Correlation Clustering — Minimizing Disagreements on Arbitrary Weighted Graphs

Dotan Emanuel and Amos Fiat

Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel

There is a correlation between the creative and the screwball.
So we must suffer the screwball gladly.

Kingman Brewster, Jr. (1919 - 1988) President Yale University (1963-1977), US Ambassador to Great Britan (1977-1981), Master of University College, London (1986-1988).

Abstract. We solve several open problems concerning the correlation clustering problem introduced by Bansal, Blum and Chawla [1]. We give an equivalence argument between these problems and the multicut problem. This implies an $O(\log n)$ approximation algorithm for minimizing disagreements on weighted and unweighted graphs. The equivalence also implies that these problems are APX-hard and suggests that improving the upper bound to obtain a constant factor approximation is non trivial. We also briefly discuss some seemingly interesting applications of correlation clustering.

1 Introduction

1.1 Problem definition

Bansal, Blum and Chawla [1] present the following clustering problem. We are given a **complete** graph on n vertices, where every edge (u, v) is labelled either $\langle + \rangle$ or $\langle - \rangle$ depending on whether u and v have been deemed to be similar or different. The goal is to produce a partition of the vertices (a clustering) that agrees as much as possible with the edge labels. The number of clusters is not an input to the algorithm and will be determined by the algorithm. *I.e.*, we want a clustering that maximizes the number of $\langle + \rangle$ edges within clusters, plus the number of $\langle - \rangle$ edges between clusters (equivalently, minimizes the number of disagreements: the number of $\langle - \rangle$ edges inside clusters plus the number of $\langle + \rangle$ edges between clusters).

Bansal *et. al.*, [1], show the problem to be *NP*-hard. They consider the two natural approximation problems:

– Given a complete graph on n vertices with $\langle + \rangle / \langle - \rangle$ labels on the edges, find a clustering that maximizes the number of agreements.

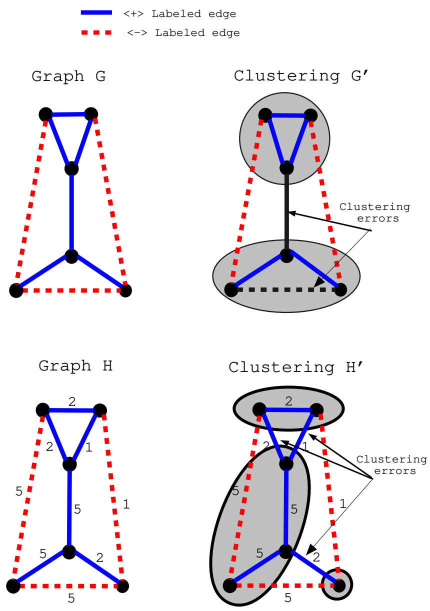

Fig. 1. Two clustering examples for unweighted and weighted (general) graphs. In the unweighted case we give an optimal clustering with two errors: one error on an edge labelled $\langle + \rangle$ and one error on an edge labelled $\langle - \rangle$. For the weighted case we get a different optimal clustering with three errors on $\langle + \rangle$ edges and total weight 5

– Given a complete graph on n vertices with $\langle + \rangle / \langle - \rangle$ labels on the edges, find a clustering that minimizes the number of disagreements.

For the problem of maximizing agreements Bansel *et. al.* ([1]) give a polynomial time approximation scheme. For the problem of minimizing disagreements they give a constant factor approximation. Both of these results hold for complete graphs.

Bansal *et. al.* pose several open problems, including the following:

1. What can one do on general graphs, where not all edges are labelled either $\langle + \rangle$ or $\langle - \rangle$? If $\langle + \rangle$ represents attraction and $\langle - \rangle$ represents the opposite, we may have only partial information on the set of all pairs, or there may be pairs of vertices for which we are indifferent.
2. More generally, for some pairs of vertices, one may be able to quantify the strength of the attraction/rejection. Is it possible to approximate the agreement/disagreement in this case?

