CORRELATION CLUSTERING WITH CONSTRANDED CLUSTER SIZES AND EXTENDED WEIGHTS BOUNDS

GREGORY J. PULEO AND OLGICA MILENKOVIC

Abstract. We consider the problem of correlation clustering on graphs with constraints on both the cluster sizes and the positive and negative weights of edges. Our contributions are twofold: First, we introduce the problem of correlation clustering with bounded cluster sizes. Second, we extend the region of weight values for which the clustering may be performed with constant approximation guarantees in polynomial time and apply the results to the bounded cluster size problem.

1. Introduction

The correlation clustering problem was introduced by Bansal, Blum, and Chawla [4, 5]. In the most basic form of the clustering model, one is given a set of objects and, for some pairs of objects, one is also given an assessment as to whether the objects are “similar” or “dissimilar”. This information is described using a graph $G$ with labeled edges: each object is represented by a vertex of the graph, and the assessments are represented by edges labeled with either a + (for similar objects) or a − (for dissimilar objects). The goal is to partition the objects into clusters so that the edges within clusters are mostly positive and the edges between clusters are mostly negative. Unlike many other clustering models, such as $k$-means [22], the number of clusters is not fixed ahead of time. Instead, finding the optimal number of clusters is part of the problem. For this reason, correlation clustering has been used in machine learning as a model of “agnostic learning” [32].

In a perfect clustering, all the edges within clusters would be positive and all the edges between clusters would be negative. However, the assessments need not be mutually consistent: for example, if $G$ contains a triangle with two positive edges and one negative edge, then we must either group the endpoints of the negative edge together (erroneously putting a negative edge inside a cluster, resulting in a “negative error”) or else we must group them separately (forcing one of the positive edges to erroneously go between clusters, resulting in a “positive error”). When a perfect clustering is not possible, we seek an optimal clustering: one that minimizes the total number of “errors”.

Correlation clustering is closely related to a number of other graph optimization problems, including the cluster editing problem, introduced by Shamir, Sharan, and Tsur [31]; the graph partitioning problem [3]; and the rank aggregation problem [10].

This work was supported in part by NSF Grants CIF 1218764, CIF 1117980, and STC Class 2010, CCF 0939370, and the Strategic Research Initiative (SRI) Grant conferred by the University of Illinois, Urbana-Champaign. Research of the first author is supported by the IC Postdoctoral Research Fellowship.
In the cluster editing problem, one is given a graph \( H \) and wishes to determine the smallest number of edge insertions and deletions that must be carried out in order to transform \( H \) into a disjoint union of cliques. This problem is equivalent to the special case of correlation clustering where \( G \) is a complete graph: taking \( H \) to be the subgraph of \( G \) consisting of the edges labeled \(+\), we see that the edges added to \( H \) correspond to negative errors in a clustering of \( G \), while the edges removed from \( H \) correspond to positive errors. The variant of cluster editing where we are only allowed to delete edges, rather than adding them, is called the cluster deletion problem \[\text{(31)}\]. Cluster editing has found a number of important applications in bioinformatics, including the analysis of putative gene co-regulation \[\text{(12)}\].

In the graph partitioning problem, one is asked to partition the vertices of a graph \( H \) into a fixed number of parts, minimizing the number of edges between different parts \[\text{(3)}\]. If we again consider a complete graph \( G \) whose positive edges are given by the edges of \( H \), one can view graph partitioning as a variant of correlation clustering where the number of clusters is fixed and only “positive errors” are penalized.

In the rank aggregation problem, and in particular, the feedback arc set problem, one seeks a globally consistent ranking (i.e., permutation) based on possibly inconsistent pairwise comparisons \[\text{(8)}\]. Equivalently, the problem is to compute the centroid of a set of permutations under a suitable ordinal distance measure, usually taken to be the Kendall or weighted Kendall distance \[\text{(15)}\]. Ailon, Charikar, and Newman \[\text{(1, 2)}\] showed that essentially the same algorithm may be used to obtain good constant-factor approximations for both the correlation-clustering problem and the rank-aggregation problem.

Finding optimal solutions to all the aforementioned problems is computationally hard. In particular, optimal correlation clustering is computationally hard even when \( G \) is a complete graph: Bansal, Blum, and Chawla \[\text{(4, 5)}\] proved that it is NP-hard to determine whether a labeled complete graph \( G \) has a clustering with at most \( k \) errors, where \( k \) is an input parameter. This was also proved independently by Shamir, Sharan, and Tsur \[\text{(31)}\] in the context of cluster editing. Shamir, Sharan, and Tsur also proved that for some \( \epsilon > 0 \), the cluster deletion problem is NP-hard to approximate within a factor of \((1 + \epsilon)\). Rank aggregation under the Kendall \(\tau\)-distance was proved to be NP-hard by Bartholdi \[\text{(6)}\]. Research has therefore focused on finding efficient approximation algorithms for this family of problems.

The objective function for correlation clustering can be viewed in two different ways, leading to two different formulations of the problem with different approximability properties. In the MAXAGREE formulation, we seek a clustering that maximizes the number of edges which agree with the clustering. Bansal, Blum, and Chawla \[\text{(4, 5)}\] gave a PTAS for MAXAGREE in the case where \( G \) is a complete graph. On general graphs, MAXAGREE is APX-hard \[\text{(9, 10)}\], although constant-factor approximation algorithms are known \[\text{(9, 11, 22)}\].

In our work, we focus on the MINDISAGREE formulation. Here, we seek a clustering that minimizes the number of edges which disagree with the clustering. MINDISAGREE on general graphs is APX-hard, and the best known approximation ratio is \(O(\log n)\), given by \[\text{(9, 10, 13)}\]. Any improvement on this approximation ratio would also improve the approximation ratio for the minimum multicut problem \[\text{(11)}\]. Thus, finding a constant-factor approximation algorithm for general graphs is likely to be difficult.
However, constant-factor approximation algorithms for \textsc{MinDisagree} have been found for the case where $G$ is a complete graph. In their original paper, Bansal, Blum, and Chawla \cite{Bansal2004} gave a 17+29-approximation algorithm. Charikar, Guruswami, and Wirth \cite{Charikar2005,Guruswami2005} gave a 4-approximation algorithm for the problem, based on a two-stage procedure that rounds the solution of a linear program. Applying a similar algorithm to general graphs yields a $O(\log n)$-approximation \cite{Demaine2004,Fiat2004}; this was also independently proved by Demaine, Emanuel, Fiat, and Immorlica \cite{Demaine2004,Fiat2004,Guruswami2005}. This $O(\log n)$ ratio is in some sense best possible for techniques based on linear programming, since the natural linear programming formulation has integrality gap $\Omega(\log n)$ on general graphs \cite{Demaine13,Immorlica19,Immorlica20}.

Later on, Ailon, Charikar, and Newman \cite{Ailon2005,Ailon2006} introduced a simple randomized 3-approximation algorithm for \textsc{MinDisagree} on complete graphs, i.e., an algorithm where the expected cost of the output is at most 3 times the optimal cost.

Bansal, Blum, and Chawla \cite{Bansal2004} also proposed a weighted version of the correlation-clustering problem, where the edges of the graph $G$ receive \emph{weights} between $-1$ and $1$ rather than simply receiving + or − labels: an edge with weight $x$ incurs cost $\frac{1}{1-x^2}$ if it is placed between clusters and cost $\frac{1}{1-x^2}$ if it is placed within a cluster. Another weighted formulation was considered by Demaine, Emanuel, Fiat, and Immorlica \cite{Demaine2004,Fiat2004,Guruswami2005}: here, the edges are still labeled + or −, but each edge also receives a nonnegative weight $w$, with positive edges costing $w$ when placed between clusters and negative edges costing $w$ when placed within clusters.

