A Polynomial Time Approximation Scheme for k-Consensus Clustering[*]

Tom Coleman[†]   Anthony Wirth[‡]

Abstract

This paper introduces a polynomial time approximation scheme for the *metric* Correlation Clustering problem, when the number of clusters returned is bounded (by k). Consensus Clustering is a fundamental aggregation problem, with considerable application, and it is analysed here as a metric variant of the Correlation Clustering problem. The PTAS exploits a connection between Correlation Clustering and the k-cut problems. This requires the introduction of a new rebalancing technique, based on minimum cost perfect matchings, to provide clusters of the required sizes.

Both Consensus Clustering and Correlation Clustering have been the focus of considerable recent study. There is an existing dichotomy between the k-restricted Correlation Clustering problems and the unrestricted versions. The former, in general, admit a PTAS, whereas the latter are, in general, APX-hard. This paper extends the dichotomy to the metric case, responding to the result that Consensus Clustering is APX-hard to approximate.

1 Introduction

In this paper, we present a polynomial-time approximation scheme for the k-Consensus Clustering problem. The interest in this problem, and our solution to it, lies in the following three areas.

Application Consensus Clustering, in which we are asked to produce a single clustering that best summarizes a number of input clusterings, is a natural example of an aggregation problem. It bears some similarity to Rank Aggregation, which is useful for combining the outputs of a collection of search tools. There will be situations in which a user wishes to restrict the number of clusters returned from this process, perhaps due to some prior knowledge, which motivates the study of k-Consensus Clustering.

Answering a theoretical question With the particular metric we use to describe the distance between two clusterings, the Consensus Clustering problem becomes a special case of the metric-Correlation Clustering problem. There is a dichotomy between variants of the Correlation Clustering problem in which there is no restriction on the number of clusters returned, and those that have a (k-) bound on the number of clusters returned. The latter are frequently easier to approximate (see below), and we show that this dichotomy extends to metric variants. There has been considerable recent interest in the approximability of the Consensus Clustering [1] and Rank Aggregation problems [6, 20]. In particular, Bonizzoni *et al.* showed that Consensus Clustering is APX-hard [5]; our paper is a response to this result.

We have recently learnt that, in work simultaneous to ours, two other papers have each produced a PTAS for k-Consensus Clustering [4, 19]. Bonizzoni *et al.*'s result is an extension of Giotis and Guruswami's [16] PTAS for Correlation Clustering on 0/1-weighted graphs. Karpinski and Schudy's PTAS is part of a broader study of PTASes for Constraint Satisfaction Problems. The PTAS in this paper extends to the (more general) metric-Correlation Clustering problem.

Theoretical techniques This paper is the first to develop an explicit link between results and proof approaches to k-cut problems and those for k-

CORRELATION CLUSTERING (also called k-CC) problems. We observe that once the optimum cluster sizes have been determined—in a k-restricted problem this can be done by *guessing*—we ought to be able to transform k-CUT-inspired algorithms to k-CC. Developing the techniques to achieve this is the nub of this paper.

With these points in mind, we open the paper with some background concerning the CONSENSUS CLUSTERING and CORRELATION CLUSTERING problems, and their link to k-CUT problems.

2 Background

2.1 Problem definitions In the CONSENSUS CLUSTERING problem, we are given m input clusterings $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$ of some collection of items and a metric Δ on the clusterings. The task is to return a single clustering, $\mathcal{C}$, of the collection that minimizes $\sum_i \Delta(\mathcal{C}, \mathcal{C}_i)$, that is, a median clustering.

The Mirkin metric, commonly used in the theory community, states that the distance between two clusterings is the number of pairs of items that are separated in one clustering yet co-clustered in the other clustering. We now cast this as a graph problem and, therefore, view pairs of items as edges in a graph. If we let $w^c(e)$ be the proportion of the $\{\mathcal{C}_i\}$ that separate the endpoints of e and $w^u(e)$ be the proportion that cluster them together, then our objective is to minimize

$$(2.1) \qquad \sum_{e \in E_c(\mathcal{C})} w^u(e) + \sum_{e \in E_u(\mathcal{C})} w^c(e),$$

where $E_c(\mathcal{C})$ are the edges that are cut by $\mathcal{C}$ and $E_u(\mathcal{C})$ those not cut by $\mathcal{C}$. In fact, Equation 2.1 is exactly the MIN-CC objective [1]; we note in passing that the complementary MAX-CC objective is

$$\sum_{e \in E_c(\mathcal{C})} w^c(e) + \sum_{e \in E_u(\mathcal{C})} w^u(e).$$

Now, when the edge weights w^u and w^c are generated from a CONSENSUS CLUSTERING instance, the w^c quantities obey the *triangle inequality*. We can therefore view CONSENSUS CLUSTERING as a specific case of the metric version of MIN-CC, which is the problem that will be the principal focus of this paper. Henceforth, we will refer to w^c as δ and w^u as $1 - \delta$. The CONSENSUS CLUSTERING MIN-CC and MAX-CC problems, and relatives, can be specified to have an upper bound, k, on the number of clusters produced, which is of utmost relevance to us here.

2.2 Our observations There has been some prior work on PTASes for the metric k-CUT and MIN-k-UNCUT problems, which we introduce below. The standard objectives for these problems are MAX-k-CUT$(\mathcal{C}) =$

$\sum_{e \in E_c(\mathcal{C})} \delta(e)$ and MIN-k-UNCUT$(\mathcal{C}) = \sum_{e \in E_u(\mathcal{C})} \delta(e)$. The inspiration for the proof approach in this paper came from the following observation.

(2.2a)

$$\text{MAX-}k\text{-CC}(\mathcal{C})$$
$$= 2 \sum_{E_c(\mathcal{C})} \delta(e) + \sum_{E_u(\mathcal{C})} (1 - \delta(e)) - \sum_{E_c(\mathcal{C})} \delta(e)$$
$$= 2 \,\text{MAX-}k\text{-CUT}(\mathcal{C}) + |E_u(\mathcal{C})| - \Delta$$

(2.2b)

$$\text{MIN-}k\text{-CC}(\mathcal{C})$$
$$= 2 \sum_{E_u(\mathcal{C})} \delta(e) + \sum_{E_c(\mathcal{C})} (1 - \delta(e)) - \sum_{E_u(\mathcal{C})} \delta(e)$$
$$= 2 \,\text{MIN-}k\text{-UNCUT}(\mathcal{C}) + |E_c(\mathcal{C})| - \Delta,$$

where Δ is the sum of the edge weights: $\sum_e \delta(e)$ (a constant). If we find a good approximation to MIN-k-UNCUT with the same cluster sizes (and thus the same number of edges cut) as the optimal MIN-k-CC solution, we have a good approximation to MIN-k-CC. The existing algorithms for the metric MIN-k-UNCUT problem do not, however, provide a sufficient guarantee concerning the consequent cluster sizes.