In this paper we address these two open questions with respect to minimizing disagreements for unweighted general graphs and for weighted general graphs.

1.2 Problem variants

Following Bansal *et. al.*, we define three problem variants. For all of these variants, the goal is to find a clustering that maximizes the number of agreements (alternately, minimizes the number of disagreements). In the weighted case one seeks to find a clustering that maximizes the number of agreements weighted by the edge weights (alternately, minimizes the number of disagreements weighted by edge weights).

- **Unweighted Complete Graphs**
  Every pair of vertices has an edge between them, and every edge is labelled either $\langle + \rangle$ or $\langle - \rangle$. An edge labelled $\langle + \rangle$ stands for attraction; the two vertices should be in the same cluster. An edge labelled $\langle - \rangle$ stands for rejection; the two vertices should be in different clusters.
- **Unweighted General Graphs**
  Two vertices need not necessarily have an edge between them, but if so then the edge is labelled either $\langle + \rangle$ or $\langle - \rangle$. If two vertices do not have an edge between them then this represents indifference (or no information) as to whether they should be in the same cluster or not.
- **Weighted General Graphs**
  Two vertices need not necessarily have an edge between them. Edges in the graph have *both* labels $\{\langle + \rangle, \langle - \rangle\}$ *and* positive real weights. An edge labelled $\langle + \rangle$ with a large weight represents strong attraction (the vertices should be in the same cluster), an edge labelled $\langle - \rangle$ and a large value represents strong rejection (the vertices should not be in the same cluster). No edge, or a weight of zero for an edge represents indifference or no prior knowledge.

For each of these problem variants we focus on minimizing disagreements (as distinct from the easier goal of maximizing agreements). We seek to minimize the number of edges labelled $\langle - \rangle$ within the clusters plus the number of the edges labelled $\langle + \rangle$ that cross cluster boundaries. In the weighted version we seek to minimize the sum of the weights of edges labelled $\langle - \rangle$ within the clusters plus the sum of the weights of the edges labelled $\langle + \rangle$ that cross cluster boundaries.

In the rest of this paper when we refer to the "correlation clustering problem" or "the clustering problem" we mean the problem of minimizing disagreements in one of the problem variants above. We will also say "positive edge" when referring to an edge labelled $\langle + \rangle$ and "negative edge" when referring to an edge labelled $\langle - \rangle$. Note that for both positive and negative edges, the weights are always ≥ 0.

Remarks:

1. We remark that although the optimal solution to maximizing agreements is the same as the optimal solution to minimizing disagreements, in terms of approximation ratios these two goals are obviously distinct.
2. It is not hard to see that for all problem variants, a trivial algorithm for maximizing agreements gives a factor of two approximation. Simply consider one of the two clusterings: every vertex is a distinct cluster or all vertices are in the same cluster.

3. It should be obvious that the problem of minimizing disagreements for unweighted complete graphs is a special case of minimizing disagreements for unweighted general graphs, which is itself a special case of minimizing disagreements for weighted general graphs.
4. We distinguish between the different problems because the approximation results and the hardness of approximation results are different or mean different things in the different variants.

1.3 Our Contributions

In [1] the authors presented a constant factor approximation algorithm for the problem of unweighted complete graphs, and proved that the problem for the weighted general graphs is APX-Hard. They gave the problem of finding approximation algorithms and hardness of approximation results for the two other variants (unweighted and weighted general graphs)as open questions.

Problem class	Approximation	Hardness of Approximation	Equivalence
Unweighted complete graphs	$c \in O(1)$	Open	
Unweighted general graphs	Open	Open	
Weighted general graphs	Open	APX-hard	

Fig. 2. Previous Results [BBC 2002] — Minimizing Disagreements

Problem class	Approximation	Hardness of Approximation	Equivalence
Unweighted general graphs	$O(\log n)$	APX-hard	Unweighted multicut
Weighted general graphs	$O(\log n)$		Weighted multicut

Fig. 3. Our Contributions. The equivalence column is to say that any c-approximation algorithm for one problem will translate into a c'-approximation approximation for the other, where c and c' are constants.