Another weighted formulation was considered by Ailon, Charikar, and Newman, and this is the formulation we subsequently consider. In the Ailon–Charikar–Newman model, each edge $e$ is assigned two nonnegative weights, $w_e^+$ and $w_e^-$. A clustering incurs cost $w_e^+$ if $e$ is placed between clusters, and incurs cost $w_e^-$ if $e$ is placed within a cluster. This model includes both the Bansal–Blum–Chawla formulation and the Demaine–Emanuel–Fiat–Immorlica formulation as special cases. The cluster deletion problem can also be represented in this framework, by giving negative edges a prohibitively high value of $w_e^-$. If no restrictions are placed on the weights $w_e^+$ and $w_e^-$, then it is possible to have edges with $w_e^+ = w_e^- = 0$; these edges are effectively absent from the graph, so there is no loss of generality in assuming that $G$ is a complete graph. Since correlation clustering is hard to approximate on general graphs, this suggests that we must place restrictions on $w_e^+$ and $w_e^-$ in order to obtain a class of problems for which a constant-factor approximation is possible.

The \textit{probability constraints}, which we abbreviate to \textit{pc}, give a natural restriction on the edge weights: in this case, we require $w_e^+ + w_e^- = 1$ for every edge $e$. The weight functions satisfying \textit{pc} are exactly the weight functions that can be represented in the Bansal–Blum–Chawla weighting model. Bansal, Blum, and Chawla \cite{Bansal2004} showed that any $k$-approximation algorithm for the unweighted correlation-clustering problem yields a $(2k+1)$-approximation algorithm to the weighted problem under \textit{pc}. In conjunction with the 4-approximation algorithm of Charikar, Guruswami, and Wirth for the unweighted problem, this yields a 9-approximation algorithm for the weighted problem under \textit{pc}. Ailon, Charikar, and Newman obtained a randomized expected 5-approximation algorithm for the weighted problem under \textit{pc}.

Another widely studied restriction is the \textit{triangle inequality} restriction (abbreviated \textit{ti}), where one requires $w_{e_1}^+ + w_{e_2}^+ \leq w_{e_3}^+$ for all $e_1 = uv$, $e_2 = vw$, and $e_3 = uw$. 
where \( u, v, w \in V(G) \). Such constraints arise naturally when correlation clustering is used as a model for clustering and rank aggregation problems. Gionis, Mannila, and Tsaparas [21] obtained a 3-approximation algorithm for correlation clustering under PC and TI, while Ailon, Charikar, and Newman obtained a randomized expected 2-approximation algorithm under the same constraints. Little appears to be known about the approximability of correlation clustering with only TI, without PC or other constraints involved. Indeed, Ailon, Charikar, and Newman leave this approximability question as an open problem.

Previous treatments of the correlation-clustering problem have allowed any clustering to be considered as a feasible solution. However, in some applications there may be restrictions on which clusterings can actually be implemented as solutions. In particular, the application may demand that no cluster should be very large: say, each cluster has size at most \( K + 1 \), for some fixed integer \( K \). Such constraints frequently arise in community detection problems, as described in [33, 29, 15, 28, 7, 30], where one may have a priori information about the largest possible size of a subcommunity. Similar constraints were also considered for clustering models unrelated to correlation clustering by Khandekar, Hildrum, Parekh, Rajan, Sethuraman, and Wolf [25]. The approximation algorithms we present in the next section are able to accommodate such constraints in the correlation-clustering case, as well as accommodating “soft constraints” that merely impose a penalty on the objective function for oversized clusters, without outright forbidding them.

2. Our Contributions

The contributions of our work are two-fold. First, we generalize the 4-approximation algorithm of Charikar, Guruswami, and Wirth [9, 10] in order to handle more general weighted graphs than previously considered. When \( G \) is a complete graph, and when the weight functions satisfy the following constraints:

- \( w_e^+ \leq 1 \) for every edge \( e \), and
- \( w_e^- \leq \tau \) for every edge \( e \), where \( \tau \) is a positive real number, and
- \( w_e^+ + w_e^- \geq 1 \) for every edge \( e \),

our algorithm achieves an approximation ratio of \( 5 - 1/\tau \). Thus, when the weights satisfy the probability constraints \( w_e^+ + w_e^- = 1 \) for all \( e \), the algorithm guarantees a 4-approximation ratio and reduces to the Charikar–Guruswami–Wirth approach. (It does not appear to have been previously recognized that the Charikar–Guruswami–Wirth algorithm, originally stated for unweighted instances, works without modification for weights obeying the probability constraints and gives the same approximation ratio; cf. Table 1 of [2].) In the limit \( \tau \to \infty \) where \( w_e^- \) is allowed to be arbitrarily large, the algorithm still guarantees a constant approximation ratio of 5.

Note, in particular, that the cluster-deletion problem for a graph \( G \) is equivalent to the correlation-clustering problem for the graph \( G' \) where all edges \( e \in E(G) \) satisfy \( w_e^+ = 1, w_e^- = 0 \) and all edges \( e \in E(G) \) satisfy \( w_e^+ = 0, w_e^- = \infty \). Charikar, Guruswami, and Wirth showed that a slight modification of their algorithm gives a 4-approximation for cluster deletion [9, 10]; in contrast, our algorithm gives a 5-approximation but also applies to problems that are intermediate between correlation-clustering and cluster-deletion, where negative errors are permissible but may be costly.
Our second contribution is to extend the correlation-clustering model to allow for bounds on the sizes of the clusters. We assume that there is a “soft constraint” of at most $K + 1$ vertices per cluster. To model the constraint, each vertex $v$ is assigned a non-negative “penalty” $\mu_v$ that it pays for each vertex beyond the $(K + 1)$st that it is clustered with. Thus, the total “penalty cost” paid by a cluster $C$ with $|C| > K$ is given by $(|C| - (K + 1)) \sum_{v \in C} \mu_v$.

If instead a “hard constraint” is desired, it suffices to set all the penalties to $\mu_v = 1$: any clusters in the resulting solution which are too large can then be split arbitrarily into clusters of size $K + 1$ and a “remainder cluster”, yielding no net gain in cost since we assume that $w^+ + e \leq 1$ for all $e$. Similarly, if no size constraint at all is desired, we can take all $\mu_v = 0$.