As far as the authors are aware, there is no previous work on metric MIN-k-UNCUT cut problems with constrained cluster sizes. Fernandez de la Vega *et al.* [10] made some progress with a PTAS for the metric-MIN-BISECTION problem; they *conjecture* that it could lead to an algorithm for metric-MIN-2-UNCUT with constrained cluster sizes.

Our contribution, therefore, is to add significant components to these algorithms so that we can obtain appropriate cluster sizes. We achieve this principally through a novel *re-balancing step*, based on a BIPARTITE MATCHING problem, which we analyse in Section 5.3. There are many other technical details that we develop to incorporate the re-balancing.

2.3 Applications Gionis *et al.* highlight a number of applications of CONSENSUS CLUSTERING algorithms. These include: clustering categorical data, where we view each category as a separate clustering of the same data, clustering heterogeneous data, dealing with missing values in some attribute, inferring the correct number of clusters (though not in the k-restricted case), detecting outliers, making more robust clusterings, and preserving privacy [15]. Filkov and Skiena apply CONSENSUS CLUSTERING to the task of combining clustering solutions from diverse microarray data sources, incorporating noise elimination [13].

2.4 Prior work on Consensus Clustering and Correlation Clustering Table 1 summarizes the approximation results for the variants of Consensus Clustering, providing a context for our contributions. We note a few results that are of particular relevance to this paper.

The Correlation Clustering problem was first introduced to the theory community by Bansal *et al.* [3]. Giotis and Guruswami [16] were the first to investigate the fixed-k (number of clusters) case, providing a PTAS for 0/1 instances for both MIN-k-CC and MAX-k-CC. It is was not clear to us how to extend their result to arbitrary, or even metric or consensus-based weights; Bonizzoni *et al.* were, however, able to exploit a connection to the Giotis and Guruswami PTAS [4]. In the context of *metric* weights, Ailon *et al.* [1] showed that their LP-based algorithm is a 2-approximation for general metric-MIN-CC and, further, an 11/7-approximation for Consensus Clustering. The minimisation version of Consensus Clustering is known to be APX-hard even with just three input clusterings [5].

2.5 Outline of paper In the next section, we then provide an overview of the theoretical techniques, before developing them in full in the remainder of the paper. In Section 4 we describe the algorithms in sufficient detail to understand their operation. Their full details, and the justification of their properties, are developed in Section 5. The Appendix contains all proofs of theorems, lemmas, *et cetera* that are not presented in the main text.

3 Proof Techniques

In order to understand our PTASes for MAX-k-CC and MIN-k-CC, we must grasp the basic intuition behind the PTASes for the metric MIN-k-UNCUT problem [18, 11].

3.1 Maximization The maximization PTAS for metric-MAX-k-CUT was developed by Fernandez de la Vega and Kenyon [12], based on (amongst others) the dense-MAX-CUT PTAS of Arora *et al.* [2]. For k-CC, we can use another dense-MAX-k-CUT PTAS, by Frieze and Kannan [14], in a straightforward way to give a PTAS for the metric-MAX-k-CC problem, exploiting that PTAS's ability to work on negatively-weighted instances. We demonstrate this approach in Section 4.1.

3.2 Minimization The idea is to employ a maximization PTAS for a minimization problem. For some instances, a good maximization approximation is not good for the minimization objective, as the optimal minimization objective is very low. However, such instances have (other) special properties. Indeed, Indyk [18], and subsequently Fernandez de la Vega *et al.* [11], exploit a *well-separatedness* quality of MIN-k-UNCUT instances that are not approximable by the MAX-k-CUT PTAS. They show that an algorithm based on representative sampling will lead to a good-quality solution to the metric-MIN-k-UNCUT problem. In Section 5.4, we will show that the pairs of clusters in MIN-k-CC instances that the MAX-k-CC PTAS does not approximate well also share *the same* well-separatedness quality, and thus can be separated by representative sampling.

Phases of the PTAS Fernandez de la Vega *et al.*'s algorithm for metric MIN-k-UNCUT operates in three basic phases. In phase one, they sample representatives in order to approximately separate *large* clusters from one another.[1] In phase two, they extract *small* clusters (which could not be obtained by sampling). In phase three they apply a maximisation PTAS to properly separate the large clusters that could not be easily separated in phase one, because they were too *close* to one another.

For the MIN-k-CC problem, in addition to exploiting the closeness of the clusters, a PTAS would need to follow Equation 2.2b. That is, the PTAS would need to ensure that the number of edges cut is the same as the optimal solution; this is ensured by obtaining the *correct* cluster sizes. Fernandez de la Vega *et al.*'s algorithm makes no guarantee about cluster sizes, and there is no straightforward way to ensure that it does. We therefore develop a new phase two, which achieves Fernandez de la Vega *et al.*'s purpose (separating small clusters), but has the additional, more important property of ensuring correct sizes of all clusters.

A Perfect Matching We achieve the correct cluster sizes in phase two by solving a perfect matching problem instance. In order to analyse this matching and bound the cost of the clustering that it generates, we develop a novel approach, which could be used for any application of such a matching step. The idea is to bound the *mistakes* that the matching makes in assigning points to clusters.

The key technique here is the use of *pairing functions*. These functions pair a point v that, at the end of phase two, should have, according to the optimal solution, been in a particular cluster, but was not, with another point, $p(v)$, that was placed in that cluster at the end of phase two, but should not have been. Through judicious choice of such a pairing function p, we obtain for each relevant mis-classified point, a small orbit of points that could be *simultaneously* re-classified to reach an alternative clustering that is also a potential solution to the matching problem. In this way, we can

---

[1]Concepts such as *large* will be defined carefully in Section 5.

