arXiv:2404.17509v3 [cs.DS] 31 Oct 2025

Understanding the Cluster LP for Correlation Clustering*

Nairen Cao†
New York University
nc1827@nyu.edu

Vincent Cohen-Addad
Google Research
cohenaddad@google.com

Euiwoong Lee‡
University of Michigan
euiwoong@umich.edu

Shi Li §
Nanjing University
shili@nju.edu.cn

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

Lukas Vogl ¶
EPFL
lukas.vogl@epfl.ch

November 3, 2025

Abstract

In the classic Correlation Clustering problem introduced by Bansal, Blum, and Chawla [BBC04], 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 [CMSY15] gave a 2.06-approximation by providing a near-optimal rounding of the standard LP, and Cohen-Addad, Lee, Li, and Newman [CLN22, CLLN23] 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 [GS11, BRS11, RT12].

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 for Correlation Clustering. It is exponential-sized, but we show that it can be $(1+\varepsilon)$-approximately solved in polynomial time for any $\varepsilon > 0$, 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 new rounding algorithms, and providing two analyses, one analytically proving a 1.56-approximation and the other solving a factor-revealing SDP to show a 1.485-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 1.485-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.

*The conference version of this paper [CCL+24] claimed an approximation ratio of 1.437 whose proof currently has a gap. This version fixes the gap with a slightly worse approximation ratio of 1.485.

†Supported by NSF grant CCF-2008422.

‡Supported in part by NSF grant CCF-2236669 and Google.

§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.

¶Supported by the Swiss National Science Foundation project 200021-184656 "Randomness in Problem Instances and Randomized Algorithms".

Contents

1 Introduction **3**
- 1.1 Our Results . . . 3
- 1.2 Further Related Work . . . 5

2 Overview and Algorithm **5**
- 2.1 Formulating and Solving Cluster LP . . . 5
- 2.2 Rounding Cluster LP . . . 7
- 2.3 Gaps and Hardness . . . 11

3 Preclustering **12**
- 3.1 Constructing Atoms $\mathcal{K}$ . . . 13
- 3.2 Defining Admissible Edges E_{adm} . . . 14

4 Solving Cluster LP Approximately **16**
- 4.1 Bounded Sub-Cluster LP Relaxation for Preclustered Instances . . . 16
- 4.2 Sampling One Cluster Using LP Solution to the Bounded Sub-Cluster LP . . . 17
- 4.3 Using Independently Sampled Clusters to Construct Cluster LP Solution . . . 19

5 Triangle Analysis used in 1.56-Approximation of Cluster LP **21**
- 5.1 $+++$ Triangles . . . 23
- 5.2 $---$ Triangles . . . 25
- 5.3 $+--$ Triangles . . . 25
- 5.4 $++-$ Triangles . . . 26

6 1.485-Approximation: Computer-assisted proof **31**

7 1.33-gap for Cluster LP **37**

8 1.04-NP Hardness **38**

A Standard SDP Relaxation **44**

B Detail of Lemma 52 **45**

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 [BBC04], 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. 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 [BGU13], duplicate detection [ARS09], community mining [CSX12], disambiguation tasks [KCMNT08], automated labelling [AHK+09, CKP08] and many more.

This problem is APX-Hard [CGW05], and various $O(1)$-approximation algorithms [BBC04, CGW05] 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 [ACN08]. The LP-based rounding was improved by Chawla, Makarychev, Schramm and Yaroslavtsev to a 2.06-approximation [CMSY15], nearly matching the LP integrality gap of 2 presented in [CGW05].

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 [CLN22] 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 [CLLN23] to $(1.73 + \varepsilon)$ in $n^{\mathrm{poly}(1/\varepsilon)}$-time, which combines *pivot-based rounding* and *set-based rounding*.

One undesirable feature of [CLLN23] 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 $1.485 + \varepsilon$. 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 [BS06, FGMS06]. 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.

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

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

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

$$z_S \geq 0 \qquad \forall S \subseteq V, S \neq \emptyset \tag{3}$$

The objective of the LP is to minimize $\text{obj}(x)$, 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 [CLLN23] 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 $n^{\text{poly}(1/\varepsilon)}$, we can output a solution $\left((z_S)_{S\subseteq V}, (x_{uv})_{uv\in\binom{V}{2}}\right)$ to the cluster LP with $\text{obj}(x) \leq (1+\varepsilon)\text{opt}$, described using a list of non-zero coordinates.* [1]

The cluster LP is one of the most powerful LPs that has been considered for the problem. A large portion of the algorithms and analysis in [CLN22] and [CLLN23] is devoted to handle the additive errors incurred by the correlated rounding procedure, which is inherited from the Raghavendra-Tan rounding technique [RT12]. 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 [Swa04, CGW05]. For the minimization version, the standard SDP has integrality gap at least 1.5 (see Appendix A), 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 $(1.56+\varepsilon)$-approximation algorithm for Correlation Clustering that runs in time $O(n^{\text{poly}(1/\varepsilon)})$.*

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 [CLLN23]. 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 a similar 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 $(1.485+\varepsilon)$-approximation algorithm for Correlation Clustering that runs in time $O(n^{\text{poly}(1/\varepsilon)})$.*

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.

[1] We remark that $\text{obj}(x)$ 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 $\mathcal{C}^*$ 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 $w^T Y w = \sum_{u,v\in V} y_{uv} w_u w_v = \sum_{u,v} \sum_{S\supseteq\{u,v\}} z_S w_u w_v = \sum_{u,v} \sum_{S\subseteq V} z_S \cdot (w_u \cdot 1_{u\in S}) \cdot (w_v \cdot 1_{v\in S}) = \sum_{S\subseteq V} z_S \left(\sum_{u\in S} w_u\right)\left(\sum_{v\in S} w_v\right) \geq 0$.

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. Apart from the APX-hardness [CGW05], it is the first hardness with an explicit hardness ratio.

Theorem 5. *Unless* **P** = **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 [DEFI06], implying that there is an $O(\log n)$-approximation but no constant factor approximation is possible under the Unique Games Conjecture [CKK+06].

In the unweighted case, a PTAS exists when the number of clusters is a fixed constant [GG06, KS09]. Much study has been devoted to the minimization version of Correlation Clustering in various computational models, for example in the online setting [MSS10, LMV+21, CLMP22], as well as in other practical settings such as distributed, parallel or streaming [CDK14, ACG+15, CLM+21, PPO+15, CCMU21, Vel22, VGW18, AW22, BCMT22, BCMT23, CHS24, CM23, CKL+24]. Other recent work involves settings with fair or local guarantees [AN23, DMN23, HIA23].

2 Overview and Algorithm

In this section, we provide an overview of our algorithms and tools for the analysis. Section 2.1 introduces some techniques used for formulating and solving the cluster LP. Section 2.2 presents our full rounding algorithm and provides high-level ideas for the analysis. Section 2.3 discusses ideas behind our gap and hardness results. These techniques were introduced over the course of three conference papers [CLN22, CLLN23, CCL+24], so we try to highlight their connections and developments over time. For instance, since the cluster LP was formalized in the last conference paper [CCL+24], several of the following techniques were initially presented in relation to a direct rounding of our *initial* (Sherali-Adams-based) relaxation.

2.1 Formulating and Solving Cluster LP

Previous Algorithms. To understand in more detail our contribution and the new techniques introduced, we need to provide a brief summary of previous approaches. We first recall the classic LP relaxation (whose integrality gap is known to be in $[2, 2.06]$). There is a variable x_{uv} for each pair of vertices whose intended value is 1 if u, v are not in the same cluster and 0 otherwise. The goal is thus to minimize $\sum_{uv \in E^+} x_{uv} + \sum_{uv \in E^-} (1 - x_{uv})$, under the classic triangle inequality constraint: $\forall u, v, w$, $x_{uv} \leq x_{uw} + x_{wv}$.

To obtain the 2.5-approximation algorithm, Ailon, Charikar and Newman [ACN08] designed a pivot-based rounding method that proceeds as follows: (1) Pick a random vertex p, called the *pivot*, and create the cluster containing p; and (2) Recurse on the rest of the instance (the vertices not in the cluster). This hence builds the clustering in a sequential manner. Once we commit to this scheme, the main design question that remains is how to construct the cluster containing the pivot. The solution of [ACN08] was to go over all the other vertices and for each other vertex u place it in the cluster of the pivot p with probability $(1 - x_{pu})$, independently of the other random decisions made for the other vertices.

This was further improved by Chawla, Makarychev, Schramm and Yaroslavtsev [CMSY15] who kept the same scheme but provided a better rounding approach: For +edges, they replaced the probability $(1 - x_{pu})$ with a new rounding function f^+ to apply to the quantity $(1 - x_{pu})$ yielding the probability of incorporating u in p's cluster. Similar to the approach of [ACN08], the (streamlined) analysis of this rounding scheme was triangle-based: The analysis is a charging scheme that charges the cost paid by each pair u, v to the triangle p, u, v where p is the pivot which decided edge u, v. Then the crux of the analysis is to show that for any triangle p, u, v, the charge is bounded compared to the LP cost.

Sherali-Adams and Correlated Rounding. One potential place for an improvement in the previous algorithms is the *independent rounding* given a pivot p; in both [ACN08, CMSY15], once we choose p, other vertices are put into the cluster of p independently of one another. It might be more advantageous to make these decisions in a correlated way. However, the standard LP, which only has variables $\{x_{uv}\}_{uv\in\binom{V}{2}}$ for the pairs, is not suited to provide information about such correlations.

Motivated by this fact, our first new idea, which appeared in the first conference version [CLN22], is the usage of *correlated rounding* based on the *Sherali-Adams hierarchy*. Namely, given a pivot p, the set of +neighbors of p that join the cluster of p is chosen in a correlated manner, using the techniques of Raghavendra and Tan [RT12]. Concretely, the Sherali-Adams hierarchy provides variables of the form y_S for any constant-sized set of vertices S that indicates the probability that all the vertices in S are in the same cluster.[3] Given a pivot p, the correlated rounding then allows to sample in such a way that the probability that u, v join the cluster of p is $y_{puv} \pm \varepsilon$, where ε is an arbitrarily small constant. Applying this rounding to a set of vertices S, it ensures that we can sample a subset C of S both being of arbitrary size (importantly not necessarily constant) such that pairs u, v ends up in C with probability y_{puv} on average, up to losing an additive $\varepsilon|S|^2$ error in the cost.

The main issue we need to face with the above plan of attack is that the additive error of $\varepsilon|S|^2$, which we pay when we create a cluster C, may end up being much larger than the cost of the optimum solution. In fact, an optimum solution may have cost zero. The correlated rounding was initially developed for *maximization problems* like Max-CSPs [dlVM07, BRS11, GS11, RT12, YZ14] where this additive error can be easily ignored by only additively sacrificing ε in the approximation ratio, but this is not the case for Correlation Clustering. [4]

[CLN22] applied this rounding to +neighbors of p to get a $(2+\varepsilon)$-approximation (with respect to a $O(1/\varepsilon^2)$-level of Sherali-Adams) and further combined it with the *global triangle charging scheme* to achieve a $(1.994+\varepsilon)$-approximation. This charging scheme was later further developed and used to analyze *global triangle distributions*, which is discussed at the end of Section 2.2.

Dealing with additive errors from the correlated rounding is one of the most technical parts [CLN22] and perhaps a limitation to getting an improved approximation ratio (e.g., special attention must be paid to edges with values close to 0 and 1 in both the algorithm and in the analysis). That brings us to our second main idea of *preclustering*, so that we can (1) precluster the instance so that the cost of the optimal solution becomes relatively large compared to the size of the "unsure part", and (2) use correlated rounding only for the unsure part of the instance so that the error from the correlated rounding is negligible.

Preclustering. Preclustering is a subroutine that aims at identifying "clear clusters". Formally, our preclustering outputs a pair $(\mathcal{K}, E_{\mathrm{adm}})$, where $\mathcal{K}$ is a partition of V into so-called *atoms* and $E_{\mathrm{adm}} \subseteq \binom{V}{2}$ is a set of *admissible* edges. The guarantee is that for any $\varepsilon > 0$, there is a $(1+\varepsilon)$-approximate solution that is *consistent* with the preclustering; an atom is not broken, and if u and v are not in the same atom and $uv \notin E_{\mathrm{adm}}$, then u and v are in different clusters. We also ensure that $|E_{\mathrm{adm}}| \le O_\varepsilon(\mathrm{opt})$, so the number of edges on which we apply the correlated rounding (and pay the additive error) is comparable to opt.

The algorithm first constructs atoms. Intuitively, consider a cluster of the optimum (integral) solution that has s vertices but contributes significantly smaller than s^2 to the objective function, say εs^2. (Otherwise, an $\varepsilon^2 s^2$-additive error will already be a ε-multiplicative error.) For this to happen, it must be that the number of $-$edges internal to S is small compared to $\binom{s}{2}$ and that the number +edges with exactly one endpoint in S is also much smaller than $\binom{s}{2}$. This cluster is thus almost a clique with tiny +edge-expansion, which is easy to (approximately) identify. This idea was used by Cohen-Addad, Lattanzi, Mitrovic, Norouzi-Fard, Parotsidis, and Tarnawski for a massively-parallel algorithm with a large constant factor approximation [CLM+21]. Our first version of preclustering [CLLN23] used [CLM+21] as a black-box to construct atoms, but in this paper, we present the simplified version of [CCL+24], which starts from an arbitrary $O(1)$-approximate clustering

[3] In which case, $y_{uv} = 1 - x_{uv}$.

[4] The only exception is [GS11] that studied minimization versions including Min Bisection and Min Uncut, but their technique heavily relies on the property of cuts not easily adaptable to Correlation Clustering.

and removes a vertex v from a cluster C when the +neighborhood of v is somewhat different from C (and makes v a singleton cluster).

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 (up to a factor of $O(1/\varepsilon)$). Then an edge uv is admissible if u and v have many (at least an $\Omega(\varepsilon)$ fraction) common neighbors in $E^+ \cap E^1$. Here, the somewhat relaxed notions of degree-similarity and large-common-neighborhood (compared to the ones for the atoms, where we wanted +neighborhood of u and v to be almost identical for u and v to be in the same atom) ensure that there is a near-optimal solution respecting the admissible edges, while also guaranting that the number of the admissible edges (crossing atoms) is not too large compared to opt.

Solving Cluster LP on Preclustered Instance by Sampling. After constructing a preclustered instance $(\mathcal{K}, E_{\text{adm}})$, we formulate an LP relaxation aimed at finding a $(1+\varepsilon)$-approximate *good clustering* for $(\mathcal{K}, E_{\text{adm}})$, that we call the *bounded sub-cluster LP*. First introduced in [CLLN23], it is a strengthened version of the Sherali-Adams hierarchy used in [CLN22]. 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 correlated rounding. 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 correlated rounding procedure can be bounded in terms of $|E_{\text{adm}}|$, and the error from sampling can be bounded using concentration bounds.

2.2 Rounding Cluster LP

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 $\text{obj}(x) \le (1+\varepsilon)\text{opt}$. 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 $\tau = 0.4$. We select the better result as the final clustering. The two procedures are defined in Algorithms 1 and 3 respectively. We use $N^+(u)$ and $N^-(u)$ to denote the sets of $+$ and $-$neighbors of a vertex $u \in V$ respectively. They are stated as randomized algorithms, but they can be derandomized by the standard method of conditional expectation.

Algorithm 1 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** $\mathcal{C} \leftarrow \mathcal{C} \cup \{V' \cap S\}$, $V' \leftarrow V' \setminus S$
return $\mathcal{C}$

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 $\mathcal{C}$ 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 $\frac{\sum_{|S\cap\{u,v\}|=1} z_S}{\sum_{S\cap\{u,v\}\neq\emptyset} z_S} = \frac{2x_{uv}}{1+x_{uv}}$. □

Algorithm 2 General pivot-based rounding with hybrid independent/dependent updates. Parameters: $I_c^+ \subseteq [0,1]$ and functions $f^+, f^- : [0,1] \to [0,1]$. For a pivot u, +edges (u,v) with $x_{uv} \notin I_c^+$ are rounded independently (include v with probability $1 - f^+(x_{uv})$); +edges with $x_{uv} \in I_c^+$ are rounded dependently via a sample $S \ni u$ drawn from $\{z_S\}$; $-$edges are always rounded independently (include v with probability $1 - f^-(x_{uv})$).

$\mathcal{C} \leftarrow \emptyset$, $V' \leftarrow V$
while $V' \neq \emptyset$ **do**
  pick a pivot $u \in V'$ uniformly at random; $C \leftarrow \{u\}$
  sample a set $S \ni u$ according to probabilities $\{z_S\}$ ▷ $\sum_{S \ni u} z_S = 1$
  for each $v \in V' \cap N^+(u)$ **do**
    if $x_{uv} \notin I_c^+$ **then** ▷ independent rounding on out-of-range + edges
      independently add v to C with probability $1 - f^+(x_{uv})$
    else if $v \in S$ **then** ▷ dependent rounding on in-range + edges
      add v to C
  for each $v \in V' \cap N^-(u)$ **do**
    independently add v to C with probability $1 - f^-(x_{uv})$ ▷ independent rounding on $-$ edges
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$; $V' \leftarrow V' \setminus C$
return $\mathcal{C}$

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 3.[5] Our high-level setup of the analysis is based on [ACN08] and [CMSY15] with some additional notions. We consider a general *budget* for every edge. We shall define two *budget functions*:

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

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

Algorithm 3 presents a generalized pivot-based rounding scheme parameterized by an interval $I_c^+ \subseteq [0,1]$ and functions f^+, f^-. Unlike the standard pivot rounding method, our scheme uses I_c^+ to designate the regime in which we apply *dependent* rounding (and applies independent rounding otherwise). By choosing different parameter triples (I_c^+, f^+, f^-), the algorithm instantiates a family of rounding procedures. We give two concrete instantiations that achieve different approximation guarantees: one more elementary and amenable to by-hand analysis, and one slightly more involved that yields a stronger approximation ratio.