In addition to the changes mentioned previously, we show that the Charikar–Guruswami–Wirth algorithm can be further modified in order to deal with both the weight and cluster size constraints. Taking $\mu^* = \max_{u \neq v} (\mu_u + \mu_v)$ and assuming that $0 < \mu^* \leq 4$, we obtain an approximation ratio of $2/\alpha$ for the aforementioned clustering scenario, where $\alpha$ is the unique positive solution to the equation

$$\frac{2\alpha \mu^*}{1 - 2\alpha} + \frac{1}{1 - 2\alpha + \frac{\alpha}{\tau}} = \frac{2}{\alpha}.$$

This rearranges to a cubic equation in $\alpha$, which may be solved numerically. However, for some values of $\tau$ and $\mu^*$, the approximation ratio is easy to compute; these values are summarized in Table 1.

| $\mu^*$ | $\tau = 1$ (prob. constraints) | $\tau \in [1, \infty)$ | $\tau \to \infty$ (arbitrarily large $w^-$) |
|---------|---------------------------------|-------------------------|-----------------------------------------|
| $\mu^* = 0$ (no size bound) | 4 | $5 - (1/\tau)$ | $5$ |
| $\mu^* \in (0, 2)$ (soft size bound) | $\star$ | $\star$ | $-5 + \sqrt{25 + 16 \mu^*}$ |
| $\mu^* = 2$ (hard size bound) | 6 | $\star$ | $\approx 6.275$ |

Table 1. Approximation ratios for special parameter values. Starred entries are solutions to a cubic equation.
minimize \[
\sum_{e \in E(G)} \left( w_e^+ x_e + w_e^- (1 - x_e) \right) + \sum_{v \in V(G)} \mu_v y_v.
\]

subject to \[
x_{uv} \leq x_{uz} + x_{zv} \quad \text{(for all distinct } u, v, z \in V(G))
\]
\[
\sum_{v \neq u} (1 - x_{uv}) \leq K + y_v \quad \text{(for all } u \in V(G))
\]
\[
0 \leq x_e \leq 1 \quad \text{(for all } e \in E(G))
\]
\[
0 \leq y_v \quad \text{(for all } v \in V(G))
\]

**Figure 1.** The linear program \( P \).

**Assumption A.** We assume that \( uv \in E(G) \) for every pair of distinct vertices \( u, v \in V(G) \). We also assume that \( \mu_u + \mu_v \leq 4 \) for all distinct vertices \( u, v \); furthermore, for every edge \( e \) we assume that \( w_e^+ \leq 1 \), that \( w_e^- \leq \tau \), and that \( w_e^+ + w_e^- \geq 1 \).

Let \( P \) denote the linear program shown in Figure 1. The integer restriction of \( P \), where \( x_e \in \{0, 1\} \), represents the exact model for our weighted and cluster-size-bounded correlation-clustering problem. We interpret \( x_e = 1 \) as meaning “the endpoints of \( e \) lie in different clusters” and we interpret \( x_e = 0 \) as meaning “the endpoints of \( e \) lie in the same cluster”. The triangle inequality \( x_{uv} \leq x_{uz} + x_{zv} \) models the fact that if two edges \( uz \) and \( zv \) are in the same cluster, then the edge \( uv \) should also belong to the same cluster. The new restriction on cluster sizes is represented by the constraints \( \sum_{v \neq u} (1 - x_{uv}) \leq K + y_v \) together with the penalty term \( \sum_{v \in V(G)} \mu_v y_v \) in the objective function; here, \( y_v \) represents the amount by which the cluster containing \( v \) exceeds the size bound. To simplify notation, we adopt the convention that \( x_{uu} = 0 \) for all \( u \).

Since any actual clustering yields a feasible integer solution to \( P \), the optimal value of \( P \) is a lower bound for the optimal cost of the correlation-clustering problem. The idea of our algorithm is to start with an optimal solution to \( P \) and “round” the solution to produce a clustering. We prove that this rounding process only increases the cost of the solution by a constant multiplicative factor.

Given any feasible solution \((x, y)\) to the linear program, where \( x \) denotes the vector of all edge costs \( x_e \), while \( y \) denotes the vector of all vertex penalties \( y_v \), we produce a clustering \( C \) via the following algorithm, where \( \alpha \) is a parameter to be determined later. Note that the only difference between Algorithm 1 and the second stage of the Charikar–Guruswami–Wirth algorithm is the use of the parameter \( \alpha \) rather than the fixed value of \( 1/2 \). As we subsequently prove, the choice of this threshold allows for extensions in the range of values for the weights of edges that can be accommodated by our algorithm.

We show next that the resulting clustering \( C \) has cost at most \( c_\alpha \text{cost}(x) \), where \( c_\alpha \) is a constant depending only on \( \alpha \) and \( c_\alpha \text{cost}(x) \) denotes the optimal solution of the integer linear program. We follow the outline of the analysis performed by Charikar–Guruswami–Wirth [9] [10], while incorporating the constraints on weights and cluster sizes at hand.

In our derivations, we make frequent use of the LP-cost and cluster-cost of an edge: the LP-cost refers to the cost of that edge using the LP solution \((x, y)\), while
Algorithm 1: Round LP solution to obtain clustering, using a threshold parameter $\alpha$.

Let $S = V(G)$.

while $S \neq \emptyset$ do
    Let the “pivot vertex” $u$ be an arbitrary element of $S$.
    Let $T = \{w \in S - \{u\} : x_{uw} \leq \alpha\}$.
    if $\sum_{w \in T} x_{uw} \geq \alpha |T|/2$ then
        Output the singleton cluster $\{u\}$.
        Let $S = S - \{u\}$.
    else
        Output the cluster $\{u\} \cup T$.
        Let $S = S - (\{u\} \cup T)$.
    end if
end while

The cluster-cost refers to the actual cost incurred by that edge in the clustering produced by Algorithm 1.

Observation 1. Let $x$ be a feasible solution to $P$, and let $wz$ be an edge. For any vertex $u$, we have $x_{wz} \geq x_{uz} - x_{uw}$ and $1 - x_{wz} \geq 1 - x_{uz} - x_{uw}$.

The next lemma bounds the LP-cost of a subset of edges used in the region-growing Algorithm 1. All notation is per the definition of the algorithm.

Lemma 2. Let $(x, y)$ be a feasible solution to $P$, and let $k \geq 0$. Suppose that $z \in S$ at the beginning of the iteration, let $u$ be the pivot vertex, and let $R \subseteq \{u\} \cup T$. For any $\zeta \in [0, 1]$, if $\sum_{v \in R} x_{uv} \leq \alpha |R|/2$ and $x_{uv} \leq \zeta$ for all $v \in R$, then the total LP-cost of the edges joining $z$ and $R$ is at least

$$\sum_{v \in R} \left[w_{vz}^+ x_{uz} + w_{vz}^- (1 - x_{uz}) - \zeta(w_{vz}^+ + w_{vz}^-) + (\zeta - \frac{\alpha}{2})\right].$$

Proof. By Observation 1, we have $x_{vz} \geq x_{uz} - x_{uv}$ and $1 - x_{vz} \geq 1 - x_{uz} - x_{uv}$ for each edge $vz$. It follows that the total LP-weight of the edges joining $z$ and $R$ is at least

$$\sum_{v \in R} \left[w_{vz}^+ (x_{uz} - x_{uv}) + w_{vz}^- (1 - x_{uz} - x_{uv})\right],$$

which rearranges to

$$\sum_{v \in R} \left[w_{vz}^+ x_{uz} + w_{vz}^- (1 - x_{uz})\right] - \sum_{v \in R} \left[(w_{vz}^+ + w_{vz}^-) x_{uv}\right].$$
Using $\sum_{v \in R} x_{uv} \leq \alpha |R|/2$ and $x_{uv} \leq \zeta$ for $v \in R$, we bound the subtracted sum as follows:

$$\sum_{v \in R} [(w_{vz}^+ + w_{vz}^-)x_{uv}] = \sum_{v \in R} x_{uv} + \sum_{v \in R} [(w_{vz}^+ + w_{vz}^- - 1)x_{uv}]$$

$$\leq \frac{\alpha |R|}{2} + \sum_{v \in R} [(w_{vz}^+ + w_{vz}^- - 1)x_{uv}]$$

$$= \sum_{v \in R} [(w_{vz}^+ + w_{vz}^- - 1)x_{uv} + \frac{\alpha}{2}]$$

$$\leq \sum_{v \in R} \left[ \zeta(w_{vz}^+ + w_{vz}^-) + \left(\frac{\alpha}{2} - \zeta\right)\right].$$

