r)$
2: Let $D_r^1, \ldots, D_r^\eta$ be an arbitrary partition of the vertices of $D(r)$ into equal-size parts
3: **for all** $i = 1, \ldots, \eta$ **do**
4: **for all** $v \in D_r^i$ **do**
5: **if** EstMarg$(S^i, \tilde{t}_i, v) + 6\eta^{-1}|N_{\text{cand}}(r)| \leq w(v)$ **then**
6: $T \leftarrow T \cup \{v\}$
7: **end if**
8: **end for**
9: $\hat{T}_{i+1} \leftarrow T$
10: **end for**

Let C^* be the cluster such that $K(r) \subseteq C^* \subseteq N_{\text{cand}}(r)$ and $\Delta(C^*) = \text{cost}(C^*) - w(C^*)$ is minimized. For C, which exists by Assumption (i) in Lemma 18, we have $\Delta(C^*) \leq \Delta(C) \leq -\gamma^2 d_{\text{adm}}(C)$. To analyze the algorithm, recall $\hat{T}_i$ is the set of vertices in set $\hat{T}$ at the beginning of the i-th iteration of the **for** loop on Line 3 of Algorithm 5. Notice that $\hat{T}_{i-1} \subseteq \hat{T}_i$. We define

$$C_i^* = \hat{T}_i \cup \text{argmin} \left\{ \text{cost}(\hat{T}_i \cup B) - w(\hat{T}_i \cup B) \;\middle|\; B \subseteq \bigcup_{j=i}^{\eta} D_r^j \right\}.$$

Assume now that S^i is a uniform sample of the cluster C_i^*. Let $t_i = |C_i^*|$ be the size of $|C_i^*|$, and assume that $\tilde{t}_i \in [t_i, (1 + \frac{1}{\eta})t_i]$. Our main lemma regarding Algorithm 5 is given as follows.

Lemma 21. *Suppose we are given a graph $G = (V, E)$, vertex weights w and vertex r.*

- *Assume there exists a cluster C that is ε-large cluster with $K(r) \subseteq C \subseteq N_{\text{cand}}(r)$ such that $\text{cost}(C) + \gamma^2 d_{\text{adm}}(C) \leq w(C)$.*

Let C_i^* be the set defined above and S^i be a uniform sample with size $\eta_0 = \Omega(\eta^3)$ from C_i^*, $\tilde{t}_i \in [t_i, (1 + \frac{1}{\eta})t_i]$ is the guess for the size of C_i^*. Then with probability $1 - 2\eta \exp(-2\eta)$,

GenerateCluster$(r, D(r), S^1, \ldots, S^\eta, \tilde{t}_1, \ldots, \tilde{t}_\eta)$ produces a cluster $\hat{T}$ such that $\text{cost}(\hat{T}) \leq w(\hat{T})$.

Proof. We will later show the following claim,

Claim 22. *With probability at least $1 - 2\exp(-2\eta)$,*

$$\Delta(C_{i+1}^*) - \Delta(C_i^*) = O(\eta^{-2}\varepsilon^{-8} d_{\text{adm}}(C)).$$

Assuming the claim is true, a union bound implies that with probability at least $1 - 2\eta \exp(-2\eta)$, this claim holds for all $i = 1, \ldots, \eta$. Lemma 21 follows by observing that $\hat{T} = C_{\eta+1}^*$ and $C_1^* = C^*$. Recall, by definition, $\Delta(C_i^*) = \text{cost}(C_i^*) - w(C_i^*)$. For any C_i^*, where $i \in [\eta]$, we have

$$\begin{aligned}
\Delta(C_i^*) &= \Delta(C_1^*) + \sum_{j=1}^{i-1} (\Delta(C_{j+1}^*) - \Delta(C_j^*)) \\
&\leq \Delta(C^*) + \eta \cdot O(\eta^{-2}\varepsilon^{-8} d_{\text{adm}}(C)) \\
&\leq \Delta(C^*) + \frac{\gamma^2}{2} d_{\text{adm}}(C) \\
&\leq -\frac{\gamma^2}{2} d_{\text{adm}}(C).
\end{aligned}$$

The next-to-last inequality follows from the assumption $\eta = \Omega(\gamma^{-2}\varepsilon^{-8})$. □

Claim 23. *Either* $|C_i^\star| = s_i = \Omega(\gamma^2\varepsilon^8|N_{\text{cand}}(r)|)$ *for all* $i \in [\eta]$ *or there exists a vertex* $u \in N_{\text{cand}}(r)$ *such that* $\text{cost}(u) \le w(u)$.

Proof. Note that for any $i \in [\eta]$, we have,

$$\Delta(C_i^*) \le -\frac{\gamma^2}{2} d_{\text{adm}}(C),$$

where C is the ε-large cluster such that $K(r) \subseteq C \subseteq N_{\text{cand}}(r)$. By Lemma 19, $|N_{\text{cand}}(r)| = O(\varepsilon^{-3}d(r))$. If r is in a non-singleton atom $K(r) \subseteq C_i^\star$, then $|C_i^\star| \ge |K(r)| \ge (1 - O(\varepsilon))d(r) \ge \Omega(\varepsilon^3|N_{\text{cand}}(r)|)$. $|K(r)| \ge (1-\varepsilon)d(r)$ is due to preclustering, every non-singleton atom has at most $O(\varepsilon)$ fraction $+$neighbors outside of $K(r)$.

Otherwise, all vertices in $N_{\text{cand}}(r)$ are in singleton-atoms. Assume that $|C_i^\star| \le \gamma^2\varepsilon^8|N_{\text{cand}}(r)| = O(\gamma^2\varepsilon^5 d(r))$. In this case,

$$-\frac{\gamma^2}{2} d_{\text{adm}}(C) \ge \Delta(C_i^\star) = \text{cost}(C_i^\star) - \sum_{v \in C_i^\star} w(v) \ge \frac{1}{2}\sum_{v \in C_i^\star} d(v) - |C_i^\star|^2 - \sum_{v \in C_i^\star} w(v).$$

The last inequality uses $\text{cost}(C') + |C'|^2 \ge \frac{1}{2}\sum_{v \in C'} d(v)$, which holds for any cluster C'. (The right side counts the $+$edges incident on C'. Either such an edge leaves C' and contributes $1/2$ to the $\text{cost}(C')$ or it is inside C'. There are at most $|C'|^2/2$ edges inside C'.) Since $C \subseteq N_{\text{cand}}(r)$, C only contains vertices in singleton atoms. Since C is ε-large, all edges in C are admissible. Thus, $d_{\text{adm}}(C) \ge \frac{1}{2}|C|^2 \ge \frac{\varepsilon^2}{2}d^2(r)$. Hence, $|C_i^\star|^2 - \frac{\gamma^2}{2}d_{\text{adm}}(C) \le \varepsilon^{10}\gamma^6 d^2(r) - \frac{\varepsilon^2\gamma^2}{4}d^2(r) \le 0$. In particular,

$$\frac{1}{2}\sum_{v \in C_i^\star} d(v) - \sum_{v \in C_i^\star} w(v) \le |C_i^\star|^2 - \frac{\gamma^2}{2} d_{\text{adm}}(C) \le 0.$$

There has to exist a vertex $v \in C_i^\star$ such that $\frac{1}{2}d(v) - w(v) \le 0$. □

Now, we are able to show the main lemma 18 of this section.

Proof of Lemma 18. Lemma 21 outputs $\hat{T}$ such that $\text{cost}(\hat{T}) \le w(\hat{T})$. However, we still need to satisfy the input conditions for Lemma 21.

This is provided by Claim 23, noting that each $|C_i^*|$ has size $\Omega(\gamma^2\varepsilon^8|N_{\text{cand}}(r)|)$. In Algorithm 4, we uniformly sample $\Theta(\eta^4\gamma^{-2}\varepsilon^{-8})$ vertices from $N_{\text{cand}}(r)$. The expected number of nodes we hit in C_i^* is $\Theta(\eta^4)$. With probability at least $1 - \exp(-\eta)$, we obtain $\eta_0 = \eta^3$ sampled nodes, allowing the algorithm to enumerate all subsets of A_i; at least one run will contain all sampled nodes from C_i^*. By a union bound, with probability at least $1 - \eta\exp(-\eta)$, a call to GenerateCluster will satisfy Lemma 21. Once we make the correct call to GenerateCluster, with probability at least $1 - 2\eta\exp(-2\eta)$, GenerateCluster outputs a correct answer. The high-probability guarantee comes from the fact that Algorithm 4 repeats the whole process $\log n$ times.

To argue about the runtime for finding a small ratio cluster, we use the statements in Theorem 7 about deciding admissibility. We will compute $N_{\text{cand}}(r)$ as follows. We can find $N_{\text{adm}}(r)$ in time $\widetilde{O}(d(r))$. Then we can iterate through all vertices in $N_{\text{adm}}(r)$ and check if they are in $N_{\text{cand}}(r)$. This takes time $\widetilde{O}(d^2(r))$. Since η and ε are constants, the runtime of GenerateClusterBySampling, Algorithm 4, is $\log(n)$ times the runtime of GenerateCluster, Algorithm 5. Here, we compute $\text{EstMarg}(S, t, v)$ for all $v \in N_{\text{cand}}$ for a constant number of constant sized sets S. This takes at most $\tilde{O}(d(r))$ since we only need to iterate over the neighbors of vertices in S. We can compute $\text{cost}(T)$ in time $O(|T| \cdot d(r)) = O(d^2(r))$. Overall, we spend at most $\widetilde{O}(d^2(r))$. □

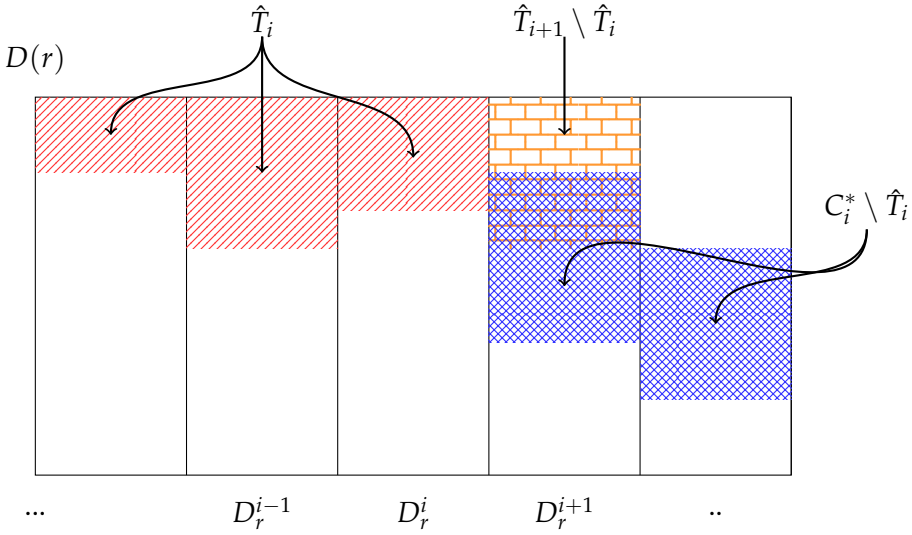

Figure 1: Illustration of the sets $\hat{T}_i$, C_i^*, and Q_i. The rectangle represents $D(r)$, divided into η parts. The red region denotes $\hat{T}_i$, containing all vertices already added to $\hat{T}$. The set C_i^* includes both the red and blue regions. In D_r^{i+1}, the algorithm attempts to include as many vertices as possible in C_i^*; the yellow region represents the newly added vertices in $\hat{T}$. Claim 22 states that the yellow and blue regions have significant overlap.

5.2 Bounding $\Delta(C_{i+1}^*) - \Delta(C_i^*)$

In this section, we will prove Claim 22.

Proof of Claim 22. Let

$$Q_i := \hat{T}_{i+1} \cup \left(C_i^* \cap \bigcup_{j=i+1}^{\eta} D_r^j \right).$$

Recall that

$$C_i^* = \hat{T}_i \cup \operatorname{argmin} \left\{ \operatorname{cost}(\hat{T}_i \cup B) - w(\hat{T}_i \cup B) \;\middle|\; B \subseteq \bigcup_{j=i}^{\eta} D_r^i \right\}.$$

By the optimality of C_{i+1}^*, we know that $\Delta(C_{i+1}^*) \le \Delta(Q_i)$. To bound $\Delta(C_{i+1}^*) - \Delta(C_i^*)$, it is sufficient to prove that

$$\Delta(Q_i) - \Delta(C_i^*) = O(\eta^{-2}\varepsilon^{-8} d_{\mathrm{adm}}(C)).$$

Note that

$$\begin{aligned} \Delta(Q_i) - \Delta(C_i^*) &= \operatorname{cost}(Q_i) - w(Q_i) - \operatorname{cost}(C_i^*) + w(C_i^*) \\ &= \operatorname{cost}(Q_i) - \operatorname{cost}(C_i^*) - w(Q_i \setminus C_i^*) + w(C_i^* \setminus Q_i) \end{aligned}$$

Intuitively, removing one vertex from Q_i will incur $\mathrm{Marginal}(Q_i, v)$ to $\operatorname{cost}(Q_i) - \operatorname{cost}(C_i^*)$, so we can use Marginal value to bound $\operatorname{cost}(Q_i) - \operatorname{cost}(C_i^*)$. More precisely, consider the process that we change Q_i to C_i^*. we first remove nodes from Q_i one by one, at the end, we get the set $Q_i \cap C_i^*$, then we try to add nodes to this set and make the final set C_i^*, $\operatorname{cost}(Q_i) - \operatorname{cost}(C_i^*)$ is bounded by the marginal in each step. Let's consider some moment in this process, we first consider the removal process. assume that at step j, our set is $Q_{i,j}$, then we choose an

arbitrary element from $v \in Q_{i,j} \setminus C_i^*$, and remove v from $Q_{i,j}$, the new set is $Q_{i,j+1}$. At the beginning, we have $Q_{i,1} = Q_i$. By Claim 20, we know that

$$\text{Marginal}(Q_{i,j}, v) - \text{Marginal}(C_i^*, v) \le 2|C_i^* \oplus Q_{i,j}| \le 2|C_i^* \oplus Q_i|$$

so the cost difference to change $Q_{i,j}$ to $Q_{i,j+1}$ is at most

$$\text{cost}(Q_{i,j}) - \text{cost}(Q_{i,j+1}) = \text{Marginal}(Q_{i,j}, v) \le \text{Marginal}(C_i^*, v) + 2|Q_i \oplus C_i^*|.$$

Similarly, in the process of changing $C_i^* \cup Q_i$ to C_i^*, the cost change in each step is bounded by $\text{Marginal}(C_i^*, v) - 2|Q_i \oplus C_i^*|$. So, in order to change Q_i to C_i^*, the total marginal changes

$$\begin{aligned}\text{cost}(Q_i) - \text{cost}(C_i^*) &\le \sum_{v \in Q_i \setminus C_i^*} (\text{Marginal}(C_i^*, v) + 2|Q_i \oplus C_i^*|) - \sum_{v \in C_i^* \setminus Q_i} (\text{Marginal}(C_i^*, v) - 2|C_i^* \oplus Q_i|) \\ &\le \sum_{v \in Q_i \setminus C_i^*} \text{Marginal}(C_i^*, v) - \sum_{v \in C_i^* \setminus Q_i} \text{Marginal}(C_i^*, v) + 2|Q_i \oplus C_i^*|^2\end{aligned}$$

$\Delta(Q_i) - \Delta(C_i^*)$ is bounded by

$$\begin{aligned}\Delta(Q_i) - \Delta(C_i^*) &= \text{cost}(Q_i) - \text{cost}(C_i^*) - w(Q_i \setminus C_i^*) + w(C_i^* \setminus Q_i) \\ &\le \sum_{v \in Q_i \setminus C_i^*} (\text{Marginal}(C_i^*, v) - w(v)) - \sum_{v \in C_i^* \setminus Q_i} (\text{Marginal}(C_i^*, v) - w(v)) + 2|Q_i \oplus C_i^*|^2.\end{aligned}$$

Now we will bound these three parts one by one.

