Understanding the Cluster Linear Program for Correlation Clustering*

Nairen Cao
Boston College
Brighton, USA
caonc@bc.edu

Vincent Cohen-Addad
Google Research
Grenoble, France
cohenaddad@google.com

Euiwoong Lee
University of Michigan
Ann Arbor, USA
euiwoong@umich.edu

Shi Li
Nanjing University
Nanjing, China
shili@nju.edu.cn

Alantha Newman
CNRS - Université Grenoble Alpes
Grenoble, France
alantha.newman@grenoble-inp.fr

Lukas Vogl
École polytechnique fédérale de Lausanne
Lausanne, Switzerland
lukas.vogl@epfl.ch

ABSTRACT

In the classic Correlation Clustering problem introduced by Bansal, Blum, and Chawla [7], the input is a complete graph where edges are labeled either + or −, and the goal is to find a partition of the vertices that minimizes the sum of the +edges across parts plus the sum of the -edges within parts. In recent years, Chawla, Makarychev, Schramm and Yaroslavtsev [21] gave a 2.06-approximation by providing a near-optimal rounding of the standard LP, and Cohen-Addad, Lee, Li, and Newman [27, 28] finally bypassed the integrality gap of 2 for this LP giving a 1.73-approximation for the problem.

While introducing new ideas for Correlation Clustering, their algorithm is more complicated than *typical* approximation algorithms in the following two aspects: (1) It is based on two different relaxations with separate rounding algorithms connected by the round-or-cut procedure. (2) Each of the rounding algorithms has to separately handle seemingly inevitable *correlated rounding errors*, coming from *correlated rounding* of Sherali-Adams and other strong LP relaxations [9, 33, 41].

In order to create a simple and unified framework for Correlation Clustering similar to those for *typical* approximate optimization tasks, we propose the *cluster LP* as a strong linear program that might tightly capture the approximability of Correlation Clustering. It unifies all the previous relaxations for the problem. It is exponential-sized, but we show that it can be $(1+\epsilon)$-approximately solved in polynomial time for any $\epsilon > 0$, providing the framework for designing rounding algorithms without worrying about correlated rounding errors; these errors are handled uniformly in solving the relaxation.

We demonstrate the power of the cluster LP by presenting a simple rounding algorithm, and providing two analyses, one analytically proving a 1.49-approximation and the other solving a factor-revealing SDP to show a 1.437-approximation. Both proofs introduce principled methods by which to analyze the performance of the algorithm, resulting in a significantly improved approximation guarantee.

Finally, we prove an integrality gap of 4/3 for the cluster LP, showing our 1.437-upper bound cannot be drastically improved. Our gap instance directly inspires an improved NP-hardness of approximation with a ratio $24/23 \approx 1.042$; no explicit hardness ratio was known before.

CCS CONCEPTS

• **Theory of computation → Facility location and clustering**; *Discrete optimization*; *Rounding techniques*; *Lower bounds and information complexity*.

KEYWORDS

Clustering, approximation algorithms, exponential size linear programming, semi-definite programming.

ACM Reference Format:
Nairen Cao, Vincent Cohen-Addad, Euiwoong Lee, Shi Li, Alantha Newman, and Lukas Vogl. 2024. Understanding the Cluster Linear Program for Correlation Clustering. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3618260.3649749

1 INTRODUCTION

Clustering is a classic problem in unsupervised machine learning and data mining. Given a set of data elements and pairwise similarity information between the elements, the task is to find a partition of the data elements into clusters to achieve (often contradictory) goals of placing similar elements in the same cluster and separating different elements in different clusters. Introduced by Bansal, Blum, and Chawla [7], Correlation Clustering elegantly models such tension and has become one of the most widely studied formulations for graph clustering. The input of the problem consists of a complete graph $(V, E^+ \uplus E^-)$, where $E^+ \uplus E^- = \binom{V}{2}$, E^+ representing the so-called *positive* edges and E^- the so-called *negative* edges.

*N.C. was supported by NSF grant CCF-2008422. S.L. is affiliated with the Department of Computer Science and Technology in Nanjing University, and supported by the State Key Laboratory for Novel Software Technology, and the New Cornerstone Science Laboratory. E.L. was supported in part by NSF grant CCF-2236669 and Google.

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649749

The goal is to find a clustering (partition) of V, namely $(V_1, \ldots, V_k)$, that minimizes the number of unsatisfied edges, namely the +edges between different clusters and the −edges within the same cluster. Thanks to the simplicity and modularity of the formulation, Correlation Clustering has found a number of applications, e.g., finding clustering ensembles [12], duplicate detection [5], community mining [22], disambiguation tasks [36], automated labelling [1, 16] and many more.

This problem is APX-Hard [18], and various $O(1)$-approximation algorithms [7, 18] have been proposed in the literature. Ailon, Charikar and Newman introduced an influential *pivot-based* algorithm, which leads to a combinatorial 3-approximation and a 2.5-approximation with respect to the standard LP relaxation [4]. The LP-based rounding was improved by Chawla, Makarychev, Schramm and Yaroslavtsev to a 2.06-approximation [21], nearly matching the LP integrality gap of 2 presented in [18].

It turns out that (a high enough level of) the Sherali-Adams hierarchy can be used to design a strictly better than 2-approximation. Cohen-Addad, Lee, and Newman [28] showed that $O(1/\varepsilon^2)$ rounds of the Sherali-Adams hierarchy have an integrality gap of at most $(1.994+\varepsilon)$. This approximation ratio was improved by Cohen-Addad, Lee, Li, and Newman [27] to $(1.73 + \varepsilon)$ in $n^{\mathrm{poly}(1/\varepsilon)}$-time, which combines *pivot-based rounding* and *set-based rounding*.

One undesirable feature of [27] is the lack of a single convex relaxation with respect to which the approximation ratio is analyzed. For technical reasons, it combines the two rounding algorithms via a generic *round-or-cut* framework. Given $x \in [0,1]^E$, each of the two rounding algorithms outputs either an integral solution with some guarantee or a hyperplane separating x from the convex hull of integral solutions; if both algorithms output integral solutions, one of them is guaranteed to achieve the desired approximation factor. Though each of the rounding procedures is based on some LP relaxations, they are different, so there is no single relaxation that can be compared to the value of the final solution.

In this work, we propose the *cluster LP* as a single relaxation that captures all of the existing algorithmic results. Based on this new unified framework, we design a new rounding algorithm as well as principled tools for the analysis that significantly extend the previous ones, ultimately yielding a new approximation ratio of $1.437 + \varepsilon$. The study of the cluster LP sheds light on the hardness side as well, as we prove a $4/3 \approx 1.33$ gap for the cluster LP and a $24/23 \approx 1.042$ NP-hardness of approximation.

1.1 Our Results

We first state the cluster LP here. It is similar to *configuration LPs* used for scheduling and assignment problems [8, 31]. In the cluster LP, we have a variable z_S for every $S \subseteq V, S \neq \emptyset$, that indicates if S is a cluster in the output clustering or not. As usual, x_{uv} for every $uv \in \binom{V}{2}$ indicates if u and v are separated in the clustering or not. For any $x \in [0,1]^{\binom{V}{2}}$, we define $\mathrm{obj}(x) := \sum_{uv \in E^+} x_{uv} + \sum_{uv \in E^-} (1 - x_{uv})$ to be the fractional number of edges in disagreement in the solution x.

$$\min \quad \mathrm{obj}(x) \quad \text{s.t.} \qquad \text{(cluster LP)}$$

$$\sum_{S \ni u} z_S = 1 \qquad \forall u \in V \qquad (1)$$

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

$$z_S \geq 0 \qquad \forall S \subseteq V, S \neq \emptyset \qquad (3)$$

The objective of the LP is to minimize $\mathrm{obj}(x)$, which is a linear function. (1) requires that every vertex u appears in exactly one cluster, (2) gives the definition of x_{uv} using z variables.

The idea behind this LP was used in [27] to design their set-based rounding algorithm, though the LP was not formulated explicitly in that paper. Moreover, the paper did not provide an efficient algorithm to solve it approximately. Our first result shows that we can approximately solve the cluster LP in polynomial time, despite it having an exponential number of variables. We remark that unlike the configuration LPs for many problems, we do not know how to solve the cluster LP simply by considering its dual.

Theorem 1. *Let $\varepsilon > 0$ be a small enough constant and* opt *be the cost of the optimum solution to the given Correlation Clustering instance. In time $n^{\mathrm{poly}(1/\varepsilon)}$, we can output a feasible cluster LP solution $\left((z_S)_{S \subseteq V}, (x_{uv})_{uv \in \binom{V}{2}}\right)$ with $\mathrm{obj}(x) \leq (1+\varepsilon)\mathrm{opt}$, described using a list of non-zero coordinates.* [1]

The cluster LP is the most powerful LP that has been considered for the problem. Indeed, previous algorithms in [28] and [27] can be significantly simplified if one is given a $(1+\varepsilon)$-approximate solution to the LP. A large portion of the algorithms and analysis in [28] and [27] is devoted to handle the additive errors incurred by the correlated rounding procedure, which is inherited from the Raghavendra-Tan rounding technique [41]. Instead, we move the complication of handling rounding errors into the procedure of solving the cluster LP relaxation.

With this single powerful relaxation, we believe that Theorem 1 provides a useful framework for future work that may use more ingenious rounding of the exponential-sized cluster LP without worrying about errors. Indeed, the constraints in the cluster LP imply that the matrix $(1 - x_{uv})_{u,v \in V}$ is PSD, [2] and thus the LP is at least as strong as the natural SDP for the problem. For the complementary version of maximizing the number of correct edges, the standard SDP is known to give a better approximation guarantee of 0.766 [18, 42]. For the minimization version, the standard SDP has integrality gap at least 1.5 (see full paper), but it is still open whether this program has an integrality gap strictly below 2 or not.

We demonstrate the power of the cluster LP by presenting and analyzing the following algorithm, significantly improving the previous best 1.73-approximation.

Theorem 2. *There exists a $(1.49 + \varepsilon)$-approximation algorithm for Correlation Clustering that runs in time $O(n^{\mathrm{poly}(1/\varepsilon)})$.*

[1] We remark that $\mathrm{obj}(x)$ given by the theorem is at most $1 + \varepsilon$ times opt, instead of the value of the cluster LP. This is sufficient for our purpose. One should also be able to achieve the stronger guarantee of $(1+\varepsilon)$-approximation to the optimum fractional solution. Instead of dealing with the optimum clustering C^* in the analysis, we deal with the optimum fractional clustering to the LP. For simplicity, we choose to prove the theorem with the weaker guarantee.

[2] Consider the matrix $Y \in [0,1]^{V \times V}$ where $y_{uv} = 1 - x_{uv}$ for every $u, v \in V$ ($Y_{uu} = 1, \forall u \in V$). For every $w \in \mathbb{R}^V$, we have $w^T Y w = \sum_{u,v \in V} y_{uv} w_u w_v = \sum_{u,v} \sum_{S \supseteq \{u,v\}} z_S w_u w_v = \sum_{u,v} \sum_{S \subseteq V} z_S \cdot (w_u \cdot 1_{u \in S}) \cdot (w_v \cdot 1_{v \in S}) = \sum_{S \subseteq V} z_S \left(\sum_{u \in S} w_u\right)\left(\sum_{v \in S} w_v\right) \geq 0$.