Note that in the last inequality we used the fact that $w_{vz}^+ + w_{vz}^- \geq 1$. \hfill \Box

**Theorem 3.** Let $G$ be a correlation-clustering instance satisfying Assumption [A]. If $(x, y)$ is any solution to the linear program $P$, then the clustering produced by Algorithm 1 using $(x, y)$ has cost at most $c_\alpha$ cost$(x, y)$, where

$$c_\alpha = \max \left\{ \mu^*, \frac{2\alpha \mu^*}{1 - 2\alpha} + \frac{1}{1 - 2\alpha + \frac{\alpha}{2}}, \frac{2}{\alpha} \right\}.$$

**Proof.** The idea behind the proof is as follows. After each cluster output $\{u\} \cup T$ is produced, one is asked to “pay for” the cluster-cost of the edges which are both incident to $\{u\} \cup T$ and were contained within the pool of vertices $S$ available at the beginning of the given iteration of the algorithm. We pay for the cluster-cost by “charging” some quantity to each of these edges. Subsequently, we show that the total amount charged to each edge is at most $c_\alpha$ times its LP-cost. While some edges may be charged multiple times, one can appropriately bound the total amount of charge accrued by each edge.

Suppose that some cluster $\{u\} \cup T$ has just been selected as an output of Algorithm 1. We split the analysis of the claimed result into two cases, according to whether or not $\{u\} \cup T$ is a singleton set.

**Case 1:** The output is a singleton cluster $\{u\}$. The total cluster cost when outputting a singleton cluster $\{u\}$ is $\sum_{v \in S - u} w_{uv}^+$, while the total LP-cost accrued by edges incident to $u$ is $\sum_{v \in S - u} w_{uv}^+ x_{uv}$.

If the singleton $\{u\}$ is output, then we have

$$\sum_{v \in T} x_{uv} \geq \frac{\alpha |T|}{2}.$$

For each $v \in T$, we have $x_{uv} \leq \alpha$. For such $x_{uv}$, we also have $1 - x_{uv} \geq x_{uv}$, since $\alpha < 1/2$. This yields the following lower bound on the LP-cost of $uv$:

$$w_{uv}^+ x_{uv} + w_{uv}^- (1 - x_{uv}) \geq w_{uv}^+ x_{uv} + w_{uv}^- x_{uv} \geq x_{uv},$$

where the last inequality follows from the bound $w_{uv}^+ + w_{uv}^- \geq 1$. Thus, each edge $uv$ has LP-cost at least $x_{uv}$, and so the edges joining $u$ and $T$ have total LP-cost at least $\alpha |T|/2$. Each such edge $uv$ incurs cluster-cost $w_{uv}^+$, which is at most equal to 1. Thus, charging $(2/\alpha)x_{uv}$ to each edge $uv$ for $v \in T$ produces enough “charge” to pay for the cluster-cost of edges $v \in T$, and at the same time, each edge is charged at most $2/\alpha$ times its LP-cost.
For \( v \in S - (T \cup \{u\}) \), we have \( x_{uv} > \alpha \), so each edge \( uw \) incurs an LP-cost at least \( \omega w^+_{uv} \) and a cluster-cost at most \( w^+_{uv} \). Thus, charging such edge \((1/\alpha)w^+_{uv}x_{uv}\) compensates for the cluster-cost of the edges under consideration.

**Case 2:** The output is a nonsingleton cluster \( \{u\} \cup T \). We first consider edges in \( \{u\} \cup T \), and then consider edges joining \( \{u\} \cup T \) with \( S - (\{u\} \cup T) \). We conclude by describing how to pay the possible penalties for allowing the set \( T \) to be too large.

**Edges within** \( \{u\} \cup T \). Let \( \gamma = \alpha - \frac{\alpha}{2z} \), and note that \( \frac{3\alpha}{4} \leq \gamma \leq \alpha \). Assume that the vertices of \( \{u\} \cup T \) are linearly ordered so that \( v \leq z \) implies \( x_{uv} \leq x_{uz} \). Let \( vz \) be an edge contained in \( \{u\} \cup T \), so that \( vz \) incurs cluster-cost \( w^-_{vz} \) and LP-cost at least \( w^-_{vz}(1 - x_{vz}) \). If \( x_{uv} \) and \( x_{vz} \) are both at most \( \gamma \), then Observation 1 yields

\[
1 - x_{vz} \geq 1 - x_{uv} - x_{vz} \geq 1 - 2\gamma,
\]

so that charging \( \frac{1}{1-2\gamma}w^-_{vz}(1 - x_{vz}) \) to the edge \( vz \) pays for its cluster-cost.

Otherwise, suppose that \( x_{vz} > \gamma \), and let \( R = \{v \in \{u\} \cup T: v < z\} \). We charge to \( z \) all edges \( vz \) with \( v \in R \). The ordering on \( \{u\} \cup T \) implies that \( \sum_{v \in R} x_{uv} \leq \alpha |R|/2 \) and that \( x_{uv} \leq x_{vz} \) for all \( v \in R \). Applying Lemma 2 with \( \zeta = x_{vz} \) and rearranging the terms in the sum shows that the total LP-cost of edges \( vz \) with \( v \in R \) is at least

\[
\sum_{v \in R} w^-_{vz}(1 - 2x_{u}z) + (x_{uz} - \frac{\alpha}{2}).
\]

This lower bound is linear in \( x_{uz} \). When \( x_{uz} = \gamma \), we see that the total LP-cost is at least \((1 - 2\gamma)\sum_{v \in R} w^-_{vz} \). When \( x_{uz} = \alpha \), the total LP-cost is at least

\[
\sum_{v \in R} w^-_{vz}(1 - 2\alpha + \frac{\alpha}{2\gamma}),
\]

which is equal to \((1 - 2\gamma)\sum_{v \in R} w^-_{vz} \) by the choice of \( \gamma \). Thus, charging \( \frac{1}{1-2\gamma}w^-_{vz}(1 - x_{vz}) \) to each edge \( vz \) pays for the cluster-cost of each edge.

**Edges joining** \( \{u\} \cup T \) **with** \( S - (\{u\} \cup T) \). Let \( z \) be a vertex such that \( z \notin \{u\} \cup T \). A cross-edge for \( z \) is an edge from \( z \) to \( \{u\} \cup T \). We show that the cross-edges for \( z \) have total cluster-cost that is at most \( \max\{\frac{1}{1-2\gamma}, \frac{\alpha}{2\gamma}\} \) times their total LP-cost. Note that whenever \( vz \) is a cross-edge, we have \( x_{uv} \leq \alpha \), by the definition of \( T \). Each cross-edge \( vz \) incurs cluster-cost \( w^+_{vz} \) and LP-cost \( w^+_{vz}x_{uz} + w^-_{vz}(1 - x_{vz}) \).

Let \( \eta = \alpha - \frac{\alpha}{2\gamma} \). If \( x_{uz} \geq 1 - \eta \), then based on Observation 1 we have

\[
x_{vz} \geq x_{uz} - x_{uv} \geq 1 - \eta - \alpha = 1 - 2\gamma
\]

for every cross-edge \( v \), where \( \gamma \) is defined as in the previous subcase. These edges have LP-cost at least \((1 - \eta - \alpha)w^+_{vz} \), and therefore have cluster-cost at most \( \frac{1}{1-2\gamma} \) times their LP-cost.