Table 1: Summary of approximation results for k-CC and k-Cut type problems. Our contributions are indicated by (∗). Note that in most cases a problem is APX-hard if the number of output clusters is not restricted, but has a PTAS if there is a k-bound.

	MAX		MIN	
	unrestricted	fixed-k	unrestricted	fixed-k
k-Cut	N/A		N/A	
⊢ 0/1		0.878 [17, 21]		
⊢ dense		PTAS [2]		
⊢ metric		PTAS [12]		PTAS [18, 11]
k-CC				
⊢ 0/1	PTAS [3]	PTAS [16]	3 [3, 7, 1] APX-hard [7]	PTAS [16]
⊢ weighted	0.7666 [7, 22] APX-hard [7]		APX-hard [9, 7]	
⊢ metric		PTAS (∗)	2 [1]	PTAS (∗)
⊢ consensus		PTAS (∗)	11/7 [1] APX-hard [5]	PTAS (∗) [4, 19]

make the smallest number of changes possible to the optimal clustering in order to achieve the essential features of our output clustering (that is, the observance of group boundaries).

4 The Algorithms

4.1 Maximization The first step for the MAX-k-CC algorithm is to form a new set of edge weights (some of which may be negative) thus. Define $\bar{G} = (V, E, \bar{\delta})$, where $\bar{\delta} = 2\delta - 1$. The second step is to apply Frieze and Kannan's PTAS for dense MAX-k-CUT to $\bar{G}$ [14].

The following theorem of Frieze and Kannan, combined with the fact that the optimum solution must be at least $|E|/2$, shows that the two comprise a PTAS for metric MAX-k-CC.

THEOREM 1. (FRIEZE AND KANNAN [14], THM 1)
There is an algorithm, that given a graph $\bar{G} = (V, E, \bar{w})$, with weight function $\bar{w} : E \to [-1, 1]$, and a fixed $\epsilon > 0$, computes in polynomial time a k-clustering $\mathcal{C}$ such that:

$$\text{MAX-}k\text{-CUT}(\mathcal{C}) \geq \text{MAX-}k\text{-CUT}^* - \epsilon n^2$$

COROLLARY 1. *If $|E| \in \Omega(n^2)$, then the Frieze-Kannan algorithm, acting on $\bar{G}$ provides a PTAS for MAX-k-CC (on G).*

4.2 Minimization Recall that, following the style of Fernandez de la Vega *et al.*'s PTAS for MIN-k-UNCUT, our algorithm operates in a number of phases.

Phase Zero: Guess the size $|C_i|$ of each cluster in the optimal solution $\mathcal{C}$; we assume $|C_i| \geq |C_{i+1}|$. Given these sizes, let k_0 be the number of clusters

that are large, also guessed. Guess a function g that assigns the large clusters to groups; a group consists of a contiguous subsequence of the $\{C_i\}$ in which C_i and C_{i+1} are close. The small clusters are assigned to a single group, separate from the large clusters. For each large cluster C_i, uniformly sample a point $c_i \in V$. If c_i satisfies some centrality property, defined in Section 5, we say that c_i is a *representative* of C_i. Section 5 shows that this happens with sufficiently high probability for all large clusters.

Phase One: Define a clustering $\mathcal{D}^1$ by assigning each point $v \in V$ to the large cluster such that the quantity $|C_i|\delta(v, c_i)$—an estimate of $\sum_{x \in C_i} \delta(v, x)$—is minimised.

Phase Two: Define $\mathcal{D}^2$ by solving the following BIPARTITE MATCHING problem:

1. Guess the number of mistakes made by Phase One. That is, let $F_{ii'} = C_i \cap D^1_{i'}$, and guess the size of each $F_{ii'}$. For each i, i' we therefore wish to place $|F_{ii'}|$ points from $D^1_{i'}$ in D^2_i.

2. The cost of placing v in D^2_i we define to be

$$\tilde{\delta}_i(v) = \begin{cases} |C_i|\delta(v, c_i) & \text{if } C_i \text{ is large, or} \\ 0 & \text{if } C_i \text{ is small.} \end{cases}$$

3. We can choose a $\mathcal{D}^2$, subject to the $|F_{ii'}|$ terms, to minimise the total $\tilde{\delta}$ using an application of BIPARTITE MATCHING.

Phase Three: Consider the *cluster groups* formed by $\mathcal{D}^2$. Cluster those points assigned to the group of small clusters by recursing this algorithm. Each group of large clusters will in fact have its internal clusters determined by the MAX-k-CC PTAS described above; note that the group boundaries of $\mathcal{D}^2$ are observed.

5   Analysis of the PTAS for metric min-k-CC

To understand the analysis, we first need to understand the purpose of phases two and three. Phase three makes no *inter-group* changes: we need to ensure that $\mathcal{D}^2$ creates clusters that observe the optimal groups well. Phase three totally recreates the clusters within each group, forming clustering $\mathcal{D}^3$, so we are not so concerned with $\mathcal{D}^2$'s *intra-group* boundaries.

Instead of $\mathcal{D}^2$, we will consider another clustering $\mathcal{C}^2$: a combination of $\mathcal{D}^2$ and $\mathcal{C}$. $\mathcal{C}^2$ makes the same inter-group mistakes as $\mathcal{D}^2$, and is thus a candidate solution for phase three. However, within each group, $\mathcal{C}^2$ will be like $\mathcal{C}$, and thus will be a good solution to the CORRELATION CLUSTERING problem.

Our first aim is to show that $\mathcal{C}^2$ is a good MIN-k-CC approximation to $\mathcal{C}$. Our second is to prove that $\mathcal{D}^3$, the outcome of Phase Three, will indeed approximate $\mathcal{C}^2$, and thus $\mathcal{C}$, well.

5.1   Phase Zero   We must now define carefully a number of quantities and constructs.

- c^* is the MIN-k-UNCUT cost of the clustering $\mathcal{C}$ (which is optimal for the MIN-k-CC objective).

- If $\delta(u, v)$ is the distance between u and v, define $\delta(A, B) = \sum_{a \in A, b \in B} \delta(a, b)$, and $\delta(A) = \delta(A, A)$. Let $\delta_i(v) = \delta(\{v\}, C_i)$ be the distance between v and C_i from the optimal solution.