This is achieved by a key modification of the pivot-based rounding algorithm that is used in conjunction with the set-based algorithm as in [27]. In combination with more careful analysis, which involves principled methods to obtain the best *budget function*, we obtain a significantly improved approximation ratio.

In order to obtain an even tighter analysis of the same algorithm, we introduce the new *factor revealing SDP* that searches over possible global distributions of triangles in valid Correlation Clustering instances. By numerically solving such an SDP, we can further improve the approximation ratio of the same algorithm.

Theorem 3. *There exists a* $(1.437+\varepsilon)$*-approximation algorithm for Correlation Clustering that runs in time* $O(n^{\mathrm{poly}(1/\varepsilon)})$.

While the proof includes a feasible solution to a large SDP and is not human-readable, we prove that our SDP gives an *upper bound* on the approximation ratio, so it is a complete proof modulo the SDP feasibility of the solution. Our program and solution can be found at https://github.com/correlationClusteringSDP/SDP1437code/.

We also study lower bounds and prove the following lower bound on the integrality gap of the cluster LP.

Theorem 4. *For any* $\varepsilon > 0$*, the integrality gap of the cluster LP is at least* $4/3-\varepsilon$.

This integrality gap for the cluster LP, after some (well-known) loss, directly translates to NP-hardness. It is the first hardness with an explicit hardness ratio apart from the APX-hardness [18].

Theorem 5. *Unless* $\mathbf{P}=\mathbf{BPP}$*, for any* $\varepsilon > 0$*, there is no* $(24/23-\varepsilon)$*-approximation algorithm for Correlation Clustering.*

1.2 Further Related Work

The weighted version of Correlation Clustering, where each pair of vertices has an associated weight and unsatisfied edges contribute a cost proportional to their weight to the objective, is shown to be equivalent to the Multicut problem [30], implying that there is an $O(\log n)$-approximation but no constant factor approximation is possible under the Unique Games Conjecture [20].

In the unweighted case, a PTAS exists when the number of clusters is a fixed constant [32, 37]. Much study has been devoted to the minimization version of Correlation Clustering in various computational models, for example in the online setting [24, 38, 39], as well as in other practical settings such as distributed, parallel or streaming [3, 6, 10, 11, 13–15, 17, 23, 25, 40, 43, 44]. Other recent work involves settings with fair or local guarantees [2, 29, 35].

2 ALGORITHMIC FRAMEWORK AND SETUP FOR ANALYSIS

In this section, we describe our algorithm for obtaining the improved approximation ratio for Correlation Clustering. We solve the cluster LP using Theorem 1 to get a fractional solution $z=(z_S)_{S\subseteq V}$, which determines $x\in[0,1]^{\binom{V}{2}}$ as in (2): $x_{uv}:=1-\sum_{S\supseteq\{u,v\}} z_S$ for every $uv\in\binom{V}{2}$. We have $\mathrm{obj}(x)\le(1+\varepsilon)\mathrm{opt}$. The theorem will be proved in Section 4. With z, we then run two procedures: the cluster-based rounding and the pivot-based rounding with threshold $1/3$. We select the better result as the final clustering. The two procedures are defined in Algorithms 1 and 2 respectively. We use $N^+(u)$ and $N^-(u)$ to denote the sets of $+$ and $-$neighbors of a vertex $u\in V$ respectively.

Algorithm 1 Cluster-Based Rounding

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

Algorithm 2 Pivot-Based Rounding with Threshold $1/3$

1: $C\leftarrow\emptyset, V'\leftarrow V$
2: **while** $V'\neq\emptyset$ **do**
3: randomly choose a pivot $u\in V'$
4: $C\leftarrow\{v\in V'\cap N^+(u): x_{uv}\le\frac{1}{3}\}$
5: **for** every $v\in V'\cap N^-(u)$ **do** independently add v to C with probability $1-x_{uv}$
6: randomly choose a set $S\ni u$, with probabilities z_S ▷ We have $\sum_{S\ni u} z_S=1$
7: $C\leftarrow C\cup(S\cap V'\cap N^+(u)), C\leftarrow C\cup\{C\}, V'\leftarrow V'\setminus C$
8: **return** C

Analysis of Cluster-Based Rounding Procedure. The cluster-based rounding procedure is easy to analyze. The following lemma suffices.

Lemma 6. *For every* $uv\in\binom{V}{2}$*, the probability that* u *and* v *are separated in the clustering* C *output by the cluster-based rounding procedure is* $\frac{2x_{uv}}{1+x_{uv}}$*. So the probability they are in the same cluster is* $\frac{1-x_{uv}}{1+x_{uv}}$.

Proof. We consider the first set S chosen in the cluster-based rounding algorithm such that $\{u,v\}\cap S\neq\emptyset$. u and v will be separated iff $|S\cap\{u,v\}|=1$. The probability that this happens is precisely $\frac{\sum_{|S\cap\{u,v\}|=1} z_S}{\sum_{S\cap\{u,v\}\neq\emptyset} z_S}=\frac{2x_{uv}}{1+x_{uv}}$. □

Therefore, a $+$edge uv will incur a cost of $\frac{2x_{uv}}{1+x_{uv}}$ in expectation in the cluster-based rounding procedure, and a $-$edge will incur a cost of $\frac{1-x_{uv}}{1+x_{uv}}$. The approximation ratios for a $+$edge uv and a $-$edge uv are respectively $\frac{2}{1+x_{uv}}$ and $\frac{1}{1+x_{uv}}$. Notice that the latter quantity is at most 1.

Notations and Analysis for Pivot-Based Rounding Procedure. We now proceed to the pivot-based rounding procedure in Algorithm 2. We remark that to recover the correlated rounding algorithm in [28] and [27], we can use $C\leftarrow\emptyset$ in Step 4. Then we can obtain their approximation ratios without the complication of handling rounding errors. The errors are handled in [28] by distinguishing between the short, median and long $+$edges. In our algorithm, we also distinguish between *short* $+$edges (those with $x_{uv}\le\frac{1}{3}$) and *long* $+$edges (those with $x_{uv}>\frac{1}{3}$); however, the purpose of this distinction is to get an improved approximation ratio, instead of to bound the rounding errors.

Our high-level setup of the analysis follows from [27, 28], which in turn is based on [4] and [21]. We consider a general *budget* for every edge. We shall define two *budget functions*:

- $b^+ : [0,1] \to \mathbb{R}_{\geq 0}$ and $b^- : [0,1] \to \mathbb{R}_{\geq 0}$.

They determine the *budget* b_{uv} for the edge uv: if $uv \in E^+$, then $b_{uv} := b^+(x_{uv})$, and if $uv \in E^-$, then $b_{uv} := b^-(x_{uv})$.

We now focus on one iteration of the while loop in Algorithm 2. Suppose $u, v, w \in V'$ at the beginning of the iteration, and let C be the cluster constructed at the end. We use u to denote the event that u is chosen as the pivot. We say vw incurs a cost in the iteration, if $vw \in E^+$ and $|C \cap \{v, w\}| = 1$, or $vw \in E^-$ and $\{v, w\} \subseteq C$. Then, we define

$$\mathrm{cost}_u(v, w) := \Pr[vw \text{ incurs a cost} \mid u],$$

and

$$\Delta_u(v, w) := \Pr[C \cap \{v, w\} \neq \emptyset \mid u] \cdot b_{vw}.$$

$\mathrm{cost}_u(v, w)$ is the probability that vw incurs a cost conditioned on the event u. When an edge vw disappears, we say vw *releases* its budget. So, $\Delta_u(v, w)$ is the expected budget released by vw in the iteration when u is the pivot. Notice that both $\mathrm{cost}_u(v, w)$ and $\Delta_u(v, w)$ do not depend on V', provided that $u, v, w \in V'$.

We call a set of three distinct vertices a *triangle*. A set of two distinct vertices is called a *degenerate triangle*. For triangle (u, v, w), let

$$\mathrm{cost}(u, v, w) := \mathrm{cost}_u(v, w) + \mathrm{cost}_v(u, w) + \mathrm{cost}_w(u, v), \quad \text{and}$$
$$\Delta(u, v, w) := \Delta_u(v, w) + \Delta_v(u, w) + \Delta_w(u, v).$$

For degenerate triangle (u, v), let

$$\mathrm{cost}(u, v) := \mathrm{cost}_u(u, v) + \mathrm{cost}_v(u, v), \quad \text{and}$$
$$\Delta(u, v) := \Delta_u(u, v) + \Delta_v(u, v).$$

We show the following lemma in our full paper.

Lemma 7. *Suppose that for every $V' \subseteq V$, we have*

$$\sum_{(u,v,w) \in \binom{V'}{3}} \mathrm{cost}(u, v, w) + \sum_{(u,v) \in \binom{V'}{2}} \mathrm{cost}(u, v) \leq \sum_{(u,v,w) \in \binom{V'}{3}} \Delta(u, v, w) + \sum_{(u,v) \in \binom{V'}{2}} \Delta(u, v). \tag{4}$$

Then, the expected cost of the clustering output by Algorithm 2 is at most $\sum_{uv \in \binom{V}{2}} b_{uv}$.

To obtain an approximation ratio of $\alpha \in [1, 2)$, we consider a variant of our algorithm, in which we run the cluster-based rounding procedure (Algorithm 1) with probability $\frac{\alpha}{2}$, and the pivot-based rounding procedure with threshold 1/3 (Algorithm 2) with the remaining probability $1 - \frac{\alpha}{2}$. Clearly, the actual algorithm that picks the better of the two clusterings generated can only be better. We set up the budget functions b^+ and b^- such that every edge pays a cost of at most α times its LP cost in expectation. That is, the following properties are satisfied for every $x \in [0, 1]$:

$$\frac{\alpha}{2} \cdot \frac{2x}{1+x} + \left(1 - \frac{\alpha}{2}\right) b^+(x) = \alpha x,$$
$$\frac{\alpha}{2} \cdot \frac{1-x}{1+x} + \left(1 - \frac{\alpha}{2}\right) b^-(x) = \alpha (1-x).$$

This gives us the following definitions:

$$b^+_\alpha(x) := \frac{\alpha}{1 - \alpha/2} \cdot \frac{x^2}{1+x}, \quad \text{and}$$
$$b^-_\alpha(x) := \frac{\alpha}{1 - \alpha/2} \cdot \frac{(1+2x)(1-x)}{2(1+x)}, \quad \forall x \in [0, 1]. \tag{5}$$

Lemma 8. *If the budget functions b^+_α and b^-_α satisfy (4) for some $\alpha \in [1, 2)$, then our algorithm has an approximation ratio of α.*