It remains to handle the case \( x_{uz} \in (\alpha, 1 - \eta) \). Here, we seek a lower bound on the total LP-cost of the cross-edges for \( z \). Note that the total cluster-cost of these edges is \( \sum_{v \in \{u\} \cup T} w^+_{vz} \), which is at most \(|T|\) since each \( w^+_{vz} \leq 1 \).

Using Lemma 2 with \( R = \{u\} \cup T \) and \( \zeta = \alpha \), we see that the total LP-cost of the cross-edges for \( z \) is at least

\[
\sum_{v \in \{u\} \cup T} \left[ w^+_{vz}x_{uz} + w^-_{vz}(1 - x_{uz}) - \alpha(w^+_{vz} + w^-_{vz}) + \frac{\alpha}{2}\right].
\]
This lower bound is a linear function in $x_{uz}$. We consider next the behavior of this function on the interval $(\alpha, 1 - \eta)$.

When $x_{uz} = \alpha$, the lower bound simplifies to

$$\sum_{v \in \{u\} \cup T} \left[ \alpha w^+_{vz} + (1 - \alpha)w^-_{vz} - \alpha(w^+_{vz} + w^-_{vz}) + \frac{\alpha}{2} \right],$$

which is at least $\alpha(T + 1)/2$ since $\alpha < 1/2$ implies $(1 - \alpha)w^-_{vz} \geq \alpha w^-_{vz}$.

When $x_{uz} = 1 - \eta$, the lower bound simplifies to

$$\sum_{v \in \{u\} \cup T} \left[ (1 - \eta - \alpha)w^+_{vz} - \frac{\alpha}{2}w^-_{vz} + \frac{\alpha}{2} \right] \geq \sum_{v \in \{u\} \cup T} \left[ (1 - \eta - \alpha)w^+_{vz} \right] = \sum_{v \in \{u\} \cup T} \left[ (1 - 2\eta)w^+_{vz} \right],$$

where we used $w^-_{vz} \leq \tau$. In all cases, charging $\max\left\{ \frac{1}{1 - 2\eta}, \frac{\alpha}{\tau} \right\}$ times the LP-cost of the edges pays for their cluster-cost.

**Paying for vertex penalties.** Finally, if $|\{u\} \cup T| > K + 1$, then we must pay for the “penalty cost” incurred by the cluster $\{u\} \cup T$. For any $v \in \{u\} \cup T$, the LP-penalty paid by $v$ is $\mu_v y_v$, while the cluster-penalty paid by $v$ is $\mu_v(|T| - K)$. The idea is to charge all the edges incident to $v$, as well as the vertex $v$ itself, in order to pay the cluster-penalty.

Let $\beta = \frac{1 - 2\alpha}{2\alpha}$. For each vertex $z \in T$, we charge $\mu_v(w^+_{vz} x_{uz} + \frac{1}{\beta} w^-_{vz} (1 - x_{uz}))$ to the edge $vz$. Observe that each edge in $T$ is only charged this way at its endpoints. The total charge from the edges is

$$\mu_v \left[ \sum_{z \in T} w^+_{vz} x_{uz} + \frac{1}{\beta} \sum_{z \in T} w^-_{vz} (1 - x_{uz}) \right].$$

We also charge $\mu_v y_v$ to the vertex $v$ itself. We wish to show that the total charge is at least $|T| - K - y_v$. Observe that the inequality $\sum_{z \in T}(1 - x_{uz}) \leq K + y_v$ implies that $\sum_{z \in T} x_{uz} \geq |T| - K - y_v$. Furthermore, Observation 1 together with the inequalities $x_{uu}, x_{uz} \leq \alpha$ yields

$$1 - x_{uz} \geq 1 - x_{uv} - x_{uz} \geq 1 - 2\alpha = \beta(2\alpha) \geq \beta x_{uz}.$$

Thus, we may write

$$\sum_{z \in T} w^+_{vz} x_{uz} + 2 \sum_{z \in T} w^-_{vz} (1 - x_{uz}) \geq \sum_{z \in T} x_{uz} + \sum_{z \in T} (w^+_{vz} - 1)x_{uz} + \frac{1}{\beta} \sum_{z \in T} w^-_{vz} (1 - x_{uz}) \geq |T| - K - y_v + \sum_{z \in T} (w^+_{vz} + w^-_{vz} - 1)x_{uz} \geq |T| - K - y_v,$$

where the last inequality is based on our assumption that $w^+_{vz} + w^-_{vz} \geq 1$. It follows that charging $\mu_v(w^+_{vz} x_{uz} + \frac{1}{\beta} w^-_{vz} (1 - x_{uz}))$ to each edge $vz$, and charging $\mu_v y_v$ to $v$ itself pays for the cluster-penalty of $v$.

In total, we have paid for all the cluster-costs by making the following charges to the edges, where $\mu^* = \max_{u \neq v \in V} (\mu_u + \mu_v)$:

- Edges $vz$ within $T$ were charged at most $\frac{1}{1 - 2\eta} w^-_{vz} (1 - x_{uz})$ to pay for their own cost, in addition to a cost of at most $\mu^*(w^+_{vz} x_{uz} + \frac{1}{\beta} w^-_{vz} (1 - x_{uz}))$ to
pay for the cluster-penalties of their endpoints. Since \( \frac{1}{\beta} = \frac{2\alpha}{1-2\alpha} \), their total charge is at most \( \max\{\mu^*, \frac{2\alpha\mu^*}{1-2\alpha} + \frac{1}{1-2\gamma}\} \) times their total LP-cost.

- Edges \( vz \) for which \( v \in \{u\} \cup T \) and \( z \in S - (\{u\} \cup T) \) were charged at most \( \max\{\frac{1}{1-2\gamma}, \frac{1}{\alpha}\} \) times their LP-cost.
- Vertices \( v \in T \) were charged exactly their LP-cost to pay for their cluster-penalty.

It follows that the clustering \( C \) has cost at most \( c_{\alpha} \text{cost}(x,y) \).

We now determine the optimal value of \( \alpha \) in terms of \( \mu^* \) and \( \tau \). Theorem 3 shows that Algorithm 1 achieves the approximation ratio

\[
\max \left\{ \mu^*, \frac{2\alpha\mu^*}{1-2\alpha} + \frac{1}{1-2\gamma} \cdot \frac{2}{\alpha} \right\}.
\]

Since we only consider \( \alpha \leq \frac{1}{2} \) and \( \mu^* \leq 4 \), we have \( \mu^* \leq \frac{2}{\alpha} \). Thus, the total approximation ratio is minimized when

\[
\frac{2\alpha\mu^*}{1-2\alpha} + \frac{1}{1-2\alpha + \frac{2}{\alpha}} = \frac{2}{\alpha}.
\]

It is straightforward to see that for any fixed \( \tau \geq 0 \) and any \( \mu^* \in [0, 4] \), this cubic equation in \( \alpha \) has a unique solution on \((0, \frac{1}{2})\]. Instead of solving it explicitly over other intervals, we study its behavior for extreme parameter values, in which case the equation simplifies.