We now focus on one iteration of the while loop in Algorithm 3. 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

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

and

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

[5] We remark that to recover the correlated rounding algorithm in [CLN22] and [CLLN23], 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 [CLN22] by distinguishing between the short, median and long +edges. In our algorithm, we also distinguish between *short* +edges (those with $x_{uv} \leq \frac{1}{3}$) 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.

$\text{cost}_u(v,w)$ 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 $\text{cost}_u(v,w)$ 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

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

For degenerate triangle (u,v), let

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

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

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

Then, the expected cost of the clustering output by Algorithm 3 is at most $\sum_{uv\in\binom{V}{2}} b_{uv}$.

Proof. Focus on any iteration of Algorithm 3; V' is the V' at the beginning of the iteration.

The expected cost incurred by all edges in the iteration is

$$\frac{1}{|V'|}\sum_{u\in V'}\sum_{(v,w)\in\binom{V'}{2}} \text{cost}_u(v,w) = \frac{1}{|V'|}\sum_{(u,v,w)\in\binom{V'}{3}} \text{cost}(u,v,w) + \frac{1}{|V'|}\sum_{(u,v)\in\binom{V'}{2}} \text{cost}(u,v).$$

The expected budget released at this iteration is

$$\frac{1}{|V'|}\sum_{u\in V'}\sum_{(v,w)\in\binom{V'}{2}} \Delta_u(v,w) = \frac{1}{|V'|}\sum_{(u,v,w)\in\binom{V'}{3}} \Delta(u,v,w) + \frac{1}{|V'|}\sum_{(u,v)\in\binom{V'}{2}} \Delta(u,v).$$

Therefore, if the condition of the lemma holds, then at every iteration of Algorithm 3, the expected cost incurred is at most the expected budget released. Overall, the expected cost of the final clustering is at most the expected total budget released by all edges during the whole procedure, which is $\sum_{uv\in\binom{V}{2}} b_{uv}$. This finishes the proof of the lemma. □

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 $\tau = 0.4$ (Algorithm 3) 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 budget^+ and budget^- 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 $x \in [0,1]$:

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

This gives us the following definitions:

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

Lemma 8. *If the budget functions budget^+_α and budget^-_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 $\tau = 0.4$ 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

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

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

Algorithm 3 Pivot-Based Rounding with Threshold $\tau = 0.4$ and $I_c^+ = (\tau, 1]$

$\mathcal{C} \leftarrow \emptyset, V' \leftarrow V$
while $V' \neq \emptyset$ **do**
  pick a pivot $u \in V'$ uniformly at random; $C \leftarrow \{u\}$
  sample a set $S \ni u$ according to probabilities $\{z_S\}$ ▷ $\sum_{S \ni u} z_S = 1$
  for each $v \in V' \cap N^+(u)$ **do**
    if $x_{uv} \leq \tau$ **then** ▷ independent rounding on out-of-range + edges
      independently add v to C ▷ $f^+(x) = 1$
    else if $v \in S$ **then** ▷ dependent rounding on in-range + edges
      add v to C
  for each $v \in V' \cap N^-(u)$ **do**
    independently add v to C with probability $1 - x_{uv}$) ▷ $f^-(x) = x$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$; $V' \leftarrow V' \setminus C$
return $\mathcal{C}$

As a baseline, we instantiate Algorithm 2 to obtain Algorithm 3. In this instantiation we set $I_c^+ = (0.4, 1]$: consequently, all +-edges (u, v) with $x_{uv} \leq 0.4$ are rounded independently, while +-edges with $x_{uv} \in I_c^+$ participate in the dependent rounding step. (All −-edges are rounded independently.)

We provide a per-triangle analysis leading to an approximation ratio of 1.56 in Section 5:

Lemma 9. *For Algorithm 3 and budget functions* $\text{budget}^+ \equiv \text{budget}^+_{1.56}$ *and* $\text{budget}^- \equiv \text{budget}^-_{1.56}$*, we have* $\text{cost}(T) \leq \Delta(T)$ *for every triangle* T*.*

Clearly, the lemma implies that (4) holds for $\text{budget}^+ \equiv \text{budget}^+_{1.56}$ and $\text{budget}^- \equiv \text{budget}^-_{1.56}$. By Lemma 8, Algorithm 3 gives an approximation ratio of 1.56.

To get a better approximation ratio, we provide an analysis that uses global distributions of triangles. It relies on solving a factor-revealing SDP. The resulting instantiation is given in Algorithm 4. For a + edge (u, v), we proceed as follows: (i) if $x_{uv} \leq 0.40$, include v in the pivot cluster deterministically; (ii) if $0.40 < x_{uv} \leq 0.57$, include v via the dependent step using the pivot-centered set distribution $\{z_S\}$; (iii) if $x_{uv} > 0.57$, include v independently with probability $1 - x_{uv}$. For a − edge (u, v), we split the endpoints with probability x_{uv}^2 (equivalently, add v to the pivot's cluster with probability $1 - x_{uv}^2$). For algorithm 4, the following lemma is proved in Section 6.

Lemma 10. *For Algorithm 4,* (4) *holds for budget functions* $\text{budget}^+ \equiv \text{budget}^+_{1.485}$ *and* $\text{budget}^- \equiv \text{budget}^-_{1.485}$*.*

Combined with Lemma 8, it implies Theorem 3.

Algorithm 4 Pivot-Based Rounding (with correlated mid-range for + edges)

$\mathcal{C} \leftarrow \emptyset$, $V' \leftarrow V$
while $V' \neq \emptyset$ **do**
  pick a pivot $u \in V'$ uniformly at random; $C \leftarrow \{u\}$
  sample a set $S \ni u$ according to probabilities $\{z_S\}$ ▷ $\sum_{S \ni u} z_S = 1$
  for each $v \in V' \cap N^+(u)$ **do**
    if $x_{uv} \leq 0.4$ **then** ▷ independent rounding on out-of-range + edges
      independently add v to C ▷ $f^+(x) = 1$ for $x \leq 0.4$
    else if $x_{uv} > 0.57$ **then** ▷ independent rounding on out-of-range + edges
      add v to C with probability $1 - x_{uv}$ ▷ $f^+(x) = x$ for $x > 0.57$
    else if $v \in S$ **then** ▷ dependent rounding on in-range + edges
      add v to C
  for each $v \in V' \cap N^-(u)$ **do**
    independently add v to C with probability $1 - x_{uv}^2$ ▷ $f^-(x) = x^2$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$; $V' \leftarrow V' \setminus C$
return $\mathcal{C}$

Global Triangle Distributions. The idea of global triangle distributions, which is a basis for Lemma 10, is systematically expressed by a *factor-revealing SDP*. Given a cluster LP solution z_S and vertices u, v, w, let us 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 $T = (a, b, c, d) \in [0, 1]^4$, let η_T represent the number of triangles (u, v, w) such that of $y_{uv} = a, y_{uw} = b, y_{vw} = c, y_{uvw} = d$.

Consider a hypothetical 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 they 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 semidefinite 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 $\sum \eta_T(\Delta(T) - \text{cost}(T)) \geq 0$.

2.3 Gaps and Hardness

A high-level intuition for the cluster LP is the following: (any) LP 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 $e, f \in V_G$ are connected by a +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 [CMM09].

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 [CKL22] 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. In Section 3, we describe our preclustering procedure. It will be used in our algorithm to solve the cluster LP described in Section 4. This finishes the proof of Theorem 1. We prove Lemma 10 in Section 6, which proves Theorem 3. It is a computer-assisted proof, and we present an analytic proof of Lemma 9, giving a bound of 1.56, in Section 5. We give the $(\frac{4}{3} - \varepsilon)$-integrality gap of the cluster LP (Theorem 4) in Section 7, and the improved hardness of $24/23 - \varepsilon$ (Theorem 5) in Section 8.

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 $\mathcal{C}$ of V, we define $\mathrm{obj}(\mathcal{C})$ to be the objective value of $\mathcal{C}$. For any $x \in [0,1]^{\binom{V}{2}}$, we already defined $\mathrm{obj}(x) = \sum_{uv \in E^+} x_{uv} + \sum_{uv \in E^-} (1 - x_{uv})$. Recall that we defined $\mathrm{cost}_u(v,w), \Delta_u(v,w), \mathrm{cost}(T)$ and $\Delta(T)$ for a triangle $T = (u,v,w)$ or a degenerate triangle $T = (u,v)$ in this section; they depend on the budget functions budget^+ and budget^-.

3 Preclustering

An important technique we use to solve the cluster LP is preclustering. We define a preclustered instance and a good clustering as follows.

Definition 11. *Given a Correlation Clustering instance $(V, E^+ \uplus E^-)$, a* preclustered instance *is defined by a pair $(\mathcal{K}, E_{\mathrm{adm}})$, where $\mathcal{K}$ is a partition of V (which can also be viewed as a clustering), and $E_{\mathrm{adm}} \subseteq \binom{V}{2}$ is a set of pairs such that for every $uv \in E_{\mathrm{adm}}$, 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.*

Definition 12. *Given a preclustered instance $(\mathcal{K}, E_{\mathrm{adm}})$ for some Correlation Clustering instance $(V, E^+ \uplus E^-)$, a clustering $\mathcal{C}$ of V is called* good *with respect to $(\mathcal{K}, E_{\mathrm{adm}})$ if*

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

We show that we can efficiently produce a preclustered instance for which there is a $(1+\epsilon)$-approximate good clustering. The rest of this section is dedicated to the proof of the following theorem:

Theorem 13. *For any sufficiently small $\varepsilon > 0$, there exists a $\mathrm{poly}(n, \frac{1}{\varepsilon})$-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 $(\mathcal{K}, E_{\mathrm{adm}})$ such that*

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

For convenience, we assume every u has a self-loop in E^+. Let N_u^+ be the set of +neighbors of u; so we have $u \in N_u^+$. Let $d_u^+ = |N_u^+|$ be its +degree. Let $\mathcal{C}^*$ be the optimum clustering, and $\text{opt} = \text{obj}(\mathcal{C}^*)$ be its cost. We assume $\varepsilon > 0$ is at most a small enough constant.

3.1 Constructing Atoms $\mathcal{K}$

In the first step of the algorithm for the proof of Theorem 13, we define the set $\mathcal{K}$ of atoms. We use any known $O(1)$-approximation algorithm for the Correlation Clustering problem to obtain a clustering; for example, we can use the 3-approximation combinatorial algorithm of [ACN08] to obtain a clustering $\mathcal{C}$ with $\text{obj}(\mathcal{C}) \le 3 \cdot \text{opt}$.

Our $\mathcal{K}$ is obtained from $\mathcal{C}$ by marking some vertices and creating singletons for them. We view $\mathcal{K}$ both as a clustering and as the set of atoms. The algorithm is described in Algorithm 5, where $\beta = 0.1$.

Algorithm 5 Construction of $\mathcal{K}$

1: **for** every non-singleton $C \in \mathcal{C}$ **do**
2: **for** every $u \in C$ **do**: mark u if $|N_u^+ \triangle C| > \frac{\beta}{2} \cdot |C|$
3: **if** at least $\frac{\beta|C|}{3}$ vertices in C are marked **then** mark all vertices in C
4: let $\mathcal{K}$ be the clustering obtained from $\mathcal{C}$ by removing marked vertices and creating a singleton cluster for each of them
5: **return** $\mathcal{K}$

Lemma 14. $\text{obj}(\mathcal{K}) \le O(1) \cdot \text{opt}$, *where we view* $\mathcal{K}$ *as a clustering.*

Proof. If a cluster C has fewer than $\frac{\beta|C|}{3}$ marked vertices before Step 3, then the cost incurred by separating the marked vertices in C is at most $\frac{2}{\beta} = O(1)$ times the cost of edges incident to these vertices in $\mathcal{C}$. On the other hand, if C has at least $\frac{\beta|C|}{3}$ marked vertices before Step 3, the cost of edges incident to C in $\mathcal{C}$ is at least $\frac{\beta}{2} \cdot |C| \cdot \frac{\beta|C|}{3} \cdot \frac{1}{2} = \Omega(|C|^2)$. The cost incurred by breaking C into singletons is at most $\binom{|C|}{2}$. As every edge is charged at most twice, the cost incurred by creating singletons for all marked vertices is at most $O(1) \cdot \text{obj}(\mathcal{C})$. So $\text{obj}(\mathcal{K}) \le O(1) \cdot \text{opt}$ as $\text{obj}(\mathcal{C}) \le 3 \cdot \text{opt}$. □

Lemma 15. *For every non-singleton atom* $K \in \mathcal{K}$*, and every* $u \in K$*, we have* $|N_u^+ \triangle K| < \beta|K|$.

Proof. Assume $K \subseteq C$ for some $C \in \mathcal{C}$. So the vertices in $C \setminus K$ are marked, the vertices in K are unmarked, and $|C \setminus K| < \frac{\beta|C|}{3}$. For every $u \in K$, we have $|N_u^+ \triangle C| \le \frac{\beta}{2} \cdot |C|$ because it is unmarked. Then $|N_u^+ \triangle K| \le \frac{\beta}{2} \cdot |C| + \frac{\beta|C|}{3} = \frac{5\beta}{6} \cdot |C| \le \frac{5\beta}{6} \cdot \frac{|K|}{1-\beta/3} \le \beta|K|$ for our choice of β. □

Lemma 16. *Consider the optimum clustering* $\mathcal{C}^*$*. Any atom* $K \in \mathcal{K}$ *is completely inside a cluster in* $\mathcal{C}^*$.

Proof. The lemma holds trivially if K is a singleton. Assume towards the contradiction that K is a non-singleton and not inside a cluster in $\mathcal{C}^*$. By Lemma 15, we have that $|N_u^+ \triangle K| < \beta|K|$ for every $u \in K$. In particular, this implies that $|K \setminus N_u^+| < \beta|K|$ and $|N_u^+ \setminus K| < \beta|K|$. We consider two cases as in [CLLN23].

First, suppose no cluster in $\mathcal{C}^*$ contains at least $\frac{2|K|}{3}$ vertices in K. We consider the operation of removing all vertices in K from their respective clusters in $\mathcal{C}^*$, and creating a single cluster K. The saving in cost is at least $\frac{1}{2} \cdot |K| \cdot (\frac{|K|}{3} - \beta|K|) - \frac{1}{2} \cdot |K| \cdot \beta|K| = (\frac{1}{6} - \beta)|K|^2 > 0$, as $\beta = 0.1$. This contradicts that $\mathcal{C}^*$ is optimum.

Then, consider the other case where there is some $C \in \mathcal{C}$ with $|C \cap K| \ge \frac{2|K|}{3}$. Let u be any vertex in $K \setminus C$; it must exist as K is not completely inside C. Then, we consider the operation of moving u from its cluster to C. The saving in cost is at least $(\frac{2}{3} - 2\beta)|K| - \frac{|K|}{3} = (\frac{1}{3} - 2\beta)|K| > 0$ as $\beta = 0.1$, contradicting that $\mathcal{C}^*$ is optimum. □

With Lemma 16, we now restrict ourselves to clusterings that do not break atoms. As a result, we can then assume all the edges between two vertices in a same atom $K \in \mathcal{K}$ are +edges. The assumption only decreases opt and can only make the two properties of Theorem 13 harder to satisfy. We use K_u for every $u \in V$ to denote the atom that contains u. We let $k_u = |K_u|$.

It is convenient for us to distribute the +edges incident to an atom $K \in \mathcal{K}$ equally to the vertices in K. For every $u, v \in V$, we define

$$w_{uv} := \frac{1}{k_u k_v} \sum_{u' \in K_u, v' \in K_v} 1_{u'v' \in E^+}.$$

to be the probability that an edge between a random vertex in K_u and a random vertex in K_v is a +edge. So $w_{uv} \in [0, 1]$ for every pair uv; in particular, if $v \in K_u$, we have $w_{uv} = 1$. We shall call w_{uv} the weight of the edge uv. For any set $u \in V, V' \subseteq V$, we define $w(u, V') := \sum_{v \in V'} w_{uv}$ be the total weight of edges between u and V'. Let $w_u := w(u, V)$ be the total weight of all edges incident to u. In this new instance, w_{uv} fraction of edge uv has a + sign, and the remaining $1 - w_{uv}$ fraction has a $-$ sign; w_u is the total fractional number of +edges incident to u, which can be treated as the +degree of u. This new instance is equivalent to the original one. Till the end of this section we focus on this instance and call it the *averaged instance* to distinguish it from the original one.

3.2 Defining Admissible Edges E_{adm}

In this section, we define the set E_{adm} of admissible edges, such that the following holds. There is a $(1+O(\varepsilon))$-approximate clustering $\mathcal{C}_1^*$ that does not break atoms, and all edges between two different clusters in $\mathcal{C}_1^*$ are admissible. We shall bound $|E_{\text{adm}}|$ in terms of $\text{obj}(\mathcal{K})$, which is at most $O(1) \cdot \text{opt}$. Notice that the construction of E_{adm} is a part of our algorithm for the proof of Theorem 13, but the construction of $\mathcal{C}_1^*$ is not. So, the latter procedure can depend on the optimum clustering $\mathcal{C}^*$.