We give an $O(\log n)$ approximation algorithm for minimizing disagreements for both the weighted and unweighted general graph problems, and prove that the problem is APX-hard even for the unweighted general graph problem, thus admitting no polynomial time approximation scheme (PTAS). We do this by reducing the correlation clustering problems to the multicut problem.

We further show that the correlation clustering problem and the multicut problem are equivalent for both weighted and unweighted versions, and that any constant approximation algorithm or hardness of approximation result for one problem implies the same for the other. Note that the question of whether there exists a constant factor approximation for general weighted and unweighted graphs remains open. This is not very surprising as the multicut problem has been studied at length, and no better approximation found, this suggests that the problem is not trivial.

1.4 Some background regarding the multicut problem

The weighted multicut problem is the following problem: Given an undirected graph G, a weight function w on the edges of G, and a collection of k pairs of distinct vertices (s_i, t_i) of G, find a minimum weight set of edges of G whose removal disconnects every s_i from the corresponding t_i.

The problem was first stated by Hu in 1963 [8]. For $k = 1$, the problem coincides of course with the ordinary min cut problem. For $k = 2$, it can be also solved in polynomial time by two applications of a max flow algorithm [16]. The problem was proven NP-hard and MAX SNP-hard for any $k \geq 3$ in by Dahlhaus, Johnson, Papadimitriou, Seymour and Yannakakis [5]. The best known approximation ratio for weighted multicut in general graphs is $O(\log k)$ [7] . For planar graphs, Tardos and Vazirani [13] give an approximate MaxFlow MinCut theorem and an algorithm with a constant approximation ratio. For trees, Garg, Vazirani and Yannakakis give an algorithm with an approximation ratio of two [6].

1.5 Structure of this paper

In section 2 we give notations and definitions, in section 3 we prove approximation results, and in section 4 we establish the equivalence of the multicut and correlation clustering problems. Section 5 gives the APX-hardness proofs.

2 Preliminaries

Let $G = (V, E)$ be a graph on n vertices. Let $e(u, v)$ denote the label $(\langle + \rangle, \langle - \rangle)$ of the edge (u, v). Let $E^{\langle + \rangle}$ be the set of positive edges and let $G^{\langle + \rangle}$ be the graph induced by $E^{\langle + \rangle}$, $E^{\langle + \rangle} = \{(u, v) | e(u, v) = \langle + \rangle\}$, $G^{\langle + \rangle} = (V, E^{\langle + \rangle})$. Let $E^{\langle - \rangle}$ be the set of negative edges and $G^{\langle - \rangle}$ the graph induced by $E^{\langle - \rangle}$, $E^{\langle - \rangle} = \{(u, v) | e(u, v) = \langle - \rangle\}$, $G^{\langle - \rangle} = (V, E^{\langle - \rangle})$

Definition 2.01 *We will call a cycle* $(v_1, v_2, v_3 ..., v_k)$ *in* G *a* **erroneous cycle** *if it is a simple cycle, and it contains exactly one negative edge.*

We let OPT denote the optimal clustering on G. In general, for a clustering $\mathcal{C}$, let $C(v)$ be the set of vertices in the same cluster as v. We call an edge (u, v) a positive mistake if $e(u, v) = \langle + \rangle$ and yet $u \notin C(v)$. We call an edge (u, v) a negative mistake if $e(u, v) = \langle - \rangle$ and $u \in C(v)$. The number of mistakes of a clustering $\mathcal{C}$ is the sum of positive and negative mistakes. The weight of the clustering is the sum of the weights of mistaken edges in C;

$$w(\mathcal{C}) = \sum_{e(u,v)=\langle - \rangle, u \in C(v)} w(u, v) + \sum_{e(u,v)=\langle + \rangle, u \notin C(v)} w(u, v).$$

For a general set of edges $T \subset E$ we will define the weight of T to be the sum of the weights in T , $w(T) = \sum_{e \in T} w(e)$.