- Since we are describing a PTAS, we have been given an ϵ, so let $I_j = (\epsilon^{j+1}, \epsilon^j]$. Let $j_0 < k^2$ be the minimum j such that for every i, i', the ratio $|C_i|/|C_{i'}| \notin I_j$. Let $k_0 = \arg\max_i |C_i| \geq \epsilon^{j_0} |C_1|$. A cluster C_i is *large* if $i \leq k_0$ and *small* otherwise, and m is the size of the smallest large cluster. We can see by the above definition that the size of all small clusters is in $O(\epsilon)m$.

- A point c_i is a *representative* point of cluster C_i if $\delta_i(c_i) \leq 2\delta(C_i)/n_i$. There is a good chance that our sampled representatives will satisfy this definition:

  LEMMA 1. (F. DE LA VEGA *et al.*, LEMMA 5) *For each $i \leq k_0$ large, let c_i be chosen uniformly at random, and independently, from V. Then with*

probability at least $[\epsilon^{j_0}/(2k)]^k$: *each c_i chosen is a representative of C_i.*

If all c_i are representatives, then the following lemma holds:

LEMMA 2. (F. DE LA VEGA *et al.*, LEMMA 4) *For all $i \leq k_0$ large, $|\tilde{\delta}_i(v) - \delta_i(v)| \leq 2\delta(C_i)/|C_i|$.*

- Let $\beta = |C_1|/(m\epsilon)$. Let C_i, C_j be two large clusters. Then i and j are *close* if $\delta(C_i, C_j) \leq \beta[\delta(C_i) + \delta(C_j)]$. We now define a group mapping function $g : [k] \to \{0, 1, \dots, \gamma\}$, for some $\gamma \leq k_0$. If $i > k_0$, $g(i) = 0$. Otherwise, if $i, i' \leq k_0$, and there exists a sequence of indices $x_1 = i, x_2, \dots, x_\ell = i'$ such that C_{x_z} and $C_{x_{z+1}}$ are close, and $z < \ell$, then $g(i) = g(i')$. We say two clusterings $\mathcal{X}$ and $\mathcal{Y}$ are g-equivalent if $g(\mathcal{X}(v)) = g(\mathcal{Y}(v))$ for all $v \in V$.

5.2   Phase One   Phases one and two together form good group boundaries. The aim of phase one is to separate large clusters in a MIN-k-UNCUT sense (it is a step from a MIN-k-UNCUT algorithm). Phase one makes no attempt to separate small clusters, nor to ensure cluster sizes are correct (which would lead to a good MIN-k-CC solution). The only result we need to borrow from Fernandez de la Vega *et al.* concerns the number of items that are in *incorrect* clusters at the end of phase one.

LEMMA 3. (F. DE LA VEGA *et al.* LEMMA 11) *Let $i, i' \leq k_0, g(i) \neq g(i')$ be large clusters in different groups. Then $|F_{ii'}| \leq O(\epsilon)m$.*

This lemma arises from the definition of *closeness* and the properties of cluster representatives.

5.3   Phase Two   To analyse phase two, we define a clustering $\mathcal{C}^2$ and show that it is a good MIN-k-CC clustering of V. Remember that we want $\mathcal{C}^2$ to make the same inter-group mistakes as $\mathcal{D}^2$, whilst imitating $\mathcal{C}$ within groups. It is tempting to define $\mathcal{C}^2$ in a similar fashion to Fernandez de la Vega *et al.*'s $\mathcal{C}$ as:
(5.3)
$$\mathcal{C}_{\text{FKKR}}^2(v) = \begin{cases} \mathcal{D}^2(v) & \text{if } v \text{ grouped wrongly by } \mathcal{D}^2, \\ \mathcal{C}(v) & \text{otherwise.} \end{cases}$$

However, this clustering is unfair to $\mathcal{D}^2$; it ignores one main aim of $\mathcal{D}^2$, which is to get the cluster sizes right. If we instead choose a $\mathcal{C}^2$ that also has correct cluster sizes, we will be able to use (2.2) to relate $\mathcal{C}^2$ MIN-k-UNCUT performance to its MIN-k-CC performance. In addition, since we will use the fact that $\mathcal{D}^2$ is a minimum-cost matching, we will require that $\mathcal{C}^2$ also

be the result of a matching, albeit not one of minimum cost.

In order to achieve correct cluster sizes, we will have to relax the condition (5.3) slightly. As $\mathcal{D}^2$ has correct cluster sizes, for each v that is mis-clustered across group boundaries from cluster i to cluster i' by $\mathcal{D}^2$, there is another point u (which we will later call $p^{-1}(v)$) which is mis-clustered into cluster i by $\mathcal{D}^2$. Although u may *not* be mis-clustered across group boundaries, we will make sure that $\mathcal{C}^2$ also mis-clusters u.

In the next section we will precisely define a set V_1, with the property that

$$(5.4) \quad \{v \text{ is placed in the wrong group by } \mathcal{D}^2\} \subseteq V_1$$

Then we define:

$$\mathcal{C}^2(v) = \begin{cases} \mathcal{D}^2(v) & \text{if } v \in V_1, \\ \mathcal{C}(v) & \text{otherwise.} \end{cases}$$

Our main focus will be to ensure V_1 is not too large—we want $\mathcal{C}^2$ to be as similar to $\mathcal{C}$ as possible.

Defining V_1. Define $G_{ii'}$ to be $D_i^2 \cap D_{i'}^1$. The elements of $G_{ii'}$ are the vertices that move from cluster i' to cluster i in Phase Two. By definition, $|G_{ii'}| = |F_{ii'}|$. Indeed, $G_{ii'}$ is our attempt at finding $F_{ii'}$; if we had $G_{ii'} = F_{ii'}$ for all i, i', we would have $\mathcal{D}^2 = \mathcal{C}$.

For each (i, i'), the sets $F_{ii'}$ and $G_{ii'}$ are of the same size. We can therefore find a pairing function p (a bijection) between them. There are many choices of such pairing functions, but we will choose one that has the following property.