We give the definition of E_{adm} upfront:

$$E^1 := \left\{ uv : \varepsilon w_v < w_u < \frac{w_v}{\varepsilon} \right\}, \tag{6}$$

$$E^2 := \left\{ uv : K_u = K_v \text{ or } uv \in E^1 \text{ and } \sum_{p \in N_u^1 \cap N_v^1} w_{up} w_{vp} > \varepsilon(w_u + w_v) \right\}, \tag{7}$$

$$E_{\text{adm}} := \left\{ uv \in E^2 : K_u \neq K_v \right\}. \tag{8}$$

Above, uv is an unordered pair in V, with $u = v$ allowed. As a sanity check, notice that the conditions for the three definitions are symmetric between u and v. Notice that $E_{\text{adm}} \subseteq E^2 \subseteq E^1$. We use N_u^1, N_u^2 to denote the neighbor sets of u in graphs (V, E^1) and (V, E^2) respectively. We use $d_u^2 = |N_u^2|$ to denote the degree of u in (V, E^2).

We then proceed to show the existence of a clustering $\mathcal{C}_1^*$ with desired properties.

Lemma 17. *There is a clustering $\mathcal{C}_1^*$ of cost at most $(1 + O(\varepsilon))\text{opt}$ that does not break any atom $K \in \mathcal{K}$, and satisfies the following condition: For every u, C with $K_u \subsetneq C \in \mathcal{C}_1^*$, we have $w_{u,C} > \frac{|C|}{2} + \varepsilon \cdot w_u$.*

Proof. Start with $\mathcal{C}_1^* = \mathcal{C}^*$; notice that it does not break any atom. While the condition does not hold for some $C \in \mathcal{C}_1^*$ and $u \in C$, i.e., $w_{u,C} \leq \frac{|C|}{2} + \varepsilon \cdot w_u$, we update $\mathcal{C}_1^* \leftarrow \mathcal{C}_1^* \setminus \{C\} \cup \{C \setminus K_u, K_u\}$. This finishes the construction of $\mathcal{C}_1^*$; clearly it does not break any atom.

We then consider how much cost increment the procedure incurs. Focus on any iteration of the while loop. At the beginning of the iteration, the cost of edges incident to any vertex in K_u in the clustering $\mathcal{C}_1^*$ (w.r.t the averaged instance) is

$$|C| - w(u, C) + w(u, V \setminus C) = w(u, V) + |C| - 2w(u, C) \geq w_u + |C| - 2\left(\frac{|C|}{2} + \varepsilon \cdot w_u\right) = (1 - 2\varepsilon) w_u.$$

The inequality is by the condition of while loop.

The cost increment incurred by separating K_u and $C \setminus K_u$ in the iteration is

$$k_u \cdot \big(w(u, C \setminus K_u) - (|C \setminus K_u| - w(u, C \setminus K_u))\big) = k_u(2w(u, C \setminus K_u) - |C \setminus K_u|)$$
$$\leq k_u \cdot \big(2w(u, C) - |C|\big) \leq 2\varepsilon k_u \cdot w_u.$$

Again, the second inequality is by the condition of the while loop.

We can charge the cost increment using the $(1-2\varepsilon)k_u \cdot w_u$ cost from edges incident to u: every unit cost is used to charge $\frac{2\varepsilon}{1-2\varepsilon}$ unit cost increment. Notice that every edge is charged at most twice. So, we have $\mathrm{obj}(\mathcal{C}_1^*) - \mathrm{obj}(\mathcal{C}) \leq \frac{4\varepsilon}{1-2\varepsilon} \cdot \mathrm{obj}(\mathcal{C}_1^*)$. This implies that $\mathrm{obj}(\mathcal{C}_1^*) \leq \big(1 + O(\varepsilon)\big)\mathrm{obj}(\mathcal{C}^*) = (1 + O(\varepsilon))\mathrm{opt}$. □

We let $\mathcal{C}_1^*$ be the $(1 + O(\varepsilon))$-approximate clustering satisfying the properties of Lemma 17.

Lemma 18. *For every cluster $C \in \mathcal{C}_1^*$ and two vertices $u, v \in C$, we have $w_u > \varepsilon w_v$.*

Proof. Assume $v \notin K_u$ since otherwise $w_u = w_v$. So C contains both K_u and K_v. Lemma 17 implies $w_u \geq w_{u,C} > \frac{|C|}{2}$. On the other hand, it implies $\varepsilon \cdot w_v < w(v, C) - \frac{|C|}{2} \leq |C| - \frac{|C|}{2} = \frac{|C|}{2}$. Therefore, $w_u > \varepsilon w_v$. □

By Lemma 18 and the definition of E^1 in 6, we have

Corollary 19. *Every $C \in \mathcal{C}_1^*$ forms a clique (with self-loops) in (V, E^1).*

Lemma 20. *For any cluster $C \in \mathcal{C}_1^*$, any two vertices $u, v \in C$ with $v \notin K_u$ have $\sum_{p \in N_u^1 \cap N_v^1} w_{up} w_{vp} > \varepsilon \cdot (w_u + w_v)$.*

Proof. Notice that $K_u, K_v \subseteq C$. By Corollary 19, we have $C \subseteq N_u^1$ and $C \subseteq N_v^1$. By Lemma 17, we have $w(u, C) > \frac{|C|}{2} + \varepsilon w_u$, and $w(v, C) > \frac{|C|}{2} + \varepsilon w_v$. Therefore,

$$\sum_{p \in N_u^1 \cap N_v^1} w_{up} w_{vp} \geq \sum_{p \in C} w_{up} w_{vp} \geq \sum_{p \in C} (w_{up} + w_{vp} - 1) = w(u, C) + w(v, C) - |C|$$
$$> \frac{|C|}{2} + \varepsilon w_u + \frac{|C|}{2} + \varepsilon w_v - |C| = \varepsilon(w_u + w_v).$$

The second inequality is by that $w_{vp}, w_{up} \in [0, 1]$. □

Corollary 19, Lemma 20 and the definition of E^2 in (7) imply

Corollary 21. *Every $C \in \mathcal{C}_1^*$ forms a clique (with self-loops) in (V, E^2).*

So, by the definition of E_{adm} in (8), $\mathcal{C}_1^*$ is a good clustering w.r.t the preclustered instance $(\mathcal{K}, E_{\mathrm{adm}})$.

We then proceed to bound $|E_{\mathrm{adm}}|$.

Lemma 22. $d_u^2 - k_u \leq O(\frac{1}{\varepsilon^2})(w_u - k_u)$.

Proof. Notice that $d_u^2 - k_u$ is precisely the number of edges $uv \in E^2$ with $v \notin K_u$. Consider the graph (V, E^1) with edge weights w: Every edge $vp \in E^1$ has weight w_{vp}. We define the weight of a 2-edge path u-p-v in (V, E^1) to be $w_{up} w_{pv}$. By the definition of E^2, if $v \notin K_u$, then $uv \in E^2$ only if the total weight of 2-edge paths of the form u-p-v is at least $\varepsilon(w_u + w_v) > \varepsilon w_u$.

First, consider the paths u-p-v with $p \notin K_u$. The total weight of all such paths (over all p and v) is at most $(w_u - k_u) \cdot \frac{w_u}{\varepsilon}$. This holds since the total weight of edges between u and $V \setminus K_u$ in the complete graph is $w_u - k_u$, and any neighbor p of u in (V, E^1) has $w_p < \frac{w_u}{\varepsilon}$ by the definition of E^1. Then consider the paths u-p-v with $p \in K_u$ and $v \notin K_u$. The total weight of all such paths is at most $k_u(w_u - k_u)$ as any $p \in K_u$ has $w_p = w_u$ and $k_p = k_u$.

Therefore, the total number of verticies $v \notin K_u$ with $uv \in E^2$ is at most

$$\frac{(w_u - k_u) \cdot \frac{w_u}{\varepsilon} + k_u(w_u - k_u)}{\varepsilon w_u} \leq \frac{1}{\varepsilon^2}(w_u - k_u) + \frac{1}{\varepsilon}(w_u - k_u) = O\big(\frac{1}{\varepsilon^2}\big)(w_u - k_u).$$ □

The total number of admissible edges is

$$|E_{\text{adm}}| = \frac{1}{2}\sum_{u\in V}(d_u^2 - k_u) \le O(\frac{1}{\varepsilon^2})\sum_{u\in V}(w_u - k_u) = O(\frac{1}{\varepsilon^2})\cdot \text{obj}(\mathcal{K}) \le O(\frac{1}{\varepsilon^2})\cdot \text{opt}.$$

The second equality is by that $\text{obj}(\mathcal{K}) = \frac{1}{2}\sum_{u\in V}(w_u - k_u)$ and the last inequality is by Lemma 14. We can scale ε down by a constant at the beginning, so that the cost of $\mathcal{C}_1^*$ is at most $(1+\varepsilon)$opt. This finishes the proof of Theorem 13.

4 Solving Cluster LP 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 solution $\left((z_S)_{S\subseteq V}, (x_{uv})_{uv\in\binom{V}{2}}\right)$ to the cluster LP with* $\text{obj}(x) \le (1+\varepsilon)\text{opt}$*, 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.

We apply Theorem 13 to obtain a preclustered instance $(\mathcal{K}, E_{\text{adm}})$, with the unknown good clustering $\mathcal{C}_1^*$. 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 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 $\mathcal{C}_1^*$ using the following procedure. It is not a part of our algorithm; it is only for analysis purpose.

1: **while** there exists some K_u in a cluster $C \in \mathcal{C}_1^*$ with $k_u < |C| \le k_u + \varepsilon_1 \cdot |N_{\text{adm}}(u)|$ **do**
2: $\quad \mathcal{C}_1^* \leftarrow \mathcal{C}_1^* \setminus \{C\} \cup \{K_u, C \setminus K_u\}$

Claim 23. *The procedure increases* $\text{obj}(\mathcal{C}_1^*)$ *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 $\mathcal{C}_1^*$ 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 $\mathcal{C}_1^*$, or in a cluster of size more than $k_u + \varepsilon_1 \cdot |N_{\text{adm}}(u)|$.

4.1 Bounded Sub-Cluster LP Relaxation for Preclustered Instances

We form an LP relaxation aiming at finding the good clustering $\mathcal{C}_1^*$. 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 $\mathcal{C}_1^*$ 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 $\mathcal{C}_1^*$. 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 $\mathcal{C}_1^*$ 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 $\mathcal{C}_1^*$. For every $uv \in \binom{V}{2}$, we have a variable x_{uv} indicating if u and

v are separated or not in $\mathcal{C}_1^*$. 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.

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

$$\sum_{s=1}^{n} y_S^s = y_S \qquad \forall S \qquad (9)$$

$$y_u = 1 \qquad \forall u \qquad (10)$$

$$y_{uv} + x_{uv} = 1 \qquad \forall uv \qquad (11)$$

$$\frac{1}{s}\sum_{u} y_{Su}^s = y_S^s \qquad \forall s, S \qquad (12)$$

$$y_S^s \ge 0 \qquad \forall s, S \qquad (13)$$

$$x_{uv} = 0 \qquad \forall u, v \text{ in a same } K \in \mathcal{K} \qquad (14)$$

$$x_{uv} = 1 \qquad \forall \text{non-admissible edge } uv \qquad (15)$$

$$y_u^s = 0 \qquad \forall u, s \in [k_u - 1] \cup \left[k_u + 1, k_u + \varepsilon_1 |N_{\mathrm{adm}}(u)|\right] \qquad (16)$$

$$\sum_{T' \subseteq T} (-1)^{|T'|} y_{S\cup T'}^s \in [0, y_S^s] \qquad \forall s, S \cap T = \emptyset \qquad (17)$$

(9) gives the definition of y_S, (10) requires u to be contained in some cluster, and (11) gives the definition of x_{uv}. (12) says if $y_S^s = 1$, then there are exactly s elements $u \in V$ with $y_{Su}^s = 1$. (An exception is when $S = \emptyset$; the equality also holds in this case.) (13) is the non-negativity constraint. (14) and (15) follows from that $\mathcal{C}_1^*$ is a good clustering, and (16) follows from (A1). The left side of (17) is the number of clusters of size s containing S but does not contain any vertex in T by inclusion-exclusion principle. So the inequality holds. This corresponds to a Sherali-Adams relaxation needed for the correlated rounding [RT12], see Lemma 24. The running time for solving the LP is $n^{O(r)} = n^{O(1/\varepsilon^{12})}$.

4.2 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 6, to produce a random cluster C. Notice that steps 1 and 2 of the algorithm are well-defined due to equalities (9) and (12).

Algorithm 6 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 $\frac{y_u^s}{s y_\emptyset^s}$; call u the *pivot*
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 (17), the Raghavendra-Tan technique can be applied:

Lemma 24 ([RT12])**.** *In Step 4 of Algorithm 6, one can sample a set $C \subseteq V$ that does not break atoms in time $n^{O(r)}$ such that*

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

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

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.

We prove the following two lemmas.

Lemma 25. *For any $v \in V$, we have $\Pr[v \in C] = \frac{1}{y_\emptyset}$.*

Proof. The probability is

$$\sum_s \frac{y^s_\emptyset}{y_\emptyset} \sum_{u \in V'} \frac{y^s_u}{s y^s_\emptyset} \cdot \frac{y^s_{uv}}{y^s_u} = \frac{1}{y_\emptyset} \sum_s \frac{1}{s} \sum_{u \in V'} y^s_{uv} = \frac{1}{y_\emptyset} \sum_s y^s_v = \frac{1}{y_\emptyset} y_v = \frac{1}{y_\emptyset}.$$

The second equality is by (12). The third and the last inequalities are by (9) and (10) respectively. □

Lemma 26. *Focus on an edge $vw \in \binom{V}{2}$.*

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

Proof. We focus on the first statement. The probability that $v \in C$ and $w \notin C$ conditioned on s is at most

$$\begin{aligned}\sum_{u \in V} \frac{y^s_u}{s y^s_\emptyset} \cdot \left(\frac{1}{y^s_u} \cdot \left(y^s_{uv} - y^s_{uvw}\right) + \mathrm{err}^s_{vw|u} \right) &= \sum_{u \in V} \left(\frac{1}{s y^s_\emptyset} \cdot \left(y^s_{uv} - y^s_{uvw}\right) + \frac{y^s_u}{s y^s_\emptyset} \cdot \mathrm{err}^s_{vw|u} \right) \\ &= \frac{1}{y^s_\emptyset} (y^s_v - y^s_{vw}) + \mathrm{err}^s_{vw}.\end{aligned}$$

To see the second equality, we apply (12) with $S = \{v\}, \{w\}$ and $\{v, w\}$ respectively, and use the definition of err^s_{vw}. Deconditioning on s, we have that the probability vw is split by C is at most

$$\begin{aligned}\sum_s \frac{y^s_\emptyset}{y_\emptyset} \cdot \left(\frac{1}{y^s_\emptyset} (y^s_v - y^s_{vw}) + \mathrm{err}^s_{vw} \right) &= \frac{1}{y_\emptyset} \sum_s (y^s_v - y^s_{vw}) + \mathrm{err}_{vw} \\ &= \frac{1}{y_\emptyset} (1 - y_{vw}) + \mathrm{err}_{vw} = \frac{1}{y_\emptyset} (x_{vw}) + \mathrm{err}_{vw}.\end{aligned}$$

The second equality used that $\sum_s y^s_v = y_v = 1$ and $\sum_s y^s_{vw} = y_{vw} = 1 - x_{vw}$.

For the second statement, the probability that $v, w \in C$ conditioned on s is at most

$$\begin{aligned}\sum_{u \in V} \frac{y^s_u}{s y^s_\emptyset} \cdot \left(\frac{1}{y^s_u} \cdot y^s_{uvw} + \mathrm{err}^2_{vw|u} \right) &= \sum_{u \in V} \left(\frac{1}{s y^s_\emptyset} \cdot y^s_{uvw} + \frac{y^s_u}{s y^s_\emptyset} \cdot \mathrm{err}^s_{vw|u} \right) \\ &= \frac{1}{y^s_\emptyset} \cdot y^s_{vw} + \mathrm{err}^s_{vw}.\end{aligned}$$

Deconditioning on s, as before, results in the desired bound of $y_{vw}/y_\emptyset + \mathrm{err}_{vw}$. □

We can then bound the total unconditional error over all vw pairs:

Lemma 27. $\sum_{vw\in\binom{V}{2}} \mathrm{err}_{vw} \le O(\varepsilon_1)\cdot\frac{1}{y_\emptyset}|E_{\mathrm{adm}}|$.

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 (16), we have that $s>k_u+\varepsilon_1\cdot|N_{\mathrm{adm}}(u)|$, since otherwise we shall have $y^s_u=0$. By the second property of Lemma 24, 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,

$$\sum_{vw\in\binom{V}{2}}\mathrm{err}^s_{vw|u}\le\frac{\varepsilon_{\mathrm{rt}}}{2}\cdot|N_{\mathrm{adm}}(u)|^2\le\frac{\varepsilon_{\mathrm{rt}}}{2\varepsilon_1}\cdot|N_{\mathrm{adm}}(u)|\cdot(s-k_u)$$
$$=\frac{\varepsilon_1}{2}\cdot|N_{\mathrm{adm}}(u)|\cdot\sum_{v\in N_{\mathrm{adm}}(u)}\frac{y^s_{uv}}{y^s_u}=\frac{\varepsilon_1}{2}\cdot\sum_{v,w\in N_{\mathrm{adm}}(u)}\frac{y^s_{uv}}{y^s_u}. \tag{18}$$

The first equality is by (12) 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 (17). We have $y_{uv}=\sum_s y^s_{uv}$, $y_u=\sum_s y^s_u$, and $y_{uv}=y_u=1$ if $v\in K_u$.) The second equality is obtained by replacing $|N_{\mathrm{adm}}(u)|$ with $\sum_{w\in N_{\mathrm{adm}}(u)}1$.