Bound for $|Q_i \oplus C_i^*|^2$. We start by bounding $|Q_i \oplus C_i^*|^2$. The only difference between C_i^* and Q_i is the vertices in D_r^{i+1}, the algorithm replace the elements in $C_i^* \cap D_r^{i+1}$ with $\hat{T}_{i+1} \setminus \hat{T}_i$. So

$$|Q_i \oplus C_i^*|^2 \le |D_r^{i+1}|^2 \le \eta^{-2}|D(r)|^2.$$

Bound for $\sum_{v \in Q_i \setminus C_i^*} (\text{Marginal}(C_i^*, v) - w(v))$. By definition of C_i^*, for any vertex $v \in D_r^{i+1} \setminus C_i^*$, adding v to C_i^* does not yield a strictly better cluster. In particular, for any $v \in D_r^{i+1} \setminus C_i^*$, we have $\text{Marginal}(C_i^*, v) - w(v) \ge 0$. The difficulty of the proof is that $\text{Marginal}(C_i^*, v) - w(v)$ could be very large. Let

$$\ell(v) = \frac{\text{Marginal}(C_i^*, v) - w(v)}{\eta^{-1}|N_{\text{cand}}(r)|}$$

be the ratio of contribution, so if we add v to Q_i, v contributes $\ell(v)\eta^{-1}|N_{\text{cand}}(r)|$ to $\Delta(Q_i) - \Delta(C_i^*)$.

We need to bound the estimation error between EstMarg and Marginal. We provide the following lemma regarding this bound, and its proof is given at the end of this section.

Lemma 24. *Let $\eta_0 = \eta^3$ and $\ell \ge 1$. Consider a vertex v and an arbitrary set of vertices T of size $t = \Omega(\eta)$. Let S be a set consisting of η_0 random samples (with repetition) from T. Let $\tilde{t} \in [t, (1 + \frac{1}{\eta})t]$ Then, with probability at least $1 - 2\exp\left(-2\ell^2\eta\right)$. we have the following inequality holds:*

$$\text{Marginal}(T, v) - \frac{(4+\ell)t}{\eta} \le \text{EstMarg}(S, \tilde{t}, v) \le \text{Marginal}(T, v) + \frac{(4+\ell)t}{\eta} \tag{12}$$

According to Lemma 24, with probability at least $1-2\exp\left(-2(\frac{\ell(v)}{2}+1)^2\eta\right)$, we have

$$\text{EstMarg}(S^i,\tilde{t}_i,v)-\text{Marginal}(C^*,v)\geq -(4+2(\frac{\ell(v)}{2}+1))\eta^{-1}|C_i^*|\geq -(6+\ell(v))\eta^{-1}|N_{\text{cand}}(r)|,$$

and

$$\text{EstMarg}(S^i,\tilde{t}_i,v)+6\eta^{-1}|N_{\text{cand}}(r)|\geq \text{Marginal}(C^*,v)-\ell\eta^{-1}|N_{\text{cand}}(r)|\geq w(v),$$

and Algorithm 5 will not add v to $\hat{T}$. Therefore $v\in Q_i$ with probability at most $2\exp\left(-2(\frac{\ell(v)}{2}+1)^2\eta\right)$, and the expected contribution of v to $\sum_{v\in Q_i\setminus C_i^*}(\text{Marginal}(C_i^*,v)-w(v))$ is at most

$$\begin{aligned}
&2\exp\left(-2(\frac{\ell(v)}{2}+1)^2\eta\right)\cdot(\text{Marginal}(C_i^*,v)-w(v))\\
&\leq 2\ell(v)\exp\left(-2(\frac{\ell(v)}{2}+1)^2\eta\right)\eta^{-1}|N_{\text{cand}}(r)|\\
&\leq 2\exp(-2\eta)\eta^{-1}|N_{\text{cand}}(r)|
\end{aligned}$$

and

$$\mathbb{E}[\sum_{v\in Q_i\setminus C_i^*}(\text{Marginal}(C_i^\star,v)-w(v))]\leq 2\exp(-2\eta)\eta^{-1}|N_{\text{cand}}(r)|\eta^{-1}|D(r)|.$$

By Markov inequality, with probability at most $\exp(-2\eta)$, we have

$$\sum_{v\in Q_i\setminus C_i^*}(\text{Marginal}(C_i^\star,v)-w(v))\geq 2\eta^{-2}|N_{\text{cand}}(r)||D(r)|.$$

Bound for $\sum_{v\in C_i^*\setminus Q_i}(\text{Marginal}(C_i^\star,v)-w(v))$. If $v\in C_i^*\setminus Q_i$, we know that $\text{Marginal}(C_i^*,v)-w(v)\geq 0$. We will distinguish two cases.

1. $\text{Marginal}(C_i^\star,v)-w(v)\leq -12\eta^{-1}|N_{\text{cand}}(r)|$
   Again, we use the same strategy of bounding $\sum_{v\in Q_i\setminus C_i^*}(\text{Marginal}(C_i^*,v)-w(v))$. Let

   $$\ell(v)=\frac{w(v)-\text{Marginal}(C_i^*,v)}{\eta^{-1}|N_{\text{cand}}(r)|}$$

   be the ratio of contribution, so if we decide not to add v to Q_i, v contributes $\ell(v)\eta^{-1}|N_{\text{cand}}(r)|$ to $\Delta(Q_i)-\Delta(C_i^*)$. Note that $\ell(v)\geq 12$. According to Lemma 24, with probability at least $1-2\exp\left(-2(\frac{\ell(v)}{2}-5)^2\eta\right)$, we have

   $$\text{EstMarg}(S^i,\tilde{t}_i,v)-\text{Marginal}(C^*,v)\leq(4+2(\frac{\ell(v)}{2}-5))\eta^{-1}|C_i^*|\leq(\ell(v)-6)\eta^{-1}|N_{\text{cand}}(r)|.$$

   and

   $$\text{EstMarg}(S^i,\tilde{t}_i,v)+6\eta^{-1}|N_{\text{cand}}(r)|\leq \text{Marginal}(C^*,v)+\ell\eta^{-1}|N_{\text{cand}}(r)|\leq w(v),$$

and Algorithm 5 will add v to $\hat{T}$. Therefore $v \notin Q_i$ with probability at most $2\exp\left(-2(\frac{\ell(v)}{2}-5)^2\eta\right)$, and the expected contribution of v to $\sum_{v\in Q_i\setminus C_i^*}(w(v)-\text{Marginal}(C_i^*,v))$ is at most

$$\begin{aligned}
&2\exp\left(-2(\frac{\ell(v)}{2}-5)^2\eta\right)\cdot(w(v)-\text{Marginal}(C_i^*,v))\\
&\le 2\ell(v)\exp\left(-2(\frac{\ell(v)}{2}-5)^2\eta\right)\eta^{-1}|N_{\text{cand}}(r)|\\
&\le 24\exp(-2\eta)\eta^{-1}|N_{\text{cand}}(r)|
\end{aligned}$$

and

$$\mathbb{E}[\sum_{v\in Q_i\setminus C_i^*}(w(v)-\text{Marginal}(C_i^*,v))]\le 24\exp(-2\eta)\eta^{-1}|N_{\text{cand}}(r)|\eta^{-1}|D(r)|.$$

By Markov inequality, with probability at most $\exp(-2\eta)$, we have

$$\sum_{v\in Q_i\setminus C_i^*}(w(v)-\text{Marginal}(C_i^*,v))\ge 24\eta^{-2}|N_{\text{cand}}(r)|\eta^{-1}|D(r)|.$$

With probability at least $1-\exp(-2\eta)$, we have

$$\sum_{v\in Q_i\setminus C_i^*}(\text{Marginal}(C_i^*,v)-w(v))\ge -24\eta^{-2}|N_{\text{cand}}(r)|\eta^{-1}|D(r)|$$

2. $\text{Marginal}(C_i^\star,v)-w(v)\ge -12\eta^{-1}|N_{\text{cand}}(r)|$. Recall that $C_i^*\setminus Q_i\subseteq D_r^{i+1}$. We have,

$$\sum_{v\in Q_i\setminus C_i^*}(\text{Marginal}(Q_i,v)-w(v))\ge -12\eta^{-1}|N_{\text{cand}}(r)|\cdot|D_r^{i+1}|=-12\eta^{-1}|N_{\text{cand}}(r)|\cdot\eta^{-1}|D(r)|.$$

Now, we are ready to give the final bound on $\Delta(Q_i)-\Delta(C_i^*)$. Using the bounds from the three cases above and we have, with probability at least $1-2\exp(-2\eta)$, we have

$$\begin{aligned}
\Delta(Q_i)-\Delta(C_i^*)\le&\sum_{v\in Q_i\setminus C_i^*}(\text{Marginal}(Q_i,v)-w(v))-\sum_{v\in C_i^*\setminus Q_i}(\text{Marginal}(C_i^*,v)-w(v))+|Q_i\oplus C_i^*|^2\\
\le&2\eta^{-2}|N_{\text{cand}}|\cdot|D(r)|+24\eta^{-2}|N_{\text{cand}}|\cdot|D(r)|+12\eta^{-2}|N_{\text{cand}}|\cdot|D(r)|+2\eta^{-2}|D(r)|^2\\
=&O(\eta^{-2}|N_{\text{cand}}|\cdot|D(r)|)
\end{aligned}$$

We still have to bound $|N_{\text{cand}}|\cdot|D(r)|$, which actually is the number of two hop candidates. Fortunately, after preclustering, the number of two hop candidates $|D(v)|\cdot|N_{\text{cand}}(v)|$ can be bounded by following lemma, which is also used in [CLP+24].

Lemma 25 (Lemma 34 of [CLP+24]). *For any vertex r and any ε-large cluster C such that $K(r)\subseteq C\subseteq N_{\text{cand}}(r)$, we have*

$$|D(r)|\cdot|N_{\text{cand}}(r)|=O(\varepsilon^{-8}d_{\text{adm}}(C)).$$

Using Lemma 25, we obtain the claimed bound,

$$\Delta(Q_i)-\Delta(C_i^*)=O(\eta^{-2}|N_{\text{cand}}|\cdot|D(r)|)=O(\eta^{-2}\varepsilon^{-8}d_{\text{adm}}(C)).$$

By the union bound, all bounds hold simultaneously with probability at least $1-2\exp(-2\eta)$. □

Proof of Lemma 24. Recall that $\tilde{t}$ is the guess for size of T.

$$\text{Marginal}(T, v) = \frac{d^+(v) - 1}{2} + |T| - 2d^+(v, T) + \mathbb{1}(v \in T).$$

and

$$\text{EstMarg}(S, \tilde{t}, v) := \frac{d^+(v) - 1}{2} + \tilde{t} - 2\frac{d^+(v, S)}{|S|}\tilde{t}$$

Consider the i-th sampled node u in S, let $X_i = \frac{t}{\eta_0} \cdot \mathbb{1}(uv \in E)$ be a random variable, so $X_i \in \{0, \frac{t}{\eta_0}\}$.

$$\mathbb{E}[\sum_{i=1}^{\eta_0} X_i] = \frac{\eta_0}{|T|} \cdot \frac{t}{\eta_0} \cdot d^+(v, T) = d^+(v, T).$$

By Hoeffding's inequality, we have,

$$\Pr[\left|\sum_{i=1}^{\eta_0}(X_i - \mathbb{E}[X_i])\right| \geq \frac{\ell t}{\eta}] \leq 2\exp\left(-\frac{2(\ell t/\eta)^2}{\eta_0 \cdot (t/\eta_0)^2}\right) \leq 2\exp\left(-2\ell^2\eta\right).$$

Now, consider the difference of Marginal and EstMarg, we have

$$\begin{aligned}
&|\text{EstMarg}(S, \tilde{t}, v) - \text{Marginal}(T, v)\,| \\
&\leq \left|\tilde{t} - 2\frac{d^+(v, S)}{|S|}\tilde{t} - |T| + 2d^+(v, T) - \mathbb{1}(v \in T)\right| \\
&\leq |\tilde{t} - |T|| + 2\frac{d^+(v, S)}{|S|}|\tilde{t} - t| + 2\left|\sum_{i=1}^{\eta_0}(X_i - \mathbb{E}[X_i])\right| + 1 \\
&\leq \frac{t}{\eta} + \frac{2t}{\eta} + \frac{2\ell t}{\eta} + 1 \\
&\leq \frac{(4 + 2\ell)t}{\eta}.
\end{aligned}$$

The last inequality is because $t = \Omega(\eta)$.

□

6 Refinements to Reach Nearly Linear Time

In this section, we present an algorithm to solve cluster LP in nearly linear time (i.e., $\widetilde{O}(m)$), where m is the number of +edges. In a later section, we will present a sublinear-time algorithm. The main theorem regarding the nearly linear algorithm is stated as follows.

Theorem 26 (Nearly Linear Time cluster LP)**.** *Let $\varepsilon > 0$ be a sufficiently small constant and let* cost(OPT) *be the cost of the optimum solution to the given Correlation Clustering instance. Then there is a small $\delta = \text{poly}(\varepsilon)$ such that the following statement holds: One can output a solution $(z_S)_{S \subseteq V}$ to the cluster LP, described using a list of non-zero coordinates, with* obj$(x) \leq (1 + \varepsilon)$cost(OPT) *in expectation such that each coordinate of z is either 0 or at least δ. The running time to compute z is $\widetilde{O}(2^{\text{poly}(1/\varepsilon)}m)$.*

We need to address two problems when aiming for nearly linear time. First, we cannot afford to compute all the clusters C_r, one for each vertex $r \in V$. Instead, we will sample a subset of vertices $U \subseteq V$ such that U is not too large and we can compute a cluster C_r for each

vertex $r \in U$. In particular, we will include each vertex v in U with probability $\log(n)/d(v)$. Since we assume that the optimal solution is ε-large , we will hit every cluster $C \in \mathrm{OPT}$ with a vertex (i.e., $U \cap C \neq \emptyset$) with high probability. Because of this, there will still be a cluster among $\{C_r\}_{r \in U}$ that achieves the ratio R.

Second, we cannot update a cluster C_r as soon as it loses one vertex. However, we can wait until a cluster C_r has lost a constant fraction $\gamma \cdot p(C_r)$ of its probability mass before we update it. If a cluster C_r did not lose this constant fraction, the ratio did not change by too much. Moreover, we can show that if the probability distribution p is a distribution from the MWU Algorithm 2, then $N_{\mathrm{cand}}(r)$ has to lose at least a constant fraction of vertices in order for C_r to lose a constant fraction of probability mass. Thus, we only need to update a cluster C_r at most a constant number of times.