Proof. Consider the variant of the algorithm where we run the cluster-based rounding procedure with probability $\frac{\alpha}{2}$, and the pivot-based procedure with threshold 1/3 with the remaining probability of $1 - \frac{\alpha}{2}$. By Lemma 7, the expected cost of the clustering given by the variant is at most

$$\sum_{uv \in E^+} \left(\frac{\alpha}{2} \cdot \frac{2x_{uv}}{1+x_{uv}} + \left(1 - \frac{\alpha}{2}\right) \cdot b^+_\alpha(x_{uv}) \right) + \sum_{uv \in E^-} \left(\frac{\alpha}{2} \cdot \frac{1-x_{uv}}{1+x_{uv}} + \left(1 - \frac{\alpha}{2}\right) \cdot b^-_\alpha(x_{uv}) \right)$$
$$= \alpha \left(\sum_{uv \in E^+} x_{uv} + \sum_{uv \in E^-} (1 - x_{uv}) \right) = \alpha \cdot \mathrm{obj}(x).$$

The actual algorithm we run can only be better than this variant. □

As a baseline, we provide a per-triangle analysis leading to an approximation ratio of 1.5 in the full paper:

Lemma 9. *For budget functions $b^+ \equiv b^+_{1.5}$ and $b^- \equiv b^-_{1.5}$, we have $\mathrm{cost}(T) \leq \Delta(T)$ for every triangle T.*

Clearly, the lemma implies that (4) holds for $b^+ \equiv b^+_{1.5}$ and $b^- \equiv b^-_{1.5}$. By Lemma 8, our algorithm gives an approximation ratio of 1.5. We remark that 1.5 is the best possible ratio we can achieve using the per-triangle analysis. For a $++-$ triangle with length $\frac{1}{2}$ for $+$edges and length 1 for the $-$edge, we need to pay a factor of 2 for each of the $\frac{1}{2}$-length $+$edge. Then the cluster-based rounding algorithm gives factors of 2 and $\frac{4}{3}$ for $+$edges of lengths 0 and $\frac{1}{2}$ respectively. For the pivot-based rounding algorithm, the factors are at least 0 and 2. A combination of the two algorithms can only lead to a factor of 1.5.

To get a better approximation ratio, we provide two analyses that use global distributions of triangles. The former is purely analytic and the latter relies on solving a factor-revealing SDP. The following two lemmas are proved in the full paper.

Lemma 10. (4) *holds for budget functions $b^+ \equiv b^+_{1.49}$ and $b^- \equiv b^-_{1.49}$.*

Lemma 11. (4) *holds for budget functions $b^+ \equiv b^+_{1.437}$ and $b^- \equiv b^-_{1.437}$.*

Combined with Lemma 8, the two lemmas imply Theorems 2 and 3 respectively.

3 OVERVIEW OF TECHNIQUES

In this section, we provide overviews of the techniques used in our results.

Simpler and Better Preclustering Procedure. The concept of preclustering was introduced in [27]. In a preclustered instance, we predetermine the fate of some edges: for some edges uv, u and v must be in the same cluster; for some other edges uv, u and v must be separated. Since the relation of being in the same cluster is transitive, we define a preclustered instance using a pair $(\mathcal{K}, E_{\text{adm}})$, where $\mathcal{K}$ is a partition of V into so called *atoms* and $E_{\text{adm}} \subseteq \binom{V}{2}$ is a set of *admissible* edges. An atom can not be broken. If u and v are not in the same atom and $uv \notin E_{\text{adm}}$, then u and v must be separated. [27] showed how to construct a preclustered instance $(\mathcal{K}, E_{\text{adm}})$, losing only a $(1+\varepsilon)$ factor in the optimum cost, while at the same time guaranteeing that $|E_{\text{adm}}| \leq O(\text{opt}/\varepsilon^{12})$. This is crucial for their correlated rounding algorithm, as it loses an additive error depending on $|E_{\text{adm}}|$. In this work, we still need the preclustering procedure to bound the rounding error, but now it is inside the procedure of solving the cluster LP.

We greatly simplify the preclustering procedure from [27], and as a result, we achieve a much better bound of $O(\text{opt}/\varepsilon^2)$ on $|E_{\text{adm}}|$. [27] used the agreement graph to construct the atoms; roughly speaking, two vertices are in agreement if their neighborhood sets are similar to each other. The analysis uses many technical structural lemmas from [25], which solves Correlation Clustering in the online setting. In contrast, our construction of atoms is simple: we construct an $O(1)$-approximate clustering C, mark vertices whose costs are large, and then $\mathcal{K}$ is obtained from C by removing marked vertices and creating singletons for them. The set of admissible edges is roughly defined as follows: we construct a graph (V, E^1) where two vertices are neighbors if their +degrees are similar. Then an edge uv is admissible if u and v have many common neighbors in $E^+ \cap E^1$.

Solving Cluster LP by Preclustering. As we mentioned, we move the complication of handling rounding errors to the step of solving the cluster LP. As in [27], we construct a preclustered instance $(\mathcal{K}, E_{\text{adm}})$, and formulate an LP relaxation aimed at finding the $(1+\varepsilon)$-approximate *good clustering* for $(\mathcal{K}, E_{\text{adm}})$, that we call the *bounded sub-cluster LP*. In contrast to [27], which solves many instances of this LP embedded in their round-or-cut framework, we only solve the LP once, therefore avoiding this heavy framework. With a solution (x, y) to the LP, we run a procedure that constructs a single cluster C randomly. The probability that any vertex is in C is precisely $1/y_\emptyset$, where $y_\emptyset$ is the fractional number of clusters in y. The probabilities that exactly one of u and v is in C, and both of them are in C, are respectively $\frac{x_{uv}}{y_\emptyset}$ and $\frac{1-x_{uv}}{y_\emptyset}$ up to some error terms arising from the Raghavendra-Tan rounding procedure. As usual, x_{uv} is the extent in which u and v are separated.

To construct the solution $z = (z_S)_{S \subseteq V}$ for the cluster LP, we generate $y_\emptyset \Delta$ many clusters C independently, for a large enough polynomial Δ. Roughly speaking, the solution z is $\frac{1}{\Delta}$ times the multi-set of clusters C we generated. The error incurred by the Raghavendra-Tan rounding procedure can be bounded in terms of $|E_{\text{adm}}|$, and the error from sampling can be bounded using concentration bounds.

1.49-approximation. We start with the algorithm of [27], but make several key modifications both in the design and in the analysis. This allows us to significantly improve the approximation ratio, first to 1.5 and, eventually, to 1.49, which shows that, perhaps surprisingly, even the rather low approximation factor of 1.5 is not tight for Correlation Clustering. The first key ingredient is to use a principled budget function for the pivot-based rounding procedure, defined earlier in (5), which is designed to optimally balance the approximation factor of edges between the two rounding procedures. This new budget function is better than the one used in [27], but does not allow us to reach 1.5 without changing the algorithm. Indeed, the budget for the short +edges in +++ triangles is still too low to reach the approximation ratio 1.5. Thus, the second key ingredient is to add the threshold step to the pivot-based rounding procedure for the short +edges (i.e., +edges uv with $x_{uv} \leq 1/3$). By adding this threshold step, the cost of the triangles containing such edges decreases; for example, a +++ triangle with all short edges now has cost zero. This allows us to use the new budget function and still reach 1.5. Notice that making the threshold too large would result in too much cost for ++− triangles.

Finally, we observe that, analogous to the correlated rounding approach of [28], only the *bad triangles* are tight, meaning their cost equals their budget. Roughly speaking, a bad triangle is a ++− triangle whose two +edges have value very close to half and whose −edge has value close to one. This allows us to apply a charging argument, in which tight triangles have part of their cost paid for by triangles that are not tight (i.e., that have extra budget). Now there are no tight triangles (i.e., all triangles have some unused budget), and we can decrease the α in the budget function from 1.5 to 70/47. As previously [4, 21, 27, 28], the analysis necessary to reach 1.5 and go below requires a case-by-case analysis of triangle types to ensure that the budget allocated to each triangle covers its cost. Both the new threshold step and the new budget functions result in an analysis that is more involved than what was required in [27], but is still feasible.

1.437-approximation. The above charging argument between different types of triangles can be more systematically expressed by a *factor-revealing SDP*. Given a cluster LP solution z_S and vertices u, v, w, we define $y_{uv} := \sum_{S \supseteq \{u,v\}} z_S$ (resp. $y_{uvw} := \sum_{S \supseteq \{u,v,w\}} z_S$) be the probability that u, v (resp. u, v, w) are in the same cluster. Given any quadruple $T = (a, b, c, d) \in [0, 1]^4$ and a cluster LP solution z_S, let η_T represent the number of triangles (u, v, w) such that of $y_{uv} = a, y_{uw} = b, y_{vw} = c, y_{uvw} = d$. The above 1.49-approximation analysis can be regarded as putting one constraint on the distribution of η_T. To enhance the approximation ratio and reduce the budget function, we opt for a more detailed categorization of triangles, imposing stronger constraints on η_T.

Consider an imaginary rounding procedure, where given a pivot u, the cluster C that contains u is simply chosen with probability z_C (note that $\sum_{C \ni u} z_C = 1$). Let X_v denote the event that node v is included in the cluster of node u in this rounding. We can show $\mathbb{E}[X_v \cdot X_w] = y_{uvw}$ and $\mathbb{E}[X_v] \cdot \mathbb{E}[X_w] = y_{uv} y_{uw}$. The covariance matrix COV_u, where $COV_u(v, w) = \mathbb{E}[X_v \cdot X_w] - \mathbb{E}[X_v] \cdot \mathbb{E}[X_w] = y_{uvw} - y_{uv} y_{uw}$, must be positive semidefinite (PSD). This PSD constraint on the covariance matrix enforces a stronger constraint on η_T. For instance, if all non-degenerate triangles centered at u are ++− triangles with y value ($y_{uv} = 0.5, y_{uw} = 0.5, y_{wv} = 0, y_{uvw} = 0$), then the covariance matrix of COV_u cannot be PSD because

$COV_u(v,w) = y_{uvw} - y_{uv}y_{uw} = -0.25$ for almost all non-diagonal entries.

For a triangle $T = (y_{uv}, y_{uw}, y_{vw}, y_{uvw})$, we discretize y_{uv}, y_{uw}, y_{vw} to incorporate the PSD constraint. We partition the interval $[0,1]$ into numerous subintervals $I_1, I_2, ..., I_t$. Each triangle with y value $(y_{uv} \in I_i, y_{uw} \in I_j, y_{vw} \in I_k, y_{uvw})$ is placed in one of these interval combinations. We can rearrange COV_u as $Q_u \in \mathbb{R}^{t\times t}$, where $Q_u(I_i, I_j) = \sum_{y_{uv}\in I_i, y_{uw}\in I_j}(y_{uvw} - y_{uv}y_{uw})$. Considering $Q = \sum_{u\in V} Q_u$, we can represent Q using T and η_T. The PSD property of Q_u implies Q is PSD, thus enforcing a constraint on η_T.