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

$$\sum_{vw}\mathrm{err}^s_{vw}=\frac{1}{sy^s_\emptyset}\sum_u y^s_u\cdot\sum_{vw}\mathrm{err}^s_{vw|u}\le\frac{1}{sy^s_\emptyset}\sum_u y^s_u\cdot\sum_{v,w\in N_{\mathrm{adm}}(u)}\frac{\varepsilon_1}{2}\cdot\frac{y^s_{uv}}{y^s_u}=\frac{\varepsilon_1}{2}\cdot\frac{1}{sy^s_\emptyset}\cdot\sum_{u\in V,v,w\in N_{\mathrm{adm}}(u)}y^s_{uv}$$
$$=\frac{\varepsilon_1}{2}\cdot\sum_{v\in V}\frac{y^s_v}{sy^s_\emptyset}\sum_{u\in N_{\mathrm{adm}}(v),w\in N_{\mathrm{adm}}(u)}\frac{y^s_{uv}}{y^s_v}\quad\le\quad\frac{\varepsilon_1}{2}\cdot\sum_{v\in V}\frac{y^s_v}{sy^s_\emptyset}\sum_{uw\in E_{\mathrm{adm}}}\left(\frac{y^s_{uv}+y^s_{vw}}{y^s_v}\right)$$
$$\le\varepsilon_1\cdot\sum_{v\in V}\frac{y^s_v}{sy^s_\emptyset}\sum_{uw\in E_{\mathrm{adm}}}\Pr[C\cap\{u,w\}\neq\emptyset\mid s,v\text{ is pivot}]$$
$$=\varepsilon_1\sum_{uw\in E_{\mathrm{adm}}}\Pr[C\cap\{u,w\}\neq\emptyset\mid s]. \tag{19}$$

The first inequality is by (18). For the second inequality in the above sequence, notice that we view uw as an unordered pair. Thus we keep both y^s_{uv} and y^s_{vw} in the numerator. To see the third inequality, notice that $\frac{y^s_{uv}}{y^s_v}=\Pr[u\in C|s,v\text{ is pivot}]\le\Pr[C\cap\{u,w\}\neq\emptyset|s,v\text{ is pivot}]$. The same inequality holds for $\frac{y^s_{vw}}{y^s_v}$.

Finally, we take all s into consideration:

$$\sum_{vw}\mathrm{err}_{vw}=\sum_s\frac{y^s_\emptyset}{y_\emptyset}\cdot\sum_{vw}\mathrm{err}^s_{vw}\le\varepsilon_1\cdot\sum_s\frac{y^s_\emptyset}{y_\emptyset}\sum_{uw\in E_{\mathrm{adm}}}\Pr[C\cap\{u,w\}\neq\emptyset|s]$$
$$=\varepsilon_1\cdot\sum_{uw\in E_{\mathrm{adm}}}\Pr[C\cap\{u,w\}\neq\emptyset]\le\frac{2\varepsilon_1}{y_\emptyset}|E_{\mathrm{adm}}|+3\varepsilon_1\sum_{uw\in\binom{V}{2}}\mathrm{err}_{uw}.$$

The first inequality is by (19). To see the second inequality, we notice that $C\cap\{u,w\}\neq\emptyset$ is the union of the 3 disjoint events: $u\in C$ and $w\notin C$, $u\notin C$ and $w\in C$, and $\{u,w\}\subseteq C$. By Lemma 26, we have $\Pr[C\cap\{u,w\}\neq\emptyset]\le\frac{2x_{uw}+y_{uw}}{y_\emptyset}+3\cdot\mathrm{err}_{uw}\le\frac{2}{y_\emptyset}+3\cdot\mathrm{err}_{uw}$. 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. □

4.3 Using Independently Sampled Clusters to Construct Cluster LP Solution

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 $|E_{\text{adm}}| \geq 1$ since otherwise the preclustered instance is trivial.) We run Algorithm 6 $\Delta y_\emptyset$ times independently to obtain clusters $C_1, C_2, \cdots, C_{\Delta y_\emptyset}$.

We use the following variant of Chernoff bound.

Theorem 28. *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 $\mu' \geq \mu$ be a real. Then for any $\delta \in (0,1)$, we have*

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

For every $u \in V$, let $R_u = \{t : u \in C_t\}$. Notice that $\Delta y_\emptyset \cdot \frac{|E_{\text{adm}}|}{y_\emptyset n^2} = \Theta(\frac{\log n}{\varepsilon_1^2})$, 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 $|R_u| \geq (1-\varepsilon_1)\Delta y_\emptyset \cdot \frac{1}{y_\emptyset} = (1-\varepsilon_1)\Delta$.
- For every $u, v \in V$ such that $uv \in E^+$, we have

$$\begin{aligned} |R_u \setminus R_v| &\leq \Delta y_\emptyset \left(\frac{x_{uv}}{y_\emptyset} + \text{err}_{uv} + \varepsilon_1 \cdot \max\left\{ \frac{x_{uv}}{y_\emptyset} + \text{err}_{uv}, \frac{|E_{\text{adm}}|}{y_\emptyset n^2} \right\} \right) \\ &\leq (1+\varepsilon_1)\Delta(x_{uv} + y_\emptyset \text{err}_{uv}) + \frac{\varepsilon_1 \Delta |E_{\text{adm}}|}{n^2}. \end{aligned} \tag{20}$$

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

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

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 (20).

Claim 29. *For every $uv \in E^+$ we have $\max\{|R'_u \setminus R'_v|, |R'_v \setminus R'_u|\} \leq (1+\varepsilon_1)\Delta(x_{uv} + y_\emptyset \text{err}_{uv}) + \frac{\varepsilon_1 \Delta |E_{\text{adm}}|}{n^2}$.*

Proof. For convenience, we use B to denote the upper bound $(1+\varepsilon_1)\Delta(x_{uv} + y_\emptyset \text{err}_{uv}) + \frac{\varepsilon_1 \Delta |E_{\text{adm}}|}{n^2}$. 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 $|R_u| = |R_v| > \lceil (1-\varepsilon_1)\Delta \rceil$, 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

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

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

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

Therefore,

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

The second inequality is due to Lemma 27, and the third one used that $|E_{\mathrm{adm}}| \le O\big(\frac{1}{\varepsilon^2}\big)\cdot \mathrm{opt}$. By scaling ε, the upper bound can be made to $(1+\varepsilon)\mathrm{opt}$. This finishes the proof of Theorem 1.

5 Triangle Analysis used in 1.56-Approximation of Cluster LP

The goal of this section is to prove Lemma 9, which is restated here.

Lemma 9. *For Algorithm 3 and budget functions* $\mathrm{budget}^+ \equiv \mathrm{budget}^+_{1.56}$ *and* $\mathrm{budget}^- \equiv \mathrm{budget}^-_{1.56}$*, we have* $\mathrm{cost}(T) \le \Delta(T)$ *for every triangle* T*.*

The proof of Lemma 9 follows from Lemma 35, 36, 37, 38, 39 which we prove in this section. First, we recall some notation.

Notation and Useful Observations. We define some notation and make some observations that will be useful in this section and in Section 6. Recall that we are given a polynomial-sized LP solution $(z = (z_S)_{S\subseteq V}, x \in [0,1]^{\binom{V}{2}})$ to (cluster LP). Moreover, remember that we choose the threshold $\tau = 0.4$. For a +edge uv, we say uv is *short* if $x_{uv} \le \tau$ and *long* if $x_{uv} > \tau$. Notice that the pivot-based rounding procedure (Algorithm 1) treats short and long +edges differently. For vertices $u, v, w \in V$, we define

$$y_{uv} := \sum_{S\supseteq\{u,v\}} z_S \in [0,1], \qquad y_{uvw} := \sum_{S\supseteq\{u,v,w\}} z_S \in [0,1], \qquad y_{uv|w} := y_{uv} - y_{uvw} \in [0,1],$$
$$y_{u|v|w} := 1 - y_{uv|w} - y_{uw|v} - y_{vw|u} - y_{uvw}.$$

In an integral solution z, $y_{uv}, y_{uvw}, y_{uv|w}$ and $y_{u|v|w}$, respectively, indicate if u and v are in the same cluster, if u, v and w are all in the same cluster, if u and v are in the same cluster not containing w, and if u, v, w are in three different clusters. These definitions also hold when u, v and w are not distinct vertices.

If a solution (z_S) satisfies the 3-round Sherali-Adams constraints, then we would have

$$y_{uv} + y_{uw} + y_{vw} = 1 + (2y_{uvw} - y_{u|v|w})$$

and $y_{u|v|w} \in [0,1]$. However, for a solution to cluster LP, we have the following, weaker, claim.

Lemma 30. *The following holds in a solution of the cluster LP. For any* $\{u, v, w\}$*, we have*

$$3y_{uvw} \le y_{uv} + y_{uw} + y_{vw} \le \frac{3}{2} + \frac{3}{2}y_{uvw}.$$

Proof. For a vertex u, we have: $1 - y_{uv} - y_{uw} + y_{uvw} \in [0,1]$. This follows from the fact that

$$\sum_{S\ni u} z_S \ge \sum_{S\ni u,v} z_S + \sum_{S\ni u,w} z_S - \sum_{S\ni u,v,w} z_S \quad\Longleftrightarrow\quad 1 \ge y_{uv} + y_{uw} - y_{uvw}.$$

Now we consider three equalities, one each for u, v and w. Adding up and dividing by 3 implies the lemma. □

In the following two claims, we focus on one iteration of the while loop in Algorithm 3. V' is the vertex set at the beginning of the iteration, and C is the cluster obtained at the end. We use u to denote the event that u is the chosen pivot.

Claim 31. *Let $u, v \in V'$ (it is possible that $u = v$). Then*

$$\Pr[v \in C \mid u] = \begin{cases} 1 & \text{if } uv \text{ is a short } + \text{edge} \\ 1 - x_{uv} = y_{uv} & \text{otherwise} \end{cases}.$$

Claim 32. *Let $u, v, w \in V'$ with $v \neq w$. Then*

$$\Pr[\{v, w\} \subseteq C \mid u] = \begin{cases} y_{uvw} & \text{if both } uv \text{ and } uw \text{ are long } + \text{edges} \\ \Pr[v \in C \mid u] \cdot \Pr[w \in C \mid u] & \text{otherwise} \end{cases},$$

$$\Pr[v \in C \wedge w \notin C \mid u] = \begin{cases} y_{uv|w} & \text{if both } uv \text{ and } uw \text{ are long } + \text{edges} \\ \Pr[v \in C \mid u] \cdot \big(1 - \Pr[w \in C \mid u]\big) & \text{otherwise} \end{cases}.$$

We now prove some useful facts about the budget functions initially defined in Section 2.2.

$$\text{budget}^+_\alpha(x) = \frac{\alpha}{1 - \alpha/2} \cdot \frac{x^2}{1 + x}, \qquad \text{and} \qquad \text{budget}^-_\alpha(x) = \frac{\alpha}{1 - \alpha/2} \cdot \frac{(1 + 2x)(1 - x)}{2(1 + x)}, \qquad \forall x \in [0, 1].$$

Lemma 33. *For any $\alpha \in [1, 2)$, $\text{budget}^+_\alpha(x)$ is convex and increasing. Moreover, $\text{budget}^-_\alpha(x)$ is decreasing.*

Proof. Consider the first and second derivative of $\text{budget}^+_\alpha(x)$,

$$\frac{d}{dx}\text{budget}^+_\alpha(x) = \frac{\alpha}{1 - \alpha/2} \cdot \frac{x(2 + x)}{(1 + x)^2} \geq 0,$$
$$\frac{d^2}{d^2x}\text{budget}^+_\alpha(x) = \frac{\alpha}{1 - \alpha/2} \cdot \frac{2}{(x + 1)^3} \geq 0.$$

Here, we used that $\frac{\alpha}{1-\alpha/2} \geq 0$ and $x \geq 0$. Similarly, for $\text{budget}^-_\alpha(x)$,

$$\frac{d}{dx}\text{budget}^-_\alpha(x) = \frac{\alpha}{1 - \alpha/2} \cdot \frac{-x(x + 2)}{(1 + x)^2} \leq 0. \qquad \square$$

We will use the following claim several times.

Claim 34. *For $x \geq \tau$, $\text{budget}^+_{1.56}(x) \geq 2x$.*

Proof. Our goal is to show that

$$\frac{2\alpha}{2 - \alpha}\frac{x^2}{1 + x} \geq 2x.$$

Simplifying, we have

$$\frac{x}{1 + x} \geq \frac{2 - \alpha}{\alpha}.$$

Solving this inequality, we have

$$x \geq \frac{2 - \alpha}{2\alpha - 2}.$$

When $\alpha = 1.56$, the inequality holds when $x \geq 0.392857$. $\lozenge$

We will prove the lemma by considering each possible triangle type separately. That is, degenerate triangles, triangles with no +edge ($---$ triangles), triangles with one +edge ($+--$ triangles), triangles with two +edges ($++-$ triangles) and lastly, triangles with three +edges ($+++$ triangles). Moreover, recall that the pivot rounding algorithm treats short +edges differently than long +edges. Therefore, for each type of triangle, we consider different cases that specify how many short and long +edges the triangle contains.

Degenerate triangles. For the case when two of the vertices u, v, w are identical, we want to make sure that $\text{cost}(u, v) \leq \Delta(u, v)$.

Lemma 35. *Fix budget functions* $\text{budget}^+ \equiv \text{budget}^+_\alpha$ *and* $\text{budget}^- \equiv \text{budget}^-_\alpha$*, where* $\alpha \geq 4/3$*. For any degenerate triangle* T*, we have* $\text{cost}(T) \leq \Delta(T)$*.*

Proof. For a $-$edge with value x this is equivalent to

$$(1 - x) \leq \frac{\alpha(1 + 2x)(1 - x)}{(2 - \alpha)(1 + x)}.$$

This holds true since $\frac{\alpha(1+2x)}{(2-\alpha)(1+x)} \geq \frac{\alpha}{(2-\alpha)} \geq 1$, for any $\alpha \geq 1$. For short $+$edges the cost is 0 and the inequality holds trivially. For long $+$edges, with value $x > 0.4$, we need,

$$x \leq \frac{2\alpha x^2}{(2 - \alpha)(1 + x)}.$$

This holds true since $\frac{2\alpha x}{(2-\alpha)(1+x)} \geq \frac{2\alpha/3}{(2-\alpha)(1+1/3)} \geq 1$ for any $\alpha \geq 4/3$. □

5.1 +++ Triangles

Lemma 36. *Fix the budget function* $\text{budget}^+ \equiv \text{budget}^+_\alpha$*, where* $\alpha = 1.56$*. For any +++ triangle* T*, we have* $\text{cost}(T) \leq \Delta(T)$*.*

Proof. We distinguish between the following four cases: the triangle has three short edges, two short edges, one short edge or no short edge.

(s,s,s) $\text{cost}(T) = 0$ and $\Delta(T) \geq 0$.

(s,s,l) For a $+++$ triangle $T = (u, v, w)$, assume that uv is long and uw, vw are short. Let $a = x_{uv}, b = x_{uw}$ and $c = x_{vw}$. Then

$$\begin{aligned}\text{cost}(T) &= 2a,\\ \Delta(T) &= \text{budget}^+(a) + \text{budget}^+(b) + \text{budget}^+(c) \geq \text{budget}^+(a) \geq 2a.\end{aligned}$$

Here, we used that $\text{budget}^+(a) \geq 2a$ by Claim 34.

(s,l,l) For a $+++$ triangle $T = (u, v, w)$, assume that uv is short and uw and vw are long, and again let $a = x_{uv}, b = x_{uw}$ and $c = x_{vw}$.

$$\begin{aligned}\text{cost}(T) &= b + c + y_{vw} + y_{uw} - 2 \cdot y_{uvw} = 2 - 2 \cdot y_{uvw}\\ \Delta(T) &= \text{budget}^+(b) + \text{budget}^+(c) + \text{budget}^+(a) \cdot (y_{vw} + y_{uw} - y_{uvw})\\ &\geq \text{budget}^+(b) + \text{budget}^+(c).\end{aligned}$$

Because $b > \tau$, we have $\text{budget}^+(b) \geq 2b$, by Claim 34. Hence,

$$\begin{aligned}\Delta(T) - \text{cost}(T) &\geq 2 \cdot (b + c) - 2 + 2 \cdot y_{uvw}\\ &= 2 \cdot (2 - y_{uw} - y_{vw} + y_{uvw}) - 2\\ &\geq 0.\end{aligned}$$

In the last inequality, we used that $y_{uw} + y_{vw} - y_{uvw} \leq 1$

(l,l,l) In this case, we use correlated rounding. For a +++ triangle $T = (u, v, w)$, let $a = x_{uv}, b = x_{uw}$ and $c = x_{vw}$. We have

$$\text{cost}(T) = 2(y_{uv} + y_{uw} + y_{vw}) - 6y_{uvw},$$
$$\Delta(T) = \text{budget}^+(a)(y_{uw} + y_{vw} - y_{uvw}) + \text{budget}^+(b)(y_{uv} + y_{vw} - y_{uvw}) + \text{budget}^+(c)(y_{uv} + y_{uw} - y_{uvw}).$$

Let us fix $s = y_{uv} + y_{uw} + y_{vw}$ (which implies $a + b + c = 3 - s$), and let us also fix y_{uvw}. Then we have

$$\Delta(T) = \text{budget}^+(a)(s - 1 + a - y_{uvw}) + \text{budget}^+(b)(s - 1 + b - y_{uvw}) + \text{budget}^+(c)(s - 1 + c - y_{uvw}).$$

Since $\Delta(T)$ is convex in terms of a, b and c, then subject to the constraint $a + b + c = 3 - s$, the function $\Delta(T)$ is minimized when $a = b = c = \frac{3-s}{3}$. Let us set $p = 1 - \frac{s}{3}$. We have

$$\Delta(T) \geq 3 \cdot \text{budget}^+(p)(s - 1 + p - y_{uvw}) = 3 \cdot \text{budget}^+(p)(\frac{2s}{3} - y_{uvw}) = 2s \cdot \text{budget}^+(p) - 3y_{uvw} \cdot \text{budget}^+(p).$$

Now we want to show that $\Delta(T) - \text{cost}(T) \geq 0$.

$$\Delta(T) - \text{cost}(T) \geq 2s \cdot \text{budget}^+(p) - 3y_{uvw} \cdot \text{budget}^+(p) - 2s + 6y_{uvw}.$$

So we want to show the following inequality.

$$2s \cdot \text{budget}^+(p) + 6y_{uvw} \geq 2s + 3y_{uvw} \cdot \text{budget}^+(p). \tag{21}$$

Since all edges are long, we have $x_{uv} + x_{uw} + x_{vw} \geq 3\tau$, which implies that $s \leq 3 - 3\tau = 1.8$, since $\tau = 0.4$. Hence, $s \in [0, 1.8]$ and $p \in [0.4, 1]$, so $\text{budget}^+(p)$ is in the range $[\text{budget}^+(\tau), \text{budget}^+(1)]$. We will consider three cases, according to the value of $\text{budget}^+(p)$. The first case is when $\text{budget}^+(p) \in [1, 2]$, in which case Inequality (21) clearly holds.