Algorithm 6 (Nearly Linear) Algorithm to find the family $\mathcal{F}$

Let R be the guess for $\mathrm{cover}(\mathrm{OPT})$ such that $R \in [\mathrm{cover}(\mathrm{OPT}), (1+\gamma)\mathrm{cover}(\mathrm{OPT}))$.
$\hat{p} \leftarrow p, \mathcal{F} \leftarrow \emptyset$
for $t = 1, \ldots, \log(n)/\varepsilon^2$ **do**
  Add each vertex v with probability $\frac{1}{d(v)}$ to U.
end for
for all $v \in V$ **do**
  If $\frac{\mathrm{cover}(K(v))}{p(K(v))} \leq (1+6\gamma)\, R$, add $K(v)$ to $\mathcal{F}$ and set $\hat{p}_w = 0$ for all $w \in K(v)$.
  If $p_v \leq \frac{\gamma d_{\mathrm{cross}}(v)}{4 d_{\mathrm{cross}}(V)}$, set $\hat{p}_w = 0$ for all $w \in K(v)$.
end for
for all $u \in U$ **do**
  Find a small ratio cluster C_u such that $K(u) \subseteq C_u \subseteq N_{\mathrm{adm}}(u)$ with vertex weights $\hat{p} > 0$ and target ratio $(1+3\gamma)R$ (Lemma 18).
end for
while $p(\mathcal{F}) \leq \gamma$ **do**
  Choose C with the smallest ratio $\frac{\mathrm{cover}(C)}{\hat{p}(C)}$ among clusters $\{C_v\}_{v \in U}$.
  Remove all w such that $\hat{p}_w = 0$ from C, add the new C to $\mathcal{F}$.
  Set $\hat{p}_v$ to 0 for all $v \in C$.
  for all $v \in U$ **do**
    if $\hat{p}(C_v) \leq (1-\gamma)p(C_v)$ **then**
      Find a new small ratio cluster C_v with vertex weights $\hat{p} > 0$ and target ratio $(1+3\gamma)R$ (Lemma 18).
    end if
  end for
end while
return $\mathcal{F}$

6.1 Approximate Ratio of Algorithm 6

Lemma 27. *Given vertex weights $p_v > 0$, Algorithm 6 finds a family $\mathcal{F} = \{S_1, S_2, ..., S_l \mid S_i \subseteq V\}$ such that,*

1. *for any distinct $S, T \in \mathcal{F}$, $S \cap T = \emptyset$,*
2. $\frac{\mathrm{cover}(\mathcal{F})}{p(\mathcal{F})} \leq (1+8\gamma)\mathrm{cover}(\mathrm{OPT})$,
3. *$p(\mathcal{F})$ is at least γ,*

4. *no $S \in \mathcal{F}$ splits an atom, i.e. $K(v) \subseteq S$ for all vertices $v \in S$.*

Proof. First, observe that Property 3. holds by the condition in the while loop. Property 2. holds since all small ratio clusters $\{C_v\}_{v \in U}$ do not split atoms. Next, we will prove the family $\mathcal{F}$ consists of disjoint sets. Observe that after we added a cluster $C \subseteq V$ to $\mathcal{F}$ we set the weight $\hat{p}_v$ to 0 for all $v \in C$. Note that if the weight $\hat{p}_v$ is set to 0 it remains 0 throughout the execution of the algorithm. Moreover, before we add a cluster $C \subseteq V$ to $\mathcal{F}$ we remove all vertices $v \in C$ with weight $p_v = 0$ from C. Thus, when we add C to $\mathcal{F}$ we have $\hat{p}_v > 0$ for all $v \in C$ and $\hat{p}_v = 0$ for all $v \in S, S \in \mathcal{F}$. Assume that the guess R satisfies $\text{cover}(\text{OPT}) \leq R \leq (1+\gamma)\text{cover}(\text{OPT})$. It remains to prove the bound on the ratio $\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})}$. To do so, we will make use of the following claim.

Claim 28. *For each cluster $C \in \text{OPT}$ with $|C| \geq 2$, $C \cap U \neq \emptyset$ with high probability. Moreover, if there exists a non-singleton atom $K \subseteq C$ then $K \cap U \neq \emptyset$ with high probability.*

Proof. We can assume that the optimal clustering is ε-large with respect to our preclustering. Thus, for any non-singleton atom K we have $|K| \geq O(\varepsilon)d(v)$ for each $v \in K$. Otherwise, v has more than $O(\varepsilon|K|)$ neighbors outside of K. The probability that we don't include a vertex from K in U in one iteration is

$$\prod_{v \in K}\left(1 - \frac{1}{d(v)}\right) \leq \exp\left(-\sum_{v \in K}\frac{1}{d(v)}\right) \leq \exp\left(-O(\varepsilon)\right).$$

Thus, after $\log(n)/\varepsilon^2$ rounds, we include a vertex from K in U with high probability. If $|C| > 1$, then $|C| \geq \varepsilon d(v)$. By the same arguments as above, we will include a vertex from C in U with high probability. ◊

Let $C \in \text{OPT}$. By Claim 28, there exists a vertex $v \in C \cap U$ such that $K = K(v)$ if K is a non-singleton atom in C. Now, we are able to analyze the approximation ratio of Algorithm 6. We first show the existence of a cluster that satisfies the conditions of Lemma 18.

Lemma 29. *In the algorithm 6, if $p(\mathcal{F}) \leq \gamma$, then there always be a cluster $C^* \in \text{OPT}$ such that*

$$\frac{\text{cover}(C^\star) + \gamma^2 d_{\text{adm}}(C^\star)}{\hat{p}(C^\star)} \leq (1+3\gamma)R.$$

Proof. Let $C^\star$ be the cluster among clusters in OPT with the smallest ratio $\frac{\text{cover}(C^\star)}{\hat{p}(C^\star)}$. Notice that the algorithm will set some $\hat{p}_v$ to 0 without adding u to $\mathcal{F}$, let

$$\text{Small} = \{v \in \hat{p}_v \text{ is set to 0 at line 8 in Algorithm 6}\}$$

We have $p(\text{Small}) \leq \sum_{v \in V}\frac{\gamma d_{\text{cross}}(v)}{16 d_{\text{cross}}(V)} \leq \gamma/4$. By the condition in loop 13, we have $\hat{p}(V) = p(V) - p(\mathcal{F}) - p(\text{Small}) \geq 1 - \gamma - \gamma/4 \geq 1 - 1.5\gamma$. Thus,

$$\begin{aligned}(1+3\gamma)R &\geq \frac{1+\gamma}{1-1.5\gamma}R \geq \frac{(1+\gamma)\text{cover}(\text{OPT})}{\hat{p}(V)} \geq \frac{\text{cover}(\text{OPT}) + \gamma^2|E_{\text{adm}}|}{\hat{p}(V)} \\ &= \frac{\sum_{C \in \text{OPT}}(\text{cover}(C) + \gamma^2 d_{\text{adm}}(C))}{\sum_{C \in \text{OPT}}\hat{p}(C)} \geq \frac{\text{cover}(C^\star) + \gamma^2 d_{\text{adm}}(C^\star)}{\hat{p}(C^\star)}.\end{aligned}$$

For the second inequality, we use that $\gamma\,\text{cover}(\text{OPT}) \geq \gamma\,\text{cost}(\text{OPT}) \geq \gamma^2|E_{\text{adm}}|$ because of the preclustering (Theorem 7). The last inequality holds by the definition of $C^\star$. □

Based on Lemma 29, we know that if we use $(1+3\gamma)R$ as the ratio and apply Lemma 18, we can always find a cluster C_v with ratio at most $(1+3\gamma)R$. One issue remains: we might set the $\hat{p}$ value of some nodes in C_v to 0. This is captured by the following invariant.

Lemma 30. *For any C_v added to $\mathcal{F}$ in Algorithm 6, we have*

$$\frac{\text{cover}(C_v)}{\hat{p}(C_v)} \leq (1+5\gamma)R.$$

Proof. When C_v is created, we have

$$\frac{\text{cover}(C_v)}{\hat{p}(C_v)} = \frac{\text{cover}(C_v)}{p(C_v)} \leq (1+3\gamma)R.$$

Note that at this moment, C_v only contains nodes with $\hat{p}$ values greater than 0. Later, the algorithm may add other clusters to $\mathcal{F}$ and set some $\hat{p}$ values to 0, which will increase the ratio of C_v. However, whenever at least a γ-fraction of the $\hat{p}$ value in C_v is decreased to 0, we renew C_v. Thus, we always maintain the bound:

$$\frac{\text{cover}(C_v)}{\hat{p}(C_v)} \leq \frac{\text{cover}(C_v)}{(1-\gamma)p(C_v)} \leq \frac{1+3\gamma}{1-\gamma}R \leq (1+5\gamma)R.$$

□

Now, we only need to show that whenever $p(\mathcal{F}) \leq \gamma$, there always exists a non-empty cluster C among the clusters $\{C_v\}_{v\in U}$ that we can consider. We prove this by contradiction.

Assume that at some iteration of Line 13, we have $p(\mathcal{F}) \leq \gamma$ and $C_v = \emptyset$ for all $v \in U$. By Lemma 29, there exists a cluster C^* such that

$$\text{cover}(C^*) + \gamma^2 d_{\text{adm}}(C^*) \leq (1+3\gamma)R\hat{p}(C^*).$$

By Claim 28, there exists a node $u \in U$ such that $u \in C^*$. Moreover, if C^* contains a non-singleton atom, then u is in the non-singleton atom. Since $\hat{p}$ is non-increasing, we can always find a cluster C_u such that the ratio is at most $(1+3\gamma)R$.

Remember that we might add single vertices with ratio $(1+6\gamma)R$. We can conclude that,

$$\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})} = \frac{\sum_{S\in\mathcal{F}}\text{cover}(S)}{\sum_{S\in\mathcal{F}}p(S)} \leq \frac{\text{cover}(C_v)}{\hat{p}(C_v)} \leq (1+6\gamma)R \leq (1+8\gamma)\,\text{cover}(\text{OPT})$$

□

6.2 Runtime of Algorithm 6

We now prove that Algorithm 6 runs in nearly linear time.

Lemma 31. *For each vertex v, let p_v be the normalized weight $w_v^{(t)}$ during a round $t = 1, \ldots, T$ of the MWU Algorithm 2 for $T = \frac{-\log(\gamma)}{\gamma^2}$. Then, the expected runtime of Algorithm 6 is $\widetilde{O}(m)$.*

Proof. We add each vertex to U with probability at most $\frac{1}{d(v)}\log(n)$. By Lemma 18, computing one C_u for $u \in U$ takes time $\widetilde{O}(d^2(u))$. Computing all $\{C_u\}_{u\in U}$ takes expected time,

$$\sum_{v\in V}\frac{\widetilde{O}(d^2(v))}{d(v)}\log(n) = \widetilde{O}(m).$$

We maintain a priority queue of the clusters $\{C_u\}_{u\in U}$. We can pick the cluster C with the best ratio and remove it from the queue in time $\log(n)$. If we update a C_u, we can insert it into the queue in time $\log(n)$. When we set the weight of a vertex to zero we have to remove this vertex from each cluster C_u. In the following, we will show that v is contained in at most $\log(n)$ clusters C_u in expectation. Thus, this takes time at most $\log(n)$. Moreover, we set the weight of a vertex to zero at most once.

Claim 32. *With high probability, each vertex is contained in at most $O(\log(n))$ clusters from $\{C_u\}_{u\in U}$.*

Proof. If a vertex v is included in the cluster C_u, the v and u have to be connected by an admissible edge. Thus the number of clusters C_u with $v \in C_u$ is bounded by the cardinality of $|U \cap N_{\text{adm}}(v)|$. Since the preclustering is ε-similar , we have that $|N_{\text{adm}}(v)| = d_{\text{adm}}(v) \leq 2\varepsilon^{-3}d(v)$ and $d(u) \geq \frac{1}{2}\varepsilon d(v)$ for each vertex $u \in N_{\text{adm}}(v)$. Thus, we have for the expectation,

$$\mathbb{E}[|U \cap N_{\text{adm}}(v)|] = \sum_{u\in N_{\text{adm}}(v)} \frac{1}{d(u)} \log(n) \leq 2\varepsilon^{-1} \sum_{u\in N_{\text{adm}}(v)} \frac{1}{d(v)} \log(n) \leq 4\varepsilon^{-4} \log(n).$$

The high-probability argument follows from a standard Chernoff bound and the union bound. $\Diamond$

It is left to show that we don't have to update the C_u's too often. To do so, we first proof that each C_u has size proportional to the degree $d(u)$.

Claim 33. *For any cluster $C_u \in N_{\text{cand}}(u)$ such that $\frac{\text{cover}(C_u)}{\hat{p}(C_u)} \leq (1+5\gamma)R$, then*

- *either $|C_u \cap \{v \in C_u \mid \hat{p}_v > 0\}| \geq \gamma^3 d(u)$,*
- *or $\frac{\text{cover}(\{v\})}{p_v} \leq (1+6\gamma)R$ for some $v \in C_u$.*

Proof. Note that cover is monotonic, so we can always remove all vertices with $\hat{p} = 0$ at the beginning and this will only decrease the ratio. We will show that if a cluster C_u has size $|C_u \cap \{v \in C_u \mid \hat{p}_v > 0\}| \leq \gamma^3 d(u)$ then there exists a vertex $v \in C_u$ with

$$\frac{\text{cover}(\{v\})}{p_v} \leq (1+6\gamma)R$$

Remember that the edge (u, v) is admissible for all $v \in C_u$. If C_u contains a non-singleton atom $K(u) \subseteq C_u$, note that $|K(u)| \geq d(u)/2 \geq |C_u|$, this contradicts to $K(v) \subseteq C_u$. so we can assume that C_u does not contain any non-singleton atom and

$$\begin{aligned}
\text{cover}(C_u) &\geq \sum_{v\in C_u} \left(\frac{1}{2}(d(v) - |C_u|) + d_{\text{cross}}(v)\right) \\
&\geq \sum_{v\in C_u} \left(\frac{1}{2}(d(v) - \gamma^2 d(v)) + d_{\text{cross}}(v)\right) \\
&\geq (1-\gamma^2) \sum_{v\in C_u} \text{cover}(\{v\}).
\end{aligned}$$

The second inequality holds since we assumed that $|C_u| \leq \gamma^3 d(u) \leq \gamma^3\varepsilon^{-1}d(v) \leq \gamma^2 d(v)$. Here, we used that the preclustering is ε-similar and (u, v) is admissible. If C_u is considered in the algorithm, then the ratio of C_u is bounded by $(1+5\gamma)R$. One can safely remove u from U otherwise.

$$(1+5\gamma)R \geq \frac{\text{cover}(C_u)}{\hat{p}(C_u)} \geq (1-\gamma^2)\frac{\sum_{v\in C_u} \text{cover}(\{v\})}{\sum_{v\in C_u} \hat{p}_v}.$$

However, than there has to exist a vertex $v \in C_u$ that achieves the ratio

$$\frac{1+5\gamma}{1-\gamma^2}R \leq (1+6\gamma)R$$

This vertex would have been added to $\mathcal{F}$ before $\{C_u\}_{u\in U}$ are computed. Then, $\hat{p}_v = 0$, a contradiction. ◊

The next lemma will give us an upper and lower bound of p value each round, this can help us bound the number of updates for C_u.

Lemma 34. *Invariant: In Algorithm 2, for any $t \in [1, T_{\mathrm{MW}}]$ and any node $u \in V$, we have*

$$\frac{\gamma d_{\mathrm{cross}}(u)}{16 d_{\mathrm{cross}}(V)} \leq p_u^{(t)} \leq \frac{16 d_{\mathrm{cross}}(u)}{d_{\mathrm{cross}}(V)}$$

Proof. We prove the statement by induction. The base case $t = 1$ holds automatically.

Now, assume that the inequality holds at round t. First, note that if a node u is not added to $\mathcal{F}$, then $m_u^{(t)} = -1$; otherwise, we have $1 \leq m_u^{(t)} \leq 1/\gamma$. Therefore, the value $w_u^{(t)}$ will increase by at most $e^{-\gamma^3 m_u^{(t)}} = e^{\gamma^3} \leq 2$ if u is not added to $\mathcal{F}$, and decrease by at most $e^{-\gamma^2} \geq 1/2$ if u is added to $\mathcal{F}$. Consequently, we obtain:

$$p_u^{(t)} \leq p_u^{(t+1)} \leq 4p_u^{(t)}, \quad \text{if } u \text{ is not added to } \mathcal{F},$$

and

$$\frac{p_u^{(t)}}{4} \leq p_u^{(t+1)} \leq p_u^{(t)}, \quad \text{if } u \text{ is added to } \mathcal{F}.$$

It remains to show that: u is never added to $\mathcal{F}$ if $p_u^{(t)} \leq \frac{\gamma d_{\mathrm{cross}}(u)}{4 d_{\mathrm{cross}}(V)}$, and u is always added to $\mathcal{F}$ if $p_u^{(t)} \geq \frac{4 d_{\mathrm{cross}}(u)}{d_{\mathrm{cross}}(V)}$. For the first argument, by Line 8, we always set $\hat{p}_u = 0$ if $p_u^{(t)} \leq \frac{\gamma d_{\mathrm{cross}}(u)}{4 d_{\mathrm{cross}}(V)}$, and we never add a node with $\hat{p}_u = 0$.