When \( \mu^* = 0 \), we do not enforce any constraints on the cluster sizes: in particular, there is no need to pay for vertex penalties, so that the cluster size cost of \( \frac{2\alpha\mu^*}{1-2\alpha} \) never needs to be considered in our analysis. In particular, one can use the growth threshold \( \alpha = \frac{1}{2} \). The optimal value of \( \alpha \) equals \( \frac{2^2}{3 \cdot \tau} \), and the resulting approximation ratio is \( 5 - (1/\tau) \). If additionally \( \tau = 1 \), then we are effectively enforcing the probability constraints, and the resulting approximation ratio is 4, as in the original paper of Charikar, Guruswami, and Wirth [9, 10]. Indeed, in this special case our LP-rounding algorithm reduces exactly to the aforementioned algorithm.

In the limit \( \tau \to \infty \), we are allowing the weights \( w_e^- \) to be arbitrarily large. In this case, when \( \mu^* > 0 \), the total cost is minimized when \( \alpha = \frac{-5 + \sqrt{25+16\mu^*}}{4\mu^*} \). When \( \mu^* = 2 \), we are effectively enforcing a hard constraint on the cluster sizes, yielding an approximation ratio roughly equal to 6.275. When \( \mu^* = 0 \), the total cost is instead minimized when \( \alpha = 2/5 \), yielding an approximation ratio 5.

Finally, when \( \mu^* = 2 \) and \( \tau = 1 \), we are effectively enforcing the probability constraints with a hard constraint on the cluster sizes. In this case, the optimal choice for \( \alpha \) is 1/3, yielding an approximation ratio 6.

As a final note, we remark on the complexity of the LP and Algorithm 1 procedures. Interior point solvers based on Karmarkar’s method \([24, 27]\) require \( O(N^{3.5} S^2 \log S \log \log S) \) operations in the worst case, where \( N \) denotes the number of variables, and \( S \) denotes the input size of the problem. For the case of correlation clustering, \( N = O(|V(G)|^2) \). The complexity of Algorithm 1 is \( O(|V(G)| + |E(G)|) \), and the higher computational cost is clearly incurred by the LP solver.
4. A Random Pivoting Algorithm

In this section, we describe a randomized 6-approximation algorithm for the unweighted correlation clustering problem with bounded cluster sizes, based on the cc-pivot algorithm of Ailon, Charikar, and Newman [1, 2], shown in Algorithm 2. Unlike the method described in Section 3, this method does not require solving a linear program, and has lower asymptotic complexity than the best known LP solvers. The cc-pivot method is also amenable to parallelization, which is a significant advantage for large-scale problems encountered in social and biological network analysis [19].

Algorithm 2 cc-pivot algorithm [1, 2].

Let $S = V(G)$.

while $S \neq \emptyset$ do

Pick a uniform random $v \in S$.

Let $T = \left(\{v\} \cup N^+(v)\right) \cap S$.

Output the cluster $T$.

Let $S = S - T$.

end while

We now require an unweighted instance of the correlation-clustering problem, and we require hard constraints on the cluster sizes. Expressed in the language of weights, we require the following assumptions:

Assumption B. We assume that $uv \in E(G)$ for every pair of distinct vertices $u, v \in V(G)$. Furthermore, we assume that the edges can be partitioned as $E(G) = E^+(G) \cup E^-(G)$, where $w^+ = 1$ and $w^- = 0$ for all $e \in E^+(G)$, while $w^+ = 0$ and $w^- = 1$ for all $e \in E^-(G)$. Finally, we assume that $\mu_v = 1$ for all $v \in V(G)$.

For $X \subseteq E(G)$, we let $d_X(v)$ denote the number of edges of $X$ incident to $v$, and we let $d^+(v)$ denote $E^+(G)(v)$. We also use $N^+(v)$ to denote the set of neighbors of $v \in V(G)$ connected by positively labeled edges.

Algorithm 3 Approximation algorithm for correlation clustering with cluster size at most $K + 1$.

Take $X \subseteq E^+(G)$ to be the smallest set such that $d^+(v) - d_X(v) \leq K$ for all $v$.

Let $H$ be the labeled graph with $E^+(H) = E^+(G) - X$ and $E^-(H) = E^-(G) \cup X$.

Run cc-pivot on $H$.

Our algorithm for solving the correlation-clustering problem with cluster-size constraints, given in Algorithm 3, is in some sense the obvious algorithm: we first remove positive edges from $G$, if necessary, so that every vertex has positive degree at most $K$, and then run cc-pivot on the resulting graph $H$, taking advantage of the fact that cc-pivot always clusters its pivot vertex $v$ with at most $d^+(v)$ vertices. It is known that the set $X$ required in the first step can be found in polynomial time [20, 23, 26] – more precisely, that it can be found in time $O(\sqrt{K}|V(G)||V(G)|^2)$. Since cc-pivot itself runs in time $O(V(G) + E(G))$, the running time of Algorithm 3 is dominated by finding the set $X$. We remark...
CONSTRAINED CLUSTER SIZES

Q: minimize \( \sum_{e \in E} x_e \)
subject to \( x_{uv} + x_{vw} + x_{uw} \geq 1 \) (for all \( uvw \in T(G) \))
\[ \sum_{w \in N^+(v)} x_{vw} \geq d^+(v) - K \] (for all \( v \in V(G) \))
x_e integer
x_e \leq 1 (for all \( e \in E(G) \))
x_e \geq 0 (for all \( e \in E(G) \))

(a) The integer linear program \( Q \).

\( Q^1: \) minimize \( \sum_{e \in E} x_e \)
subject to \( \sum_{w \in N^+(v)} x_{vw} \geq d^+(v) - K \) (for all \( v \in V(G) \))
x_e integer
x_e \leq 1 \quad \text{(for all } e \in E(G) \text{)}
x_e \geq 0 \quad \text{(for all } e \in E(G) \text{)}

(b) The integer linear program \( Q^1 \) and linear program \( Q^2 \). Free variables in \( Q^1 \) range over the same sets as the corresponding variables in \( Q \).

Figure 2. The linear programs \( Q, Q^1, \) and \( Q^2 \).

here without proof that choosing \( X \) greedily (by independently deleting all but \( K \) incident edges for each vertex of \( v \)) may not find an optimal \( X \), but runs faster and still yields a 9-approximation algorithm. (The proof given here can be easily modified, taking advantage of the fact that the output of the greedy algorithm has at most twice as many edges as the optimal set.)

It is clear that Algorithm 3 produces a clustering in which all clusters have size at most \( K + 1 \). We claim that Algorithm 3 is in fact an expected 6-approximation algorithm. In the rest of this section, we prove the claim.

Following Ailon, Charikar and Newman, we say that a triangle in \( G \) is “bad” if it consists of two positively and one negatively labeled edge. We let \( T(G) \) denote the family of (vertex sets of) bad triangles in \( G \), and likewise for \( T(H) \).