For a graph $G = (V, E)$ and a set of edges $T \subset E$ we define the graph $G \setminus T$ to be the graph $(V, E \setminus T)$.

Definition 2.02 *We will call a clustering a* **consistent clustering** *if it contains no mistakes.*

3 A logarithmic approximation factor for minimizing disagreements

3.1 Overview

We now show that finding an optimal clustering is equivalent to finding a minimal weight covering of the erroneous cycles. An edge is said to cover a cycle if the edge disconnects the cycle.

Guided by this observation will define a multicut problem derived from our original graph by replacing the negative edges with source-sink pairs (and some other required changes). We show that a solution to the newly formed multicut problem induces a solution to the clustering problem, that this solution and the multicut solution have the same weight, and that optimal solution to the multicut problem induces an optimal solution to the clustering problem.

These reductions imply that the $O(\log k)$ approximation algorithm for the multicut problem [7] induces an $O(\log n)$ approximation algorithm for the correlation clustering problem. We prove this for weighted general graphs, which imply the same result for unweighted general graphs. We start by stating two simple lemmata:

Lemma 3.11 *A graph contains no erroneous cycles if and only if it has a consistent clustering.*

Proof. Omitted.

Lemma 3.12 *The weight of mistakes made by the optimal clustering is equal to the minimal weight set of edges whose removal will eliminate all erroneous cycles in G.*

Proof. Omitted.

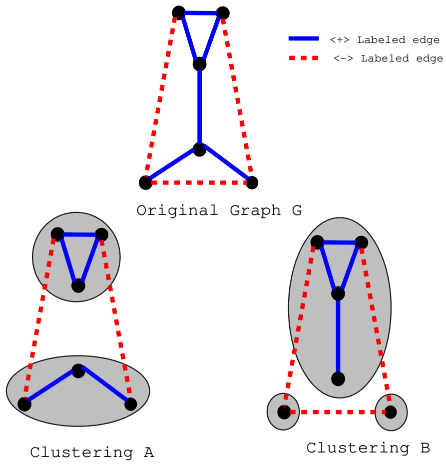

Fig. 4. Two optimal clusterings for G. For both of these clusterings we have removed two edges (different edges) so as to eliminate all the erroneous cycles in G. After the edges were removed every connected component of $G^{\langle + \rangle}$ is a cluster. Note that the two clusterings are consistent; no positive edges connect two clusters and no negative edges connect vertices within the same cluster.

3.2 Reduction from correlation clustering to weighted multicut

We give a reduction from the problem of correlation clustering to the weighted multicut problem. The reduction translates an instance of unweighted correlation clustering into an instance of unweighted graph multicut, and an instance of weighted correlation clustering into an instance of weighted graph multicut.

Given a weighted graph G whose edges are labelled $\{\langle + \rangle, \langle - \rangle\}$ we construct a new graph H_G and a collection of source-sink pairs $S_G = \{\langle s_i, t_i \rangle\}$ as follows:

- For every negative edge $(u, v) \in E^{\langle - \rangle}$ we introduce a new vertex $v_{\widehat{u,v}}$, a new edge $(v_{\widehat{u,v}}, u)$ with weight equal to that of (u, v), and a source-sink pair $\langle v_{\widehat{u,v}}, v \rangle$.
- Let V_{new} denote the set of new vertices, E_{new}, the set of new edges, and S_G, the set of source-sink pairs. Let $V^{'} = V \cup V_{\text{new}}$, $E^{'} = E^{\langle + \rangle} \cup E_{\text{new}}$, $H_G = (V^{'}, E^{'})$. The weight of the edges in $E^{\langle + \rangle}$ remains unchanged. We now have a multicut problem on (H_G, S_G).

We claim that given any solution to the multicut problem, this implies a solution to the correlation clustering problem with the exact same value, and that an approximate solution to the former gives an approximate solution to the later.