The second argument is slightly more involved. A node u is added to $\mathcal{F}$ at Line 7. Note that $R \geq \mathrm{cover}(\mathrm{OPT}) \geq d_{\mathrm{cross}}(V)$. Whenever $p_u^{(t)} \geq \frac{4 d_{\mathrm{cross}}(u)}{d_{\mathrm{cross}}(V)}$, we have

$$\begin{aligned}\frac{\mathrm{cover}(K(u))}{p(K(u))} &\leq \frac{d_{\mathrm{cross}}(K(u))}{4|K(u)| \cdot d_{\mathrm{cross}}(u)/d_{\mathrm{cross}}(V)} \\ &\leq \frac{|K(u)| \cdot d_{\mathrm{cross}}(u)}{4|K(u)| \cdot d_{\mathrm{cross}}(u)/d_{\mathrm{cross}}(V)} \leq \frac{d_{\mathrm{cross}}(V)}{2} \leq R.\end{aligned}$$

Thus, we will add $K(u)$ to $\mathcal{F}$, which completes the proof. □

Now, using Lemma 34, we can give a lower bound and upper bound of the $\hat{p}$ value.

Lemma 35. *Let $N_{\mathrm{cand}}(u) \subseteq K(u) \cup \left(\bigcup_{w\in K(u)} N_{\mathrm{adm}}(w)\right)$ be the candidate set considered in Lemma 18, and define $D(u) = N_{\mathrm{cand}}(u) \setminus K(u)$. If $|D(u)| \leq \frac{d_{\mathrm{adm}}(K(u))}{|K(u)|}$, then for any $v \in N_{\mathrm{cand}}(u)$*

$$p_v = \Omega\left(\frac{\gamma\varepsilon^4|D(u)|}{d_{\mathrm{cross}}(V)}\right), \quad p_v = O\left(\frac{\varepsilon^{-1}d(u)}{d_{\mathrm{cross}}(V)}\right)$$

Proof. For any node $v \in K(u)$, we have

$$p_v = p_u \geq \frac{\gamma d_{\text{cross}}(u)}{16 d_{\text{cross}}(V)} \geq \frac{\gamma d_{\text{cross}}(K(u))}{16|K(u)| d_{\text{cross}}(V)} = \Omega\left(\frac{\gamma \varepsilon^3 d_{\text{adm}}(K(u))}{|K(u)| d_{\text{cross}}(V)}\right) = \Omega\left(\frac{\gamma \varepsilon^3 |D(u)|}{d_{\text{cross}}(V)}\right).$$

The first inequality follows from Lemma 34, the second inequality is based on the definition of $d_{\text{cross}}(u)$, and the first equality follows from Lemma 8.

For any node $v \in D(u)$, we have

$$\begin{aligned} p_v &\geq \frac{\gamma d_{\text{cross}}(v)}{16 d_{\text{cross}}(V)} \geq \frac{\gamma d(v)}{32 d_{\text{cross}}(V)} = \Omega\left(\frac{\gamma \varepsilon d(w)}{d_{\text{cross}}(V)}\right) \text{ (for some } w \in K(u)) \\ &= \Omega\left(\frac{\gamma \varepsilon d_{\text{cross}}(w)}{d_{\text{cross}}(V)}\right) = \Omega\left(\frac{\gamma \varepsilon d_{\text{cross}}(u)}{d_{\text{cross}}(V)}\right) = \Omega\left(\frac{\gamma \varepsilon^4 |D(u)|}{d_{\text{cross}}(V)}\right). \end{aligned}$$

The first inequality follows from Lemma 34, the second inequality is based on the definition of $d_{\text{cross}}(v)$. The first equality follows from ε-similar preclustering. Note that we do not require the edge uv to be an admissible edge; instead, we only require that there exists some node w such that vw is admissible. The second equality follows from ε-similar preclustering and the definition of $d_{\text{cross}}(u)$.

Next, we focus on the upper bound. The upper bound for $v \in K(u)$ holds based on the definition of $d_{\text{cross}}(v)$. From Lemma 34, each node $v \in D(u)$ satisfies

$$\hat{p}_v \leq \frac{16 d_{\text{cross}}(v)}{d_{\text{cross}}(V)} = O\left(\frac{d(v)}{d_{\text{cross}}(V)}\right) = O\left(\frac{\varepsilon^{-1} d(w)}{d_{\text{cross}}(V)}\right) = O\left(\frac{\varepsilon^{-1} d(u)}{d_{\text{cross}}(V)}\right).$$

The first equality follows from the definition of $d_{\text{cross}}(v)$, the second equality follows from the fact that there exists some $w \in K(u)$ such that vw is admissible and ε-similar preclustering, and the last equality holds due to ε-similar preclustering. □

We remark that in the sublinear model, it is impossible to compute $\bigcap_{v \in K(u)} N_{\text{adm}}(v)$ exactly. Therefore, we must relax the definition of N_{cand} in the sublinear model. That's the main reason that we only require $|D(u)| \leq \frac{d_{\text{adm}}(K(u))}{|K(u)|}$ in Lemma 35.

Now, we are able to argue that we do not need to update C_u too many times. By Lemma 18, we know that for each u, we only consider nodes in $N_{\text{cand}}(u)$ for inclusion in C_u.

Lemma 36. *Consider Algorihtm 6, for each $u \in U$, C_u is updated at most $O(\gamma^{-5}\varepsilon^{-5})$ times.*

Proof. Since all nodes in $K(u)$ are always chosen simultaneously, at most one round is required to set the $\hat{p}$ value of nodes in $K(u)$ to zero. Our goal is to bound the number of rounds needed to remove all nodes from $D(u)$. To do this, we derive a lower bound on $\hat{p}(C_u)$.

The first thing is the condition for Lemma 35. Note that we set

$$N_{\text{cand}}(u) = K(u) \cup \left(\bigcap_{w \in K(u)} N_{\text{adm}}(w)\right)$$

if u is not a singleton. The candidate set is the intersection of all admissible neighbors, so we must have

$$|D(u)| \leq \frac{d_{\text{adm}}(K(u))}{|K(u)|}.$$

so from Lemma 35, we know that $\hat{p}_v = \Omega\left(\frac{\gamma\varepsilon^4|D(u)|}{d_{\text{cross}}(V)}\right)$. From Claim 33, we know that $|C_u| \geq \gamma^3 d(u)$. To remove a γ-fraction of the $\hat{p}$ value of C_u, we must remove a total weight of at least

$$\Omega\left(\gamma \cdot \gamma^3 d(u) \cdot \frac{\gamma\varepsilon^4|D(u)|}{d_{\text{cross}}(V)}\right) = \Omega\left(\frac{\gamma^5\varepsilon^4 d(u)|D(u)|}{d_{\text{cross}}(V)}\right).$$

On the other hand, from Lemma 35, we know that $\hat{p}_v = O\left(\frac{\varepsilon^{-1}d(u)}{d_{\text{cross}}(V)}\right)$. If we do not set the $\hat{p}$ value of nodes in $K(u)$ to zero, we must set at least

$$\Omega\left(\frac{\gamma^5\varepsilon^4 d(u)|D(u)|/d_{\text{cross}}(V)}{\varepsilon^{-1}d(u)/d_{\text{cross}}(V)}\right) = \Omega(\gamma^5\varepsilon^5|D(u)|)$$

nodes from $D(u)$ to $\hat{p} = 0$ to reduce γ fractional value of $\hat{p}(C_u)$. Therefore, after at most $O(\gamma^{-5}\varepsilon^{-5})$ rounds, the algorithm will have set all nodes in $D(u)$ to zero. □

6.3 Wrap-Up: Proof of Nearly Linear Time Algorithm for cluster LP

We are now ready to prove Theorem 26.

Proof of Theorem 26. We can obtain the preclustering $(\mathcal{K}, E_{\text{adm}})$ in time $\widetilde{O}(n)$ by Theorem 7. By Lemma 12, we can obtain a solution z to the covering cluster LP after constant rounds $T_{\text{MW}} = O(\text{poly}(1/\varepsilon))$ of Algorithm 2 such that $\text{cover}(z) \leq (1 + O(\gamma))\,\text{cover}(\text{OPT})$. Moreover, z_S is at least $\frac{1}{T_{\text{MW}}}$ for each $S \in \text{supp}(z)$ since each coordinate of the points $z^{(t)}$ is either 0 or at least 1. The solution z is simply the average of all the points $z^{(t)}$. Similar, the solution z does not split atoms since each $z^{(t)}$ does not split atoms by Lemma 11. Furthermore, each vertex v is contained in at most T sets $S \in \text{supp}(z)$ since the solutions $z^{(t)}$ have disjoint support by Lemma 11.

By Lemma 11, in time $O(n)$, we can convert z into a solution $\widetilde{z}$ to the cluster LP such that $\text{cover}(\widetilde{z}) \leq \text{cover}(z)$ and $\widetilde{z}_S \geq \frac{1}{cT_{\text{MW}}}$ for all $S \in \text{supp}(z)$. Observe that for a solution z to the cluster LP, we have $\text{cover}(z) = \text{cost}(z) + d_{\text{cross}}(V)$. Thus,

$$\begin{aligned}
\text{cost}(\widetilde{z}) &\leq \text{cover}(\widetilde{z}) - d_{\text{cross}}(V) \\
&\leq (1 + O(\gamma))\text{cover}(z) - d_{\text{cross}}(V) \\
&\leq (1 + O(\gamma))\text{cover}(\text{OPT}) - d_{\text{cross}}(V) \\
&= (1 + O(\gamma))\text{cost}(\text{OPT}) + O(\gamma)d_{\text{cross}}(V) \\
&\leq (1 + O(\gamma))\text{cost}(\text{OPT}) + O(\varepsilon)\text{cost}(\text{OPT}) \\
&\leq (1 + O(\varepsilon))\text{cost}(\text{OPT}).
\end{aligned}$$

Here, we used that $d_{\text{cross}}(V) = O(\varepsilon^{-12})\text{cost}(\text{OPT})$ by Lemma 8. In each round of Algorithm 2, we have to construct the point $z^{(t)}$. To do so, we will find the family $\mathcal{F}$ from Lemma 27 in expected time $\widetilde{O}(m)$ by Lemma 31. □

□

7 Finding a Partial Clustering with Small Ratio in Sublinear Time

In order to achieve sublinear runtime, we will have to address the following problems.

1. First, it is not clear how to compute $d_{\text{cross}}(v)$ for vertices in non-singleton atoms. However, we can estimate $d_{\text{cross}}(v)$ in sublinear time for all vertices in atoms.

2. We need to implement the algorithm that finds one good ratio cluster, GenerateClusterBySampling (Algorithm 4) in time $\widetilde{O}(d(r))$ instead of $\widetilde{O}(d^2(r))$. More specifically,
   - We need to be able to find $N_{\text{cand}}(r)$,
   - We need to estimate the cost of the cluster T.
3. We need to determine the best solution z since we compute one for each guess R of the optimal cost.

Now, we address each problem one by one.

7.1 Compute $d_{\text{cross}}(v)$

Actually, it is impossible to compute $d_{\text{cross}}(v)$ in sublinear time if $d_{\text{cross}}(v)$ is very small. Consider the following two scenarios, where we have two graphs:

- The first graph consists of two cliques, each containing n nodes.
- The second graph also consists of two cliques, but with a slight modification: we remove one random edge from each clique and then add two edges crossing the cliques, connecting the vertices from which we removed edges.

Suppose we are given one of these two graphs with probability $1/2$. If we could compute $d_{\text{cross}}(v)$ for these graphs in sublinear time, we would be able to distinguish which graph we were given. However, it is not difficult to show that distinguishing between these two graphs requires $\Omega(n^2)$ queries for any (randomized) algorithm. See [AW22] B.1 for similar reduction.

The issue in the above scenario arises when $d_{\text{cross}}(v)$ is too small, making it impossible to compute in sublinear time. To address this, whenever $d_{\text{cross}}(v)$ is too small, we set $K(v)$ as a cluster. For large $d_{\text{cross}}(v)$, instead of computing it exactly, we estimate $d_{\text{cross}}(v)$ within a $(1+\beta)$ approximation, for any small enough constant $\beta = O(\varepsilon^3)$.

This is formally captured by the following lemma.

Lemma 37. *Let $\beta > 0$ be a small enough constant. In time $\widetilde{O}(|K|)$, we can find, with high probability, either an estimate $\hat{d}_{\text{cross}}(K)$ such that*

$$(1-\beta)d_{\text{cross}}(K) \le \hat{d}_{\text{cross}}(K) \le (1+\beta)d_{\text{cross}}(K)$$

or a certificate that K is a cluster in OPT*.*

Proof. Note that $d_{\text{cross}}(K) = 2 \cdot \text{cost}(K) = |E^+(K, V \setminus K)| + 2|E^-(K)|$ by definition. We will estimate both terms in the sum individually.

Claim 38. *If $|E^+(K, V \setminus K)| \ge \beta^2|K|$, then we can find an estimate X in time $\widetilde{O}(|K|)$ such that*

$$|X - |E^+(K, V \setminus K)|| \le \frac{\beta}{4} \cdot |E^+(K, V \setminus K)|$$

with high probability.

Proof. We will sample $s = 32\beta^{-4}|K| \log n$ edges incident on K uniformly at random. To sample one such edge we can sample a vertex proportional to his degree, then an edge adjacent to

the vertex uniformly at random and if the edge is inside K we will discard it with probability $1/2$. Let m_K be the number of edges incident on K. We set the estimate to be

$$X = \frac{m_K}{s} \sum_{i=1}^{s} X_i,$$

where X_i is a random variable indicating whether the i-the edge leaves the atom K. Observe that the estimator is unbiased,

$$\mathbb{E}[X] = \frac{m_K}{s} \sum_{i=1}^{s} \mathbb{E}[X_i] = \frac{m_K}{s} \sum_{i=1}^{s} \frac{|E^+(K, V \setminus K)|}{m_K} = |E^+(K, V \setminus K)|.$$

By our preclusetring, there are at most $\frac{1}{2}|K|^2$ edges leaving the atom K. Thus, $m_K \leq |K|^2$. By Chernoff,

$$\Pr\left[\left|\frac{s}{m_K} X - \frac{s}{m_K} |E^+(K, V \setminus K)|\right| \geq \frac{\beta}{4} \cdot \frac{s}{m_K} |E^+(K, V \setminus K)|\right] \leq 2\exp\left(-\frac{\beta^2}{16} \cdot \frac{s}{m_K} |E^+(K, V \setminus K)|\right)$$
$$= O\left(\frac{1}{n^2}\right).$$

The last inequality holds since $m_K \leq |K|^2$ and $|E^+(K, V \setminus K)| \geq \beta^2 |K|$ by assumption. ◊

We can estimate $E^-(K)$ in a similar manner. The only difference is that we sample $s = 32\beta^{-4}|K| \log n$ pairs uv and check whether $uv \in E^+$.

Recall that in the sublinear model, we can determine whether $uv \in E^+$ in $O(1)$ time. The following lemma states the result, and we omit the proof for brevity.