Despite there being infinitely many types of triangles in each range I_i, I_j, I_k, our key observation is that $y_{uvw} - y_{uv}y_{uw}$ is multilinear. Therefore, we only need a few triangles in each range to represent all possible triangles. We want to mention the triangles we need are fixed so can be precomputed and the only unsure variable is η_T. To compute a lower bound $\sum \eta_T(\Delta(T) - \text{cost}(T))$, we set up a semi-definite program (SDP) under the constraint that Q is PSD. This SDP is independent of cluster LP and relies on the chosen interval and budget function. By employing a practical SDP solver, we demonstrate that $\sum \eta_T(\Delta(T) - \text{cost}(T)) \geq 0$.

Gaps and Hardness. A high-level intuition for the cluster LP is the following: (any) LPs cannot distinguish between a random graph and a nearly bipartite graph. For the cluster LP, given a complete graph $H = (V_H, E_H)$ with $n = |V_H|$, our Correlation Clustering instance is $G = (V_G, E_G)$ where $V_G = E_H$ and $e, f \in V_G$ have a plus edge in G if they share a vertex in V. Consider vertices of H as *ideal clusters* in G containing their incident edges. The LP fractionally will think that it is nearly bipartite, implying that the entire E_H can be partitioned into $n/2$ ideal clusters of the same size. Of course, integrally, such a partition is not possible in complete graphs.

For the cluster LP, it suffices to consider a complete graph instead of a random graph. We believe (but do not prove) that such a gap instance can be extended to stronger LPs (e.g., Sherali-Adams strengthening of the cluster LP), because it is known that Sherali-Adams cannot distinguish a random graph and a nearly bipartite graph [19].

The idea for the NP-hardness of approximation is the same. The main difference, which results in a worse factor here, is that other polynomial-time algorithms (e.g., SDPs) can distinguish between random and nearly bipartite graphs! So, we are forced to work with slightly more involved structures.

Still, we use a similar construction for 3-uniform hypergraphs; let $H = (V_H, E_H)$ be the underlying 3-uniform hypergraph and $G = (V_G, E_G)$ be the plus graph of the final Correlation Clustering instance where $V_G = E_H$ and $e, f \in E_H$ has an edge in G if they share a vertex in H. We use the hardness result of Cohen-Addad, Karthik, and Lee [26] that shows that it is hard to distinguish whether H is *nearly bipartite*, which implies that half of the vertices intersect every hyperedge, or close to a random hypergraph.

Organization. We show how to solve the cluster LP in Section 4, proving Theorem 1. We give the $(\frac{4}{3}-\varepsilon)$-integrality gap of the cluster LP (Theorem 4) in Section 5, and the improved hardness of $24/23-\varepsilon$ (Theorem 5) in Section 6.

Global Notations. For two sets A and B, we use $A \triangle B = (A \setminus B) \cup (B \setminus A)$ to denote the symmetric difference between A and B. We used N_u^+ and N_u^- to denote the sets of + and −neighbors of a vertex u respectively in the Correlation Clustering instance. For a clustering C of V, we define $\text{obj}(C)$ to be the objective value of C. For any $x \in [0,1]^{\binom{V}{2}}$, we already defined $\text{obj}(x) = \sum_{uv\in E^+} x_{uv} + \sum_{uv\in E^-}(1 - x_{uv})$. Recall that we defined $\text{cost}_u(v,w)$, $\Delta_u(v,w)$, $\text{cost}(T)$ and $\Delta(T)$ for a triangle $T = (u,v,w)$ or a degenerate triangle $T = (u,v)$ in Section 2; they depend on the budget functions b^+ and b^-.

4 SOLVING CLUSTER LP RELAXATION APPROXIMATELY

In this section, we show how to solve the cluster LP in polynomial time, by proving Theorem 1, which is repeated below.

Theorem 1. *Let $\varepsilon > 0$ be a small enough constant and* opt *be the cost of the optimum solution to the given Correlation Clustering instance. In time $n^{\text{poly}(1/\varepsilon)}$, we can output a feasible cluster LP solution $\left((z_S)_{S\subseteq V}, (x_{uv})_{uv\in\binom{V}{2}}\right)$ with $\text{obj}(x) \leq (1+\varepsilon)$opt, described using a list of non-zero coordinates.*

We define some global parameters used across this section. Let $\varepsilon_1 = \varepsilon^3, \varepsilon_{\text{rt}} = \varepsilon_1^2 = \varepsilon^6$, and $r = \Theta(1/\varepsilon_{\text{rt}}^2) = \Theta(1/\varepsilon^{12})$ be an integer, with some large enough hidden constant. The subscript "rt" stands for Raghavendra-Tan.

4.1 Preclustering

We use the definition of a preclustered instance from [27], with some minor modifications.

Definition 12. *Given a Correlation Clustering instance $(V, E^+ \uplus E^-)$, a* preclustered instance *is defined by a pair $(\mathcal{K}, E_{\text{adm}})$, where $\mathcal{K}$ is a partition of V (which can also be viewed as a clustering), and $E_{\text{adm}} \subseteq \binom{V}{2}$ is a set of pairs such that for every $uv \in E_{\text{adm}}$, u and v are not in a same set in $\mathcal{K}$.*

Each set $K \in \mathcal{K}$ is called an atom. *An (unordered) pair uv between two vertices u and v in a same $K \in \mathcal{K}$ is called an* atomic edge*; in particular, a self-loop uu is an atomic edge. A pair that is neither an atomic nor an admissible edge is called a* non-admissible *edge.*

There are two minor differences between our definition and the one in [27]. First, we require that $\mathcal{K}$ forms a partition; this can be guaranteed by adding singletons. Second, we do not require an edge between two different non-singleton atoms to be non-admissible. Our construction can guarantee this condition, but it is not essential.

Definition 13. *Given a preclustered instance $(\mathcal{K}, E_{\text{adm}})$ for some Correlation Clustering instance $(V, E^+ \uplus E^-)$, a clustering C of V is called* good *with respect to $(\mathcal{K}, E_{\text{adm}})$ if*

- *u and v are in the same cluster in C for an atomic edge uv, and*
- *u and v are not in the same cluster in C for a non-admissible edge uv.*

The following theorem with a worse bound on $|E_{\text{adm}}|$ was proved in [27]. We give a cleaner proof of the theorem in the full paper; as a byproduct, it achieves a better bound on $|E_{\text{adm}}|$.

Theorem 14. *For any sufficiently small $\varepsilon > 0$, there exists a $\text{poly}(n, \frac{1}{\varepsilon})$-time algorithm that, given a Correlation Clustering instance $(V, E^+ \uplus E^-)$ with optimal value* opt *(which is not given to us), produces a preclustered instance $(\mathcal{K}, E_{\text{adm}})$ such that*

- *there exists a good clustering w.r.t* $(\mathcal{K}, E_{\text{adm}})$*, whose cost is at most* $(1+\varepsilon)\text{opt}$*, and*
- $|E_{\text{adm}}| \le O(\frac{1}{\varepsilon^2}) \cdot \text{opt}$.

We can assume in the preclustered instance $(\mathcal{K}, E_{\text{adm}})$, the edges between two different atoms K and K' are all admissible, or all non-admissible. If one edge between them is non-admissible, we can change all other edges to non-admissible edges. This will not change the set of good clusterings, and it will decrease $|E_{\text{adm}}|$.

We apply Theorem 14 to obtain a preclustered instance $(\mathcal{K}, E_{\text{adm}})$, with the unknown good clustering C_1^*. We define K_u to be the atom that contains u, and $k_u = |K_u|$. We shall use $N_{\text{adm}}(u)$ to be the set of vertices v such that $uv \in E_{\text{adm}}$; so $N_{\text{adm}}(u) = N_{\text{adm}}(v)$ if $v \in K_u$. We further process the good clustering C_1^* using the following procedure in [27]. This procedure is not a part of our algorithm; it is only for analysis purpose.

1: **while** there exists some K_u in a cluster $C \in C_1^*$ with $k_u < |C| \le k_u + \varepsilon_1 \cdot |N_{\text{adm}}(u)|$ **do**
2: $\quad C_1^* \leftarrow C_1^* \setminus \{C\} \cup \{K_u, C \setminus K_u\}$

Claim 15. *The procedure increases* $\text{obj}(C_1^*)$ *by at most* $2\varepsilon_1 \cdot |E_{\text{adm}}|$.

Proof. Whenever we break C into K_u and $C \setminus K_u$ in the procedure, the cost increase is at most $k_u \cdot (|C| - k_u) \le k_u \cdot \varepsilon_1 \cdot |N_{\text{adm}}(u)| = \varepsilon_1 \sum_{v \in K_u} |N_{\text{adm}}(v)|$. We separate each atom K_u at most once. Therefore, the total cost increase is at most $\varepsilon_1 \sum_{v \in V} |N_{\text{adm}}(v)| = 2\varepsilon_1 \cdot |E_{\text{adm}}|$. □

So, the cost of C_1^* after the procedure will be at most $(1+\varepsilon)\text{opt} + O(\varepsilon_1)|E_{\text{adm}}|$. Crucially, the following property is satisfied:

(A1) For every $u \in V$, K_u is either a cluster in C_1^*, or in a cluster of size more than $k_u + \varepsilon_1 \cdot |N_{\text{adm}}(u)|$.

4.2 Bounded Sub-Cluster LP Relaxation for Preclustered Instances

Following [27], we form an LP relaxation aiming at finding the good clustering C_1^*. In the LP, we have a variable y_S^s, for every $s \in [n]$, and $S \subseteq V$ of size at most r (recall that $r = \Theta(1/\varepsilon^{12})$), that denotes the number of clusters in C_1^* of size s containing S as a subset. When $S \neq \emptyset$, there is at most one such cluster and thus $y_S^s \in \{0,1\}$ indicates if S is a subset of a cluster of size s in C_1^*. For every $S \subseteq V$ of size at most r, let $y_S := \sum_s y_S^s$ denote the number of clusters (of any size) in C_1^* containing S as a subset. Again, if $S \neq \emptyset$, then $y_S \in \{0,1\}$ indicates if S is a subset of a cluster in C_1^*. For every $uv \in \binom{V}{2}$, we have a variable x_{uv} indicating if u and v are separated or not in C_1^*. We call the LP the *bounded sub-cluster LP relaxation*, as we have variables indicating if a small set S is a subset of a cluster or not.