Theorem 3.21 *(H_G, S_G) has a cut of weight W if and only if G has a clustering of weight W, and we can easily construct one from the other. In particular, the optimal clustering in G of weight W implies an optimal multicut in (H_G, S_G) of weight W and vice versa.*

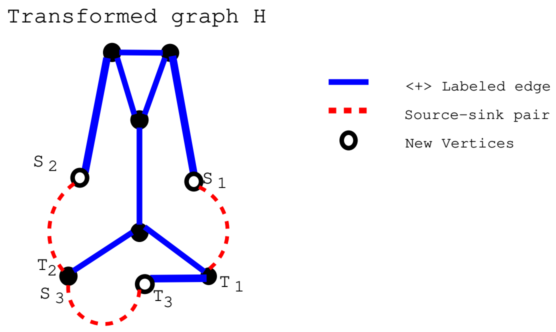

Fig. 5. The original graph from Figure 4 after the transformation

Proof. **Proposition 3.22** *Let $\mathcal{C}$ be a clustering on G with weight W then there exists a multicut T' in (H_G, S_G) with weight W.*

Proof. Let $\mathcal{C}$ be a clustering of G with weight W, where T is the set of mistakes made by $\mathcal{C}$ ($w(T) = W$). Let $T' = \{(u,v)|(u,v) \in T, (u,v) \in G^{\langle + \rangle}\} \cup \{(v_{\widehat{u,v}}, u)|(u,v) \in T, (u,v) \in G^{\langle - \rangle}\}$, *i.e.*, we replace every negative edge $(u,v) \in T$, with the edge $(v_{\widehat{u,v}}, u)$. Note that $w(T) = w(T')$. We now argue that T' is a multicut.

Assume not, then there exists a pair $(v_{\widehat{u,v}}, v) \in S_G$ and a path from $v_{\widehat{u,v}}$ to u that contains no edge from T'. From the construction of S_G and H_G, this implies that the edge $(v_{\widehat{u,v}}, u) \notin T'$ and that there exists a path from u to v in $G^{\langle + \rangle} \setminus T$. Note that (u,v) is a negative edge in $G \setminus T$, so the negative edge (u,v) and the path from u to v in $G^{\langle + \rangle} \setminus T$ jointly form an erroneous cycle in $G \setminus T$. This is a contradiction since $G \setminus T$ is consistent (Lemma 3.12) and contains no erroneous cycles (Lemma 3.11). Note that the proof is constructive.

Proposition 3.23 *If T' is a multicut in H_G of weight W, then there exists a clustering $\mathcal{C}$ in G of weight W.*

Proof. We construct a set T from the cut T' by replacing all edges in E_{new} with the corresponding negative edges in G, and define a clustering $\mathcal{C}$ by taking every connected component of $G^{\langle + \rangle}/T$ as a cluster. T has the same cardinality and total weight T'. Thus, if we show that $\mathcal{C}$ is consistent on $G \setminus T$ we are done (since $w(\mathcal{C}(G)) = w(\mathcal{C}(G \setminus T)) + w(T) = 0 + w(T') = W$).

Assume that $\mathcal{C}$ is not a consistent clustering on $G \setminus T$, then there exists an erroneous cycle in $G \setminus T$ (Lemma 3.11). Let (u,v) be the negative edge along this cycle. This implies a path from u to v in H_G (the path of positive edges of the cycle in $G \setminus T$). We also know that (u,v) is negative edge, which means that in the construction of H_G we replaced it with edge $(v_{\widehat{u,v}}, u)$. The edge $(v_{\widehat{u,v}}, u)$

is not in the cut (not in T') since (u, v) is not in T (as $(u, v) \in G \setminus T$). From this it follows that there is a path from $v_{\widehat{u,v}}$ to v in H_G. But the pair $\langle v_{u,v}, v\rangle$ are a source-sink pair which is in contradiction to T' being a multicut.