Claim 39. *If $|E^-(K, V \setminus K)| \geq \beta^2|K|$, then we can find an estimate Y in time $\widetilde{O}(|K|)$ such that*

$$|Y - |E^-(K)|| \leq \frac{\beta}{4} \cdot |E^-(K)|$$

with high probability.

To deal with the case where $|E^+(K, V \setminus K)|$ is small, we will repeat the estimation process $\log(n)$ times to obtain $X^{(1)}, \ldots, X^{(\log n)}$. The final estimate X will be the median of the repetitions $X^{(1)}, \ldots, X^{(\log n)}$. Now, if $|E^+(K, V \setminus K)| \leq \beta^2|K|$ then $X^{(i)} \leq 3\beta^2|K|$ with probability at least $1/3$ by Markov's inequality. Thus, $X \leq 3\beta^2|K|$ with high probability. We proceed similar for Y. Our estimate for $d_{\text{cross}}(K)$ will be $\hat{d}_{\text{cross}}(K) = X + 2Y$.

Claim 40. *If $d_{\text{cross}}(K) \geq \beta|K|$, the the estimate $\hat{d}_{\text{cross}}(K) = X + 2Y$ satisfies*

$$|\hat{d}_{\text{cross}}(K) - d_{\text{cross}}(K)| \leq \beta \cdot d_{\text{cross}}(K),$$

with high probability.

Proof. Since $d_{\text{cross}}(K) = |E^+(K, V \setminus K)| + 2|E^-(K)| \geq \beta|K|$, we know that $|E^+(K, V \setminus K)| \geq \frac{1}{2}\beta|K|$ or $|E^-(K)| \geq \frac{1}{4}\beta|K|$. We will assume that $|E^+(K, V \setminus K)| \geq \frac{1}{2}\beta|K|$. The case where $|E^-(K)| \geq \frac{1}{4}\beta|K|$ is symmetric. If $|E^-(K)| \geq \beta^2|K|$, then the bound follows from Claim 38 and Claim 39. Otherwise, with high probability,

$$\begin{aligned}
|\hat{d}_{\text{cross}}(K) - d_{\text{cross}}(K)| &\leq |X - |E^+(K, V \setminus K)|| + 2|Y - |E^-(K)|| \\
&\leq \frac{\beta}{4}|E^+(K, V \setminus K)| + 8\beta^2|K| \\
&\leq \beta \cdot d_{\text{cross}}(K).
\end{aligned}$$

◊

Claim 41. *If $d_{\text{cross}}(K) \le \beta|K|$ then $\hat{d}_{\text{cross}}(K) \le 6\beta|K|$ with hight probability.*

Proof. Since $d_{\text{cross}}(K) = |E^+(K, V \setminus K)| + 2|E^-(K)| \le \beta|K|$, we know that $|E^+(K, V \setminus K)| \le \beta|K|$ and $|E^-(K)| \le \frac{1}{2}\beta|K|$. Thus, with high probability, $X \le 3\beta|K|$ and $Y \le \frac{3}{2}\beta|K|$. Hence, $\hat{d}_{\text{cross}}(K) = X + 2Y \le 6\beta|K|$ ◊

So far we have shown that we can obtain an estimate $\hat{d}_{\text{cross}}(K)$ such that

- if $d_{\text{cross}}(K) \ge \beta|K|$ then $\hat{d}_{\text{cross}}(K)$ concentrates with high probability,
- and if $d_{\text{cross}}(K) \le \beta|K|$ then $\hat{d}_{\text{cross}}(K) \le 6\beta|K|$ with high probability.

To finish the proof, we need to show that if the estimate is small then K is a cluster in the optimal solution and on the other hand, if the estimate is large then it concentrates.

Claim 42. *If $\hat{d}_{\text{cross}}(K) \le 6\beta|K|$, then K is a cluster in the optimal solution.*

Proof. In the case where $d_{\text{cross}}(K) \ge 12\beta|K|$, we know that $\hat{d}_{\text{cross}}(K) \ge (1-\beta)12\beta|K| > 6\beta|K|$ with high probability. Thus, $d_{\text{cross}}(K) < 12\beta|K|$ with high probability since we assume $\hat{d}_{\text{cross}}(K) \le 6\beta|K|$. In that case, however, by Lemma 8,

$$12\beta|K| > d_{\text{cross}}(K) \ge \Omega(\varepsilon^3 d_{\text{adm}}(K)).$$

In particular, for a small enough β, $d_{\text{adm}}(K) < |K|$. This implies that $\bigcap_{v \in K} N_{\text{adm}}(v) = \emptyset$. Thus, K is a cluster in OPT. ◊

Claim 43. *If $\hat{d}_{\text{cross}}(K) > 6\beta|K|$, then with hight probability*

$$(1-\beta)d_{\text{cross}}(K) \le \hat{d}_{\text{cross}}(K) \le (1+\beta)d_{\text{cross}}(K).$$

Proof. In the case where $d_{\text{cross}}(K) \le \beta|K|$, we know that $\hat{d}_{\text{cross}}(K) \le 6\beta|K|$ with high probability. Thus, $d_{\text{cross}}(K) \ge \beta|K|$ with high probability since we assume $\hat{d}_{\text{cross}}(K) > 6\beta|K|$. To finish the proof, remember that

$$(1-\beta)d_{\text{cross}}(K) \le \hat{d}_{\text{cross}}(K) \le (1+\beta)d_{\text{cross}}(K)$$

with high probability if $d_{\text{cross}}(K) \ge \beta|K|$. ◊

□

Note that, based on Lemma 37, if $d_{\text{cross}}(v)$ is too small, then we make $K(v)$ a cluster and never consider it in cluster LP. For any $K(v)$ that we do consider, we can assume that $d_{\text{cross}}(K) = \Omega(\text{poly}(1/\varepsilon)|K|)$.

7.2 Approximate N_{cand}

The next question is the construction of N_{cand}. In Section 5, we defined the candidate set as the intersection of all admissible neighbors. However, it is again impossible to compute such a candidate set exactly in $\tilde{O}(n)$ time. Instead, we must relax the definition of the candidate set in a way that does not affect the validity of other proofs.

Lemma 44. *Given a vertex r that is part of a non-singleton atom $K(r)$. We can find a set $\hat{D}(r) \subseteq N_{\text{adm}}(r)$ in time $\widetilde{O}(d(r))$ such that*

$$|\hat{D}(r)| \le \frac{2 \cdot d_{\text{adm}}(K)}{|K|}$$

with hight probability.

Proof. We will sample $s = \log n$ vertices $v_1, \ldots, v_s$ from $K(r)$ uniformly at random. Let u be the vertex that minimizes $d_{\mathrm{adm}}(u)$ among the vertices $v_1, \ldots v_s$. We set $\hat{D}(r) = N_{\mathrm{adm}}(u) \cap N_{\mathrm{adm}}(r)$. Observe that

$$\mathbb{E}_{v \sim K}\left[d_{\mathrm{adm}}(v)\right] = \frac{d_{\mathrm{adm}}(K)}{K}.$$

Thus, by Markov's inequality, $d_{\mathrm{adm}}(v_1) \leq \frac{2 \cdot d_{\mathrm{adm}}(K)}{K}$ with probability at least $1/2$. Thus, after $s = \log n$ repetitions there will be a vertex v among $v_1, \ldots v_s$ such that $d_{\mathrm{adm}}(v) \leq \frac{2 \cdot d_{\mathrm{adm}}(K)}{K}$ with high probability. In particular,

$$|\hat{D}(r)| \leq d_{\mathrm{adm}}(v) \leq \frac{2 \cdot d_{\mathrm{adm}}(K)}{|K|}.$$

□

Lemma 19 and Lemma 25 still hold true for the approximate $\hat{N}_{\mathrm{cand}} := K(r) \cup \hat{D}(r)$ where $\hat{D}(r)$ is the estimate from Lemma 44.

Lemma 45. *For any $r \in V$, if $v \in \hat{N}_{\mathrm{cand}}(r)$, then $|\hat{N}_{\mathrm{cand}}(r)| = O(\varepsilon^{-4} d(v))$.*

The proof for Lemma 45 is the same as for Lemma 19 since $\hat{N}_{\mathrm{cand}} \subseteq N_{\mathrm{adm}}(r)$ remains true for the estimate.

Lemma 46. *For any vertex r and any ε-large cluster C such that $K(r) \subseteq C \subseteq \hat{N}_{\mathrm{cand}}(r)$, we have*

$$|\hat{D}(r)| \cdot |\hat{N}_{\mathrm{cand}}(r)| = O(\varepsilon^{-8} d_{\mathrm{adm}}(C)).$$

Proof. Fix $\alpha = \varepsilon^8$. We will distinguish two cases.

1. $|K(r)| \geq \alpha |\hat{N}_{\mathrm{cand}}(r)|$. In this case, we can show the required inequality immediately because the estimate satisfies $|K(r)| \cdot |\hat{D}(r)| \leq 2 d_{\mathrm{adm}}(K(r))$. In particular,

$$d_{\mathrm{adm}}(C) \geq d_{\mathrm{adm}}(K(r)) \geq \frac{1}{2}|K(r)| \cdot |\hat{D}(r)| \geq \frac{\alpha}{2}|\hat{N}_{\mathrm{cand}}(r)| \cdot |\hat{D}(r)|.$$

2. $|K(r)| \leq \alpha |\hat{N}_{\mathrm{cand}}(r)|$. We have by definition of $\hat{N}_{\mathrm{cand}}$ that $|\hat{N}_{\mathrm{cand}}(r)| \leq |K(r)| + |N_{\mathrm{adm}}(r)| = O(\varepsilon^{-3} d(r))$. Since C is ε-large, $|C| \geq \varepsilon d(r) \geq \Omega(\varepsilon^4 |N_{\mathrm{cand}}(r)|)$

$$\begin{aligned}
d_{\mathrm{adm}}(C) &\geq d_{\mathrm{adm}}(C \setminus K(r)) \\
&\geq (|C| - |K(r)|) \cdot |C| \\
&\geq \Omega((\varepsilon^4 |\hat{N}_{\mathrm{cand}}(r) - |K(r)|) \cdot \varepsilon^4 \hat{N}_{\mathrm{cand}}(r)) \\
&\geq \Omega((\varepsilon^4(\varepsilon^4 - \alpha)|\hat{N}_{\mathrm{cand}}(r)|^2)) \\
&\geq \Omega(\varepsilon^8 |\hat{N}_{\mathrm{cand}}(r)|^2).
\end{aligned}$$

□

7.3 Estimate the cost of a cluster T

We are given a vertex r and a cluster $T \subseteq \hat{N}_{\mathrm{cand}}(r)$ that contains at most one non-singleton atom $K := K(r)$. We want to find a good estimate of the cost(T). If T doesn't contains a non-singleton atom K, we will define $K = \emptyset$. We can write the cost of the cluster T as

$$\mathrm{cost}(T) = \mathrm{cost}(K) + \Delta(T).$$

Here, $\Delta(T)$ is the change in cost when adding the missing vertices from $T \setminus K$ to K. We can write $\Delta(T)$ as the sum over the individual contributions of the vertices in $T \setminus K$,

$$\Delta(T) = \sum_{v \in T \setminus K} \Delta(T, v).$$

where $\Delta(T, v)$ is given as

$$\Delta(T, v) = \frac{d^+(v) + d^-(v, T) + d^-(v, K) - d^+(v, T) - d^+(v, K)}{2}$$

We already have a good estimate for $\text{cost}(K) = d_{\text{cross}}(K)$ by Lemma 37. We can also obtain a good estimate for $\Delta(T)$.

Lemma 47. *For any small enough constant $\beta > 0$, given a vertex r and a cluster $T \subseteq \hat{N}_{\text{cand}}(r)$ that contains at most one non-singleton atom $K := K(r)$. Assume there exists an ε-large cluster C with $K(r) \subseteq C \subseteq \hat{N}_{\text{cand}}(r)$. We can find an estimate $\hat{\Delta}(T)$ in time $\widetilde{O}(d(r)\varepsilon^{20}/\beta)$ such that,*

$$|\hat{\Delta}(T) - \Delta(T)| \leq \beta \cdot \min\{d_{\text{adm}}(C), d_{\text{cross}}(T \setminus K)\}$$

with hight probability.

Proof. To estimate $\Delta(T)$ we sample $s = \varepsilon^{-20}\beta^{-2} \log n$ vertices $v_1, \ldots, v_s$ from $T \setminus K$ uniformly at random. We set the estimate to be

$$\hat{\Delta}(T) = \frac{|T \setminus K|}{s} \sum_{i=1}^{s} \Delta(T, v_i).$$

Note that the estimator is unbiased,

$$\mathbb{E}[\hat{\Delta}(T)] = \frac{|T \setminus K|}{s} \sum_{i=1}^{s} \mathbb{E}[\Delta(T, v_i)] = \sum_{v \in T \setminus K} \Delta(T, v) = \Delta(T).$$

Observe that $|\Delta(T, v)| \leq d(v) + |T| \leq d(v) + |\hat{N}_{\text{cand}}(r)| = O(\varepsilon^{-2}|\hat{N}_{\text{cand}}(r)|)$. Here, we used that the ε-large cluster C is a subset of $\hat{N}_{\text{cand}}(r)$, in particular, $|\hat{N}_{\text{cand}}(r)| \geq |C| \geq \varepsilon d(r)$. By Hoeffdings inequality,

$$\begin{aligned}
\Pr\left[\left|s \cdot \hat{\Delta}(T) - s \cdot \Delta(T)\right| \geq s \cdot \varepsilon^8 \beta \cdot |\hat{N}_{\text{cand}}(r)| \cdot |\hat{D}(r)|\right] &\leq 2\exp\left(-\Omega\left(\frac{s^2\varepsilon^{16}\beta^2 \cdot |\hat{N}_{\text{cand}}(r)|^2 \cdot |\hat{D}(r)|^2}{s \cdot \varepsilon^{-4}|\hat{N}_{\text{cand}}(r)|^2 \cdot |C \setminus K|^2}\right)\right) \\
&\leq 2\exp\left(-\Omega(s \cdot \varepsilon^{20}\beta^2)\right) \\
&= O\left(\frac{1}{n^2}\right)
\end{aligned}$$

The last inequality holds since $T \setminus K \subseteq \hat{D}(r)$. Remember that $|\hat{N}_{\text{cand}}(r)| \cdot |\hat{D}(r)| = O(\varepsilon^{-8} d_{\text{adm}}(C))$ by Lemma 46.
Similar, $|\Delta(C, v)| \leq d(v) + |C| \leq d(v) + |\hat{N}_{\text{cand}}(r)| = O(\varepsilon^{-3} d(r))$ by Lemma 45. Note that $d_{\text{cross}}(v) = d(v) = \Omega(\varepsilon d(r))$ for all $v \in T \setminus K$. Thus, $d_{\text{cross}}(T \setminus K) = \Omega(|T \setminus K| \cdot \varepsilon d(r))$ By Hoeffdings inequality,

$$\begin{aligned}
\Pr\left[\left|s \cdot \hat{\Delta}(T) - s \cdot \Delta(T)\right| \geq s \cdot \beta \cdot d_{\text{cross}}(T \setminus K)\right] &\leq 2\exp\left(-\frac{s^2\beta^2 d^2_{\text{cross}}(T \setminus K)}{s \cdot \varepsilon^{-6} \cdot d^2(r) \cdot |T \setminus K|^2}\right) \\
&\leq 2\exp\left(-\Omega\left(\frac{s \cdot |T \setminus K|^2 \cdot \beta^2 \cdot \varepsilon^2 \cdot d^2(r)}{|T \setminus K|^2 \cdot \varepsilon^{-6} d^2(r)}\right)\right) \\
&= O\left(\frac{1}{n^2}\right).
\end{aligned}$$

□

Now, we are ready to show that we can combine $\hat{d}_{\text{cross}}(K)$ and $\hat{\Delta}(T)$ to get a good estimate for $\widehat{\text{cost}}(T)$.