DEFINITION 1. *A pairing function p has* small-loops, *if, for each $v \in C_i$, the orbit of v (the set of points reachable from v by repeated application of p) enters each $G_{ii'}$ at most once for each $i \in [k]$.*

REMARK 1. *There exists a pairing function p with small-loops.*

Some properties of small-loop pairing functions p are immediate. Each vertex $v \in F_{ii'} \cap G_{ii'}$ has $p(v) = v$; consequently, each value $\mathcal{D}^2(v)$ is unique in a given orbit, and so every orbit must be of length at most k. Note also, from the definition, that an orbit is a subset of some D_i^1, all $\mathcal{D}^1(v)$ values are the same, and that $\mathcal{D}^2(p(v)) = \mathcal{C}(v)$, a fact that will be used often.

DEFINITION 2. *An orbit o is* group-contained *if for all $v \in o$, $g(\mathcal{C}(v)) = g(\mathcal{D}^1(v))$.*

Let V_1 be all vertices *not* in group-contained orbits. It is not hard to show that $g(\mathcal{D}^2(v)) = g(C(v))$ for all v in a group-contained orbit, and thus (5.4) holds. We notice

a few important properties of $\mathcal{C}^2$ immediately. The first pertains to the points where $\mathcal{C}^2$ differs from $\mathcal{C}$. Using Lemma 3, we can see that:

REMARK 2. *The size of the set V_1 is in $O(\epsilon m)$.*

The second tells us that $\mathcal{C}^2$ is indeed a candidate solution for Phase Three.

REMARK 3. *For all $v \in V$, $g(\mathcal{C}^2(v)) = g(\mathcal{D}^2(v))$.*

The min-k-uncut cost of $\mathcal{C}^2$. We now have a definition of $\mathcal{C}^2$; we must work towards our first goal, and bound its MIN-k-CC cost. In this section we will bound its MIN-k-UNCUT cost, as the correctness of its cluster sizes means that we can use (2.2) turn this into a bound on MIN-k-CC cost.

To begin with, we need a Lemma which is a simple consequence of the fact that $\mathcal{D}^2$ was defined using an optimal solution to the BIPARTITE MATCHING problem we set up in Phase Two.

LEMMA 4.

$$\sum_{v \in V_1} \tilde{\delta}_{\mathcal{D}^2(v)}(v) \leq \sum_{v \in V_1} \tilde{\delta}_{\mathcal{C}(v)}(v)$$

The pairing function p linked mis-clustered vertices with vertices that would have been better choices. In order to show that our mistakes are not too costly, we will show that these points are close together.

LEMMA 5.

$$\sum_{v \in V_1} \delta(v, p(v)) \leq O(1) \frac{c^*}{m}$$

Proof. We sketch the proof here; a full proof appears in the appendix.

Fix some $v \in V_1$. We know that v and $p(v)$ are co-clustered by $\mathcal{D}^1$, onto say cluster D_I^1, so we can use the triangle inequality to write:

$$|C_I|\delta(v, p(v)) \leq |C_I|\delta(v, c_I) + |C_I|\delta(p(v), c_I)$$
$$= \tilde{\delta}_I(v) + \tilde{\delta}_I(p(v)).$$

Straightforward, but technical, arguments show that as $v \in V_1$, either $\mathcal{C}(v)$ or $\mathcal{D}^2(v)$ must be large—so by the choice made in Phase One, the first term on the right is less than $\tilde{\delta}_{\mathcal{C}(v)}(v) + \tilde{\delta}_{\mathcal{D}^2(v)}(v)$.

Applying the same argument to $p(v)$ and the second term, summing over $v \in V_1$ gives

$$\sum_{v \in V_1} \delta(v, p(v)) \leq \frac{2}{m} \sum_{v \in V_1} \left[\tilde{\delta}_{\mathcal{C}(v)}(v) + \tilde{\delta}_{\mathcal{D}^2(v)}(v) \right]$$
$$\leq \frac{4}{m} \sum_{v \in V_1} \tilde{\delta}_{\mathcal{C}(v)}(v).$$

The second inequality follows from an application of Lemma 4.

Finally, we use the fact that $\tilde{\delta}_i$ approximates δ_i and the bound on $|V_1|$ along with the fact that $\sum_{v \in V} \delta_{\mathcal{C}(v)}(v) = c^*$ to complete the proof.

It is useful to split V_1 into two different families of subsets:

DEFINITION 3. *For each i, let $In_i = C_i^2 \setminus C_i$, and $Out_i = C_i \setminus C_i^2$, so that $C_i^2 = C_i + In_i - Out_i$.*

Note that $V_1 = \bigcup_{i \in [k]} In_i = \bigcup_{i \in [k]} Out_i$, and that $In_i = \{p(v) \mid v \in Out_i\}$, and so $|C_i^2| = |C_i|$.

LEMMA 6. $\mathcal{C}^2$ *is a $(1 + O(\epsilon))$ approximation to $\mathcal{C}$ in terms of* MIN-k-UNCUT *cost. Equivalently,*

$$\sum_i \delta(C_i^2) \leq (1 + O(\epsilon))c^*.$$

Proof. Again, we sketch the proof. We begin by expanding $\delta(C^2)$ in terms of In_i and Out_i.

$$\delta(C^2) = \sum_{i \in [k]} \Big(\delta(C_i) + 2\left[\delta(C_i, In_i) - \delta(C_i, Out_i)\right] +$$

$$[\delta(In_i) - \delta(In_i, Out_i)] +$$

$$[\delta(Out_i) - \delta(Out_i, In_i)] \Big).$$

Consider the second-last term,

$$\sum_{i \in [k]} \delta(In_i) - \delta(In_i, Out_i) \leq \sum_{i \in [k]} \sum_{u \in In_i} \sum_{v \in In_i} \delta(v, p(v))$$

(by the triangle inequality)

$$\leq |V_1| O(1) \frac{c^*}{m} \leq O(\epsilon)c^*.$$

(Remark 2 and Lemma 5).

We can bound the final term the same way. Now consider the second term, remembering the definitions of In_i and Out_i, and our notational conveniences:

$$\sum_{i \in [k]} \delta(C_i, In_i) - \delta(C_i, Out_i) =$$

$$\sum_{v \in V_1} \left[\delta_{\mathcal{C}^2(v)}(v) - \delta_{\mathcal{C}(v)}(v) \right]$$

Now, we can split this sum into those v which are clustered in large clusters by $\mathcal{D}^2$ and those which are not. For the first group, we can use Lemma 2 and our bound on $|V_1|$ to replace the δs with $\tilde{\delta}$s, followed by an application of Lemma 4 to form a bound. For the second group, as the total number of points in small clusters is $O(\epsilon)m$, we can use the triangle inequality and Lemma 5.