We use the following type of shorthand: y_u^s for $y_{\{u\}}^s$, y_{uv}^s for $y_{\{u,v\}}^s$, and y_{Su}^s for $y_{S\cup\{u\}}^s$. The bounded sub-cluster LP is defined as follows. In the description, we always have $s \in [n], u \in V$ and $uv \in \binom{V}{2}$. For convenience, we omit the restrictions. By default, any variable of the form y_S or y_S^s has $|S| \le r$; if not, we do not have the variable and the constraint involving it.

$$\min \quad \text{obj}(x) \qquad \text{(bounded sub-cluster LP)}$$

$$\sum_{s=1}^{n} y_S^s = y_S \quad \forall S \quad (6) \qquad \frac{1}{s}\sum_u y_{Su}^s = y_S^s \quad \forall s, S \quad (9)$$

$$y_u = 1 \quad \forall u \quad (7) \qquad y_S^s \ge 0 \quad \forall s, S \quad (10)$$

$$y_{uv} + x_{uv} = 1 \quad \forall uv \quad (8)$$

$$x_{uv} = 0 \quad \forall u, v \text{ in a same } K \in \mathcal{K} \quad (11)$$

$$x_{uv} = 1 \quad \forall \text{non-admissible edge } uv \quad (12)$$

$$y_u^s = 0 \quad \forall u, s \in [k_u - 1] \cup \left[k_u + 1, k_u + \varepsilon_1 |N_{\text{adm}}(u)|\right] \quad (13)$$

$$\sum_{T' \subseteq T} (-1)^{|T'|} y_{S\cup T'}^s \in [0, y_S^s] \forall s, S \cap T = \emptyset \quad (14)$$

(6) gives the definition of y_S, (7) requires u to be contained in some cluster, and (8) gives the definition of x_{uv}. (9) says if $y_S^s = 1$, then there are exactly s elements $u \in V$ with $y_{Su}^s = 1$. (An exception is when $S = \emptyset$; but the equality also holds.) (10) is the non-negativity constraint. (11) and (12) follows from that C_1^* is a good clustering, and (13) follows from (A1). The left side of (14) is the number of clusters of size s containing S but does not contain any vertex in T. So the inequality holds. This corresponds to a Sherali-Adams relaxation needed for the correlated rounding [41], see Lemma 16. The running time for solving the LP is $n^{O(r)} = n^{O(1/\varepsilon^{12})}$.

4.3 Sampling One Cluster Using LP Solution to the Bounded Sub-Cluster LP

We solve the bounded sub-cluster LP to obtain the y and x vectors. Given y, we can use the procedure construct-cluster described in Algorithm 3, which is from [27], to produce a random cluster C.

Algorithm 3 construct-cluster(y)

1: randomly choose a cardinality s, so that s is chosen with probability $\frac{y_\emptyset^s}{y_\emptyset}$
2: randomly choose a vertex $u \in V$, so that u is chosen with probability $\frac{y_u^s}{s y_\emptyset^s}$
3: define a vector y' such that $y'_S = \frac{y_{Su}^s}{y_u^s}$ for every $S \subseteq V$ of size at most $r-1$
4: apply the Raghavendra-Tan correlated rounding technique over the fractional set y' to construct a cluster $C \subseteq V$ that does not break any atom, and **return** C

With (14), the Raghavendra-Tan technique can be applied:

Lemma 16 ([41]). *In Step 4 of Algorithm 3, one can sample a set* $C \subseteq V$ *that does not break atoms in time* $n^{O(r)}$ *such that*

- *For each* $v \in V$, $\Pr[v \in C] = y'_v$.
- $\frac{1}{|N_{\text{adm}}(u)|^2} \sum_{v,w \in N_{\text{adm}}(u)} \left|\Pr[v, w \in C] - y'_{vw}\right| \le \varepsilon_{\text{rt}}$.

Recall that $\varepsilon_{\text{rt}} = \Theta(1/\sqrt{r})$ and the hidden constant inside $\Theta(\cdot)$ is large enough.

As in [27], we define $\mathrm{err}^s_{vw|u}$ to be the error generated by the procedure when we choose s as the cardinality and u as the pivot:

$$\mathrm{err}^s_{vw|u} := \left|\Pr\left[v, w \in C | s, u\right] - \frac{y^s_{uvw}}{y^s_u}\right|, \forall vw \in \binom{V}{2},$$

and

$$\mathrm{err}^s_{vw} := \frac{1}{s y^s_\emptyset} \sum_{u \in V} y^s_u \cdot \mathrm{err}^s_{vw|u} \text{ and } \mathrm{err}_{vw} := \sum_s \frac{y^s_\emptyset}{y_\emptyset} \cdot \mathrm{err}^s_{vw}$$

as the error for vw conditioned on s, and the unconditioned error. Notice that all these quantities are expectations of random variables, and thus deterministic.

The following two lemmas can be proved using the same arguments as in [27].

Lemma 17 ([27]). *For any $v \in V$, we have $\Pr[v \in C] = \frac{1}{y_\emptyset}$.*

Lemma 18 ([27]). *Focus on an edge $vw \in \binom{V}{2}$.*

(1) $\Pr\left[v \in C, w \notin C\right] \le \frac{1}{y_\emptyset} \cdot x_{vw} + \mathrm{err}_{vw}$.

(2) $\Pr\left[|\{v, w\} \subseteq C\right] \le \frac{1}{y_\emptyset} \cdot y_{vw} + \mathrm{err}_{vw}$.

A similar lemma to the following is proved in [27]. The parameters we use here are slightly different and we provide a proof for completeness.

Lemma 19. $\sum_{vw \in \binom{V}{2}} \mathrm{err}_{vw} \le O(\varepsilon_1) \cdot \frac{1}{y_\emptyset} |E_{\mathrm{adm}}|$.

Proof. Throughout the proof, we assume u, v, w are all in V, vw and uw are in $\binom{V}{2}$.

Fix some $s \in [n], u \in V$ with $y^s_u > 0$, and we now bound $\sum_{vw} \mathrm{err}^s_{vw|u}$. If $s = k_u$, then $C = K_u$; no errors will be created and the quantity is 0. Assume $s > k_u$. By (13), we have that $s > k_u + \varepsilon_1 \cdot |N_{\mathrm{adm}}(u)|$, since otherwise we shall $y^s_u = 0$. By the second property of Lemma 16, we have $\sum_{vw} \mathrm{err}^s_{vw|u} \le \frac{\varepsilon_{\mathrm{rt}}}{2} |N_{\mathrm{adm}}(u)|^2$. (Notice that if one of v and w is not in $N_{\mathrm{adm}}(u)$, then $\mathrm{err}^s_{vw|u} = 0$.) Recall that $\varepsilon_{\mathrm{rt}} = \varepsilon_1^2$. Therefore,

$$\sum_{vw \in \binom{V}{2}} \mathrm{err}^s_{vw|u} \le \frac{\varepsilon_{\mathrm{rt}}}{2} \cdot |N_{\mathrm{adm}}(u)|^2 \le \frac{\varepsilon_{\mathrm{rt}}}{2\varepsilon_1} \cdot |N_{\mathrm{adm}}(u)| \cdot (s - k_u)$$

$$= \frac{\varepsilon_1}{2} \cdot |N_{\mathrm{adm}}(u)| \cdot \sum_{v \in N_{\mathrm{adm}}(u)} \frac{y^s_{uv}}{y^s_u} = \frac{\varepsilon_1}{2} \cdot \sum_{v,w \in N_{\mathrm{adm}}(u)} \frac{y^s_{uv}}{y^s_u}.$$

The first equality is by (9) and $y^s_{uv} = y^s_u$ for every $v \in K_u$. (To see this, notice that $y^s_{uv} \le y^s_u$ is implied by (14). We have $y_{uv} = \sum_s y^s_{uv}$, $y_u = \sum_s y^s_u$, and $y_{uv} = y_u = 1$ if $v \in K_u$.)

Considering the inequalities over all $u \in V$, we have

$$\sum_{vw} \mathrm{err}^s_{vw} = \frac{1}{s y^s_\emptyset} \sum_u y^s_u \cdot \sum_{vw} \mathrm{err}^s_{vw|u}$$

$$\le \frac{1}{s y^s_\emptyset} \sum_u y^s_u \cdot \sum_{v,w \in N_{\mathrm{adm}}(u)} \frac{\varepsilon_1}{2} \cdot \frac{y^s_{uv}}{y^s_u}$$

$$= \frac{\varepsilon_1}{2} \cdot \frac{1}{s y^s_\emptyset} \cdot \sum_{u \in V, v,w \in N_{\mathrm{adm}}(u)} y^s_{uv}$$

$$= \frac{\varepsilon_1}{2} \cdot \sum_{v \in V} \frac{y^s_v}{s y^s_\emptyset} \sum_{u \in N_{\mathrm{adm}}(v), w \in N_{\mathrm{adm}}(u)} \frac{y^s_{uv}}{y^s_v}$$

$$\le \frac{\varepsilon_1}{2} \cdot \sum_{v \in V} \frac{y^s_v}{s y^s_\emptyset} \sum_{uw \in E_{\mathrm{adm}}} \left(\frac{y^s_{uv} + y^s_{vw}}{y^s_v}\right)$$

$$\le \varepsilon_1 \cdot \sum_{v \in V} \frac{y^s_v}{s y^s_\emptyset} \sum_{uw \in E_{\mathrm{adm}}} \Pr[C \cap \{u, w\} \ne \emptyset \mid s, v \text{ is pivot}]$$

$$= \varepsilon_1 \sum_{uw \in E_{\mathrm{adm}}} \Pr[C \cap \{u, w\} \ne \emptyset \mid s].$$

To see the last inequality, notice that $\frac{y^s_{uv}}{y^s_v} = \Pr[u \in C | s, v \text{ is pivot}] \le \Pr[C \cap \{u, w\} \ne \emptyset | s, v \text{ is pivot}]$. The same inequality holds for $\frac{y^s_{vw}}{y^s_v}$.

Finally, we take all s into consideration:

$$\sum_{vw} \mathrm{err}_{vw} = \sum_s \frac{y^s_\emptyset}{y_\emptyset} \cdot \sum_{vw} \mathrm{err}^s_{vw}$$

$$\le \varepsilon_1 \cdot \sum_s \frac{y^s_\emptyset}{y_\emptyset} \sum_{uw \in E_{\mathrm{adm}}} \Pr[C \cap \{u, w\} \ne \emptyset | s]$$

$$= \varepsilon_1 \cdot \sum_{uw \in E_{\mathrm{adm}}} \Pr[C \cap \{u, w\} \ne \emptyset]$$

$$\le \frac{2\varepsilon_1}{y_\emptyset} |E_{\mathrm{adm}}| + 3\varepsilon_1 \sum_{uw \in \binom{V}{2}} \mathrm{err}_{uw}.$$

To see the last inequality, we notice that $C \cap \{u, w\} \ne \emptyset$ is the union of the 3 disjoint events: $u \in C$ and $w \notin C$, $u \notin C$ and $w \in C$, and $\{u, w\} \notin C$. By Lemma 18, we have $\Pr[C \cap \{u, w\} \ne \emptyset] \le \frac{2x_{vw} + y_{vw}}{y_\emptyset} + 3 \cdot \mathrm{err}_{uw} \le \frac{2}{y_\emptyset} + 3 \cdot \mathrm{err}_{uw}$. So, we have $\sum_{vw} \mathrm{err}_{vw} \le \frac{1}{1 - 3\varepsilon_1} \cdot \frac{2\varepsilon_1}{y_\emptyset} |E_{\mathrm{adm}}|$. This proves the lemma. □

4.4 Construction of Solution to the Cluster LP Using Independently Sampled Clusters

With all the ingredients, we can now describe our algorithm for solving the cluster LP approximately, finishing the proof of Theorem 1. Let $\Delta = \Theta\left(\frac{n^2 \log n}{\varepsilon_1^2 |E_{\mathrm{adm}}|}\right)$ with a large enough hidden constant, and $\Delta y_\emptyset$ being an integer. (We assume $|E_{\mathrm{adm}}| \ge 1$ since otherwise the preclustered instance is trivial.) We run Algorithm 3 $\Delta y_\emptyset$ times independently to obtain clusters $C_1, C_2, \cdots, C_{\Delta y_\emptyset}$.