Consider the integer linear program \( Q \) given in Figure 2 which represents an alternative model for the correlation-clustering problem with cluster sizes at most \( K + 1 \). Here, we interpret \( x_{vw} = 1 \) as meaning “the edge \( vw \) is an error”. The constraints of the form \( x_{uv} + x_{vw} + x_{uw} \geq 1 \) model the fact that each bad triangle must contain at least one error, while the constraints of the form \( \sum_{w \in N^+(v)} x_{vw} \geq d^+(v) - K \) model the fact that if a vertex has more than \( K \) positive neighbors, then at least \( d^+(v) - K \) of its positive edges must be mistakes, since \( v \) can only be clustered with at most \( K \) of those neighbors. Since any feasible clustering \( C \) yields a feasible solution to \( Q \) whose cost in \( Q \) is exactly the number of errors in \( C \), the optimal cost of \( Q \) is a lower bound on the cost of an optimal clustering.
The idea behind our analysis is that the two subproblems solved in Algorithm 3 can be represented by the programs $Q^1$ and $Q^2$, which respectively model the problem of finding a set $X$ to remove and the problem of clustering the resulting graph $H$. The constraints for $Q^1$ are just the cluster-size constraints of $Q$, and the constraints for $Q^2$ are almost the bad-triangle constraints of $Q$, except that the constraints for $Q^2$ range over bad triangles in $H$ while the constraints for $Q$ range over bad triangles in $G$. Nevertheless, we show that solutions to (appropriately defined) duals of $Q^1$ and $Q^2$ can be combined and slightly modified in order to yield a feasible solution $\xi$ to the dual of a suitable relaxation of $Q$. We show then that $\text{cost}(\xi) \geq \frac{1}{5}\mathbb{E}[\text{cost}(C)]$, where $C$ is the clustering produced by Algorithm 3; this shows that any feasible clustering has cost at least $\frac{1}{5}\mathbb{E}[\text{cost}(C)]$.

We remark that the program $Q^2$ is exactly the linear program introduced by Ailon, Charikar, and Newman in order to analyze the performance of cc-PVOT. In order to prove that cc-PVOT is an expected 3-approximation algorithm, they proved the following result, which we use as a lemma:

**Lemma 4** (Ailon–Charikar–Newman). The linear program $Q^2$ has a dual solution $\beta$ such that the dual cost of $\beta$ is at least $\frac{1}{5}\mathbb{E}[\text{cost}(C)]$, where $C$ is the random clustering produced by cc-PVOT.

In order to deal with the integrality constraints in $Q$ and $Q^1$, we rewrite them as additional linear constraints. Observe that the restrictions $0 \leq x_e \leq 1$ imply that $Q^1$ only has finitely many integer feasible points. Let $\mathcal{P}$ be the convex hull of the integer feasible points of $Q^1$; by standard polytope theory, there is some matrix $A$ and vector $b$ such that
\[
\mathcal{P} = \{ x \in \mathbb{R}^{E(G)} : Ax \leq b \}.
\]
Replacing the constraint “$x_e \in \{0, 1\}$” with $Ax \leq b$ in $Q^1$, we obtain a linear program $Q^{1*}$ whose optimal value is equal to the optimal value of the integer program $Q^1$. On the other hand, let $Q^*$ be the linear program obtained from $Q$ by replacing the constraint “$x_e \in \{0, 1\}$” with “$Ax \leq b$”. Since every feasible point for $Q$ is also feasible for $Q^1$, all integer feasible points of $Q$ also satisfy $Ax \leq b$. It follows that the optimal value of $Q^*$ is at most the optimal value of $Q$. It is possible for $Q^*$ to have a strictly lower optimal value, since adding the additional constraints for bad triangles may introduce non-integral vertices to the feasible polytope, but this is irrelevant for our analysis.

Solving $Q^1$ or $Q^{1*}$ directly by standard linear programming techniques would likely be intractable, since the matrix $A$ may be exponentially larger than the original linear program. Nevertheless, the algorithms given by Hell–Kirkpatrick [23] and by Gabow [20] run in polynomial time and produce an optimal solution to $Q^1$, using matching techniques. This optimal solution guarantees the existence of a dual solution with equal cost:

**Lemma 5.** If $X$ is the edge set computed in Algorithm 3 then the linear program $Q^{1*}$ has a dual solution $\psi$ such that $\text{cost}(\psi) = |X|$.

*Proof.* A feasible solution to $Q^{1*}$ is just the indicator vector of an edge set $Y$ such that $d^+(v) - d^-(v) \leq K$, for all $v \in V(G)$; the cost of such a solution is $|Y|$. Since $X$ is a smallest such edge set, the claim follows from strong duality of linear programming. \qed
Our goal now is to combine the dual solutions $\psi$ and $\pi$ for $Q_{1^*}$ and $Q^2$ and form a dual solution $\xi$ for $Q^*$. Let $R_{1^*}$, $R^2$, and $R^*$ denote the duals of $Q_{1^*}$, $Q^2$, and $Q^*$, respectively. We introduce the following notation for the variables of these dual programs:

- For each $t \in T(G)$, let $\alpha_t$ denote the variable of $R^*$ corresponding to the constraint $x_{vw} + x_{uw} + x_{uw} \geq 1$, where $u, v, w$ are the vertices of $t$.
- For each $t \in T(H)$, let $\beta_t$ denote the variable of $R^2$ corresponding to the constraint $x_{vw} + x_{uw} + x_{uw} \geq 1$, where $u, v, w$ are the vertices of $t$.
- For each vertex $v \in V(G)$, let $\gamma_v$ denote the variable of $R^*$ and of $R_{1^*}$ corresponding to the constraint $\sum_{w \in N^+(v)} x_{vw} \geq d^+(v) - K$.
- For each edge $e \in E(G)$, let $\delta_e$ denote the variable of $R^*$ and of $R_{1^*}$ corresponding to the constraint $x_e \leq 1$.

- Let $A_1, \ldots, A_h$ denote the rows of $A$, and for each $i \in \{1, \ldots, h\}$, let $\epsilon_i$ denote the variable of $R^*$ and of $R_{1^*}$ corresponding to the constraint $A_i x \leq b_i$.

We write solutions to $R^*$ as tuples of the form $(\alpha, \gamma, \delta, \epsilon)$, where $\alpha$, $\gamma$, and $\delta$ are vectors indexed as above, and likewise for $R_{1^*}$ and $R^2$. Given solutions $\beta$ and $(\gamma, \delta, \epsilon)$ to $R^2$ and $R_{1^*}$, respectively, we would like to simply use $(\beta, 0, 0, 0)$ and $(0, \gamma, \delta, \epsilon)$ as solutions for $R^*$, where each 0 stands for a zero vector of the appropriate length. In general, this is not possible, because $\beta$ may assign positive weight to triangles of $T(H)$ that do not lie in $T(G)$. The following lemma addresses this difficulty; we omit the proof, as it is straightforward but notationally tedious.

**Lemma 6.** Let $\beta$ be a solution to $R^2$ and let $(\gamma, \delta, \epsilon)$ be a solution to $R_{1^*}$. If $\alpha$ is the vector indexed by $T(G)$ defined by

$$\alpha_t = \begin{cases} 
\beta_t, & \text{if } t \in T(H), \\
0, & \text{otherwise,}
\end{cases}$$

then $(\alpha, 0, 0, 0)$ and $(0, \gamma, \delta, \epsilon)$ are feasible solutions to $R^*$.

**Lemma 6** suggests that triangles in $T(H) - T(G)$ will play an important role in the proof. We characterize these triangles.

**Lemma 7.** Let $G$, $X$, and $H$ be as in Algorithm 3. For each $t \in T(H)$, we have $t \in T(H) - T(G)$ if and only if $t$ contains exactly one edge of $X$.

**Proof.** If $t \in T(H)$, then $t$ contains exactly one negative edge $e_t$. Since $E^-(H) - E^-(G) = X$, we have $t \notin T(G)$ if and only if $e_t \in X$. □

We now construct the desired dual point and prove a lower bound on its cost.