The second case is when $\text{budget}^+(p) \in (2, \text{budget}^+(1)]$. For ease of notation, let $\bar{b} = \text{budget}^+(p)$ and assume $\bar{b} > 2$. Rearranging, we derive the next inequalities, which are all equivalent.

$$2s \cdot \bar{b} + 6y_{uvw} \geq 2s + 3y_{uvw} \cdot \bar{b}$$
$$2s + 2s(\bar{b} - 2) \geq 3y_{uvw} \cdot \bar{b} - 6y_{uvw}$$
$$2s + 2s(\bar{b} - 2) \geq 3y_{uvw}(\bar{b} - 2).$$

Since $s = y_{uv} + y_{uw} + y_{vw} \geq 3y_{uvw}$, the inequality holds.

The third and last case is when $\text{budget}^+(p) \in [\text{budget}^+(\tau), 1)$. Again, for simplicity, let $\bar{b} = \text{budget}^+(p)$. Then our goal is to prove the following for $\bar{b} \in [\text{budget}^+(\tau), 1)$.

$$2s \cdot \bar{b} + 6y_{uvw} \geq 2s + 3y_{uvw} \cdot \bar{b}.$$

Rearranging, we have

$$3y_{uvw} + 3y_{uvw}(1 - \bar{b}) \geq 2s(1 - \bar{b}).$$

Again, rearranging, we have

$$2s \leq 3y_{uvw} + \frac{3y_{uvw}}{1 - \bar{b}}. \tag{22}$$

By Lemma 30, we have $2s \leq 3 + 3y_{uvw}$. Thus, if we can show

$$\frac{y_{uvw}}{1 - \bar{b}} \geq 1 \quad \iff \quad y_{uvw} \geq 1 - \bar{b}$$

then we have proved (22). To prove this, we will show

$$1 - \bar{b} \leq \frac{2s}{3} - 1 \leq y_{uvw}.$$

The second inequality follows from Lemma 30. Recalling that $p = 1 - \frac{s}{3}$, we just need to show

$$1 - \bar{b} = 1 - \text{budget}^+(1 - \frac{s}{3}) \leq \frac{2s}{3} - 1,$$

which is established next.

We want to show

$$1 - \text{budget}^+_{1.56}(1 - \frac{s}{3}) \leq \frac{2s}{3} - 1.$$

Rearranging, we have

$$2 - \frac{2s}{3} \leq \text{budget}^+_{1.56}(1 - \frac{s}{3}),$$

which follows from Claim 34, since $1 - \frac{s}{3} \in [\tau, 1]$. □

5.2 −−− Triangles

Lemma 37. *Fix the budget function* $\text{budget}^- \equiv \text{budget}^-_\alpha$*, where* $\alpha \geq 1$*. For any* −−− *triangle* T*, we have* $\text{cost}(T) \leq \Delta(T)$*.*

Proof. We have that $\text{budget}^-_\alpha(x) = \frac{\alpha}{1-\alpha/2} \cdot \frac{(1+2x)(1-x)}{2(1+x)} \geq (1-x)$. As before, in the proof of Lemma 35, we have $\frac{\alpha(1+2x)}{(2-\alpha)(1+x)} \geq \frac{\alpha}{(2-\alpha)} \geq 1$, for any $\alpha \geq 1$. The inequality holds true if the coefficient in the budget is 1.

$$\begin{aligned}
\text{cost}(T) &= y_{uv}y_{uw} + y_{uv}y_{vw} + y_{uw}y_{vw}.\\
\Delta(T) &\geq (y_{uv} + y_{uw} - y_{uv}y_{uw})y_{vw} + (y_{uv} + y_{vw} - y_{uv}y_{vw})y_{uw} + (y_{uw} + y_{vw} - y_{uw}y_{vw})y_{uv}\\
&= 2(y_{uv}y_{uw} + y_{uv}y_{vw} + y_{uw}y_{vw}) - 3y_{uv}y_{uw}y_{vw} \geq y_{uv}y_{uw} + y_{uv}y_{vw} + y_{uw}y_{vw}.
\end{aligned}$$

□

5.3 +−− Triangles

Lemma 38. *Fix budget functions* $\text{budget}^+ \equiv \text{budget}^+_\alpha$ *and* $\text{budget}^- \equiv \text{budget}^-_\alpha$*, where* $\alpha \geq \frac{4}{3}$*. For any* +−− *triangle* T*, we have* $\text{cost}(T) \leq \Delta(T)$*.*

Proof. We will prove the following two cases.

short +edge For a +−− triangle $T = (u, v, w)$, assume that uv is a short +edge and uw and vw are −edges. The following inequality holds when $\alpha \geq \frac{4}{3}$.

$$\text{budget}^-_\alpha(x) = \frac{\alpha}{1-\alpha/2} \cdot \frac{(1+2x)(1-x)}{2(1+x)} \geq 2(1-x).$$

Therefore, we have for $\text{cost}(T)$ and $\Delta(T)$,

$$\begin{aligned}
\text{cost}(T) &= y_{uw} + y_{vw} + y_{uw} + y_{vw} - 2 \cdot y_{uw}y_{vw} \leq 2(y_{uw} + y_{vw}),\\
\Delta(T) &\geq \text{budget}^-(b) + \text{budget}^-(c) \geq 2(y_{uw} + y_{vw}).
\end{aligned}$$

long +edge For a +−− triangle $T = (u, v, w)$, assume that uv is a long +edge and uw and vw are −edges. For $x \geq \tau$, as observed in the proof of Lemma 35, we can lower bound the budget of the +edge as follows.

$$\text{budget}^+_\alpha(x) = \frac{\alpha}{1-\alpha/2} \cdot \frac{x^2}{1+x} \geq \frac{2\alpha\tau}{(2-\alpha)(1+\tau)}x \geq x.$$

As in the previous case, $\text{budget}^-(x) \geq 2(1-x)$. The inequality $\text{cost}(T) \leq \Delta(T)$ is true if the coefficient for the $+$edge is 1 and for the $-$edge is 2.

$$\begin{aligned}
\text{cost}(T) &= y_{uw}y_{uv} + y_{vw}y_{uv} + y_{uw} + y_{vw} - 2y_{uw}y_{vw} = (2-a)(y_{uw}+y_{vw}) - 2y_{uw}y_{vw}.\\
\Delta(T) &\geq (y_{uw}+y_{vw}-y_{uw}y_{vw})a + 2(y_{uw}+y_{uv}-y_{uw}y_{uv})y_{vw} + 2(y_{vw}+y_{uv}-y_{vw}y_{uv})y_{uw}\\
&= (y_{uw}+y_{vw}-y_{uw}y_{vw})a + 2(1-a+y_{uw}a)y_{vw} + 2(1-a+y_{vw}a)y_{uw}\\
&= (2-a)(y_{uw}+y_{vw}) + 3y_{uw}y_{vw}a.
\end{aligned}$$

□

5.4 ++− Triangles

Lemma 39. *Fix budget functions* $\text{budget}^+ \equiv \text{budget}^+_\alpha$ *and* $\text{budget}^- \equiv \text{budget}^-_\alpha$*, where* $\alpha = 1.56$*. For any* $++-$ *triangle* T*, we have* $\text{cost}(T) \leq \Delta(T)$*.*

Proof. Again, we distinguish cases depending on the number of short $+$edges.

(s,s) For a $++-$ triangle $T = (u, v, w)$, assume that uv is a $-$edge and uw, vw are short $+$edges. As usual, let $x_{uv} = a, x_{uw} = b$ and $x_{vw} = c$.

$$\begin{aligned}
\text{cost}(T) &= 2\cdot a + 1,\\
\Delta(T) &= \text{budget}^-(a) + \text{budget}^+(b) + \text{budget}^+(c) \geq \text{budget}^-(a) + 2\cdot\text{budget}^+\left(\frac{b+c}{2}\right)\\
&\geq \text{budget}^-(a) + 2\cdot\text{budget}^+\left(\frac{a}{2}\right) = \frac{2\alpha}{2-\alpha}\left(\frac{(1+2a)(1-a)}{2(1+a)} + \frac{a^2}{2+a}\right). \qquad (23)
\end{aligned}$$

As before, we used the convexity of budget^+ (Lemma 33) together with the triangle inequality, $a \leq b + c$. The derivative of (23) w.r.t. a is less than 0 for $a \geq 0$,

$$-\frac{2\alpha}{2-\alpha}\left(\frac{a(3a+4)}{(1+a)^2(2+a)^2}\right) \leq 0.$$

Hence, (23) is decreasing with a. Observe that $a \leq b + c \leq 2\tau$. Since the cost is increasing with a we only need to check the case $a = 2\tau$,

$$\begin{aligned}
\Delta(T) - \text{cost}(T) &\geq \text{budget}^-(2\tau) + 2\cdot\text{budget}^+(\tau) - 4\tau - 1\\
&\geq \text{budget}^-(2\tau) + 4\tau - 4\tau - 1 \geq .02.
\end{aligned}$$

Notice that we use Claim 34 to lower bound $\text{budget}^+(\tau)$.

(s,l) For a $++-$ triangle $T = (u, v, w)$, assume that uv is a $-$edge, uw is a short $+$edge and vw is a long $+$edge.

$$\begin{aligned}
\text{cost}(T) &= a + y_{vw} + y_{vw} + y_{uv} - 2y_{vw}y_{uv} = 1 + 2y_{vw}(1-y_{uv}) = 1 + 2(1-x_{vw})a = 1 + 2(1-c)a.\\
\Delta(T) &= \text{budget}^+(c) + \text{budget}^-(a) + \text{budget}^+(b)(y_{vw} + y_{uv} - y_{vw}y_{uv})\\
&= \text{budget}^+(c) + \text{budget}^-(a) + \text{budget}^+(b)(2 - c - a - (1-c)(1-a))\\
&= \text{budget}^+(c) + \text{budget}^-(a) + \text{budget}^+(b)(1-ac).
\end{aligned}$$

Case 1 $c \geq a$.

$$\Delta(T) \geq \text{budget}^+(a) + \text{budget}^-(a) = \frac{\alpha}{2-\alpha} \geq 3,$$

while $\text{cost}(T)$ is always less than 3.

Case 2 $c \le a$. We have $b \ge a - c$ by triangle inequality. Since $b \le \tau$, we also have $a - c \le \tau$. Since $\text{budget}^+(x)$ is increasing by Lemma 33,

$$\Delta(T) \ge \text{budget}^+(c) + \text{budget}^-(a) + \text{budget}^+(a-c)(1-ac).$$

Define $C_\alpha = \frac{\alpha}{2-\alpha}$. For a fixed $x := c \ge \tau$, define $z := a$. For $-$edges we can use a coefficient of C_α, since $\text{budget}^-(x) \ge C_\alpha(1-x)$. We need to argue that the difference on line (24) is non-negative.

$$\begin{aligned}
\Delta(T) - \text{cost}(T) &\ge \text{budget}^+(x) + \text{budget}^-(z) + \text{budget}^+(z-x)(1-xz) - 1 - 2(1-x)z && (24)\\
&\ge \text{budget}^+(x) + C_\alpha(1-z) + \text{budget}^+(z-x)(1-xz) - 1 - 2(1-x)z && (25)\\
&= 2C_\alpha\frac{x^2}{1+x} + C_\alpha(1-z) + 2C_\alpha\frac{(z-x)^2}{1+z-x}(1-xz) - 1 - 2(1-x)z. && (26)
\end{aligned}$$

Define

$$f(z) := C_\alpha(1-z) + 2C_\alpha\frac{(z-x)^2}{1+z-x}(1-xz) - 1 - 2(1-x)z.$$

Claim 40. *f is decreasing.*

Proof. We will show that the derivative of f is negative. We have

$$\begin{aligned}
f'(z) &= C_\alpha\left(2x^2 - 4xz + 2x + 1 + \frac{2(x-1)x-2}{(1+z-x)^2}\right) + 2(x-1)\\
&\le C_\alpha\left(-2xz + 2x + 1 + \frac{2(x-1)x-2}{(1+z-x)^2}\right).
\end{aligned}$$

We will show that $g(z) := \left(-2xz + 2x + 1 + \frac{2(x-1)x-2}{(1+z-x)^2}\right) < 0$ and thus, $f'(z) < 0$. We have,

$$g'(z) = -\frac{4(x^2-x-1)}{(1+z-x)^3} - 2x,$$

and,

$$g''(z) = \frac{12(x^2-x-1)}{(1+z-x)^4}.$$

Note that $g''(z) \le 0$ since $C > 0$, the numerator is negative for $x \in [0,1]$ and the denominator is always positive. Thus, $g'(z)$ is decreasing and minimized for $z = 1$,

$$g'(1) = -\frac{4(x^2-x-1)}{(2-x)^3} - 2x \ge 0,$$

for all $x \in [0,1]$. Hence, $g(z)$ is increasing and maximized for $z = \min\{1, x+\tau\}$. We have,

$$g(1) = \left(-2x + 2x + 1 + \frac{2(x-1)x-2}{(2-x)^2}\right) < 0$$

for $x \ge 1 - \tau$ and,

$$g(x+\tau) = \left(-2x(x+\tau) + 2x + 1 + \frac{2(x-1)x-2}{(1+\tau)^2}\right) < 0,$$

for $x \in [0,1]$. ◊

Now we return to our goal, which is to show that

$$2C_\alpha\frac{x^2}{1+x} + C_\alpha(1-z) + 2C_\alpha\frac{(z-x)^2}{1+z-x}(1-xz) - 1 - 2(1-x)z \ge 0. \quad (27)$$

By Claim 40, we can assume that $z = \min\{1, \tau + x\}$. First, we consider the case in which $\tau + x \geq 1$, in which case we assume that $z = 1$. Thus, we have

$$(27) \geq 2C_\alpha \left(x^2 + \frac{1}{x-2} + \frac{1}{x+1} \right) + 2x - 3 \geq 0 \tag{28}$$

for $x \geq 1 - \tau$.

Next, we consider the case in which $z = x + \tau < 1$,

$$(27) \geq 2C_\alpha \frac{x^2}{1+x} + C_\alpha(1 - x - \tau) + 2C_\alpha \frac{\tau^2}{1+\tau}(1 - x(x+\tau)) - 1 - 2(1-x)(x+\tau) \geq 0, \tag{29}$$

for $x \in [0, 1]$.

(1,1) For a $++-$ triangle $T = (u, v, w)$, assume that uv is a $-$edge and uw, vw are long $+$edges. Define $C_\alpha = \frac{\alpha}{2-\alpha}$. For $-$edges we can use a coefficient of C_α, since $\text{budget}^-(x) \geq C_\alpha(1-x)$. For $+$edges define $f(x) = \frac{2C_\alpha \cdot x^2}{1+x}$ so that $\text{budget}^+(x) = f(x)$. Recall that $a = x_{uv}, b = x_{uw}$ and $c = x_{vw}$. Without loss of generality, we assume that $b \leq c$, which implies $y_{uw} \geq y_{vw}$.

$$\begin{aligned}
\Delta(T) &= C_\alpha(y_{uw} + y_{vw} - y_{uvw})y_{uv} + f(c)(y_{uw} + y_{uv} - y_{uw}y_{uv}) + f(b)(y_{vw} + y_{uv} - y_{vw}y_{uv}) \\
&\geq C_\alpha(y_{uw} + y_{vw} - y_{uvw})y_{uv} + 2c(y_{uw} + y_{uv} - y_{uw}y_{uv}) + 2b(y_{vw} + y_{uv} - y_{vw}y_{uv}). \\
\text{cost}(T) &= y_{uvw} + y_{uw} + y_{uv} - 2y_{uw}y_{uv} + y_{vw} + y_{uv} - 2y_{vw}y_{uv} \\
&= y_{uvw} + y_{uw} + y_{vw} + 2y_{uv} - 2(y_{uw} + y_{vw})y_{uv}.
\end{aligned}$$

The inequality follows from Claim 34. Using $y_{uv} = 1 - a, y_{uw} = 1 - b, y_{vw} = 1 - c$ and $y = y_{uvw}$, we have

$$\begin{aligned}
\Delta(T) &\geq C_\alpha(y_{uw} + y_{vw} - y_{uvw})y_{uv} + 2c(y_{uw} + y_{uv} - y_{uw}y_{uv}) + 2b(y_{vw} + y_{uv} - y_{vw}y_{uv}) \\
&= C_\alpha(2 - b - c - y)(1 - a) + 2c + 2b - 4abc.
\end{aligned}$$

For the difference we get,

$$\begin{aligned}
\Delta(T) - \text{cost}(T) &\geq C_\alpha(2 - b - c - y)(1 - a) + 3c + 3b + 2a - 4abc - y - 4 + 2(2 - b - c)(1 - a) \\
&= 2C_\alpha + (1 - C_\alpha)b + (1 - C_\alpha)c + (-C_\alpha - 1 + C_\alpha a)y + (-2C_\alpha - 2)a + (C_\alpha + 2)ab + (C_\alpha + 2)ac - 4abc.
\end{aligned}$$

Define the previous line to be the function $g(a, b, c, y)$. Our goal is to that that

$$\Delta(T) - \text{cost}(T) \geq g(a, b, c, y) \geq 0.$$

To minimize g, we can set y as large as possible, since y has a negative coefficient in g. So we have

$$y = \min\{y_{uv}, y_{vw}\} = \min\{1 - a, 1 - c\},$$

because we assumed $b \leq c$ and hence $y_{vw} \leq y_{uw}$.