We use the following variant of Chernoff bound.

Theorem 20. *Let $X_1, X_2, X_3, \cdots, X_n$ be independent (not necessarily iid) random varibles which take values in $[0, 1]$. Let $X = \sum_{i=1}^n X_i$, $\mu = \mathbb{E}[X]$, and $\mu' \ge \mu$ be a real. Then for any $\delta \in (0, 1)$, we have*

$$\Pr[X < (1 - \delta)\mu] < e^{-\delta^2 \mu / 2} \quad \text{and} \quad \Pr[X > \mu + \delta\mu'] < e^{-\delta^2 \mu' / 3}.$$

For every $u \in V$, let $R_u = \{t : u \in C_t\}$. Notice that $\Delta y_\emptyset \cdot \frac{|E_{\mathrm{adm}}|}{y_\emptyset n^2} = \Theta(\frac{\log n}{\varepsilon_1^2})$, with a large enough hidden constant. Using Chernoff bound and union bound, we can prove that with probability at least $1 - 1/n$, the following conditions hold.

- For every $u \in V$, we have $|R_u| \ge (1 - \varepsilon_1)\Delta y_\emptyset \cdot \frac{1}{y_\emptyset} = (1 - \varepsilon_1)\Delta$.
- For every $u, v \in V$ such that $uv \in E^+$, we have

$$|R_u \setminus R_v| \le \Delta y_\emptyset \left(\frac{x_{uv}}{y_\emptyset} + \mathrm{err}_{uv} + \varepsilon_1 \cdot \max\left\{\frac{x_{uv}}{y_\emptyset} + \mathrm{err}_{uv}, \frac{|E_{\mathrm{adm}}|}{y_\emptyset n^2}\right\}\right)$$

$$\leq (1+\varepsilon_1)\Delta(x_{uv}+y_\emptyset \mathrm{err}_{uv})+\frac{\varepsilon_1\Delta|E_{\mathrm{adm}}|}{n^2}. \tag{15}$$

- For every $uv \in E^-$, we have

$$|R_u \cap R_v| \leq \Delta y_\emptyset \left(\frac{y_{uv}}{y_\emptyset}+\mathrm{err}_{uv}+\varepsilon_1\cdot\max\left\{\frac{y_{uv}}{y_\emptyset}+\mathrm{err}_{uv}, \frac{|E_{\mathrm{adm}}|}{y_\emptyset n^2}\right\}\right)$$
$$\leq (1+\varepsilon_1)\Delta(y_{uv}+y_\emptyset \mathrm{err}_{uv})+\frac{\varepsilon_1\Delta|E_{\mathrm{adm}}|}{n^2}.$$

From now on we assume the conditions hold. For every $u \in V$, we let R'_u be the set of the $\lceil(1-\varepsilon)\Delta\rceil$ smallest indices in R_u. Clearly, $|R'_u \cap R'_v| \leq |R_u \cap R_v|$. We show $|R'_u \setminus R'_v|$ is still upper bounded by (15).

Claim 21. *For every $uv \in E^+$ we have* $\max\{|R'_u \setminus R'_v|, |R'_v \setminus R'_u|\} \leq (1+\varepsilon_1)\Delta(x_{uv}+y_\emptyset \mathrm{err}_{uv})+\frac{\varepsilon_1\Delta|E_{\mathrm{adm}}|}{n^2}$.

Proof. For convenience, we use B to denote the upper bound $(1+\varepsilon_1)\Delta(x_{uv}+y_\emptyset \mathrm{err}_{uv})+\frac{\varepsilon_1\Delta|E_{\mathrm{adm}}|}{n^2}$. We think of R'_u (R'_v resp.) as obtained from the set R_u (R_v resp.) by removing the largest indices one by one. Wlog we assume $|R_u| \geq |R_v|$; and thus initially $|R_v \setminus R_u| \leq |R_u \setminus R_v| \leq B$. We remove the elements from R_u and R_v in two stages.

In the first stage we do the following. While $|R_u| > |R_v|$, we remove the largest index from R_u. This can not increase $|R_u \setminus R_v|$. After the first stage, we have $|R_u \setminus R_v| = |R_v \setminus R_u| \leq B$.

In the second stage we do the following. While $|R_u| = |R_v| > \lceil(1-\varepsilon)\Delta\rceil$, we remove the largest index in R_u from R_u, and do the same for R_v. Consider one iteration of the while loop. If the two indices are the same, then $|R_u \setminus R_v| = |R_v \setminus R_u|$ does not change. Otherwise, wlog we assume the index we removed from R_u is larger. Then removing the index in R_u will decrease $|R_u \setminus R_v|$. So the iteration can not increase $|R_u \setminus R_v| = |R_v \setminus R_u|$. □

Then, for every $t \in [1, \Delta y_\emptyset]$, we define $C'_t = \{u : t \in R'_u\} \subseteq C_t$; then every v is contained in C'_t for exactly $\lceil(1-\varepsilon)\Delta\rceil$ values of t. We define $z_S = \frac{1}{\lceil(1-\varepsilon)\Delta\rceil}\cdot|\{t : C'_t = S\}|$ for every $S \subseteq V$ with $S \neq \emptyset$. Define $\tilde{x}_{uv} = 1 - \sum_{\{u,v\}\subseteq S} z_S$ for every $uv \in \binom{V}{2}$. Then $(\tilde{x}, z)$ is a valid solution to the cluster LP.

For a $uv \in E^+$, we have

$$\tilde{x}_{uv} = \frac{1}{\lceil(1-\varepsilon)\Delta\rceil}\cdot|R'_u \setminus R'_v| \leq \frac{1+\varepsilon_1}{1-\varepsilon}(x_{uv}+y_\emptyset \mathrm{err}_{uv})+\frac{\varepsilon_1|E_{\mathrm{adm}}|}{(1-\varepsilon)n^2}.$$

For a $uv \in E^-$, we have

$$(1-\tilde{x}_{uv}) \leq \frac{1+\varepsilon_1}{1-\varepsilon}(1-x_{uv}+y_\emptyset \mathrm{err}_{uv})+\frac{\varepsilon_1|E_{\mathrm{adm}}|}{(1-\varepsilon)n^2}.$$

Therefore,

$$\mathrm{obj}(\tilde{x}) \leq (1+O(\varepsilon))\left(\mathrm{obj}(x)+y_\emptyset\sum_{uv\in\binom{V}{2}}\mathrm{err}_{uv}\right)+O(\varepsilon_1)|E_{\mathrm{adm}}|$$
$$\leq (1+O(\varepsilon))\mathrm{obj}(x)+O(\varepsilon_1)|E_{\mathrm{adm}}|$$
$$\leq (1+O(\varepsilon))\cdot\mathrm{opt}+O(\varepsilon^3)\cdot O\left(\frac{1}{\varepsilon^2}\right)\cdot\mathrm{opt} = (1+O(\varepsilon))\mathrm{opt}.$$

The second inequality is due to Lemma 19, and the third one used that $|E_{\mathrm{adm}}| \leq O\left(\frac{1}{\varepsilon^2}\right)\cdot\mathrm{opt}$. By scaling ε, the upper bound can be made to $(1+\varepsilon)\mathrm{opt}$. This finishes the proof of Theorem 1.

5 1.33-GAP FOR CLUSTER LP

In this section, we show that the cluster LP has a gap of 4/3, proving Theorem 4 restated below.

Theorem 4. *For any $\varepsilon > 0$, the integrality gap of the cluster LP is at least $4/3-\varepsilon$.*

The graph of the plus edges of our gap instance is based on the line graph of a base graph; given a based graph $H = (V_H, E_H)$, our correlation clustering instance is $G = (V_G, E_G)$ where $V_G = E_H$ and $e, f \in V_G$ have a plus edge in G if they share a vertex in V_H.

A high-level intuition is the following: LPs cannot distinguish between a random graph and a nearly bipartite graph. Consider vertices of H as *ideal clusters* in G containing their incident edges. Given a random graph H, the LP fractionally will think that it is nearly bipartite, implying that the almost entire E_H can be partitioned into $n/2$ ideal clusters. Of course, integrally, such a partition is not possible in random graphs. For the cluster LP, it suffices to consider a complete graph instead of a random graph. We believe (but do not prove) that such a gap instance can be extended to stronger LPs (e.g., Sherali-Adams strengthening of the cluster LP), because it is known that Sherali-Adams cannot distinguish a random graph and a nearly bipartite graph [19].

Proof of Theorem 4. Let $H = (V_H, E_H)$ be a complete graph on n vertices. Let $d = n-1$ be the degree of H. Our correlation clustering instance $G = (V_G, E_G)$ is the line graph of H; $V_G = E_H$ and $e, f \in E_H$ has + edge in G if and only if they share a vertex in H. The + degree of each $e \in E_H$ in G is $2d-2$.

Consider the following solution for the cluster LP: for every $v \in V_H$, let $E_v \subseteq E_H$ be the d edges containing v. The cluster LP has $z_{E_v} = 1/2$ for every $v \in v_H$. Each $e \in E_H$ belongs to two fractional clusters, each of which has its $d-1$ plus neighbors, so fractionally $d-1$ plus edges incident on it are violated. Since each violated edge is counted twice, the LP value is $\binom{n}{2}(d-1)/2$.

Let us consider the integral optimal correlation clustering of G. Consider a cluster C in the clustering. Note that every vertex in C has at least $|C|/2$ plus neighbors in C, which implies $|C| \leq 4d$. We apply the following procedure to C to partition it further.

Claim 22. *There is a partition of C into $C_1, \ldots, C_r$ such that (1) each C_i is a subset of E_v for some $v \in V_H$, and (2) replacing C by $C_1, \ldots, C_r$ in the correlation clustering solution increases the objective function by at most $35|C|$.*

Proof. For $v \in V_H$, let $n_v := |C \cap E_v|$. Note that $\sum_v n_v = 2|C|$. Without loss of generality, assume $V_H = \{v_1, \ldots, v_n\}$ with $n_{v_1} \geq \cdots \geq n_{v_n}$. If $e = (v_i, v_j) \in C$ has $i, j > 8$, then the number of its plus neighbors in C is $n_{v_i} + n_{v_j} < 2\cdot\frac{1}{8}\cdot 2|C| = |C|/2$, so it should not exist in C. So, every edge is incident on v_i for some $i \leq 8$.

Let us make at most $\binom{8}{2} = 28$ edges in C between $v_1, \ldots, v_8$ as singleton clusters; the objective function increases by at most $28|C|$. Then partition the remaining C into $E_1, \ldots, E_8$ where $E_i := C \cap E_{v_i}$. Each $e \in E_i$ has at most seven plus neighbors in $\cup_{j\neq i}E_j$, so the objective function increases by at most $7|C|$. So, we partitioned C into $C_1, \ldots, C_r$ where all the edges in C_i share a common endpoint. We increased the objective function by at most $35|C|$. □

After we apply the above procedure to every cluster C, we increased the cost by at most $35|V_H| \leq 35n^2$ and all the edges in

a cluster C share a common endpoint. For $v \in V_H$, let C_v be the cluster in the solution whose common endpoint is v. (If there are many of them, merging them will strictly improve the objective function value.) Without loss of generality, there are t such clusters $C_{v_1}, \ldots, C_{v_t}$ and let $n_i := |C_{v_i}|$ such that $n_1 \geq \cdots \geq n_t$.