**Theorem 8.** Let $\beta$ be an optimal solution to $R^2$ and let $(\gamma, \delta, \epsilon)$ be an optimal solution to $R_{1^*}$. Let $\alpha$ be defined as in Lemma 6 and let $\xi$ denote the convex combination $\frac{1}{2}(\alpha, 0, 0, 0) + \frac{1}{2}(0, \gamma, \delta, \epsilon)$. The point $\xi$ is a feasible solution to $R^*$ with value at least $\frac{1}{2}E[\text{cost}(C)]$, where $C$ is the random clustering produced by Algorithm 3.

**Proof.** Feasibility of $\xi$ follows immediately from Lemma 6. Variables which appear in both $R^*$ and $R_{1^*}$, or in both $R^*$ and $R^2$ have the same coefficient in the cost vector for each program, since the corresponding constraints in the primal programs are the same. Hence,

$$\text{value}_{R^*}(0, \gamma, \delta, \epsilon) = \text{value}_{R_{1^*}}(\gamma, \delta, \epsilon) = |X|,$$
To compute the value of $(\alpha, 0, 0, 0)$, we use the observation that each variable $\alpha_t$ or $\beta_t$ has coefficient 1 in the cost vector for $R^*$ or $R^2$, respectively. Hence,

$$value_{R^2}(\beta) - value_{R^*}(\alpha, 0, 0, 0) = \sum_{t \in \mathcal{T}(H) - \mathcal{T}(G)} \beta_t.$$

Now, for each $e \in E(H)$, we see that $R^2$ has a constraint of the form $\sum_{e \subseteq t} \beta_t \leq 1$; that is, the weights $\beta_t$ give a fractional packing of bad triangles (the same observation was used by Ailon, Charikar, and Newman [1, 2] in their proof). In particular, since each triangle in $\mathcal{T}(H) - \mathcal{T}(G)$ contains an edge of $X$, we have

$$\sum_{t \in \mathcal{T}(H) - \mathcal{T}(G)} \beta_t \leq \sum_{e \in X} \sum_{e \subseteq t} \beta_t \leq |X|.$$

Since Lemma 4 yields $value(\beta) \geq \frac{1}{6}E[cost(C)]$, we have

$$value(\xi) \geq \frac{1}{2} value(0, \gamma, \delta, \epsilon) + \frac{1}{2} value(\alpha, 0, 0, 0) \geq \frac{1}{2} |X| + \frac{1}{2} \left( \frac{1}{3} E[cost(C)] - |X| \right).$$

The claim follows.

**Corollary 9.** The optimal cost of $Q$ is at least $\frac{1}{6}E[cost(C)]$.

**Proof.** By Theorem and duality, the optimal cost of $Q^*$ is at least $\frac{1}{6}cost(C)$. Since $Q^*$ is a relaxation of $Q$, the claim follows.

**Corollary 10.** Algorithm is an expected 6-approximation algorithm for the correlation-clustering problem under Assumption [2].

**References**

1. Nir Ailon, Moses Charikar, and Alantha Newman, *Aggregating inconsistent information: ranking and clustering*, Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, ACM, 2005, pp. 684–693.
2. , *Aggregating inconsistent information: ranking and clustering*, Journal of the ACM (JACM) 55 (2008), no. 5, 23.
3. Konstantin Andreev and Harald Racke, *Balanced graph partitioning*, Theory of Computing Systems 39 (2006), no. 6, 929–939.
4. Nikhil Bansal, Avrim Blum, and Shuchi Chawla, *Correlation clustering*, Proceedings of the 43rd Symposium on Foundations of Computer Science (Washington, DC, USA), FOCS ’02, IEEE Computer Society, 2002, pp. 238–.
5. , *Correlation clustering*, Mach. Learn. 56 (2004), no. 1–3, 89–113.
6. John Bartholdi III, Craig A Tovey, and Michael A Trick, *Voting schemes for which it can be difficult to tell who won the election*, Social Choice and welfare 6 (1989), no. 2, 157–165.
7. Jean M Carlson and John Doyle, *Complexity and robustness*, Proceedings of the National Academy of Sciences 99 (2002), no. suppl 1, 2538–2545.
8. Pierre Charbit, Stéphan Thomassé, and Anders Yeo, *The minimum feedback arc set problem is np-hard for tournaments*, Combinatorics, Probability and Computing 16 (2007), no. 01, 1–4.
9. Moses Charikar, Venkatesan Guruswami, and Anthony Wirth, *Clustering with qualitative information*, Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), FOCS ’03, IEEE Computer Society, 2003, pp. 524–.
10. , *Clustering with qualitative information*, J. Comput. Syst. Sci. 71 (2005), no. 3, 360–383.
11. Marie-Christine Costa, Lucas Létocart, and Frédéric Roupin, *Minimal multicut and maximal integer multiflow: a survey*, European Journal of Operational Research 162 (2005), no. 1, 55–69.
12. Frank Dehne, Michael A Langston, Xuemei Luo, Sylvain Pite, Peter Shaw, and Yun Zhang, *The cluster editing problem: Implementations and experiments*, Parameterized and Exact Computation, Springer, 2006, pp. 13–24.

13. Erik D Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica, *Correlation clustering in general weighted graphs*, Theoretical Computer Science 361 (2006), no. 2, 172–187.

14. Erik D Demaine and Nicole Immorlica, *Correlation clustering with partial information*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Springer, 2003, pp. 1–13.

15. Raffaele Di Natale, Alfredo Ferro, Rosalba Giugno, Misael Mongiovì, Alfredo Pulvirenti, and Dennis Shasha, *Sing: Subgraph search in non-homogeneous graphs*, BMC bioinformatics 11 (2010), no. 1, 96.

16. Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar, *Rank aggregation methods for the web*, Proceedings of the 10th international conference on World Wide Web, ACM, 2001, pp. 613–622.

17. Dotan Emanuel and Amos Fiat, *Correlation clustering–minimizing disagreements on arbitrary weighted graphs*, Algorithms-ESA 2003, Springer, 2003, pp. 208–220.

18. Farzad Farnoud and Olgica Milenkovic, *An axiomatic approach to constructing distances for rank comparison and aggregation*, Transactions on Information Theory 60 (2014), no. 10, 6417–6439.

19. Robson Leonardo Ferreira Cordeiro, Caetano Traina Junior, Agma Juci Machado Traina, Julio López, U Kang, and Christos Faloutsos, *Clustering very large multi-dimensional datasets with mapreduce*, Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 690–698.

20. Harold N. Gabow, *An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems*, Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC ’83, ACM, 1983, pp. 448–456.

21. Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas, *Clustering aggregation*, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (2007), no. 1, 4.

22. John A Hartigan and Manchek A Wong, *Algorithm as 136: A k-means clustering algorithm*, Applied statistics (1979), 100–108.

23. P. Hell and D. G. Kirkpatrick, *Algorithms for degree constrained graph factors of minimum deficiency*, J. Algorithms 14 (1993), no. 1, 115–138. MR 1199249 (93m:68133)

24. NK Karmarkar and KG Ramakrishnan, *Computational results of an interior point algorithm for large scale linear programming*, Mathematical Programming 52 (1991), no. 1-3, 555–586.

25. Ron Shamir, Roded Sharan, and Dekel Tsur, *Cluster graph modification problems*, Discrete Appl. Math. 144 (2004), no. 1-2, 173–182. MR 2095392 (2005e:05819)

26. Chaitanya Swamy, *Correlation clustering: maximizing agreements via semidefinite programming*, Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2004, pp. 526–527.

27. Chong Zhang, Julian Yarkony, and Fred A Hamprecht, *Cell detection and segmentation using correlation clustering*, Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014, Springer, 2014, pp. 9–16.