Proposition 3.22 and proposition 3.23 imply that

$$\begin{aligned} w(\text{Optimal clustering}(G)) &= w(\text{Multicut induced by opt. clustering}(H_G, S_G)) \\ &\geq w(\text{Minimal Multicut}(H_G, S_G)) \\ &= w(\text{Clustering on } G \text{ induced by minimal multicut }) \\ &\geq w(\text{Optimal clustering}(G)), \end{aligned}$$

where all inequalities must hold with equalities.

We can now use the approximation algorithm of [7] to get an $O(\log k)$ approximation solution to the multicut problem (k is the number of source-sink pairs) which translates into an $O(\log |E^{\langle - \rangle}|) \leq O(\log n^2) = O(\log n)$ solution to the clustering problem. Note that this result holds for both weighted and unweighted graphs and that the reduction of the unweighted correlation clustering problem results in a multicut problem with unity capacities and demands.

4 Reduction from multicut to correlation clustering

In the previous section we argued that every correlation clustering problem can be presented (and approximately solved) as a multicut problem. We will now show that the opposite is true as well, that every instance of the multicut problem can be transformed to an instance of a correlation clustering problem, and that transformation has the following properties: any solution to the correlation clustering problem induces a solution to the multicut problem with lower or equal weight, and an optimal solution to the correlation clustering problem induces an optimal solution to the multicut problem.

In the previous section we could use one reduction for the weighted version and the unweighted version. Here we will present two slightly different reductions from unweighted multicut to unweighted correlation clustering and from weighted multicut to weighted correlation clustering.

4.1 Reduction from weighted multicut to weighted correlation clustering

Given a multicut problem instance: an undirected graph H, a weight function w on the edges of H , $w : E \to \mathcal{R}^+$, and a collection of k pairs of distinct vertices $S = \{\langle s_i, t_i\rangle, \ldots, \langle s_k, t_k\rangle)\}$ of H we construct a correlation clustering problem as follows:

- We start with $G_H = H$, all edge weights are preserved and all edges labelled $\langle + \rangle$.

– In addition, for every source-sink pair $\langle s_i, t_i \rangle$ we add to G_H a negative edge $e_i = (s_i, t_i)$ with weight $w(e_i) = \sum_{e \in H} w(e) + 1$.

Our transformation is polynomial, adds at most $O(n^2)$ edges, and increases the largest weight in the graph by a multiplicative factor of at most n.

Theorem 4.11 *A clustering on G_H with weight W induces a multicut on (H, S) with weight $\leq W$. An optimal clustering in G_H induces an optimal multicut in (H, S).*

Proof. If a clustering $\mathcal{C}$ on G_H contains no negative mistakes, then the set of positive mistakes T is a multicut on H and $w(c) = w(t)$. If $\mathcal{C}$ contains a negative mistake, say (u, v), we take one of the endpoints (u or v) and place it in a cluster of it's own, thus eliminating this mistake. Since every negative edge has weight $\geq$ the sum of all positive edges, the gain by splitting the cluster will exceed the loss introduced by new positive mistakes, therefore the new clustering $\mathcal{C}'$ on G has weight $W' < W$, and it contains no negative mistakes. Thus, we know that $\mathcal{C}$ induces a cut of weight W'.

Now let T denote the minimal multicut in (H, S). T induces a clustering on G_H (the connected components of $G^{\langle + \rangle} \setminus T$) that contains no negative mistakes. This in turn means that the weight of the clustering is the weight of the positive mistakes, which is exactly $w(T)$. We now have $w(\text{Optimal multicut}) = w(\text{Clustering induced by optimal multicut})$. Combining the above two arguments we have that

$$\begin{aligned} w(\text{Optimal multicut}) &= w(\text{Clustering induced by optimal multicut}) \\ &\geq w(\text{Optimal clustering}) \\ &\geq w(\text{Multicut induced by the optimal clustering}) \\ &\geq w(\text{Optimal multicut}). \end{aligned}$$

Thus, all inequalities must hold with equality.