Claim 23. $\sum_{i=1}^{t} n_i^2 \leq n^3/3$.

Proof. The LHS is monotone in $(n_1, \ldots, n_t)$, and if there is an edge $(v_i, v_j) \in C_j$ with $j > i$ (which implies $n_i \geq n_j$), the LHS strictly improves by moving (v_i, v_j) to C_i. Therefore, the configuration that maximizes the LHS is when $t = n$ and C_{v_i} contains all the edges of H not incident on $v_1, \ldots, v_{i-1}$. In that case, the LHS is

$$\sum_{i=1}^{n-1}(n-i)^2 = n^3 \sum_{i=1}^{n-1}(\frac{n-i}{n})^2 \cdot \frac{1}{n} \leq n^3 \int_0^1 (1-x)^2 dx$$
$$= n^3[x - x^2 + x^3/3]_0^1 = n^3/3,$$

as desired. □

Using this, we can prove a lower bound on the cost of our near-optimal clustering. Note that every cluster is a clique of +edges. Thus, the only edges violated are +edges. Moreover, there are at most $\sum_{i\in[t]} n_i^2/2 \leq n^3/6$ correctly clustered +edges. The cost of our near-optimal clustering is the total number of +edges of G minus the number of correctly clustered +edges, namely at most $\binom{n}{2}(d-1) - n^3/6 = n^3/3 - o(n^3)$. Since the cost of the optimal clustering is at most $35n^2$ lower than ours, it is still $n^3/3 - o(n^3)$. The fractional solution has the value at most $n^3/4$, so the gap is at least $4/3 - o(1)$. □

6 1.04-NP HARDNESS

In this section, we show that it is NP-hard (under randomized reductions) to obtain an algorithm with an approximation ratio of $24/23 \geq 1.043$, proving Theorem 5 restated below.

The idea is similar to the gap for the cluster LP in Section 5, which is based on the fact that the LPs generally cannot distinguish nearly bipartite graphs and random graphs. The main difference, which results in a worse factor here, is that other polynomial-time algorithms (e.g., SDPs) can distinguish between them! So, we are forced to work with slightly more involved structures.

Still, we use a similar construction for 3-uniform hypergraphs; let $H = (V_H, E_H)$ be the underlying 3-uniform hypergraph and $G = (V_G, E_G)$ be the plus graph of the final Correlation Clustering instance where $V_G = E_H$ and $e, f \in E_H$ has an edge in G if they share a vertex in H. We use the following hardness result of Cohen-Addad, Karthik, and Lee [26] that shows that it is hard to distinguish whether H is *nearly bipartite* or close to a random hypergraph.

Theorem 24. *For any $\varepsilon > 0$, there exists a randomized polynomial-time algorithm that receives a 3-CNF formula ϕ as input and outputs a simple 3-uniform hypergraph $H = (V_H, E_H)$ where the degree of each vertex is $(1 \pm o(1))d$ for some $d = \omega(|V_H|)$ such that the following properties are satisfied with high probability.*

- *(YES) If ϕ is satisfiable, there exists $U \subseteq V_H$ with $|U| = |V_H|/2$ that intersects every hyperedge in E_H. Moreover, for every $u \in U$, $|\{e \in E_H : e \cap U = \{u\}\}| \geq (1/2 - \varepsilon)d$.*
- *(NO) If ϕ is unsatisfiable, any set of $\gamma|V_H|$ vertices ($\gamma \in [0,1]$) do not intersect at least a $(1-\gamma)^3 - \varepsilon$ fraction of hyperedges in E_H.*

Proof. The same reduction in Theorem 4.1 of (the arXiv version of) [26] yields the desired hardness. In the following, we highlight the difference between the statement of Theorem 4.1 of [26] and our Theorem 24 and briefly explain how our additional properties are satisfied by their reduction.

(1) Regularity of H: Section 4.5 of [26], based on an earlier weighted hard instance, constructs the final hard instance $H = (V_H, E_H)$ as a certain random hypergraph where the degree of each vertex v is the sum of independent $\{0,1\}$ variables with the same expected value. This expected value is $\Theta(|V_H|^{1.5})$, so the standard Chernoff and union bound argument will show that the degree of each vertex is almost the same with high probability.
(2) In the (YES) case, for every $u \in U$, $|\{e \in E_H : e \cap U = \{u\}\}| \geq (1/2 - \varepsilon)d$: It follows from their construction in Section 4.1. The construction is analogous to Håstad's celebrated result on Max-3SAT [34] where in the (YES) case, almost three quarters of the clauses have one true literal and almost one quarter have three true literals, so that for each true literal ℓ, roughly half of the clauses containing ℓ has it as the only true literal.
(3) In the (NO) case: the guarantee holds for any value of $\gamma \in [0,1]$ instead of just 0.5: One can simply change $1/2$ to $1-\gamma$ in the proof of Lemma 4.4 in Section 4.3. It is analogous to the fact that all nontrivial Fourier coefficients vanish in Håstad's result on Max-3SAT and Max-3LIN [34].

□

Given such $H = (V_H, E_H)$, let $n := |V_H|$. Our correlation clustering instance $G = (V_G, E_G)$ is the line graph of H; $V_G = E_H$ and $e, f \in E_H$ have a plus edge in G if they share a vertex in H. This means that every $e \in V_G$ has $(3 \pm o(1))d$ plus edges incident on it; we used the fact that $d = \omega(n)$ and e has at most $O(n)$ other hyperedges that intersect with e with at least two points (which causes double counting).

YES case. Consider $U \subseteq V_H$ guaranteed in Theorem 24. Our (randomized) clustering is the following: randomly permute vertices to obtain $U = \{v_1, \ldots, v_{n/2}\}$, and let $E_i := \{e \in E_H : v_i \in e \text{ and } e \cap \{v_1, \ldots, v_{i-1}\} = \emptyset\}$. Since U intersects every $e \in E_H$, $(E_1, \ldots, E_{n/2})$ forms a partition of E_H.

We analyze the expected cost of this clustering. For each $e \in E_H$, let $save(e)$ be (the number of plus neighbors in the same cluster) minus (the number of minus neighbors in the same cluster). Intuitively, it is the amount of saved cost between e and its neighbors, compared to the situation where e is a singleton cluster. Then, the cost of our clustering is the total number of plus edges of G, namely $|E_H| \cdot \frac{3(1\pm o(1))d}{2} = nd^2 \cdot \frac{(1\pm o(1))}{2}$, minus $\sum_{e\in E_H} save(e)/2$.

Fix $v \in U$ and let $E_v := \{e \in E_H : v \in e\}$, $E'_v := \{e \in E_H : e \cap U = \{v\}\}$, $E''_v := E_v \setminus E'_v$. Then $|E_v| = (1 \pm o(1))d$ and $|E'_v| \geq (1/2 - \varepsilon)d$. We would like to compute $\mathbb{E}[|E_i|^2]$ over random permutations where i is defined such that $v_i = v$. It is clear that $E'_v \subseteq E_i$. For each $e \in E''_v$, the probability that $e \in E_i$ is at least $1/3$ (when v comes

before the other two vertices of e in the random permutation). And two hyperedges $e, f \in E''_v$, the probability that both are in E_i is at least $1/5$ (when v comes first among $|e \cup f| \le 5$ vertices). Therefore,

$$\mathbb{E}[|E_i|^2] \ge |E'_i|^2 + 2|E'_i||E''_i|/3 + |E''_i|^2/5 \ge d^2(1/4 + 1/6 + 1/20 - O(\varepsilon)) = d^2(7/15 - O(\varepsilon)).$$

Therefore, the total saving is at least $nd^2(7/30 - O(\varepsilon))$ and the final cost is at most $nd^2(1/2 - 7/60 + O(\varepsilon)) = nd^2(23/60 + O(\varepsilon))$.

NO case. Our analysis will be similar to that of the gap instance, slightly more complicated by the fact that we are working with a non-complete hypergraph. Consider the optimal correlation clustering and consider one cluster C. For $e \in C$, it has at most $(3 \pm o(1))d$ plus edges in G, so $|C| \le (6 + o(1))d$; otherwise, it is better to make e a singleton cluster. We prove that if C is large, then we can partition C into smaller clusters where each cluster consists of hyperedges sharing the same vertex in H. For $v \in E_H$, let $E_v \subseteq E_H$ be the set of hyperedges containing v.

Claim 25. *There is a partition of C into $C_1, \dots, C_r$ such that (1) each C_i is a subset of E_v for some $v \in V_H$, and (2) replacing C by $C_1, \dots, C_r$ in the correlation clustering solution increases the objective function by at most $O(n|C|)$.*

Proof. Without loss of generality, assume $V_H = \{v_1, \dots, v_n\}$ and define $n_i := |C \cap E_{v_i}|$ such that $n_1 \ge \cdots \ge n_n$. Note that $\sum_i n_i = 3|C|$.

If $e = (v_i, v_j, v_k)$ with $i, j, k > 20$, then $n_i + n_j + n_k < 3 \cdot (3|C|/20) < |C|/2$, which implies that e has more minus neighbors than plus neighbors in C, leading to contradiction. So, every hyperedge is incident on v_i for some $i \le 20$.

Since two vertices of H have at most n hyperedges containing both of them, let us make at most $n \cdot \binom{10}{2}$ hyperedges in C that contain at least two of $v_1, \dots, v_{20}$ as singleton clusters; the objective function increases by at most $n \cdot \binom{10}{2} \cdot |C|$. Then partition the remaining C into $E_1, \dots, E_{20}$ where $E_i := C \cap E_{v_i}$. Each $e \in E_i$ has at most $2 \cdot 20 \cdot n$ plus edges in $\cup_{j \ne i} E_j$ (20 choices for v_j, 2 choices for a vertex in $e \ni \{v_i\}$, and n choices for hyperedges containing both vertices), so the objective function increases by at most $O(n|C|)$. So, we partitioned C into $C_1, \dots, C_r$ where all the hyperedges in C_i share a common endpoint. In total, we increased the objective function by at most $O(n|C|)$. □

Applying the above procedure for every cluster C increases the objective function by at most $O(n \cdot |E_H|) = O(n^2 d)$. Then, we have a clustering where all the edges in a cluster C share a common endpoint. C forms a clique in H. For $v \in V_H$, let C_v be the cluster in the solution whose common endpoint is v. (If there are many of them, merging them will strictly improve the objective function value.) Without loss of generality, there are t such clusters $C_{v_1}, \dots, C_{v_t}$ and let $c_i := |C_{v_i}|$ such that $c_1 \ge \cdots \ge c_t$.