Lemma 48. *For any small enough constant $\beta > 0$, given a vertex r and a cluster $T \subseteq \hat{N}_{\text{cand}}(r)$ that contains at most one non-singleton atom $K := K(r)$, we can find an estimate $\widehat{\text{cost}}(T)$ in time $\widetilde{O}(d(r))$ such that,*

$$|\widehat{\text{cost}}(T) - \text{cost}(T)| \leq \beta \cdot d_{\text{cross}}(T)$$

with hight probability.

Proof. By Lemma 37 and Lemma 47, we can obtain estimates $\hat{d}_{\text{cross}}(K)$ and $\hat{\Delta}(T)$ respectively. We will set $\hat{\text{cost}}(T) = \hat{d}_{\text{cross}}(K) + \hat{\Delta}(T)$. We have by the guarantees provided,

$$\begin{aligned}
|\widehat{\text{cost}}(T) - \text{cost}(T)| &\leq |\hat{d}_{\text{cross}}(K) - d_{\text{cross}}(K)| + |\hat{\Delta}(T) - \Delta(T)| \\
&\leq \beta \cdot (d_{\text{cross}}(K) + d_{\text{cross}}(T \setminus K)) \\
&= \beta \cdot d_{\text{cross}}(T).
\end{aligned}$$

□

7.4 Finding one small ratio cluster in sublinear time

Lemma 49. *Suppose we are given a graph $G = (V, E)$, vertex weights $\hat{p}$, a target ratio R, a vertex r and the set of vertices $N_{\text{adm}}(r)$.*

(i) Assume there exists a cluster C be an ε-large cluster with $K(r) \subseteq C \subseteq N_{\text{cand}}(r)$ such that $\text{cover}(C) + \gamma^2 d_{\text{adm}}(C) \leq R \cdot \hat{p}(C)$.

(ii) Assume that $\text{cover}(\{v\}) > R \cdot \hat{p}(\{v\})$ for all $v \in C$.

Then, with high probability, in time $\widetilde{O}(d(r))$, we can find a cluster $C_r \subseteq N_{\text{cand}}(r)$ such that,

$$\text{cover}(C_r) \leq R \cdot \hat{p}(C_r).$$

Moreover, C_r does not split atoms and contains exactly one non-singleton atom $K(r) \subseteq C_r$ iff $K(r)$ is a non-singleton atom.

Algorithm 7 GenerateClusterBySampling($G, N_{\text{adm}}(r), \hat{N}_{\text{cand}}(r), w, r, R$)

Input: The graph G, $K(r)$, $N_{\text{adm}}(r)$, $\hat{N}_{\text{cand}}(r)$, w, r, ratio R.
Output: A small ratio cluster $\hat{T}$ if r satisfies Assumption (i) from Lemma 18.
Repeat the following steps $O(\log n)$ times.
for i from 1 to η **do**
    Uniformly sample $\Theta(\eta^4 \gamma^{-2} \varepsilon^{-8})$ vertices from $\hat{N}_{\text{cand}}(r)$ with replacement.
    Let the sample set be A_i. {A_i may contain some element multiple times.}
end for
$\hat{D}(r) \leftarrow \hat{N}_{\text{cand}}(r) \setminus K(r)$
$\mathcal{T} \leftarrow \emptyset$
for every $(S^1, S^2, ..., S^\eta) \subset (A_1, A_2, ..., A_\eta)$ such that $|S^i| \leq \eta$, where $i \in [\eta]$ **do**
    for every $(\tilde{t}_1, \tilde{t}_2, ..., \tilde{t}_\eta) \in (L(r), L(r), ...L(r))$, where $\tilde{t}_j \in L(r)$ for $j \in [\eta]$ **do**
        Add $T \leftarrow \text{GenerateCluster}(r, D(r), S^1, \ldots, S^\eta, \tilde{t}_1, \ldots, \tilde{t}_\eta)$ to $\mathcal{T}$.
    end for
end for
Compute the estimate $\hat{\Delta}(T)$ for each $T \in \mathcal{T}$ according to Lemma 47.
return $T \in \mathcal{T}$ that minimizes $\hat{\Delta}(T)$.

Algorithm 8 GenerateCluster$(r, \hat{D}(r), S^1, \ldots, S^\eta, \tilde{t}_1, \ldots, \tilde{t}_\eta)$

$T \leftarrow K(r), \hat{T}_1 \leftarrow K(r)$
Let $D_r^1, \ldots, D_r^\eta$ be an arbitrary partition of the vertices of $\hat{D}(r)$ into equal-size parts
for all $i = 1, \ldots, \eta$ **do**
  for all $v \in D_r^i$ **do**
    if $\text{EstMarg}(S^i, \tilde{t}_i, v) + 6\eta^{-1}|\hat{N}_{\text{cand}}(r)| \leq w(v)$ **then**
      $T \leftarrow T \cup \{v\}$
    end if
  end for
  $\hat{T}_{i+1} \leftarrow T$
end for

Proof. The proof is almost identical to that of Lemma 18. The only difference is that we use $\hat{\Delta}(T)$ to choose T. From Lemma 47, we know that the estimation introduces at most an error of $O(\beta \cdot d_{\text{adm}}(C))$.

The first requirement of the lemma provides a slackness of $\gamma^2 d_{\text{adm}}(C)$. As long as we set $\beta = O(\gamma^2)$, the estimation error remains within the allowed slackness.

□

7.5 Determine the correct guess R of the optimal cost

Our final challenge is to determine the best solution z among the solutions we computed for each guess R. Since we iterate over different granularities of R in the range $[1, n^2]$, we know that one of these values satisfies $R \in [\text{cover}(\text{OPT}), (1+\gamma)\text{cover}(\text{OPT})]$. Thus, we know that for this guess R, we will compute a solution z with small cost. To determine the this solution (or any other with a smaller cost), we will use Lemma 48 to estimate the cost. After applying Algorithm 1, we obtain a solution $\{z_S\}$, where each nonzero z_S is at least some constant. We can estimate $\text{cost}(S)$ for each nonzero z_S, and since each node appears in only a constant number of sets S, the total running time for estimating all $\text{cost}(S)$ values is $\tilde{O}(n)$.

On the other hand, each node contributes to the estimation error of $\beta \cdot d_{\text{cross}}(v)$, leading to a total error of $\beta \cdot d_{\text{cross}}(V)$. As long as β is a sufficiently small constant, we can determine a solution z that is a good approximation.

8 MPC Implementation

In this section, we present an algorithm to solve cluster LP in the MPC model. Our goal is to establish the following theorem for the MPC model.

Theorem 1 (Efficient cluster LP)**.** *Let $\varepsilon, \delta > 0$ be small enough constants and let* OPT *be the cost of the optimum solution to the given Correlation Clustering instance. Then there is a small $\Delta = \text{poly}(\varepsilon)$ such that the following statement holds. One can output a solution $(z_S)_{S \subseteq V}$ to the cluster LP with $\text{obj}(x) \leq (1+\varepsilon)\text{OPT}$ in expectation, described using a list of non-zero coordinates such that each coordinate of z is either 0 or at least Δ. In the various models, the respective procedure has the following attributes.*

- *(Sublinear model) The running time to compute z is $\widetilde{O}(2^{\text{poly}(1/\varepsilon)}n)$.*
- *(MPC model) It takes $2^{\text{poly}(1/\varepsilon)}$ rounds with $O(n^\delta)$ memory per machine and total memory $\tilde{O}(\text{poly}(\frac{1}{\varepsilon})m)$, or takes $\text{poly}(\frac{1}{\varepsilon})$ rounds with $O(n^\delta)$ memory per machine and total memory $\tilde{O}(2^{\text{poly}(1/\varepsilon)}m)$.*

While Algorithm 2 is already well-parallelized and runs in $O\left(\frac{\log(1/\gamma)}{\gamma^4}\right)$ rounds, the algorithm for finding a disjoint family of clusters is not well-parallelized. In Algorithm 6, clusters are identified sequentially and added one by one to the final output set $\mathcal{F}$. While the running time is efficient, the approach lacks parallelism. To address this, we need a method that selects multiple good ratio clusters C_u simultaneously. The MPC algorithm is presented in Algorithm 9.

Algorithm Description Algorithm 9 first removes all nodes whose ratio is either too large or too small. For nodes with a large ratio, we add them to $\mathcal{F}$ (Line 3), and for nodes with a small ratio, we remove them by setting their $\hat{q}$ values to 0 (Line 4).

Next, we attempt to find a disjoint set of clusters C_u such that, in each round, the sum of their p-values is sufficiently large. This set can be found with constant probability, so we repeat this process for $\Theta(\log n)$ rounds (Line 24).

In each round, we add a cluster center to U with probability proportional to its degree. Thus, for each cluster in the optimal solution, at least one node is chosen into U. We then consider the candidate sets of these chosen nodes: if a node appears in the candidate sets of two chosen nodes, we remove it in this round by setting its $\hat{p}$ value to 0. For each chosen node, as long as it retains enough fractional modes that have not been removed, we apply Lemma 18 to find a good ratio cluster. We set the parameter as $\gamma' = \Theta(\gamma^4\varepsilon^5)$.

Algorithm 9 MPC Algorithm to find the family $\mathcal{F}$

1: Let R be the guess for cover(OPT) such that $R \in [\text{cover}(\text{OPT}), (1+\frac{\gamma}{2})\text{cover}(\text{OPT}))$.
2: $\hat{q} \leftarrow p, \mathcal{F} \leftarrow \emptyset$
3: **for all** $v \in V$ **do**
4: If $\frac{\text{cover}(K(v))}{p(K(v))} \leq (1+6\gamma)\,R$, add $K(v)$ to $\mathcal{F}$ and set $\hat{q}_w = 0$ for all $w \in K(v)$.
5: If $p_v \leq \frac{\gamma d_{\text{cross}}(v)}{4d_{\text{cross}}(V)}$, set $\hat{q}_w = 0$ for all $w \in K(v)$.
6: **end for**
7: **while** $p(\mathcal{F}) \leq \gamma$ **do**
8: **for** $t = 1, \ldots, \Theta(\log n/(\varepsilon^5\gamma'))$ **do**
9: $\hat{p} \leftarrow \hat{q}, \mathcal{F}_t \leftarrow \emptyset$
10: $\hat{p}_v \leftarrow 0$, for all nodes v in $\mathcal{F}$
11: Add each vertex v with probability $\frac{\varepsilon^4\gamma'}{24d(v)}$ to U.
12: Mark all nodes in $D(u)$, for every $u \in U$.
13: **for all** $u \in U$ **do**
14: Let $Remove(u) = \{w \in D(u) \mid w \text{ gets more than one mark}\}$
15: If $\hat{q}(Remove(u)) \geq \gamma'\hat{q}(D(u))$, then remove u from U.
16: set $\hat{p}_w \leftarrow 0$ for all $w \in Remove(u)$.
17: **end for**
18: Let $\widetilde{U}$ be the U set.
19: **for** $u \in \widetilde{U}$ **do**
20: Find a good ratio cluster C_u such that $K(u) \subseteq C_u \subseteq N_{\text{adm}}(u)$ with vertex weights $\hat{p} > 0$ and target ratio $(1+5\gamma)R$ (Lemma 18).
21: add C_u to $\mathcal{F}_t$
22: **end for**
23: **end for**
24: $T_{\text{MPC}} \leftarrow \text{argmax}_t\, p(\mathcal{F}_t)$
25: add $\mathcal{F}_{T_{\text{MPC}}}$ to $\mathcal{F}$ and sets $\hat{q}_v = 0$ for all $v \in \mathcal{F}_{T_{\text{MPC}}}$
26: **end while**
27: **return** $\mathcal{F}$

8.1 Approximate Ratio of Algorithm 9

Our main lemma regarding approximate ratio is given as follows,

Lemma 50. *Given vertex weights $p_v > 0$, Algorithm 9 finds a family $\mathcal{F} = \{S_1, S_2, ..., S_l \mid S_i \subseteq V\}$ such that,*

1. *for any distinct $S, T \in \mathcal{F}$, $S \cap T = \emptyset$,*
2. *$\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})} \leq (1 + 8\gamma)\text{cover}(\text{OPT})$,*
3. *$p(\mathcal{F})$ is at least γ,*
4. *no $S \in \mathcal{F}$ splits an atom, i.e. $K(v) \subseteq S$ for all vertices $v \in S$.*

Proof. First, observe that Property 3 holds due to the condition in the while loop. Property 2 holds because all small-ratio clusters $\{C_v\}_{v \in U}$ do not split atoms.

Next, we prove that the family $\mathcal{F}$ consists of disjoint sets. Note that in each round, when we consider the candidate set, if a node appears in two candidate sets, we set its $\hat{p}$ value to zero, ensuring that it is not added to the final cluster.

It remains to bound the ratio $\frac{\text{cover}(\mathcal{F})}{p(\mathcal{F})}$, which follows directly from Lemma 18. □

Now, we only need to show that Algorithm 9 terminates.

8.2 Number of Iterations of Algorithm 9

In this section, we show that the for loop (Line 7 to Line 24) terminates in $O(\text{poly}(1/\varepsilon))$ rounds. More precisely, we prove the following:

Lemma 51. *Consider one round of execution of Algorithm 9 from Line 7 to Line 24. Then, with high probability, we have*

$$p(\mathcal{F}_{T_{\text{MPC}}}) = \Omega(\gamma^{10}\varepsilon^{14}).$$

Consequently, the while loop (Line 6) will be executed at most $O(1/(\gamma^{10}\varepsilon^{14}))$ rounds with high probability.

We start our proof by showing that each round for any cluster $C \in \text{OPT}$, with constant probability, we will add at least one node to U and if C contains a non-singleton atom, we will include at least one node from the non-singleton atom with constant probability.

Claim 52. *For each cluster $C \in \text{OPT}$ with $|C| \geq 2$, we have $C \cap \widetilde{U} \neq \emptyset$ with probability at least $\Omega(\varepsilon^5\gamma')$. Moreover, if there exists a non-singleton atom $K \subseteq C$ then $K \cap \widetilde{U} \neq \emptyset$ with probability at least $\Omega(\varepsilon^5\gamma')$.*

Proof. Recall the $(\mathcal{K}, E_{\text{adm}})$ is ε-similar preclustering. Thus, for any non-singleton atom K we have $|K| \geq d(v)/2$ for each $v \in K$. On the other hand, recall that OPT is ε-large cluster, so $|C| \geq \varepsilon d(v)$, and $|C| \leq d_{\text{adm}}(v) + |K(v)| \leq 3\varepsilon^{-3}d(v)$.