The min-k-CC cost of $\mathcal{C}^2$. Now that we have established that $\mathcal{C}^2$ is a good MIN-k-UNCUT solution, and that $|C_i^2| = |C_i|$, we can apply (2.2) to obtain the following theorem:

THEOREM 2. $\mathcal{C}^2$ *is a $(1 + O(\epsilon))$ approximation to $\mathcal{C}$ in terms of* MIN-k-CC *cost.*

5.4 Phase Three We have achieved our first aim. We have shown that there is a candidate solution $\mathcal{C}^2$ that phase three could find, which is a MIN-k-CC approximation to $\mathcal{C}$. We now show that in fact phase three will approximate $\mathcal{C}^2$, and thus $\mathcal{C}$.

As Phase Three operates on one group of clusters at a time, we now need to show that the approximate solutions that phase three obtains on groups are sufficient for our purposes. Let $G|_j = \bigcup \{\mathcal{D}_i^2, g(i) = j\}$. For the group of small clusters, we just recurse the MIN-k-CC algorithm; since $|G|_0| \leq \epsilon|V|$, we can apply an inductive argument.

For the groups of large clusters, phase three uses the MAX-k-CC PTAS of Section 4.1. Remember that a maximisation PTAS can be used for a minimisation problem if the minimisation cost is not too low. We will use the fact that groups of clusters are *not* well-separated to show that indeed the MIN-k-CC is high.

We begin with the relevant lemma from Fernandez de la Vega *et al.*, which shows that this is the case for the MIN-k-UNCUT objective.

LEMMA 7. (F. DE LA VEGA *et al.*) *Let j be a group of clusters. Then*

$$\sum_{g(i)=j} \delta(\mathcal{C}_i) \geq \Omega(\epsilon^{3j_0+1}) \sum_{u,v \in G|_j} \delta(u, v).$$

This lemma tells us that a MAX-k-CUT PTAS will work on a group, and this is how Fernandez de la Vega *et al.* use it. But we can turn it into a similar statement about MIN-k-CC using the following technical lemma:

LEMMA 8. *Let $\mathcal{C}$ be a clustering of G. Let $M = \max_i |C_i|$. Suppose that there exist f and g such that for all i, $|C_i| \geq f(\epsilon)M$, and that*

(5.5) $\qquad g(\epsilon)$ MAX-k-CUT$(\mathcal{C}) <$ MIN-k-UNCUT$(\mathcal{C})$,

then

(5.6) $\qquad \dfrac{f^2(\epsilon)g(\epsilon)}{2}$ MAX-k-CC$(\mathcal{C}) <$ MIN-k-CC$(\mathcal{C})$.

If we let MIN-k-CC$|_j(\mathcal{X})$ be the the MIN-k-CC cost of $\mathcal{X}$ as a clustering of the set $G|_j$, we have:

COROLLARY 2.

$$\text{MIN-}k\text{-CC}|_j(\mathcal{C}^2) \geq \Omega(\epsilon^{5j_0+2})\binom{|G|_j|}{2}.$$

Corollary 2 and the inductive argument about the group of small clusters show us that Phase Three will approximate $\mathcal{C}^2$ within each group. Finally, we need another technical lemma to ensure that a clustering which is good on each group is good on the entirety of V.

LEMMA 9. *Suppose $\mathcal{X}$ and $\mathcal{Y}$ are g-equivalent clusterings. If for each j,*

$$\text{MIN-}k\text{-CC}|_j(\mathcal{X}) \leq (1 + f(\epsilon))\text{MIN-}k\text{-CC}|_j(\mathcal{Y})$$

Then, over all of V,

$$\text{MIN-}k\text{-CC}(\mathcal{X}) \leq (1 + f(\epsilon))\,\text{MIN-}k\text{-CC}(\mathcal{Y})\,.$$

THEOREM 3. *The algorithm in Section 4.2 is a PTAS for the MIN-k-CC problem.*

Proof. Lemma 9 along with Corollary 2 is enough to show that Phase Three will find a good approximation to the best solution g-equivalent to $\mathcal{D}^2$ that exists. Theorem 2 has shown that $\mathcal{C}^2$, which is g-equivalent to $\mathcal{D}^2$, is a good approximation to $\mathcal{C}$. Thus we have the required bound on the quality of the solution of $\mathcal{D}^3$. The running time of the algorithm is polynomial, as it is the combination of PTASes, the guessing of $O(k)$ values, and the solution of a BIPARTITE MATCHING problem.

6 Conclusions

The main contribution of this paper was a PTAS for the metric-MIN-k-CC problem, a generalization of the k-CONSENSUS CLUSTERING problem, which has a number of key applications. The minimization PTAS has a novel rebalancing step, involving a minimum-cost perfect bipartite matching, and complements the APX-hardness of CONSENSUS CLUSTERING. We anticipate that this rebalancing technique will find application in other situations where the sizes of clusters are constrained.

Naturally, the PTAS presented here, like that of Giotis and Guruswami [16], is too slow to run in practice. It would be fruitful, therefore, to look for local-search approaches that have approximation guarantees, as we have done in the past [8].

References

[1] AILON, N., CHARIKAR, M., AND NEWMAN, A. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM 55*, 5 (2008), 1–27.

[2] ARORA, S., KARGER, D., AND KARPINSKI, M. Polynomial time approximation schemes for dense instances of NP-hard problems. *Journal of Computer and System Sciences 58*, 1 (1999), 193–210.

[3] BANSAL, N., BLUM, A., AND CHAWLA, S. Correlation clustering. *Machine Learning 56*, 1 (2004), 89–113.

[4] BONIZZONI, P., DELLA VEDOVA, G., AND DONDI, R. A ptas for the minimum consensus clustering problem with a fixed number of clusters. In *Proceedings of the Eleventh Italian Conference on Theoretical Computer Science* (2009).

[5] BONIZZONI, P., DELLA VEDOVA, G., DONDI, R., AND JIANG, T. On the approximation of correlation clustering and consensus clustering. *Journal of Computer and System Sciences 74*, 5 (2008), 671–96.