Case 1: $a \leq b \leq c$. First, we consider the case in which $a \leq b \leq c$. In this case, we can set $y = 1 - c$. So we have

$$\begin{aligned}
g_1(a, b, c) &:= 2C_\alpha + (1 - C_\alpha)(b + c) + (-C_\alpha - 1 + C_\alpha a)(1 - c) + (-2C_\alpha - 2)a + (C_\alpha + 2)ab + (C_\alpha + 2)ac - 4abc \\
&= (1 - C_\alpha)b + (C_\alpha - 1) + (2 - C_\alpha a)c + (-C_\alpha - 2)a + (C_\alpha + 2)ab + (C_\alpha + 2)ac - 4abc \\
&= (1 - C_\alpha)b + (C_\alpha - 1) + 2c + (-C_\alpha - 2)a + (C_\alpha + 2)ab + 2ac - 4abc.
\end{aligned}$$

Claim 41. *g_1 is increasing in c.*

Proof. The derivative of g_1 with respect to c is

$$2 + 2a - 4ab \geq 2 + 2a - 4a = 2(1 - a) \geq 0.$$

Here, we used that $b \leq 1$. ◊

So we can make c as small as possible and we thus set $c = b$.

$$g_1(a, b, c) \geq g_1(a, b, b) = (3 - C_\alpha)b + (C_\alpha - 1) + (-C_\alpha - 2)a + (C_\alpha + 4)ab - 4ab^2.$$

Claim 42. *$g_1(a, b, b)$ is decreasing with a.*

Proof. The derivative is

$$-C_\alpha - 2 + C_\alpha b + 4b - 4b^2.$$

The above quadratic is maximized for $b = \frac{C_\alpha + 4}{8}$,

$$-C_\alpha - 2 + C_\alpha b + 4b - 4b^2 \leq \frac{1}{16}(C_\alpha^2 - 8C_\alpha - 16) \leq 0$$

◊

Now we can increase a as much as possible and set $a = b$,

$$g_1(b, b, b) = (C_\alpha - 1) + (1 - 2C_\alpha)b + (C_\alpha + 4)b^2 - 4b^3. \tag{30}$$

Claim 43. *$g_1(b, b, b)$ is decreasing in b.*

Proof. The derivative is,

$$1 - 2C_\alpha + (2C_\alpha + 8)b - 12b^2.$$

The above quadratic is maximized for $b = \frac{C + 4}{12}$,

$$1 - 2C_\alpha + (2C_\alpha + 8)b - 12b^2 \leq \frac{1}{12}(C_\alpha^2 - 16C_\alpha + 28) \leq 0$$

. ◊

Thus, we have,

$$g_1(b, b, b) \geq g_1(1, 1, 1) = 0$$

Case 2: $b \leq a \leq c$**.** Our goal is to show that in this case, $g_1(a, b, c) \geq 0$. Claim 41 still applies. When we decrease c as much as possible, we have $c = a$. So we have

$$g_1(a, b, c) \geq g_1(a, b, a) = g_2(a, b) := (1 - C_\alpha)b + (C_\alpha - 1) - C_\alpha a + (C_\alpha + 2)ab + 2a^2 - 4a^2 b.$$

Claim 44. *g_2 is decreasing in b.*

Proof. The derivative is

$$1 - C_\alpha + (C_\alpha + 2)a - 4a^2.$$

The above quadratic is maximized for $a = \frac{C_\alpha + 2}{8}$,

$$1 - C_\alpha + (C_\alpha + 2)a - 4a^2 \leq \frac{1}{16}(C_\alpha^2 - 12C_\alpha + 20) \leq 0.$$

◊

So we increase b as much as possible, setting $b = a$. Then we have

$$g_2(a, b) \geq g_2(a, a) = g_3(a) := a + (C_\alpha - 1) - 2C_\alpha a + (C_\alpha + 4)a^2 - 4a^3.$$

Claim 45. *g_3 is decreasing in a.*

Proof. The derivative is

$$1 - 2C_\alpha + 2(4 + C_\alpha)a - 12a^2.$$

The above quadratic is maximized for $a = \frac{C_\alpha+4}{12}$,

$$1 - 2C_\alpha + 2(4 + C_\alpha)a - 12a^2 \leq \frac{1}{12}(C_\alpha^2 - 16C_\alpha + 28) \leq 0.$$

◊

By Claim 45, we have $g_3(a) \geq g_3(1)$, and we have that $g_3(1) = 0$.

Case 3: $b \leq c \leq a$**.** Recall that our goal is to prove the following quantity is ≥ 0.

$$2C_\alpha + (1 - C_\alpha)b + (1 - C_\alpha)c + (-C_\alpha - 1 + C_\alpha a)y + (-2C_\alpha - 2)a + (C_\alpha + 2)ab + (C_\alpha + 2)ac - 4abc.$$

In this case, we can set $y = 1 - a$, so we have

$$C_\alpha - 1 + (1 - C_\alpha)b + (1 - C_\alpha)c - a - C_\alpha a^2 + (C_\alpha + 2)ab + (C_\alpha + 2)ac - 4abc. \tag{31}$$

Claim 46. (31) *is decreasing with a.*

Proof. The derivative w.r.t. a is

$$-1 - 2C_\alpha a + (C_\alpha + 2)(b + c) - 4bc. \tag{32}$$

The coefficient of a in 32 is $-2C_\alpha \leq 0$. Similar, the coefficient of b in 32 is $C + 2 - 4c \geq C - 2 \geq 0$. Thus, we can assume that $a = b = c$,

$$(32) \leq -1 + 4c - 4c^2 = -(2c - 1)^2 \leq 0.$$

◊

Thus, to minimize (31), we can make a as large as possible, setting $a = \min\{b + c, 1\}$. Let us consider the first subcase in which $b + c > 1$ and so $a = 1$. In this case, (31) becomes

$$-2 + 3(b + c) - 4bc = -(2b - 1)(2c - 1) + b + c - 1 \geq -(2b - 1)(2c - 1). \tag{33}$$

Note that $c \geq \frac{1}{2}$ since $b + c \geq 1$ and $c \leq b$. Thus 33 is decreasing with b,

$$-(2b - 1)(2c - 1) \geq -(2(1 - c) - 1)(2c - 1) = (1 - 2c)^2 \geq 0$$

The second subcase is when $b + c \leq 1$ and we set $a = b + c$ or equivalently, $c = a - b$. Then (31) becomes

$$C_\alpha - 1 + (4b^2 - C_\alpha)a + (2 - 4b)a^2. \tag{34}$$

Claim 47. (34) *is decreasing with a.*

Proof. The derivative of (34) w.r.t. a is,

$$4b^2 - C_\alpha + 2(2-4b)a.$$

Note that $b \leq \frac{1}{2}$ since $b+c \leq 1$ and $b \leq c$. Thus the derivative is increasing with a and maximized for $a = 1$,

$$4b^2 - C_\alpha + 2(2-4b)a \leq 4b^2 - 8b - C_\alpha + 4.$$

which is decreasing for $b \leq 1$ and thus maximized for $b = \tau$,

$$4\tau^2 - 8\tau - C_\alpha + 4 \leq 0$$

$\Diamond$

By the above claim, (34) is minimized for $a = 1$,

$$C_\alpha - 1 + (4b^2 - C_\alpha)a + (2-4b)a^2 \geq 4b^2 - 4b + 1 = (2b-1)^2 \geq 0.$$

□

6 1.485-Approximation: Computer-assisted proof

In this section, by introducing and solving the *factor-revealing SDP*, we will prove Lemma 10, which is restated as follows,

Lemma 10. *For Algorithm 4, (4) holds for budget functions* $\text{budget}^+ \equiv \text{budget}^+_{1.485}$ *and* $\text{budget}^- \equiv \text{budget}^-_{1.485}$.

Let $\mathcal{T}$ be defined as the set of all possible triangles, represented by tuples of values $(y_{uv}, y_{uw}, y_{vw}, y_{uvw})$ satisfying the condition $y_{uv} \leq y_{uw} \leq y_{vw}$. We consider positive degenerate triangle uv as a $++$+ triangle and negative degenerate triangle uv as a $--$+ triangle, both having the same y value $(y_{uv}, y_{uv}, 1, y_{uv})$. This substitution simplifies our computations, and we will later show that it does not increase the value of $\Delta(T) - \text{cost}(T)$ for degenerate triangles. Consequently, $\mathcal{T}$ includes both degenerate and non-degenerate triangles and it can be used to lower bound $\Delta(T) - \text{cost}(T)$. Given graph G and the cluster LP solution z_S, along with their corresponding y values, we define $\eta_T(y)$ for any triangle $T \in \mathcal{T}$ as the number of such triangles present in graph G.

In Section 5, we showed that, for any triangle $T \in \mathcal{T}$, the inequality $\Delta(T) - \text{cost}(T) \geq 0$ holds. In this section, we show, for any solution y, it is possible to determine a reduced budget such that the summation $\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T)) \geq 0$. The key insight is that $\eta_T(y)$ for a given solution y cannot take arbitrary values. In this section, we explain how stronger constraints can be imposed on $\eta_T(y)$, enabling the use of a smaller budget for $\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T))$.

Here, we first outline the constraint we intend to impose on η_T. Subsequently, we will employ discretization techniques to encompass all possible triangles T. We will then formulate a semidefinite program to demonstrate that it is possible to achieve $\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T)) \geq 0$ within a reduced budget.

Covariance Constraint. Given cluster LP solution z_S and y, for each node u, the covariance matrix $\text{COV}_u \in \mathbb{R}^{V \times V}$ where

$$\text{COV}_u(v, w) = y_{uvw} - y_{uv}y_{uw}$$

is positive semidefinite (PSD). Actually, we can even show a stronger version of the covariance constraint.

Lemma 48. *We define* $y_S = \sum_{S' \supseteq S} z_{S'}$ *for every* $S \subseteq V$. *For any* $T \subseteq V, T \neq \emptyset$, *the matrix* $M \in \mathbb{R}^{V \times V}$ *where* $M_{uv} = y_{T \cup \{u,v\}} - y_{T \cup \{u\}}y_{T \cup \{v\}}$ *is PSD.*

Proof. Fix any vector $a \in \mathbb{R}^V$. For every $S \subseteq V$, we define $a(S) := \sum_{v \in S} a_v$. Then

$$\begin{aligned}
a^\intercal M a &= \sum_{u,v \in V} (y_{T\cup\{u,v\}} - y_{T\cup\{u\}} y_{T\cup\{v\}}) a_u a_v \\
&= \sum_{u,v \in V} a_u a_v \left(\sum_{S \supseteq T\cup\{u,v\}} z_S - \Big(\sum_{S \supseteq T\cup\{u\}} z_S \Big) \Big(\sum_{S' \supseteq T\cup\{v\}} z_{S'} \Big) \right) \\
&= \sum_{S \supseteq T} z_S \cdot a(S) \cdot a(S) - \sum_{S \supseteq T} \sum_{S' \supseteq T} z_S \cdot z_{S'} \cdot a(S) \cdot a(S') \\
&\geq \sum_{S \supseteq T, S' \supseteq T} z_S z_{S'} \cdot a^2(S) - \sum_{S \supseteq T} \sum_{S' \supseteq T} z_S \cdot z_{S'} \cdot a(S) \cdot a(S') \\
&= \sum_{S \supseteq T, S' \supseteq T} z_S z_{S'} \left(\frac{1}{2} a^2(S) + \frac{1}{2} a^2(S') - a(S) a(S') \right) \geq 0.
\end{aligned}$$

The first inequality used that $\sum_{S' \supseteq T} z_{S'} \leq 1$ as $T \neq \emptyset$. Therefore, M is PSD. □

Lemma 48 places a significant constraint on the distribution of triangles. We want to mention that the covariance matrix is implied by cluster LP, so we do not need to add extra constraints to the linear program. To obtain Lemma 48, we only require that the solution is from cluster LP.

To leverage the property of the covariance matrix being positive semidefinite, we begin by partitioning the interval $[0, 1]$ into numerous subintervals denoted as $I_1, I_2, ..., I_t$ where each subinterval $I_j \in [0, 1]$ is continuous. Given any $y_{uv} \in [0, 1]$, we use $I(y_{uv})$ to represent the interval containing y_{uv}. Let $l(I_j)$ and $r(I_j)$ be the left and right boundary of interval I_j. Given a node u, consider the matrix $Q_u \in \mathbb{R}^{t \times t}$, where $Q_u(I_j, I_k)$ is defined as

$$Q_u(I_j, I_k) = \sum_{\substack{y_{uv} \in I_j, y_{uw} \in I_k \\ v \in V, w \in V}} y_{uvw} - y_{uv} y_{uw}.$$

When $v = u$ or $w = u$, we have $y_{uvw} - y_{uv} y_{uw} = 0$ and Q_u remains well-defined. We first show that Q_u is also PSD.

Lemma 49. *Let Q_u be the matrix defined above, then $Q_u \succeq 0$.*

Proof. For any given vector $a \in \mathbb{R}^t$, let $a(I_j)$ be the value for index I_j.

$$\begin{aligned}
a^\intercal \cdot Q_u \cdot a &= \sum_{I_j, I_k} a(I_j) Q_u(I_j, I_k) a(I_k) \\
&= \sum_{I_j, I_k} \sum_{y_{uv} \in I_j, y_{uw} \in I_k} a(I_j) a(I_k) (y_{uvw} - y_{uv} y_{uw}) \\
&= \sum_{v,w} \sum_{I_j \ni y_{uv}, I_k \ni y_{uw}} a(I_j) a(I_k) (y_{uvw} - y_{uv} y_{uw}) \\
&= \sum_{v,w} \sum_{y_{uv}, y_{uw}} a(I(y_{uv})) a(I(y_{uw})) (y_{uvw} - y_{uv} y_{uw}) \\
&= b^T \cdot \mathrm{COV}_u \cdot b \geq 0
\end{aligned}$$

where $b \in \mathbb{R}^n$ and each entry of b is defined as $b(v) = a(I(y_{uv}))$. The last inequality is using the fact that the covariance matrix is PSD. □

Given a cluster LP solution y, let $Q(y) = \sum_u Q_u$ be the sum of Q_u for all vertices u in the set V. According to Lemma 49, we can assert that $Q(y) \succeq 0$.

We can now express the matrix $Q(y)$ using triangles $T \in \mathcal{T}$ and the corresponding counts $\eta_T(y)$. To achieve this, we need to consider the increase caused by T in the matrices Q_u, Q_v, and Q_w.

Specifically, a non-degenerate triangle T increases the entries $Q_u(I(y_{uv}), I(y_{uw}))$ and $Q_u(I(y_{uw}), I(y_{uv}))$ by $y_{uvw} - y_{uv}y_{uw}$ for the matrix Q_u,. Similar computations are performed for Q_v and Q_w. For degenerate triangle $T = (y_{uv}, y_{uv}, 1, y_{uv})$, it increases $Q_u(I(y_{uv}), I(y_{uv}))$ by $(y_{uv} - y_{uv}^2)$, increases $Q_v(I(y_{uv}), I(y_{uv}))$ by $(y_{uv} - y_{uv}^2)$ and does not affect Q_w. In either case, we increase $Q(I(y_{uv}), I(y_{uw}))$ and $Q_u(I(y_{uw}), I(y_{uv}))$ by $y_{uvw} - y_{uv}y_{uw}$, $Q(I(y_{vu}), I(y_{vw}))$ and $Q(I(y_{vw}), I(y_{vu}))$ by $y_{uvw} - y_{vu}y_{vw}$, $Q(I(y_{wu}), I(y_{wv}))$ and $Q(I(y_{wv}), I(y_{wu}))$ by $y_{uvw} - y_{wu}y_{wv}$.