Claim 26. *$\sum_{i=1}^t c_i^2 \le d^2 n(0.2 + O(\sqrt{\varepsilon}))$, where ε is the parameter from Theorem 24.*

Proof. Here, we use the NO case guarantee from Theorem 24: for any $\gamma \in [0, 1]$ and choice of γn vertices, it covers at most $1 - (1 - \gamma)^3 + \varepsilon = 3\gamma - 3\gamma^2 + \gamma^3 + \varepsilon$ fraction of the edges, which is equivalent to: for every $i \in [n]$,

$$\sum_{j=1}^{i} c_i \le (3(i/n) - 3(i/n)^2 + (i/n)^2 + \varepsilon)|E_H|. \tag{16}$$

Let $\delta = o(1)$ be such that every vertex of H has degree at most $(1 + \delta)d$, which means that $(1 + \delta)d \ge c_1 \ge \cdots \ge c_t$. And let $f_{i/n} := c_i/((1 + \delta)d)$. Then (16) becomes

$$\frac{1}{n}\sum_{j=1}^{i} f_{j/n} \le (3(i/n) - 3(i/n)^2 + (i/n)^2 + \varepsilon)\frac{|E_H|}{(1+\delta)dn} \le (3(i/n) - 3(i/n)^2 + (i/n)^2 + \varepsilon)/3. \tag{17}$$

(Note that $|E_H| \le (1 + \delta)dn/3$.) if we interpret $\frac{1}{n}\sum_{j=1}^{i} f_{j/n}$ as $\int_0^1 f(x)dx$ where $f(x) = c_{\lceil xn \rceil}$, we have that

$$\sum_{i=1}^{t} |c_i|^2 \le (1 + \delta)^2 d^2 n \max_f \int_0^1 f(x)^2 dx,$$

where the maximum is taken over functions $f : [0, 1] \to [0, 1]$ with the constraints that

(1) For all $y \in [0, 1]$,
$$\int_{x=0}^{y} f(x)dx \le y - y^2 + y^3/3 + \varepsilon/3. \tag{18}$$
(Compared to (17), we add more constraints for every $y \in [0, 1]$, but it is valid to do so since the step function $f(\cdot)$ defined above satisfies all these constraints; if (18) is violated for some value $y \in (i/n, (i + 1)/n)$ for some integer i, (17) is violated at $(i + 1)/n$ because $f(y)$ stays the same in the interval while the upper bound increases strictly less than linearly.)
(2) f decreasing with $f(0) \le 1$.

Then one see that the optimal f satisfies either $f(y) = 1$ or $\int_{x=0}^{y} f(x) = y - y^2 + y^3 + \varepsilon/3$ for every $y \in [0, 1)$. If it is not satisfied at some y, we can increase $f(y)$ while decreasing $f(z)$ for some $z > y$, which will still satisfy the constraints and increase $\int_0^1 f(x)^2 dx$. Therefore, we can conclude that $f(y) = 1$ for $y \le \tau$ and

$$\int_{x=0}^{y} f(x)dx = y - y^2 + y^3/3 + \varepsilon/3$$
$$\Rightarrow f(y) = (y - y^2 + y^3/3 + \varepsilon/3)' = 1 - 2y + y^2$$

for $y > \tau$, where $\tau = \Theta(\sqrt{\varepsilon})$ is the solution of $\tau = \tau - \tau^2 + \tau^3 + \varepsilon/3$. Then, we can bound

$$\int_{x=0}^{1} f(x)^2 dx \le O(\sqrt{\varepsilon}) + \int_{x=0}^{1} (1 - 2x + x^2)^2 dx \le 0.2 + O(\sqrt{\varepsilon}),$$

which implies that $\sum_i c_i^2 \le d^2 n(0.2 + O(\sqrt{\varepsilon}))$. □

Using this, we can prove a lower bound on the cost of our near-optimal clustering. Note that every cluster is a clique of +edges. Thus, the only edges violated are +edges. Moreover, there are at most $\sum_{i \in [t]} c_i^2/2 \le d^2 n(0.1 + O(\sqrt{\varepsilon}))$ correctly clustered +edges. The cost of our near-optimal clustering is the total number of +edges of G minus the number of correctly clustered +edges, namely at least $nd^2(1/2 - 0.1 - O(\sqrt{\varepsilon})) = nd^2(0.4 - O(\sqrt{\varepsilon}))$. Since the cost of

the optimal clustering is at most $O(n^2d)$ lower than ours, it is still $nd^2(0.4 - O(\sqrt{\varepsilon}))$ using $d = \omega(n)$.

Since the value in the YES case is at most $(23/60 + O(\varepsilon))nd^2$, so the gap is almost $\frac{24}{23} \geq 1.043$.

REFERENCES

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

[2] Sara Ahmadian and Maryam Negahbani. 2023. Improved approximation for fair correlation clustering. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 9499–9516.

[3] Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. 2015. Correlation Clustering in Data Streams. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. 2237–2246.

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

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

[6] Sepehr Assadi and Chen Wang. 2022. 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) (LIPIcs, Vol. 215)*. 10:1–10:20.

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

[8] Nikhil Bansal and Maxim Sviridenko. 2006. The Santa Claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*. 31–40.

[9] Boaz Barak, Prasad Raghavendra, and David Steurer. 2011. Rounding semidefinite programming hierarchies via global correlation. In *Proceedings of 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 472–481.

[10] Soheil Behnezhad, Moses Charikar, Weiyun Ma, and Li-Yang Tan. 2022. Almost 3-Approximate Correlation Clustering in Constant Rounds. In *Proceedings of 63rd Annual IEEE Symposium on Foundations of Computer Science, (FOCS)*. 720–731.

[11] Soheil Behnezhad, Moses Charikar, Weiyun Ma, and Li-Yang Tan. 2023. Single-Pass Streaming Algorithms for Correlation Clustering. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 819–849.

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

[13] Mélanie Cambus, Davin Choo, Havu Miikonen, and Jara Uitto. 2021. Massively Parallel Correlation Clustering in Bounded Arboricity Graphs. In *35th International Symposium on Distributed Computing (DISC) (LIPIcs, Vol. 209)*. 15:1–15:18.

[14] Mélanie Cambus, Fabian Kuhn, Etna Lindy, Shreyas Pai, and Jara Uitto. 2024. A $(3+\varepsilon)$-Approximate Correlation Clustering Algorithm in Dynamic Streams. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

[15] Nairen Cao, Shang-En Huang, and Hsin-Hao Su. 2024. Breaking 3-Factor Approximation for Correlation Clustering in Polylogarithmic Rounds. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

[16] Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. 2008. A graph-theoretic approach to webpage segmentation. In *Proceedings of the 17th International conference on World Wide Web (WWW)*. 377–386.

[17] Sayak Chakrabarty and Konstantin Makarychev. 2023. Single-Pass Pivot Algorithm for Correlation Clustering. Keep it simple!. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[18] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. 2005. Clustering with qualitative information. *J. Comput. System Sci.* 71, 3 (2005), 360–383.

[19] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. 2009. Integrality gaps for Sherali-Adams relaxations. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*. 283–292.

[20] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. 2006. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity* 15, 2 (2006), 94–114.

[21] Shuchi Chawla, Konstantin Makarychev, Tselil Schramm, and Grigory Yaroslavtsev. 2015. 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)*. 219–228.

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

[23] Flavio Chierichetti, Nilesh Dalvi, and Ravi Kumar. 2014. Correlation clustering in MapReduce. In *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 641–650.

[24] Vincent Cohen-Addad, Silvio Lattanzi, Andreas Maggiori, and Nikos Parotsidis. 2022. Online and Consistent Correlation Clustering. In *Proceedings of International Conference on Machine Learning (ICML)*. 4157–4179.

[25] Vincent Cohen-Addad, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub Tarnawski. 2021. Correlation Clustering in Constant Many Parallel Rounds. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 2069–2078.

[26] Vincent Cohen-Addad and Euiwoong Lee. 2022. Johnson coverage hypothesis: Inapproximability of k-means and k-median in L_p-metrics. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1493–1530.

[27] Vincent Cohen-Addad, Euiwoong Lee, Shi Li, and Alantha Newman. 2023. 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)*.

[28] Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. 2022. Correlation Clustering with Sherali-Adams. In *Proceedings of 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 651–661.

[29] Sami Davies, Benjamin Moseley, and Heather Newman. 2023. Fast Combinatorial Algorithms for Min Max Correlation Clustering. *arXiv preprint arXiv:2301.13079* (2023).

[30] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. 2006. Correlation clustering in general weighted graphs. *Theoretical Computer Science* 361, 2-3 (2006), 172–187.

[31] Lisa Fleischer, Michel X. Goemans, Vahab S. Mirrokni, and Maxim Sviridenko. 2006. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 611–620.

[32] Ioannis Giotis and Venkatesan Guruswami. 2006. Correlation Clustering with a Fixed Number of Clusters. *Theory of Computing* 2 (2006), 249–266.

[33] Venkatesan Guruswami and Ali Kemal Sinop. 2011. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 482–491.

[34] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (2001), 798–859.

[35] Holger Heidrich, Jannik Irmai, and Bjoern Andres. 2023. A 4-approximation algorithm for min max correlation clustering. *arXiv preprint arXiv:2310.09196* (2023).

[36] Dmitri V. Kalashnikov, Zhaoqi Chen, Sharad Mehrotra, and Rabia Nuray-Turan. 2008. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering* 20, 11 (2008), 1550–1565.

[37] Marek Karpinski and Warren Schudy. 2009. Linear time approximation schemes for the Gale-Berlekamp game and related minimization problems. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*. 313–322.

[38] Silvio Lattanzi, Benjamin Moseley, Sergei Vassilvitskii, Yuyan Wang, and Rudy Zhou. 2021. Robust Online Correlation Clustering. In *Advances in Neural Information Processing Systems (Neurips)*. 4688–4698.

[39] Claire Mathieu, Ocan Sankur, and Warren Schudy. 2010. Online Correlation Clustering. In *Proceedings of 27th International Symposium on Theoretical Aspects of Computer Science (STACS)*. 573–584.

[40] Xinghao Pan, Dimitris S. Papailiopoulos, Samet Oymak, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. 2015. Parallel Correlation Clustering on Big Graphs. In *Advances in Neural Information Processing Systems (Neurips)*. 82–90.

[41] Prasad Raghavendra and Ning Tan. 2012. Approximating CSPs with global cardinality constraints using SDP hierarchies. In *Proceedings of the 23d Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 373–387.

[42] Chaitanya Swamy. 2004. Correlation Clustering: Maximizing agreements via semidefinite programming.. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 526–527.

[43] Nate Veldt. 2022. Correlation Clustering via Strong Triadic Closure Labeling: Fast Approximation Algorithms and Practical Lower Bounds. In *International Conference on Machine Learning (ICML)*. 22060–22083.

[44] Nate Veldt, David F. Gleich, and Anthony Wirth. 2018. A correlation clustering framework for community detection. In *Proceedings of the 2018 ACM World Wide Web Conference (WWW)*. 439–448.

Received 13-NOV-2023; accepted 2024-02-11