[6] CHARBIT, P., THOMASSE, S., AND YEO, A. The minimum feedback arc set problem is NP-hard for tournaments. *Combinatorics, Probability and Computing 16*, 1 (2006), 1–4.

[7] CHARIKAR, M., GURUSWAMI, V., AND WIRTH, A. Clustering with qualitative information. *Journal of Computer and System Sciences 71*, 3 (2005), 360–83.

[8] COLEMAN, T., SAUNDERSON, J., AND WIRTH, A. A local-search 2-approximation for 2-correlation clustering. In *Proceedings of the Sixteenth Annual European Symposium on Algorithms* (2008), pp. 308–19.

[9] DEMAINE, E., EMANUEL, D., FIAT, A., AND IMMORLICA, N. Correlation clustering in general weighted graphs. *Theoretical Computer Science 361*, 2-3 (2006), 172–87.

[10] FERNANDEZ DE LA VEGA, W., KARPINSKI, M., AND KENYON, C. Approximation schemes for metric bisection and partitioning. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2004), pp. 506–15.

[11] FERNANDEZ DE LA VEGA, W., KARPINSKI, M., KENYON, C., AND RABANI, Y. Approximation schemes for clustering problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of Computing* (2003), pp. 50–8.

[12] FERNANDEZ DE LA VEGA, W., AND KENYON, C. A randomized approximation scheme for metric MAX-CUT. *Journal of Computer and System Sciences 63*, 4 (2001), 531–41.

[13] FILKOV, V., AND SKIENA, S. Integrating microarray data by consensus clustering. In *Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI)* (2003), pp. 418–25.

[14] FRIEZE, A., AND KANNAN, R. The regularity lemma and approximation schemes for dense problems. In *Proceedings of the Thirty-Seventh Annual IEEE Symposium on Foundations of Computer Science* (1996), pp. 12–20.

[15] GIONIS, A., MANNILA, H., AND TSAPARAS, P. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD) 1*, 1 (2007).

[16] GIOTIS, I., AND GURUSWAMI, V. Correlation clustering with a fixed number of clusters. *Theory of Computing 2*, 1 (2006), 249–66.

[17] GOEMANS, M., AND WILLIAMSON, D. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM 42* (1995), 1115–45.

[18] INDYK, P. A sublinear time approximation scheme for clustering in metric-spaces. In *Proceedings of the Fortieth Annual IEEE Symposium on Foundations of Computer Science* (1999), pp. 154–9.

[19] KARPINSKI, M., AND SCHUDY, W. Linear time approximation schemes for the Gale-Berlekamp game and related minimization problems. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (2009), pp. 313–22.

[20] KENYON-MATHIEU, C., AND SCHUDY, W. How to rank with few errors. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing* (2007), pp. 95–103.

[21] KHOT, S., KINDLER, G., MOSSEL, E., AND O'DONNELL, R. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? In *Proceedings of the Forty-Fifth Annual IEEE Symposium on Foundations of Computer Science* (2004), pp. 146–54.

[22] SWAMY, C. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2004), pp. 526–7.

A Proofs

A.1 Proof of max-k-CC PTAS First we prove that a solution to the negatively-weighted MAX-k-CUT problem that we set up in Section 4.1 will indeed provide a solution to our MAX-k-CC problem. Note that $e_c(\mathcal{C})$ is defined to be $|E_c(\mathcal{C})|$, and $e_u(\mathcal{C})$ is $|E_u(\mathcal{C})|$.

REMARK 4. *Let $G = (V, E, \delta)$ be a weighted graph. Define $\bar{G} = (V, E, \bar{\delta})$, where $\bar{\delta} = 2\delta - 1$. Then, for any k-clustering $\mathcal{C}$,*

$$\text{MAX-}k\text{-CC}(G, \mathcal{C}) = \text{MAX-}k\text{-CUT}(\bar{G}, \mathcal{C}) + |E| - \Delta.$$

Proof.

$$
\begin{aligned}
\text{MAX-}k\text{-CC}(G, \mathcal{C}) &= \text{MAX-}k\text{-CUT}(G, \mathcal{C}) + e_u(\mathcal{C}) \\
&\quad - \text{MIN-}k\text{-UNCUT}(G, \mathcal{C}) \\
&= \text{MAX-}k\text{-CUT}(G, \mathcal{C}) + \big[|E| - e_c(\mathcal{C})\big] \\
&\quad + \big[\text{MAX-}k\text{-CUT}(G, \mathcal{C}) - \Delta)\big] \\
&= 2\,\text{MAX-}k\text{-CUT}(G, \mathcal{C}) \\
&\quad - e_c(\mathcal{C}) + |E| - \Delta \\
&= \sum_{e \in E_c(\mathcal{C})} (2w(e) - 1) + |E| - \Delta \\
&= \text{MAX-}k\text{-CUT}(\hat{G}, \mathcal{C}) + |E| - \Delta.
\end{aligned}
$$

Remark 4 tells us, for instance, that the optimal solution to MAX-k-CUT on $\bar{G}$ is the same as the optimal solution of MAX-k-CC on G.

Proof of Corollary 1

Proof. Let $\mathcal{C}$ be the solution returned by algorithm FK, run on $\hat{G}$, and $\mathcal{C}^*$ be the optimal solution to MAX-k-CC. Then, by Remark 4,

$$
\begin{aligned}
\text{MAX-}&k\text{-CC}^* - \text{MAX-}k\text{-CC}(\mathcal{C}) \\
&= \text{MAX-}k\text{-CUT}(\hat{G}, \mathcal{C}^*) - \text{MAX-}k\text{-CUT}(\hat{G}, \mathcal{C}) \\
&\leq \text{MAX-}k\text{-CUT}(\hat{G})^* - \big[\text{MAX-}k\text{-CUT}(\hat{G})^* - \epsilon n^2\big] \\
&= \epsilon n^2
\end{aligned}
$$

and we are done, since MAX-k-CC* is in $\Omega(n^2)$.