More formally, to compute $Q(y)$. Let $C(T) \in \mathbb{R}^{6\times 6}$ be matrix that

$$C(T) = diag(C^{(1)}(T), C^{(2)}(T), ..., C^{(6)}(T))$$

where

$$\begin{aligned} C^{(1)}(T) = C^{(2)}(T) &= \mathrm{COV}_u(v, w) = y_{uvw} - y_{uv}y_{uw}, \\ C^{(3)}(T) = C^{(4)}(T) &= \mathrm{COV}_v(u, w) = y_{uvw} - y_{vu}y_{vw}, \\ C^{(5)}(T) = C^{(6)}(T) &= \mathrm{COV}_w(v, u) = y_{uvw} - y_{wv}y_{wu}. \end{aligned}$$

Let $Head(T) \in \mathbb{R}^{6\times t}$ be a binary matrix, defined as $Head(T)(i, I(a)) = 1$ when

$$(i, a) \in \{(1, y_{uv}), (2, y_{uw}), (3, y_{vu}), (4, y_{vw}), (5, y_{wu}), (6, y_{wv})\},$$

and 0 otherwise. Similarly, we define $Tail(T) \in \mathbb{R}^{6\times t}$ as a binary matrix, defined as $Tail(T)(i, I(a)) = 1$ when

$$(i, a) \in \{(1, y_{uw}), (2, y_{uv}), (3, y_{vw}), (4, y_{vu}), (5, y_{wv}), (6, y_{wu})\}.$$

We can then express $Q(y)$ as

$$Q(y) = \sum_{T\in\mathcal{T}} \eta_T(y) \cdot \left(Head^{\intercal}(T) \cdot C(T) \cdot Tail(T)\right).$$

Let $\mathcal{T}(I_i, I_j, I_k) = \{(y_{uv}, y_{uw}, y_{vw}, y_{uvw}) \in \mathcal{T} \mid y_{uv} \le y_{uw} \le y_{vw}, y_{uv} \in I_i, y_{vw} \in I_j, y_{uv} \in I_k\}$ be the set of triangles within the intervals I_i, I_j, I_k. For any two triangles T with $(y_{uv} = a, y_{uw} = b, y_{vw} = c)$ and T' with $(y_{uv} = a', y_{uw} = b', y_{vw} = c')$, if T and T' are located within the same interval, meaning $I(a) = I(a')$, $I(b) = I(b')$, and $I(c) = I(c')$, then we have $Head(T) = Head(T')$ and $Tail(T) = Tail(T')$. Given I_i, I_j, I_k, define $Head(I_i, I_j, I_k) = Head(T)$ (or $Tail(I_i, I_j, I_k) = Tail(T)$), where T is an arbitrary triangle such that $T \in \mathcal{T}(I_i, I_j, I_k)$. We can then express $Q(y)$ as

$$Q(y) = \sum_{I_i, I_j, I_k} Head^{\intercal}(I_i, I_j, I_k) \cdot \left(\sum_{T\in\mathcal{T}(I_i, I_j, I_k)} \eta_T(y) \cdot C(T)\right) \cdot Tail(I_i, I_j, I_k).$$

Since $Q(y) \succeq 0$, it provides a constraint for $\eta_T(y)$. Our target is to show that $\sum_T \eta_T(y)\cdot(\Delta(T) - \mathrm{cost}(T)) \ge 0$ under the constraint that $Q(y) \succeq 0$. We will consider all possible triangles in each range $\mathcal{T}(I_i, I_j, I_k)$ and there will be at most $O(t^3)$ different ranges. For any solution y, the triangle $T = \{(y_{uv}, y_{uw}, y_{vw}, y_{uvw}) \mid y_{uv} \le y_{uw} \le y_{vw}\}$ satisfies the triangle inequality, that is $1 - y_{uw} + 1 - y_{vw} \ge 1 - y_{uv}$. Therefore, given range I_i, I_j, I_k, we assume that $l(I_i) \ge l(I_j) + l(I_k) - 1$, $r(I_j) \le 1 + l(I_i) - l(I_k)$ and $r(I_k) \le 1 + l(I_i) - l(I_j)$, since there will not be any triangles in the range when the above inequality does not hold. If necessary, we can truncate intervals to ensure that these inequalities hold true.

In each range I_i, I_j, I_k, we will use $\tilde{d}(I_i, I_j, I_k) = \min_{T\in\mathcal{T}(I_i, I_j, I_k)}(\Delta(T) - \mathrm{cost}(T))$ to establish a lower bound on $\Delta(T) - \mathrm{cost}(T)$. $\tilde{d}(I_i, I_j, I_k)$ can be precomputed and there will be at most $O(t^3)$ different $\tilde{d}$ value. We first show that $\tilde{d}(I_i, I_j, I_k)$ is a lower bound even for degenerate triangle. Recall that we treat degenerate triangle as triangle with y value $(y_{uv}, y_{uv}, 1, y_{uv})$.

Lemma 50. *Let $T = (u, v)$ be a degenerate triangle, for any budget function b^+, b^- such that $b^+(0) = 0$, then*

$$\text{cost}(u, v) - \Delta(u, v) \geq \tilde{d}(I(y_{uv}), I(y_{uv}), I(1))$$

Proof. Let T_1 represent the $+++$ triangle with $(y_{uv}, y_{uv}, 1, y_{uv})$. If (u, v) is a positive edge and $y_{uv} \geq 2/3$, then $\text{cost}(u, v) = 0$ and $\Delta(u, v) = 2 \cdot \text{budget}^+(x_{uv})$. In this case, $\text{cost}(T_1) = 0$ and $\Delta(T_1) = 2 \cdot \text{budget}^+(x_{uv}) + \text{budget}^+(0)$. If (u, v) is a positive edge and $y_{uv} < 2/3$, we have $\text{cost}(u, v) = 2(1 - y_{uv})$ and $\Delta(u, v) = 2 \cdot \text{budget}^+(x_{uv})$. For T_1 in this scenario, $\text{cost}(T_1) = 2(1-y_{uv})$ and $\Delta(T_1) = 2 \cdot \text{budget}^+(x_{uv}) + \text{budget}^+(0) y_{uv}$. In either case, we observe that $\Delta(u, v) - \text{cost}(u, v) = \Delta(T_1) - \text{cost}(T_1) \geq \tilde{d}(I(y_{uv}), I(y_{uv}), I(1))$.

Let T_2 be the $--+$ triangle with $(y_{uv}, y_{uv}, 1, y_{uv})$. If (u, v) is a negative edge, then $\text{cost}(u, v) = 2y_{uv}$ and $\Delta(u, v) = 2 \cdot \text{budget}^-(x_{uv})$. For T_2, we have $\text{cost}(T_2) = 2y_{uv} + 2y_{uv} - 2y_{uv}^2$ and $\Delta(T_2) = 2 \cdot \text{budget}^-(x_{uv}) + b^+(0)(2y_{uv} - y_{uv}^2)$. Therefore, we conclude that $\Delta(u, v) - \text{cost}(u, v) \geq \Delta(T_2) - \text{cost}(T_2) \geq \tilde{d}(I(y_{uv}), I(y_{uv}), I(1))$. □

We still need to represent Q using η_T. The problem is there might be infinite types of triangles in each range $\mathcal{T}(I_i, I_j, I_k)$, and we are not able to list every possible triangle. To solve this problem, our last observation is that, for $i \in [1, 6]$, $C^{(i)}(T)$ is a multilinear function in terms of y_{uv}, y_{uw}, y_{vw} and y_{uvw}.

Since $C^{(i)}(T)$ is multilinear, we only need to consider 16 different triangles in each range to cover all possible triangles, where those triangles' y value are

$$y_{uv} \in \{l(I_i), r(I_i)\}, y_{uw} \in \{l(I_j), r(I_j)\}, y_{vw} \in \{l(I_k), r(I_k)\}$$
$$y_{uvw} \in \{\max\Big(0, l(I_i) + l(I_j) - 1, l(I_j) + l(I_k) - 1, l(I_i) + l(I_k) - 1\Big), \min\Big(r(I_i), r(I_j), r(I_k)\Big)\}.$$

The y_{uvw} constraints comes from the fact that $y_{uvw} \leq y_{vw}$ and $y_{uw} + y_{uv} - y_{uvw} \leq 1$. Each triangle might be considered in different ranges multiple times and we shall treat them as different types of triangles. Let $\mathcal{T}_\mathcal{D}(I_i, I_j, I_k)$ be the set of triangles containing this 16 triangles in $\mathcal{T}(I_i, I_j, I_k)$ and $\mathcal{T}_\mathcal{D} = \cup_{I_i, I_j, I_k} \mathcal{T}_\mathcal{D}(I_i, I_j, I_k)$, we can set up a semidefinite programming to lower bound $\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T))$. In factor-revealing SDP, the variables are η_T that represent the ratio of triangles $T \in \mathcal{T}_\mathcal{D}$ and $N \cdot \eta_T$ is the number of triangle of T, where $N = \frac{n(n-1)(n-2)}{6} + \frac{n(n-1)}{2} = \frac{n(n-1)(n+1)}{6}$. $\tilde{d}(I_i, I_j, I_k)$, $Head^\intercal(I_i, I_j, I_k)$, $C(T)$ and $Tail(I_i, I_j, I_k)$ are constant and can be precomputed.

$$\min \quad \sum_{I_i, I_j, I_k} \sum_{T \in \mathcal{T}_\mathcal{D}(I_i, I_j, I_k)} \eta_T \cdot \tilde{d}(I_i, I_j, I_k) \qquad \text{s.t.} \qquad \text{(factor-revealing SDP)}$$

$$Q = \sum_{I_i, I_j, I_k} Head^\intercal(I_i, I_j, I_k) \left(\sum_{T \in \mathcal{T}_\mathcal{D}(I_i, I_j, I_k)} \eta_T \cdot C(T) \right) \cdot Tail(I_i, I_j, I_k) \succeq 0$$
$$\eta_T \geq 0 \qquad \forall T \in \mathcal{T}_\mathcal{D}$$
$$\sum_{T \in \mathcal{T}_\mathcal{D}} \eta_T = 1,$$

Our main theorem regarding the SDP program is that

Theorem 51. *For any cluster LP solution z_S and y , let OPT_{SDP} be the optimal solution for factor-revealing SDP, then*

$$\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T)) \geq N \cdot \text{OPT}_{\text{SDP}}$$

where $N = \frac{n(n-1)(n+1)}{6}$ is the number of degenerate and non-degenerate triangles.

Proof. Note that $\tilde{d}(I_i, I_j, I_k)$ is the lower bound of $\Delta(T) - \text{cost}(T)$ for any $T \in \mathcal{T}(I_i, I_j, I_k)$. Hence, we have

$$\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T)) \geq \sum_{I_i, I_j, I_k} \sum_{T \in \mathcal{T}(I_i, I_j, I_k)} \eta_T(y) \cdot \tilde{d}(I_i, I_j, I_k).$$

The difficulty of the proof arises from the fact that we only consider triangles from $\mathcal{T}_{\mathcal{D}}(I_i, I_j, I_k)$, which contains only 16 triangles. We will show that we can use triangles from $\mathcal{T}_{\mathcal{D}}$ to represent Q and the objective value is at most $\sum_{I_i, I_j, I_k} \sum_{T \in \mathcal{T}(I_i, I_j, I_k)} \eta_T(y) \cdot \tilde{d}(I_i, I_j, I_k)$. For any triangle T with $(y_{uv} \in I_i, y_{uw} \in I_j, y_{vw} \in I_k)$ and $l(I_i) \leq l(I_j) \leq l(I_k)$, let $T_1, T_2, ..., T_{16}$ be the 16 triangles from $\mathcal{T}_{\mathcal{D}}(I_i, I_j, I_k)$. We will later show that there exist $\lambda_i \geq 0$ such that $C(T) = \sum_{i \in [1,16]} \lambda_i \cdot C(T_i)$ and $\sum_{i \in [1,16]} \lambda_i = 1$. Therefore, we can use $T_1, T_2, ..., T_{16}$ to replace T without affecting the Q value. As we use $\tilde{d}(I_i, I_j, I_k)$ in the objective function and $\sum_{i \in [1,16]} \lambda_i = 1$, this substitution does not impact the objective value either.

Combining these findings, for any solution y in cluster LP, we can deduce the existence of a feasible solution η_T from factor-revealing SDP such that $\sum \eta_t = 1$ and $N \cdot Q(\eta_T) = Q(y)$. Therefore, we have

$$\begin{aligned}\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T)) &\geq \sum_{I_i, I_j, I_k} \sum_{T \in \mathcal{T}(I_i, I_j, I_k)} \eta_T(y) \cdot \tilde{d}(I_i, I_j, I_k) \\ &\geq \sum_{I_i, I_j, I_k} \sum_{T \in \mathcal{T}_{\mathcal{D}}(I_i, I_j, I_k)} N \cdot \eta_T \cdot \tilde{d}(I_i, I_j, I_k) \geq N \cdot \text{OPT}_{\text{SDP}}.\end{aligned}$$

We still need to demonstrate the existence of $\lambda_i \geq 0$ such that $C(T) = \sum_{i \in [1,16]} \lambda_i \cdot C(T_i)$ and $\sum_{i \in [1,16]} \lambda_i = 1$. For any triangle $T = (y_{uv}, y_{uw}, y_{vw}, y_{uvw})$, consider triangle $T_1 = (l(I_i), y_{uw}, y_{vw}, y_{uvw})$ and $T_2 = (r(I_i), y_{uw}, y_{vw}, y_{uvw})$ with $\lambda = \frac{y_{uv} - l(I_i)}{r(I_i) - l(I_i)}$. We then have $C^{(i)}(T) = (1 - \lambda) C^{(i)}(T_1) + \lambda C^{(i)}(T_2)$ for $i \in [1, 6]$. We can apply the same transformation to y_{uw}, y_{vw}, y_{uvw} since $C^{(i)}(T)$ is multilinear. This completes the proof. □

Theorem 51 provides a valuable method for establishing a lower bound on $\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T))$ for any given solution and budget function. To proceed with the computation of this lower bound, we need to solve factor-revealing SDP. It is important to emphasize that solving factor-revealing SDP is completely independent of cluster LP. The value of OPT_{SDP} is determined solely by the chosen budget function and the selected intervals $I_1, I_2, ..., I_t$.

The estimation errors, which is the difference between $\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T))$ and OPT_{SDP}, mainly arise from two sources. First, we use $\tilde{d}(I_i, I_j, I_k)$ as an estimated value for all triangles within $T_D(I_i, I_j, I_k)$. Second, some of the triangles we consider may violate the triangle inequality, and there's the possibility that y_{uvw} could violate its constraint as defined in cluster LP.

Given the budget function, the accuracy of our estimation improves with a greater number of chosen intervals, but this also leads to longer computation times for solving OPT_{SDP}. Additionally, to enhance the accuracy of computing OPT_{SDP}, we employ two key techniques. Firstly, we introduce an additional SDP constraint, and secondly, we discretize the values of y_{uvw}.

Frequency Constraint. We can set up another constraint for η_T. Similarly, given a node u, consider the frequency matrix

$$F_u(I_j, I_k) = |\{v \in V, w \in V \mid y_{uv} \in I_j, y_{uw} \in I_k\}| = \sum_{y_{uv} \in I_j, y_{uw} \in I_k} 1.$$

$F_u(I_j, I_k)$ is the number of pairs such that $y_{uv} \in I_j$ and $y_{uw} \in I_k$. We want to mention v and w might be u. Let $freq \in \mathbb{R}^t$ such that $freq(I_j) = |\{v \in V \mid y_{uv} \in I_j\}|$, then we can represent F_u as

$$F_u = freq^{\intercal} \cdot freq$$

which implies F_u is PSD, too. We will follow the discretization strategy of Q matrix. Let $F = \sum_u F_u$ and $A \in \mathbb{R}^{6\times 6}$ is defined as

$$A = diag(1, 1, 1, 1, 1, 1).$$

Then the increase of F causing by T is

$$Head^{\intercal}(T) \cdot A \cdot Tail(T).$$

It is worth noting this increase holds even for degenerate triangles. For any degenerate triangle (u, v), it raises $F(I(y_{uv}), I(y_{uv}))$, $F(I(y_{uv}), I(1))$, and $F(I(1), I(y_{uv}))$ by 2, echoing the increase seen in the triangle with the y value $(y_{uv}, y_{uv}, 1, y_{uv})$. However, unlike the Q_u vector, F_u also considers cases where $v = u$ and $w = u$. In the Q_u vector, this case does not impact Q_u due to $y_{uuu} - y_{uu}y_{uu} = 0$. We need to increment $F(I(1), I(1))$ by n since $\eta(y)$ does not account for $v = u$ and $w = u$. Fortunately, we already include the $+++$ triangle $T_3 = (1, 1, 1, 1)$; given that T_3 does not influence the matrix Q value and $\Delta(T_3) - \text{cost}(T_3) = 0$, each occurrence of T_3 raises $F(I(1), I(1))$ by 6. By increasing $\eta_{T_3}(y)$ by $n/6$, we can effectively represent F as:

$$F(y) = \sum_{I_i, I_j, I_k} Head^{\intercal}(I_i, I_j, I_k) \left(\sum_{T \in \mathcal{T}(I_i, I_j, I_k)} \eta_T(y) \cdot A \right) \cdot Tail(I_i, I_j, I_k).$$

Given that A is a constant matrix, any T can be utilized to encompass all triangles from $\mathcal{T}(I_i, I_j, I_k)$ for F.

Discretization for y_{uvw}. Another technique that enhances estimation accuracy involves discretizing the y_{uvw} values. We achieve this by subdividing the interval $[max(0, (l(I_i) + l(I_j) + l(I_k) - 1)/2), r(I_k)]$ into numerous subintervals. The specific manner in which we partition this interval depends on both the interval's length and its significance in the overall estimation.