At line 11, we will add node u to U with probability $\frac{\varepsilon^4\gamma'}{24d(u)}$. Then, at line 14, we will remove the node from U if γ' fractional of $\hat{q}(D(u))$ can be considered in other candidate sets. For a given node $u \in C$, let X_u be the indicator variable that u is the only node in C that is added to U at Line 11 and is still in $\widetilde{U}$ after line 14. Let A_u be the indicator variable that u is the only node in C that is added to U at Line 11 and B_u be the total $\hat{p}$ value that are in the $\mathit{Remove}(u)$ set. so

$$\begin{aligned}\Pr[X_u = 1] &= \Pr[A_u = 1 \cap B_u < \gamma'\hat{p}(D(u))] \\ &= \Pr[B_u < \gamma'\hat{p}(D(u)) \mid A_u = 1] \cdot \Pr[A_u = 1]\end{aligned}$$

For A_u, we know that only u is added to U, so

$$\Pr[A_u = 1] = \frac{\varepsilon^4\gamma'}{24d(u)} \prod_{v \in C \setminus \{u\}} \left(1 - \frac{\varepsilon^4\gamma'}{24d(v)}\right)$$
$$\geq \frac{\varepsilon^4\gamma'}{24\varepsilon^{-1}|C|} \left(1 - \frac{\varepsilon^4\gamma'}{8|C|\varepsilon^3}\right)^{|C|} \geq \frac{\varepsilon^5\gamma'}{48|C|}$$

The first inequality is because $\varepsilon^3|C|/3 \leq d(v) \leq \varepsilon^{-1}|C|$. Now condition on $A_u = 1$, we calculate the expected value of B_u,

$$\mathbb{E}[B_u \mid A_u = 1] \leq \sum_{v \in D(u)} \hat{p}_v \sum_{w \in N_{\text{adm}}(v)} \frac{\varepsilon^4\gamma'}{24d(w)} \leq \sum_{v \in D(u)} \hat{p}_v \sum_{w \in N_{\text{adm}}(v)} \frac{\varepsilon^4\gamma'}{12\varepsilon d(v)}$$
$$\leq \sum_{v \in D(u)} \hat{p}_v \cdot d_{\text{adm}}(v) \frac{\varepsilon^4\gamma'}{12\varepsilon d(v)} \leq \sum_{v \in D(u)} \hat{p}_v \cdot 2\varepsilon^{-3} d(v) \frac{\varepsilon^4\gamma'}{12\varepsilon d(v)}$$
$$\leq \sum_{v \in D(u)} \frac{\gamma'\hat{p}_v}{6} \leq \gamma'\hat{p}(D(u))/3$$

Then by markov's inequality, we have

$$\Pr[B_u \geq \gamma'\hat{p}(D(u)) \mid A_u = 1] \leq \frac{1}{2}$$

Combining these two points together, we have

$$\Pr[X_u = 1] \geq \frac{\varepsilon^5\gamma'}{96|C|}$$

and $C \cap U \neq \emptyset$ with probability at least

$$\sum_{u \in C} \Pr[X_u = 1] \geq |C| \cdot \frac{\varepsilon^5\gamma'}{96|C|} = \Omega(\varepsilon^5\gamma')$$

The proof for $K \cap \widetilde{U} \neq \emptyset$ is almost identical, so we omit the details here. However, we note that whenever $K \cap \widetilde{U} \neq \emptyset$, we always have $|K \cap \widetilde{U}| = 1$.

This follows from the fact that if $|K \cap U| \geq 2$, then all nodes in $D(K)$ will receive two marks, causing $D(K)$ to be added to the *Remove* set, which in turn removes $K \cap U$ from U. ♢

Consider one iteration Line 7 of Algorithm 9, let

$$\mathcal{C}^*_{\leq(1+4\gamma)R} = \{C^* \in \text{OPT} \mid \frac{\text{cover}(C^*) + \gamma^2 d_{\text{adm}}(C^*)}{\hat{q}(C^*)} \leq (1+4\gamma)R\}$$

be the set of clusters from optimal clustering such that the ratio is at most $(1+4\gamma)R$. We will first show that $\mathcal{C}^*_{\leq(1+4\gamma)R}$ has a large $\hat{q}$ value.

Lemma 53. *Consider one iteration of Algorithm 9, if $p(\mathcal{F}) \leq \gamma$, then we have*

$$\hat{q}(\mathcal{C}^*_{\leq(1+4\gamma)R}) \geq \gamma$$

Proof. Notice that the algorithm will set some $\hat{q}_v$ to 0 without adding u to $\mathcal{F}$, let

$$\text{Small} = \{v \in \hat{q}_v \text{ is set to 0 at line 4 in Algorithm 9}\}$$

We have $p(\text{Small}) \leq \sum_{v \in V} \frac{\gamma d_{\text{cross}}(v)}{16 d_{\text{cross}}(V)} \leq \gamma/4$. Based on the assumption of $p(\mathcal{F}) \leq \gamma$, we have $\hat{q}(V) = p(V) - p(\mathcal{F}) - p(\text{Small}) \geq 1 - 1.25\gamma$. Thus,

$$\begin{aligned}(1+2.5\gamma)R &\geq \frac{1+\gamma}{1-1.25\gamma} R \geq \frac{(1+\gamma)\text{cover}(\text{OPT})}{\hat{q}(V)} \\ &\geq \frac{\text{cover}(\text{OPT}) + \gamma^2 |E_{\text{adm}}|}{\hat{q}(V)} = \frac{\sum_{C \in \text{OPT}} (\text{cover}(C) + \gamma^2 d_{\text{adm}}(C))}{\sum_{C \in \text{OPT}} \hat{q}(C)}.\end{aligned}$$

For the second inequality, we use that γ cover(OPT) $\geq \gamma$ cost(OPT) $\geq \gamma^2 |E_{\text{adm}}|$ because of the preclustering (Theorem 7). On the other hand, if $\hat{q}(\mathcal{C}_{\leq(1+4\gamma)R}) < \gamma$, then consider $C \in \text{OPT}$ whose ratio is at least $(1+4\gamma)R$, we have

$$\begin{aligned}&\frac{\sum_{C \in \text{OPT}} (\text{cover}(C) + \gamma^2 d_{\text{adm}}(C))}{\sum_{C \in \text{OPT}} \hat{q}(C)} \\ \geq\ & \frac{\sum_{C \in \text{OPT} \setminus C_{\leq(1+4\gamma)R}} (\text{cover}(C) + \gamma^2 d_{\text{adm}}(C))}{\sum_{C \in \text{OPT}} \hat{q}(C)} \\ \geq\ & \frac{\sum_{C \in \text{OPT} \setminus C_{\leq(1+4\gamma)R}} (\text{cover}(C) + \gamma^2 d_{\text{adm}}(C))}{\sum_{C \in \text{OPT}} \hat{q}(C)} \\ \geq\ & \frac{\sum_{C \in \text{OPT} \setminus C_{\leq(1+4\gamma)R}} (\text{cover}(C) + \gamma^2 d_{\text{adm}}(C))}{\sum_{C \in \text{OPT} \setminus C_{\leq(1+4\gamma)R}} \hat{q}(C)} \cdot \frac{\sum_{C \in \text{OPT} \setminus C_{\leq(1+4\gamma)R}} \hat{q}(C)}{\sum_{C \in \text{OPT}} \hat{q}(C)} \\ \geq\ & (1+4\gamma)R \cdot (1 - \frac{\gamma}{1-1.5\gamma}) > (1+2.5\gamma)R\end{aligned}$$

which contradicts to the fact that the total ratio is at most $(1+2.5\gamma)R$. So, we conclude that $\hat{q}(\mathcal{C}^*_{\leq(1+4\gamma)R}) \geq \gamma$. □

Consider one iteration Line 7 of Algorithm 9, let

$$\mathcal{C}_{\leq(1+4\gamma)R} = \{C \in \mathcal{C}^*_{\leq(1+4\gamma)R} \mid C \cap \widetilde{U} \neq \emptyset\}$$

be the set of good ratio cluster such that at least one of its node is chosen into $\widetilde{U}$, we can show that

Lemma 54. *For every $C^* \in C_{\leq(1+4\gamma)R}$, we have*

$$\hat{p}(C^*) \geq (1 - \frac{\gamma}{2})\hat{q}(C^*)$$

Moreover, we have

$$\Pr[\hat{p}(\mathcal{C}_{\leq(1+4\gamma)R}) = \Omega(\varepsilon^5 \gamma' \gamma)] = \Omega(\varepsilon^5 \gamma')$$

Proof. when we set up $\hat{p}$, we first copy $\hat{q}$ to $\hat{p}$ value. Then, we might set some $\hat{p}$ value to 0 if the nodes have two chosen neighbor. So, to show that $\hat{p}(C^*) \geq (1-\gamma)\hat{q}(C^*)$, we need to show that the removed nodes contributes to at most γ fractional weight to C^*.

For each $C^* \in \mathcal{C}_{\leq(1+4\gamma)R}$, let $u \in C^* \cap \widetilde{U}$ be the node in $\widetilde{U}$. Based on Claim 33, we know that $|C^*| = 1$ or $|C^*| \geq \gamma^3 d(u)$. The first case is impossible since we will set its $\hat{p}$ value to 0

and we will never consider it in $C_{\leq(1+4\gamma)R}$. For the second case, note that we remove at most γ' fractional $\hat{q}$ from $D(u)$, we will give an upper bound of this value.

Recall that based on lemma 35, we know that for any node $v \in C^* \subset N_{\text{cand}}(u)$, we have

$$\hat{q}_v = O\left(\frac{\varepsilon^{-1}d(u)}{d_{\text{cross}}(V)}\right)$$

so the upper bound $\hat{q}(D(u))$ is at most

$$\hat{q}(D(u)) = O\left(\frac{\varepsilon^{-1}d(u)\cdot|D(u)|}{d_{\text{cross}}(V)}\right)$$

based on Claim 33, we know that C^* contains at least $\gamma^3 d(u)$ vertices with positive $\hat{p}$ value, for each vertex in C^*, we have

$$\hat{q}_v = \Omega\left(\frac{\gamma\varepsilon^4|D(u)|}{d_{\text{cross}}(V)}\right)$$

so, the lower bound $\hat{p}$ value for C^* is at least

$$\hat{q}(C^*) = \Omega\left(\frac{\gamma^4\varepsilon^4|D(u)|\cdot d(u)}{d_{\text{cross}}(V)}\right)$$

For any node $u \in \widetilde{U}$, we remove at most γ' fractional of its $\hat{q}(D(u))$ value, as long as $\gamma' = \Omega(\gamma^5\varepsilon^4)$, we have

$$\hat{p}(C^*) \geq (1-\frac{\gamma}{2})\hat{q}(C^*)$$

Now for each $C^* \in \mathcal{C}^*_{\leq(1+4\gamma)R}$, based on Claim 52, we know that with probability at least $\Omega(\varepsilon^5\gamma')$, $C^* \cap \widetilde{U} \neq \emptyset$ and $C^* \in \mathcal{C}_{\leq(1+4\gamma)R}$, so we have

$$\mathbb{E}[\hat{q}(\mathcal{C}_{\leq(1+4\gamma)R})] = \Omega(\varepsilon^5\gamma'\hat{q}(\mathcal{C}^*_{\leq(1+4\gamma)R}))$$

Note that $\hat{q}(\mathcal{C}_{\leq(1+4\gamma)R}) \leq \hat{q}(\mathcal{C}^*_{\leq(1+4\gamma)R})$, by the reverse markov inequality, we have

$$\Pr[\hat{q}(\mathcal{C}_{\leq(1+4\gamma)R}) = \Omega(\varepsilon^5\gamma' q(\mathcal{C}^*_{\leq(1+4\gamma)R}))] = \Omega(\varepsilon^5\gamma')$$

By Lemma 25, we know that $\hat{q}(C^*_{\leq(1+4\gamma)R}) \geq \gamma$, so we have

$$\Pr[\hat{q}(\mathcal{C}_{\leq(1+4\gamma)R}) = \Omega(\varepsilon^5\gamma'\gamma)] = \Omega(\varepsilon^5\gamma')$$

For each $C^* \in \mathcal{C}_{\leq(1+4\gamma)R}$, we have $\hat{p}(C^*) \geq (1-\gamma)\hat{q}(C^*)$, so

$$\Pr[\hat{p}(\mathcal{C}_{\leq(1+4\gamma)R}) = \Omega(\varepsilon^5\gamma'\gamma)] = \Omega(\varepsilon^5\gamma')$$

□

From the above lemma, we know that for any $C^* \in C_{\leq(1+4\gamma)R}$, at least one node $u \in C^*$ will be fed into Lemma 18. Our final goal is to show that in each round, we cover a set of good clusters with sufficient p-value.

Next, we show that the cluster returned by Lemma 18 covers a constant fraction of the $\hat{p}$-value. This is formally stated in the following lemma.

Lemma 55. *For every $C^* \in C_{\leq(1+4\gamma)R}$, let $u \in C^* \cap \widetilde{U}$ be the node in $\widetilde{U}$ and C_u be the cluster returned by Lemma 18, then we have*

$$\hat{p}(C_u) = \Omega(\gamma^4\varepsilon^5\hat{p}(C^*))$$

Proof. Note that for any $C^* \in C_{\leq(1+4\gamma)R}$, we have

$$\begin{aligned}\frac{\mathrm{cover}(C^*) + \gamma^2 d_{\mathrm{adm}}(C^*)}{\hat{p}(C^*)} &\leq \frac{\mathrm{cover}(C^*) + \gamma^2 d_{\mathrm{adm}}(C^*)}{(1-\gamma)\hat{q}(C^*)} \\ &\leq \frac{1+4\gamma}{1-\gamma/2}R \leq (1+5\gamma)R\end{aligned}$$

This satisfy the input of Lemma 18. So we can find a cluster C_u with ratio at most $(1+5\gamma)R$.

If $\hat{p}(D(u)) \leq \hat{p}(N_{\mathrm{cand}}(u))/2$, based on Lemma 18, we know that $K(u) \subset C_u$ is alwasy added to C_u, so $\hat{p}(C_u) \geq \hat{p}(K(u)) \geq \hat{p}(N_{\mathrm{cand}}(u))/2 \geq \hat{p}(C^*)/2$.

On the other hand, if $\hat{p}(D(u)) \geq \hat{p}(N_{\mathrm{cand}}(u))/2$. Based on Lemma 35, the upper bound $\hat{p}(D(u))$ is at most

$$\hat{p}(D(u)) = O\left(\frac{\varepsilon^{-1}d(u)\cdot|D(u)|}{d_{\mathrm{cross}}(V)}\right)$$

and the upper bound for $\hat{p}(C^*)$ is

$$\hat{p}(C^*) \leq p(N_{\mathrm{cand}}(u)) \leq 2\hat{p}(D(u)) = O\left(\frac{\varepsilon^{-1}d(u)\cdot|D(u)|}{d_{\mathrm{cross}}(V)}\right)$$

By Claim 33, we know that C_u contains at least $\gamma^3 d(u)$ vertices with $\hat{p} > 0$. Based on Lemma 35, for each vertex in C_u, we have

$$\hat{p}_v = \Omega\left(\frac{\gamma\varepsilon^4|D(u)|}{d_{\mathrm{cross}}(V)}\right)$$

so, the lower bound $\hat{p}$ value for C_u is at least

$$\hat{p}(C_u) = \Omega\left(\frac{\gamma^4\varepsilon^4|D(u)|\cdot d(u)}{d_{\mathrm{cross}}(V)}\right)$$

so we have $\hat{p}(C_u) = \Omega(\gamma^4\varepsilon^5\hat{p}(C^*))$ □

Proof of Lemma 51. Note that, based on Lemma 54, for each iteration of the for loop at Line 7, we include the set $\mathcal{C}_{\leq(1+4\gamma)R}$ with total $\hat{p}$ value at least $\Omega(\varepsilon^5\gamma'\gamma) = \Omega(\varepsilon^9\gamma^6)$ with probability at least $\Omega(\varepsilon^5\gamma')$.

We repeat the for loop for $\Theta(\log n/(\varepsilon^5\gamma'))$ iterations. With high probability, we will find a set $\mathcal{C}_{\leq(1+4\gamma)R}$ with total p value at least $\Omega(\varepsilon^5\gamma'\gamma) = \Omega(\varepsilon^9\gamma^6)$.

Now, from Lemma 55, for each $C^* \in \mathcal{C}_{\leq(1+4\gamma)R}$, we can find a sufficiently large set C_u. Thus, $\mathcal{F}_{T_{\mathrm{MPC}}}$ can cover at least $\Omega(\gamma^{10}\varepsilon^{14})$ of the p value.

□

8.3 Wrap-Up: Proof of MPC Algorithm for Theorem 1

Now we are ready to show the key theorem for this section.

Proof of Theorem 1. We can obtain the preclustering $(\mathcal{K}, E_{\text{adm}})$ in time $\widetilde{O}(n)$ by Theorem 7. We now argue that this algorithm can be implemented in the MPC model.