4.2 Reduction from unweighted multicut to unweighted correlation clustering

Given an unweighted multicut problem instance: an undirected graph H and a collection of k pairs of distinct vertices $S = \{\langle s_i, t_i \rangle, \ldots, \langle s_k, t_k \rangle\}$ of H we construct an unweighted correlation clustering problem as follows:

– For every v, $\langle v, u \rangle \in S$ or $\langle u, v \rangle \in S$, ($v$ is either a source or a sink) we add $n - 1$ new vertices and connect those vertices and v in a clique with positive edges (weight 1). We denote this clique by Q_v.
– For every pair $\langle s_i, t_i \rangle \in S$ we connect all vertices of Q_{s_i} to t_i and all vertices of Q_{t_i} to s_i using edges labelled $\langle - \rangle$.
– Other vertices of H are added to the vertex set of G_H, Edges of H are added to the edge set of G_H and labelled $\langle + \rangle$.

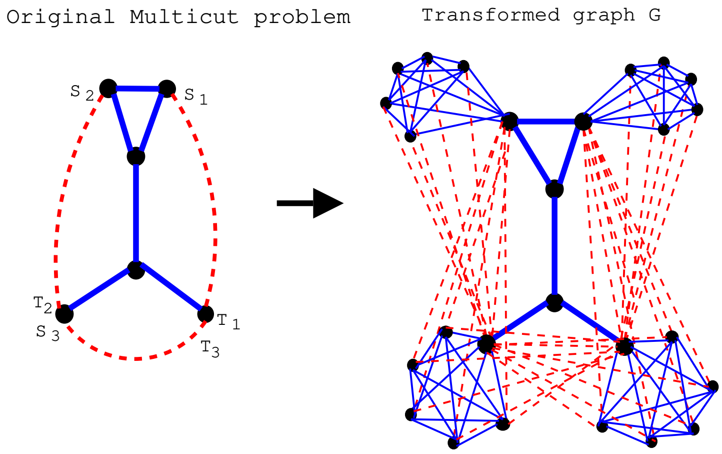

Fig. 6. Transformation from the unit capacity multicut problem (on the left) to the unweighted correlation clustering problem (on the right)

Our goal is to emulate the previous argument for weighted general graphs in the context of unweighted graphs. We do so by replacing the single edge of high weight with many unweighted negative edges. Our transformation is polynomial time, adds at most n^2 vertices and at most n^3 edges.

Theorem 4.21 *A clustering on G_H with weight W induces a multicut on (H, S) with weight $\leq W$. An optimal clustering in G of weight W induces an optimal multicut for (H, S) of weight W.*

Proof. We call a clustering *pure* if all vertices that belong to the same Q_v are in the same cluster, and that if $\langle v, w\rangle \in S$ then Q_v and Q_w are in different clusters. The following proposition implies that we can "fix" any clustering to be a pure clustering without increasing its weight.

Proposition 4.22 *Given a clustering C on G. We can "fix" that clustering to be pure thus find a pure clustering $C^{'}$ on G such that $w(C^{'}) \leq w(C)$.*

Proof. For every Q_v that is split amongst two or more cluster we take all vertices of Q_v to form a new cluster. By doing so we may be adding up to $n-1$ new mistakes, (positive mistakes, positive edges adjacent to v in original graph). Merging these vertices into one cluster component will reduce the number of errors by $n-1$ at least.

If two Q_v and Q_w are in the same cluster component, we can move one of them into a cluster of its own. As before, we we may be introducing as many as $n-1$ new positive mistakes but simultaneously eliminating $2n$ negative mistakes.

Given a clustering C on G_H we first "fix" it using the technique of proposition 4.22 to obtain a pure clustering $C^{'}$. Any mistake for pure clustering must be a positive mistake, the only negative edges are between clusters.

Let T be the set of positive mistakes for C', we now show that T is a multicut on (H, S). No source-sink pair are in the same cluster since the clustering in pure and removing the edges of T disconnects every source/sink pair. Thus, T is a multicut for (H, S).