However, it is crucial to note that discretizing y_{uvw} results in a substantial increase in the number of triangles to be considered. This surge may render the final SDP impractical to solve. The formulation of our ultimate factor-revealing SDP is provided below.

$$\min \sum_{I_i, I_j, I_k, I_l} \sum_{T \in \mathcal{T}_{\mathcal{D}}(I_i, I_j, I_k, I_l)} \eta_T \cdot \tilde{d}(I_i, I_j, I_k, I_l) \qquad \text{s.t.} \qquad \text{(factor-revealing SDP)}$$

$$Q = \sum_{I_i, I_j, I_k} Head^{\intercal}(I_i, I_j, I_k) \left(\sum_{I_l, T \in \mathcal{T}_{\mathcal{D}}(I_i, I_j, I_k, I_l)} \eta_T \cdot C(T) \right) \cdot Tail(I_i, I_j, I_k) \succeq 0$$

$$F = \sum_{I_i, I_j, I_k} Head^{\intercal}(I_i, I_j, I_k) \left(\sum_{I_l, T \in \mathcal{T}_{\mathcal{D}}(I_i, I_j, I_k, I_l)} \eta_T \cdot A \right) \cdot Tail(I_i, I_j, I_k) \succeq 0$$

$$\eta_T \geq 0 \qquad \forall T \in \mathcal{T}_{\mathcal{D}}$$

$$\sum_{T \in \mathcal{T}_{\mathcal{D}}} \eta_T = 1.$$

where $\tilde{d}(I_i, I_j, I_k, I_l) = min_{T \in \mathcal{T}(I_i, I_j, I_k, I_l)}(\Delta(T) - \text{cost}(T))$, and $\mathcal{T}_{\mathcal{D}}(I_i, I_j, I_k, I_l)$ is the set of triangles such that

$$y_{uv} \in \{l(I_i), r(I_i)\}, y_{uw} \in \{l(I_j), r(I_j)\}, y_{vw} \in \{l(I_k), r(I_k)\}, y_{uvw} \in \{l(I_l), r(I_l)\}.$$

By appropriately setting intervals, we can demonstrate that OPT_{SDP} remains non-negative, as affirmed in Lemma 52, even for a reduced budget function. The primary challenge lies in achieving a balance between estimation error and the running time of solving factor-revealing SDP in practice. For a more in-depth explanation of the specific settings of Lemma 52, please refer to the details provided in Appendix B.

Lemma 52. *For Algorithm 4, let* $\text{budget}^+ \equiv \text{budget}^+_{1.485}$ *and* $\text{budget}^- \equiv \text{budget}^-_{1.485}$. *Let* OPT_{SDP} *be the optimum of the factor-revealing SDP factor-revealing SDP below. There exists a discretization of the edge and triple variables into intervals* $I_1, \ldots, I_t$ *(together with an interval grid for* y_{uvw}*) such that* $\text{OPT}_{\text{SDP}} \geq 0$.

Proof of Lemma 10. For budget functions $\text{budget}^+ \equiv \text{budget}^+_{1.485}$ and $\text{budget}^- \equiv \text{budget}^-_{1.485}$, and for degenerate and non-degenerate triangles, based on Theorem 51 and Lemma 52, we know that

$$\sum_T \eta_T(y) \cdot (\Delta(T) - \text{cost}(T)) \geq N \cdot \text{OPT}_{\text{SDP}} = 0.$$

□

7 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 +edges of our gap instance is based on the line graph of a base graph; given a base graph $H = (V_H, E_H)$, our Correlation Clustering instance is $G = (V_G, E_G)$ where $V_G = E_H$ and $e, f \in V_G$ are connected by a +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 [CMM09].

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 53. *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 $35|V_H| \le 35n^2$ 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 $C_{v_1}, \dots, C_{v_t}$ and let $n_i := |C_{v_i}|$ such that $n_1 \ge \cdots \ge n_t$.

Claim 54. $\sum_{i=1}^{t} n_i^2 \le n^3/3$.

Proof. The LHS is monotone in $(n_1, \dots, n_t)$, and if there is an edge $(v_i, v_j) \in C_j$ with $j > i$ (which implies $n_i \ge n_j$), 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 $v_1, \dots, v_{i-1}$. In that case, the LHS is

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

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 $\sum_{i\in[t]} n_i^2/2 \le n^3/6$ 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 $4/3 - o(1)$. □

8 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 $24/23 \ge 1.043$, proving Theorem 5 restated below.

The idea is similar to the gap for the cluster LP in Section 7, 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.

We still 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 [CKL22] that shows that it is hard to distinguish whether H is *nearly bipartite* or close to a random hypergraph.

Theorem 55. *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 $u \in U$, $|\{e \in E_H : e \cap U = \{u\}\}| \ge (1/2 - \varepsilon)d$.*
- *(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) [CKL22] yields the desired hardness. In the following, we highlight the difference between the statement of Theorem 4.1 of [CKL22] and our Theorem 55 and briefly explain how our additional properties are satisfied by their reduction.

1. Regularity of H: Section 4.5 of [CKL22], 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 [Hås01] 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 [Hås01].

□

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 55. 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 $|e \cup f| \leq 5$ vertices). Therefore,

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

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 $|C| \leq (6 + o(1))d$; 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 56. *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 $O(n|C|)$.*

Proof. Without loss of generality, assume $V_H = \{v_1, \dots, v_n\}$ and define $n_i := |C \cap E_{v_i}|$ such that $n_1 \geq \cdots \geq n_n$. 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 $i \leq 20$.

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 $\cup_{j \neq i} E_j$ (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 $c_1 \geq \cdots \geq c_t$.

Claim 57. $\sum_{i=1}^{t} c_i^2 \leq d^2 n(0.2 + O(\sqrt{\varepsilon}))$, *where ε is the parameter from Theorem 55.*

Proof. Here, we use the NO case guarantee from Theorem 55: 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 $i \in [n]$,

$$\sum_{j=1}^{i} c_i \leq (3(i/n) - 3(i/n)^2 + (i/n)^2 + \varepsilon)|E_H|. \tag{35}$$

Let $\delta = o(1)$ be such that every vertex of H has degree at most $(1 + \delta)d$, which means that $(1 + \delta)d \geq c_1 \geq \cdots \geq c_t$. And let $f_{i/n} := c_i/((1 + \delta)d)$. Then (35) becomes

$$\frac{1}{n} \sum_{j=1}^{i} f_{j/n} \leq (3(i/n) - 3(i/n)^2 + (i/n)^2 + \varepsilon) \frac{|E_H|}{(1 + \delta)dn} \leq (3(i/n) - 3(i/n)^2 + (i/n)^2 + \varepsilon)/3. \tag{36}$$

(Note that $|E_H| \leq (1 + \delta)dn/3$.) Interpreting $\frac{1}{n} \sum_{j=1}^{i} f_{j/n}$ as $\int_0^1 f(x)dx$ where $f(x) = c_{\lceil xn \rceil}$, we have that $\sum_{i=1}^{t} |c_i|^2 \leq (1 + \delta)^2 d^2 n \max_f \int_0^1 f(x)^2 dx$, where the maximum is taken over functions $f : [0, 1] \to [0, 1]$ with the constraints that

1. For all $y \in [0, 1]$,
$$\int_{x=0}^{y} f(x)dx \leq y - y^2 + y^3/3 + \varepsilon/3. \tag{37}$$
(Compared to (36), 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 (37) is violated for some value $y \in (i/n, (i + 1)/n)$ for some integer i, (36) 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 $f(0) \leq 1$.

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 $y \leq \tau$ 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 $O(n^2 d)$ 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 $\frac{24}{23} \ge 1.043$.

References

[ACG+15] Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2237–2246, 2015. 5

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):1–27, 2008. 3, 5, 6, 8, 13

[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*, pages 172–181, 2009. 3

[AN23] Sara Ahmadian and Maryam Negahbani. Improved approximation for fair correlation clustering. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 9499–9516, 2023. 5

[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. 3

[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. 5

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1):89–113, 2004. 1, 3

[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. 5

[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. 5

[BGU13] Francesco Bonchi, Aristides Gionis, and Antti Ukkonen. Overlapping correlation clustering. *Knowledge and Information Systems*, 35(1):1–32, 2013. 3

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Proceedings of 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 472–481, 2011. 1, 6

[BS06] Nikhil Bansal and Maxim Sviridenko. The Santa Claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 31–40, 2006. 3

[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. ACM, 2024. 1, 5, 6

[CCMU21] Mélanie Cambus, Davin Choo, Havu Miikonen, and Jara Uitto. Massively parallel correlation clustering in bounded arboricity graphs. In *35th International Symposium on Distributed Computing (DISC)*, volume 209 of *LIPIcs*, pages 15:1–15:18, 2021. 5

[CDK14] Flavio Chierichetti, Nilesh Dalvi, and Ravi Kumar. Correlation clustering in MapReduce. In *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 641–650, 2014. 5

[CGW05] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005. 3, 4, 5, 44

[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. 5

[CKK+06] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006. 5

[CKL22] Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. 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)*, pages 1493–1530, 2022. 12, 38, 39

[CKL+24] Mélanie Cambus, Fabian Kuhn, Etna Lindy, Shreyas Pai, and Jara Uitto. A $(3+\varepsilon)$-Approximate Correlation Clustering Algorithm in Dynamic Streams. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024. 5

[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. 3

[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)*, 2023. 1, 3, 4, 5, 6, 7, 8, 13

[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. 5, 6

[CLMP22] Vincent Cohen-Addad, Silvio Lattanzi, Andreas Maggiori, and Nikos Parotsidis. Online and consistent correlation clustering. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 4157–4179, 2022. 5

[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. 1, 3, 4, 5, 6, 7, 8

[CM23] Sayak Chakrabarty and Konstantin Makarychev. Single-pass pivot algorithm for correlation clustering. Keep it simple! In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 5

[CMM09] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Integrality gaps for Sherali-Adams relaxations. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 283–292, 2009. 12, 37

[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. 1, 3, 5, 6, 8

[CSX12] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2204–2212, 2012. 3

[DEFI06] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006. 5

[dlVM07] Wenceslas Fernandez de la Vega and Claire Kenyon Mathieu. Linear programming relaxations of maxcut. In *Proceedings of the18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 53–61, 2007. 6

[DMN23] Sami Davies, Benjamin Moseley, and Heather Newman. Fast combinatorial algorithms for min max correlation clustering. In *International Conference on Machine Learning*, pages 7205–7230. PMLR, 2023. 5

[FGMS06] Lisa Fleischer, Michel X. Goemans, Vahab S. Mirrokni, and Maxim Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 611–620, 2006. 3

[GG06] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2:249–266, 2006. 5

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. 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)*, pages 482–491, 2011. 1, 6

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 39

[HIA23] Holger Heidrich, Jannik Irmai, and Bjoern Andres. A 4-approximation algorithm for min max correlation clustering. *arXiv preprint arXiv:2310.09196*, 2023. 5

[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. 3

[KS09] Marek Karpinski and Warren Schudy. 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)*, pages 313–322, 2009. 5

[LMV+21] Silvio Lattanzi, Benjamin Moseley, Sergei Vassilvitskii, Yuyan Wang, and Rudy Zhou. Robust online correlation clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4688–4698, 2021. 5

[MSS10] Claire Mathieu, Ocan Sankur, and Warren Schudy. Online correlation clustering. In *Proceedings of 27th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 573–584, 2010. 5

[PPO+15] Xinghao Pan, Dimitris S. Papailiopoulos, Samet Oymak, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. Parallel correlation clustering on big graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 82–90, 2015. 5

[RT12] Prasad Raghavendra and Ning Tan. Approximating CSPs with global cardinality constraints using SDP hierarchies. In *Proceedings of the 23d Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 373–387, 2012. 1, 4, 6, 17

[Swa04] Chaitanya Swamy. Correlation clustering: Maximizing agreements via semidefinite programming. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 526–527, 2004. 4, 44

[Vel22] Nate Veldt. Correlation clustering via strong triadic closure labeling: Fast approximation algorithms and practical lower bounds. In *International Conference on Machine Learning (ICML)*, pages 22060–22083, 2022. 5

[VGW18] Nate Veldt, David F. Gleich, and Anthony Wirth. A correlation clustering framework for community detection. In *Proceedings of the 2018 ACM World Wide Web Conference (WWW)*, pages 439–448, 2018. 5

[YZ14] Yuichi Yoshida and Yuan Zhou. Approximation schemes via sherali-adams hierarchy for dense constraint satisfaction problems and assignment problems. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 423–438, 2014. 6

A Standard SDP Relaxation

The following is a natural SDP relaxation previously used in approximation algorithms for the maximization problem [CGW05, Swa04] augmented with triangle inequality from the standard LP relaxation.

$$\begin{aligned} \min \quad \operatorname{obj}(x) = \sum_{ij \in E^+} x_{ij} + \sum_{ij \in E^-} (1 - x_{ij}) \qquad & \text{(SDP)} \\ x_{ij} &= 1 - v_i \cdot v_j \quad \forall i, j \in V \\ v_i \cdot v_j &\geq 0 \quad \forall i, j \in V \\ v_i \cdot v_i &= 1 \quad \forall i \in V \\ x_{ij} &\leq x_{ik} + x_{jk} \quad \forall i, j, k \in V \\ v_i &\in \mathbb{R}^n \quad \forall i \in V. \end{aligned}$$

Lemma 58. *The integrality gap of* (SDP) *is at least* 1.5.

Proof. Let S_k be a star with k leaves and one center vertex with degree k. Here, the sdp_k value is for the basic relaxation. Then the value $\text{opt}_k = k - 1$. Consider the sdp solution where there are k unit vectors, one for each leaf, each with a 1 in the i^{th} position. The vector corresponding to the center vertex has length k and each entry is $1/\sqrt{k}$. Then the integrality gap is at least

$$\frac{\text{opt}_k}{\text{sdp}_k} = \frac{k-1}{k - \sqrt{k}}.$$

Notice that for $k = 2, 3$, the above vector configuration leads to distances that violate the triangle inequality. Thus the k yielding the largest integrality gap that also does not violate the triangle inequality is $k = 4$, which gives a gap of $3/2$. □

B Detail of Lemma 52

Proof. We use the same factor-revealing SDP framework as in our earlier analysis. For convenience we reproduce it here:

$$\min \sum_{I_i,I_j,I_k,I_\ell} \sum_{T\in\mathcal{T}_{\mathsf{D}}(I_i,I_j,I_k,I_\ell)} \eta_T \cdot \widetilde{d}(I_i,I_j,I_k,I_\ell) \qquad \text{s.t.} \qquad \text{(factor-revealing SDP)}$$

$$Q = \sum_{I_i,I_j,I_k} \mathrm{Head}^\top(I_i,I_j,I_k) \Big(\sum_{I_\ell,\, T\in\mathcal{T}_{\mathsf{D}}(I_i,I_j,I_k,I_\ell)} \eta_T \cdot C(T) \Big) \mathrm{Tail}(I_i,I_j,I_k) \succeq 0,$$

$$F = \sum_{I_i,I_j,I_k} \mathrm{Head}^\top(I_i,I_j,I_k) \Big(\sum_{I_\ell,\, T\in\mathcal{T}_{\mathsf{D}}(I_i,I_j,I_k,I_\ell)} \eta_T \cdot A \Big) \mathrm{Tail}(I_i,I_j,I_k) \succeq 0,$$

$$\eta_T \geq 0 \quad \forall T \in \mathcal{T}_{\mathsf{D}}, \qquad \sum_{T\in\mathcal{T}_{\mathsf{D}}} \eta_T = 1.$$

Discretization. We discretize $[0, 1]$ using the following breakpoints

$$\{0,\ 0.05,\ 0.1,\ 0.2,\ 0.3,\ 0.35,\ 0.38,\ 0.39,\ 0.40,\ 0.405,\ 0.41,\ 0.42,\ 0.44, 0.45,\ 0.5,\ 0.55,\ 0.57,\ 0.58,\ 0.6,\ 0.65,$$
$$,\quad 0.7, 0.75,\ 0.78,\ 0.8,\ 0.9,\ 0.95,\ 0.96,\ 0.99,\ 1\},$$

yielding 28 subintervals. We refine around the rounding thresholds 0.40 and 0.57 to reflect the piecewise structure of the pivot rule.

For triples, when $(I_i, I_j) \subseteq [0.38, 0.65]^2$ we discretize y_{uvw} more finely. For each I_i, I_j, I_k we let

$$I_l \in [\max\Big(0, l(I_i) + l(I_j) - 1, l(I_j) + l(I_k) - 1, l(I_i) + l(I_k) - 1\Big), \min\Big(r(I_i), r(I_j), r(I_k)\Big)].$$

we split I_ℓ into 10 equal subintervals. When $(I_i, I_j) \subseteq [0.38, 0.45]^2$, we split I_ℓ into 20 equal subintervals.

Boundary coalescing. A triangle that lies exactly on interval boundaries can be enumerated in several (I_i, I_j, I_k, I_ℓ) cells (up to 8). To avoid duplicates, we coalesce these copies and retain a single representative: the one with the largest value of $\widetilde{d}$. This produces an upper bound on the true minimum of (factor-revealing SDP); consequently, certifying nonnegativity for the coalesced program is conservative (it may require a slightly larger approximation factor, which we encode in 1.485).

Numerical certificate. With the budgets $\mathrm{budget}^+_{1.485}, \mathrm{budget}^-_{1.485}$ and the rounding in Algorithm 4, the solver returns $\mathrm{OPT}_{\mathrm{SDP}}$ that is nonnegative to numerical precision (empirically within 10^{-8}). By standard smoothing and interval rounding arguments (as in our prior factor-revealing analyses), this yields an exact nonnegativity certificate for (factor-revealing SDP) under the stated discretization, which completes the proof of Lemma 52. □