In Algorithm 1, we first compute d_{cross}. This can be done given $\mathcal{K}$ by appending the cluster label to each edge to indicate whether the endpoints belong to the same cluster in the preclustering. Then, computing d_{cross} only requires counting the edges that cross different clusters in $\mathcal{K}$. This step can be implemented using sorting. We will discuss how to solve covering cluster LP later.

Finally, we need to apply Lemma 11 to compute the solution for cluster LP. Implementing Lemma 11 in the MPC model is somewhat tricky, but the key observation is that each $z_S \geq \frac{1}{T_{\text{MW}}}$. Therefore, for each node u, at most $O(T_{\text{MW}})$ clusters with positive z_S values will cover u. For each node, we collect all clusters with positive z_S values. If the final sum $\sum_{S \ni u} z_S > 1$, we sort z_S for u in any order and mark those z_S as "greater than 1 for u" for later clusters. Each S then collects these marked clusters and removes them from itself. In this process, for each non-singleton atom, we perform this operation only once for a representative, allowing S to remove the non-singleton atoms. Since at most $O(T_{\text{MW}})$ clusters cover u, this process requires at most $O(T_{\text{MW}} \cdot n)$ total space.

Algorithm 2 and Algorithm 9 are naturally parallelized, requiring only $O(\text{poly}(1/\varepsilon))$ rounds for execution. From Claim 32, we know that with high probability, each vertex is contained in at most $O(\log n)$ clusters from $\{C_u\}_{u \in U}$, so storing the candidate set N_{cand} for U requires at most $O(n \log n)$ space. We can then distribute all edge information to execute Lemma 18 for each $u \in U$.

The more challenging part is applying Lemma 18. For each $N_{\text{cand}}(u)$, we need to sample $O(\text{poly}(1/\varepsilon))$ nodes and enumerate all possible subsets of the sampled nodes. There are two methods for performing this enumeration:

1. Store all subsets of the sampled nodes. For each $N_{\text{cand}}(u)$, this requires $\tilde{O}(2^{\text{poly}(1/\varepsilon)}|N_{\text{cand}}(u)|)$ space to store the edge relations between the sampled nodes and all other nodes. Once these relations are recorded, Algorithm 5 can determine whether to add a node to T. This method requires only $O(1)$ rounds but consumes a total space of $\tilde{O}(2^{\text{poly}(1/\varepsilon)}m)$.

2. Enumerate all possible subsets one by one. This avoids the need to record edge information for the sampled nodes, but it requires an additional $O(2^{\text{poly}(1/\varepsilon)})$ rounds for each iteration.

Combining all these points, we conclude that we can either: - Spend $2^{\text{poly}(1/\varepsilon)}$ rounds with $O(n^\delta)$ memory per machine and total memory $\tilde{O}(\text{poly}(1/\varepsilon)m)$, or - Spend $\text{poly}(1/\varepsilon)$ rounds with $O(n^\delta)$ memory per machine and total memory $\tilde{O}(2^{\text{poly}(1/\varepsilon)}m)$ to solve cluster LP.

□

9 Rounding Algorithms

In this section, we present fast rounding algorithms for the cluster LP. We begin by providing intuition behind the rounding algorithms and by explaining the rounding techniques used in [CCL+24]. In Section 9.1, we show how to implement the rounding in nearly linear time. Then, in Section 9.2, we demonstrate how to perform rounding in the sublinear model.

Note that we have already obtained a solution to cluster LP. In [CCL+24], the authors showed how to round a cluster LP solution to an integral solution. The approximation ratio is given by the following theorem.

Theorem 56. *[CCL+24] There exists an algorithm that, given a feasible solution for the cluster LP, produces a clustering whose objective value is at most* 1.485 *times that of the cluster LP solution.*

To be more precise, the algorithm from Theorem 56 consists of two different rounding algorithms, one is executed with probability $p = \frac{1.485}{2}$ the other with probability $1 - p$. The two algorithms in question are the cluster-based rounding (Algorithm 10) and the pivot-based rounding (Algorithm 11). We need to show that we can implement both algorithms in nearly linear time.

Algorithm Description. The cluster-based rounding algorithm (Algorithm 10) selects clusters S based on their z_S values. In each round, we randomly choose a cluster S with probability proportional to z_S. We then add all nodes from S that are still in the graph to a new cluster and remove those nodes from the graph. This process is repeated until the graph is empty.

The pivot-based algorithm (Algorithm 11) follows a different approach. Instead of selecting a set directly, at each round, we randomly choose a node u that is still in the graph, referred to as the pivot. We then form a cluster using this pivot as follows. For each small $+$edge (u, v), we include v in the cluster. For the remaining $+$edges, we apply correlated rounding, where we randomly choose a set S containing u and include all remaining $+$edges if they are in S. For $-$edges, we use independent rounding, where the probability of adding a node to the cluster is determined by the distance metric:

$$1 - x_{uv} = \sum_{S \supseteq \{u,v\}} z_S.$$

The algorithm then removes all nodes that have been clustered in the current round and repeats the process until the graph is empty.

Algorithm 10 Cluster-Based Rounding

$\mathcal{C} \leftarrow \emptyset, V' \leftarrow V$
while $V' \neq \emptyset$ **do**
  randomly choose a cluster $S \subseteq V$, with probabilities $\frac{z_S}{\sum_{S'} z_{S'}}$
  if $V' \cap S \neq \emptyset$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{V' \cap S\}, V' \leftarrow V' \setminus S$
  end if
end while
return $\mathcal{C}$

9.1 Nearly Linear Time Rounding Algorithm

Implementation of Cluster-Based Rounding. We are given a solution z to the cluster LP with $\text{cost}(z) \leq (1 + \varepsilon)\text{cost}(\text{OPT})$ and $|\text{supp}(z)| = O(n)$. Moreover, by Lemma 11, for each $S \in \text{supp}(z)$, we have $z_S \geq \Delta$ for some constant Δ.

We will implement Algorithm 10 as follows. For each vertex we sample a random variable k_v from an exponential distribution with rate z_S. To do this, we sample a value k_S for each set with $z_S > 0$ by sampling a value uniformly at random from $p_S \in (0, 1)$ and then setting

$$k_S = \lfloor \frac{n^c}{z_S} \log \frac{1}{p_S} \rfloor,$$

Algorithm 11 Pivot-Based Rounding (with correlated mid-range for + edges)

$\mathcal{C} \leftarrow \emptyset$, $V' \leftarrow V$
while $V' \neq \emptyset$ **do**
  choose a pivot $u \in V'$ uniformly at random
  $C \leftarrow \{u\}$
  for each $v \in V' \cap N^-(u)$ **do**
    independently add v to C with probability $1 - x_{uv}^2$
  end for
  $C_\ell \leftarrow \{v \in V' \cap N^+(u) : x_{uv} \leq 0.40\}$
  $C_m \leftarrow \{v \in V' \cap N^+(u) : 0.40 < x_{uv} \leq 0.57\}$
  $C_h \leftarrow \{v \in V' \cap N^+(u) : x_{uv} > 0.57\}$
  for each $v \in C_h$ **do**
    independently add v to C with probability $1 - x_{uv}$
  end for
  sample a random set $S \ni u$ according to the set distribution $\{z_S\}_{S \ni u}$
  $C \leftarrow C \cup C_\ell \cup (C_m \cap S)$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$, $V' \leftarrow V' \setminus C$
end while
return $\mathcal{C}$

where c is a fixed constant. Then, for each vertex v, let k_v be the smallest k_S among sets S that contain the vertex v. All vertices with the same k_v are assigned to the same cluster. Algorithm 12 is equivalent to Algorithm 10. Indeed, let $X_1, \ldots, X_l$ be exponentially distributed random variables each with rate r_i. The probability that X_i is the minimum among $X_1, \ldots, X_1$ is equal to

$$Pr[X_i = \min\{X_1, \ldots, X_l\}] = \frac{r_i}{\sum_j r_j}.$$

We emphasize that, given the solution from Theorem 1, Algorithm 12 runs in $\widetilde{O}(n)$ time in the sublinear model. We conclude with the following lemma for the cluster-based rounding algorithm:

Lemma 57. *Given a solution $\{z_S\}$ to cluster LP such that for each $S \in \mathrm{supp}(z)$, we have $z_S \geq \Delta$ for some constant Δ. Then, Algorithm 10 runs in $\widetilde{O}(n/\Delta)$ time in the sublinear model.*

Algorithm 12 Cluster-Based Rounding

for $S \in \mathcal{S}$ **do**
  $k_S = \lfloor \frac{n^c}{z_s} \log \frac{1}{p_s} \rfloor$, where p_s is uniformly chosen from $(0, 1)$
end for
for $v \in V$ **do**
  $k_v = \min\{k_S \mid S \ni v\}$
end for
Put all nodes with same k_v value into same cluster

Implementation of Pivot-Based Rounding. Now we discuss how to implement Algorithm 11 in nearly linear time. Note that each vertex is contained in at most $\frac{1}{\Delta}$ clusters $S \in \mathrm{supp}(z)$. Thus a fixed pivot u belongs to at most $1/\Delta$ sets S such that $z_S > 0$. We therefore only need to consider $1/\Delta$ sets to choose one of these sets according to the respective probability

distribution. Visiting the $+$edges incident to each pivot requires at most $O(m)$ time over the course of the algorithm.

For the $-$edges, the main challenge is listing all $-$edges from $V \cap N^-(u)$ and then performing independent rounding. Our key observation is that $x_{uv} < 1$ if and only if there exists some S such that S contains both u and v and $z_S > 0$. Thus, we only need to iterate over all vertices in $\cup_{S \ni u, z_S > 0} S$.

Since each $z_S \geq \Delta$ for $S \in \mathrm{supp}(z)$, each node v with $x_{uv} < 1$ contributes at least $1/\Delta$ to the LP value. Therefore, there are at most $O(m/\Delta)$ $-$edges across all S. Since each $-$edge is visited at most once, listing all $-$edges takes at most $O(m/\Delta)$ time.

Once we process the pivot, we need to update S to remove nodes that have been clustered. Since each nonzero $z_S > \Delta$, the total update time is at most $O(n/\Delta)$.

Combining everything, we obtain the following lemma for pivot-based rounding.

Lemma 58. *Given a solution $\{z_S\}$ to cluster LP such that for each $S \in \mathrm{supp}(z)$, we have $z_S \geq \Delta$ for some constant Δ. Then, Algorithm 11 runs in $\widetilde{O}(m/\Delta)$ time.*

9.2 Rounding in Sublinear Model

To argue that Algorithm 11 runs in sublinear time, we make the following observation. For a fixed pivot r, let

$$U_r = \bigcup_{S \ni r : z_S > 0} S$$

i.e., U_r is the set of all vertices that occur together with r in some set in the support of z. If $v \notin U_r$, then $x_{uv} = 1$, meaning Algorithm 11 never considers v when r is chosen as the pivot. Thus, Algorithm 11 only considers U_r for one step of iteration.

Moreover, we can iterate over all $S \ni r$ with $z_S > 0$ and all $v \in S$ in time $\frac{1}{\Delta}|U_r|$ since each vertex is contained in at most $\frac{1}{\Delta}$ sets $S \in \mathrm{supp}(z)$. Hence, we can implement one step of the algorithm in time at most $\widetilde{O}(|U_r|/\Delta)$.

On the other hand, for each vertex $v \in U_r$, the probability that it is included in the cluster of r is at least $1 - x_{rv} \geq \Delta$ for $+$edges and $1 - x_{rv} \cdot x_{rv} \geq \Delta$ for $-$edges. Thus, in expectation, we remove at least $\Delta|U_r|$ vertices from the current graph.

Combining these points, in Algorithm 11, we pay $O(1/\Delta^2)$ for each node in expectation. Therefore, the final running time is given by the following lemma.

Lemma 59. *Given a solution $\{z_S\}$ to cluster LP such that for each $S \in \mathrm{supp}(z)$, we have $z_S \geq \Delta$ for some constant Δ. Then, Algorithm 11 runs in $\widetilde{O}(n/\Delta^2)$ time.*

Now, we can prove the main theorem regarding the rounding algorithm.

Proof of Theorem 2. The approximation ratio follows from Theorem 56. The runtime of cluster-based rounding is given in Lemma 57, and the runtime of pivot-based rounding is given in Lemma 59. Combining these results establishes the main theorem. □

Expected Guarantee We highlight one final point regarding general rounding algorithms in the sublinear model. Typically, a rounding algorithm provides an expected guarantee. To achieve a high-probability guarantee, we can run the rounding algorithm multiple times and select the best outcome. However, in the sublinear model, it is unclear how to determine which output is the best, as computing the clustering cost sublinear time is challenging.

For our rounding algorithm, we achieve a high-probability guarantee. This is because our rounding algorithm sets an atom as a cluster if its d_{cross} value is very small and never

splits an atom. We can apply Lemma 37 to estimate the cost of the clustering. The estimation cost is $\beta \cdot d_{\text{cross}}(V)$.

Since $d_{\text{cross}}(V) = O(\frac{1}{\varepsilon^{12}})\text{OPT}$, the estimation cost is always small enough compared to the optimal solution.

References

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):1–27, 2008.

[AHK+09] 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 (WSDM)*, pages 172–181, 2009.

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: A meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[ARS09] Arvind Arasu, Christopher Ré, and Dan Suciu. Large-scale deduplication with constraints using dedupalog. In *Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE)*, pages 952–963, 2009.

[AW22] 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.

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1):89–113, 2004.

[BCC+24] Soheil Behnezhad, Moses Charikar, Vincent Cohen-Addad, Alma Ghafari, and Weiyun Ma. Fully dynamic correlation clustering: Breaking 3-approximation, 2024.

[BCMT22] Soheil Behnezhad, Moses Charikar, Weiyun Ma, and Li-Yang Tan. Almost 3-approximate correlation clustering in constant rounds. In *Proceedings of 63rd Annual IEEE Symposium on Foundations of Computer Science, (FOCS)*, pages 720–731, 2022.

[BCMT23] Soheil Behnezhad, Moses Charikar, Weiyun Ma, and Li-Yang Tan. Single-pass streaming algorithms for correlation clustering. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 819–849, 2023.

[BGU13] Francesco Bonchi, Aristides Gionis, and Antti Ukkonen. Overlapping correlation clustering. *Knowledge and Information Systems*, 35(1):1–32, 2013.

[CCL+24] 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.

[CGW05] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005.

[CHS24] Nairen Cao, Shang-En Huang, and Hsin-Hao Su. Breaking 3-factor approximation for correlation clustering in polylogarithmic rounds. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024.

[CKP08] 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.

[CLLN23] 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)*, pages 1082–1104, 2023.

[CLM+21] 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)*, pages 2069–2078, 2021.

[CLN22] 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.

[CLP+24] 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 (STOC)*, pages 1617–1628, 2024.

[CMSY15] 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.

[CSX12] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In *Advances in Neural Information Processing Systems (Neurips)*, pages 2204–2212, 2012.

[DMM24] Mina Dalirrooyfard, Konstantin Makarychev, and Slobodan Mitrović. Pruned pivot: correlation clustering algorithm for dynamic, parallel, and local computation models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[KCMNT08] 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.

[KYNK14] Sungwoong Kim, Chang D Yoo, Sebastian Nowozin, and Pushmeet Kohli. Image segmentation using higher-order correlation clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1761–1774, 2014.

[MC24] Konstantin Makarychev and Sayak Chakrabarty. Single-pass pivot algorithm for correlation clustering. Keep it simple! *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[PST95] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301, 1995.

[YV18] Grigory Yaroslavtsev and Adithya Vadapalli. Massively parallel algorithms and hardness for single-linkage clustering under ℓ_p-distances. In *Proceedings of 35th International Conference on Machine Learning (ICML)*, page 5596–5605, 2018.