Let OPT be the optimal clustering on G. OPT is pure (otherwise we can fix it and get a better clustering) and therefore induces a multicut on (H, S). Let T denote the minimal multicut in (H, S). T induces a pure-clustering on G as follows: take the connected component of $G^{\langle + \rangle} \setminus T$ as clusters and for every terminal $v \in S$ add every node in Q_v to the cluster containing vertices v. It can be easily seen that this gives a pure clustering, and that the only mistakes on the clustering are the edges in T.

Thus, we can summarize:

$$\begin{aligned} w(\text{Optimal multicut}) &= w(\text{Clustering induced by optimal multicut}) \\ &\geq w(\text{Optimal clustering}) \\ &\geq w(\text{Multicut induced by optimal clustering}) \\ &\geq w(\text{Optimal multicut}). \end{aligned}$$

All inequalities must hold with equality.

5 More on correlation clustering and multicuts

The two way reduction we just presented proves that the correlation clustering problem and the multicut problem are essentially identical problems. Every exact solution to one implies an exact solution to the other. Every polynomial time approximation algorithm with a constant, logarithmic, or polylogarithmic approximation factor for either problem translates into a polynomial time approximation algorithm with a constant, logarithmic or polylogarithmic approximation factor, respectively, for the other. (We use this prove an $O(\log n)$ approximation in section 3).

From this it also follows that hardness of approximation results transfer from one problem to the other. Since the multicut problem is APX-hard and remains APX-hard even in the unweighted case it implies that unweighted correlation clustering problem is itself APX hard.

An interesting observation is that [1] give a constant factor approximation for the unweighted complete graph. This implies that the unweighted multicut problem where every two nodes u, v, are either connected by an edge or $\langle u, v \rangle$ is a source/sink pair has a constant factor approximation.

On the other hand, correlation clustering problems where $G^{\langle + \rangle}$ is a planner graph or has a tree structure has a constant factor approximation (as follows from [13, 6]).

Addendum: We recently learned that two other groups, Erik D. Demaine and Nicole Immorlica [3] and Charikar, Guruswami, and Wirth [12], have both independently obtained similar results (using somewhat different techniques).

References

1. Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Foundations of Computer Science (FOCS)*, pages 238–247, 2002.
2. Gruia Calinescu, Cristina G. Fernandes, and Bruce Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. *proceedings of the 6^{th} Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 1998.
3. Demaine Erik D. and Immorlica Nicole. Correlation clustering with partial information. *APPROX*, 2003.
4. E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, and M. Yannakakis. The complexity of multiway cuts. *Proceedings, 24^{th} ACM Symposium on Theory of Computing*, pages 241–251, 1992.
5. E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 4(23):864–894, 1994.
6. N. Garg, V. Vazirani, and M. Yannakakis. Primal$-$Dual Approximation Algorithms for Integral Flow and Multicut in Trees, with Applications to Matching and Set Cover. *Proceedings of ICLP*, pages 64–75, 1993.
7. Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. 25^{th} *STOC.*, pages 698–707, 1993.
8. T.C. Hu. Multicommodity network flows. *Operations Research*, (11):344360, 1963.
9. D. Klein, S. D. Kamvar, and C. D Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002.
10. Tom Leighton and S. Rao. An approximate max-flow mincut theorem for uniform multicommodity flow problems with applications to approximation algorithms. *In Proc. of the 29^{th} IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 422–431, 1988.
11. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Symposium on Math, Statistics,and Probability*, pages 281–297, 1967.
12. Venkat Guruswami Moses Charikar and Tony Wirth. Personal communication. 2003.
13. E. Tardos and V. V. Vazirani. Improved bounds for the maxflow minmulticut ratio for planar and $K_{r,r}-$free graphs. *Information Processing Letters*, pages 698 – 707, 1993.
14. K. Wagstaff and C. Cardie. Clustering with instance-level constraints. *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110, 2000.
15. K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.
16. M. Yannakakis, P. C. Kanellakis, S. C. Cosmadakis, and C. H. Papadimitriou. Cutting and partitioning a graph after a fixed pattern. *Proceedings, 10^{th} Intl. Coll. on Automata, Languages and Programming*, page 712722, 1983.