A.2 Proofs from Phase Two

Proof of Remark 1

Proof. It is simple to find such a pairing function; suppose we have a pairing function p such that $v_1, v_2 \in G_{i'i}$ are in the same orbit. That is, $p^x(v_1) = v_2$ for some x. Then, as v_1, v_2 are both in $G_{i'i}$, there are two nodes $w_1, w_2 \in F_{i'i}$ such that $p(w_1) = v_1$ and $p(w_2) = v_2$. Then we can define a new pairing function p' such that
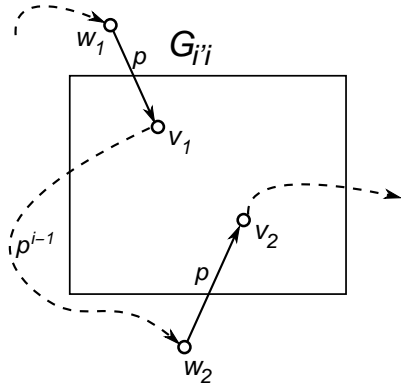
(a) Pairing function p: v_1 and v_2 are on the same orbit, yet both in $G_{i'i}$.
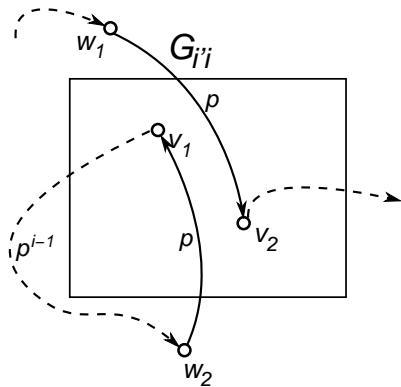
(b) Pairing function p': v_1 and v_2 are now on two distinct orbits.

Figure 1: Changing a pairing function p to p' in order to make smaller loops.

$p'(w_2) = v_1$, $p'(w_1) = v_2$, and $p'(w) = p(w)$ otherwise. The procedure is demonstrated in Figure 1.

After this change, p' is still a pairing function, and v_1 and v_2 are now on separate orbits (v_1 on a loop of length x). Repeated application of the above procedure will lead to a pairing function with small loops. Since points that were not on the same orbit before the procedure are not on the same orbit after the procedure, the repeated application must terminate.

Proof of Remark 2

Proof. By definition, each non-group contained orbit has some vertex with $g(\mathcal{D}^1(v)) \neq g(\mathcal{C}(v))$, that is, in $F_{ii'}$ with $g(i) \neq g(i')$. From Lemma 3, we know that $\sum_{g(i) \neq g(i')} |F_{ii'}| \in O(k^2 \epsilon m)$, if $i, i' \leq k_0$. In addition, if $i > k_0$, then $|F_{ii'}| \leq |C_i|$, as it is a subset, and this is less than ϵm. Each (small-loop) orbit has at most k vertices, which proves the Remark.

Proof of Remark 3

Proof. Clearly points in V_1 are g-equivalent. If $v \in V \setminus V_0$, then also $p^{-1}(v) \in V \setminus V_0$, and so we know $g(\mathcal{C}(p^{-1}(v))) = g(\mathcal{D}^1(p^{-1}(v)) = \mathcal{D}^1(v)) = g(\mathcal{C}(v))$. Now, we know by definition that $\mathcal{D}^2(v) = \mathcal{C}(p^{-1}(v))$ and that $\mathcal{C}^2(v) = \mathcal{C}(v)$, completing the proof.

Proof of Lemma 4

Proof. To prove this, we appeal to the optimality of $\mathcal{D}^2$ as a solution to the BIPARTITE MATCHING problem on $\tilde{G}$. Consider another solution to BIPARTITE MATCHING, μ, defined as $\mu(v) = \mathcal{C}(v)$ for $v \in V_1$ and $\mu(v) = \mathcal{D}^2(v)$ otherwise. This function indeed provides a matching, as $\mathcal{C}(v) = \mathcal{D}^2(p(v))$ for any v, and no orbit enters or leaves V_1. So, considering the costs of the two matchings, as $\mathcal{D}^2$ is optimal, and only differs from μ on V_1, the lemma follows.

Full Proof of Lemma 5 The only important detail that is omitted from the main text is why for any $v \in V_1$ either $\mathcal{C}(v)$ or $\mathcal{D}^2(v)$ is large. This follows from a corollary to the following remark:

REMARK 5. *Without loss of generality, if $g(\mathcal{C}(v)) = g(\mathcal{D}^2(v)) = 0$, then $\mathcal{C}(v) = \mathcal{D}^2(v)$.*

Proof. Suppose this is not the case for some v satisfying $g(\mathcal{D}^2(v)) = g(\mathcal{C}(v)) = 0$, but not $\mathcal{D}^2(v) = \mathcal{C}(v)$. We will define a new optimal solution to the BIPARTITE MATCHING problem, $\mathcal{D}^{2'}$, which is g-equivalent to $\mathcal{D}^2$. Repeating in this fashion for all v failing the condition will lead to a clustering that satisfies Remark 5. Define $\mathcal{D}^{2'}$ identically to $\mathcal{D}^2$, except let $D^{2'}(v) = C(v)$, which is of course $D^2(p^{-1}(v))$, and $D^{2'}(p^{-1}(v)) = D^2(v)$. Thus v and $p^{-1}(v)$ are exchanging roles. We can then reduce p to regain the small loops property. As $g(\mathcal{D}^2(v)) = g(\mathcal{D}^{2'}(v)) = 0$ the assignment cost in the BIPARTITE MATCHING is unchanged, so $\mathcal{D}^{2'}(v)$ is an optimal solution. Moreover, $\mathcal{D}^{2'}$ is g-equivalent to $\mathcal{D}^2$, and has clusters of the same sizes as $\mathcal{D}^2$. So none of the later of the analysis will be affected, and we assume $\mathcal{D}^2$ has the required property.

COROLLARY 3. *No orbit has two consecutive nodes mapped to small clusters by $\mathcal{C}$.*

Proof. Suppose that $g(\mathcal{C}(p^{-1}(v))) = g(\mathcal{C}(v)) = 0$ for some $v \neq p(v)$. This would imply that $g(\mathcal{D}^2(v)) = 0$ also, and therefore from Remark 5, that $\mathcal{D}^2(v) = \mathcal{C}(v)$. Finally, this implies $\mathcal{D}^2(v) = \mathcal{D}^2(p(v))$, which breaks the small loops property.

Full Proof of Lemma 6 To begin with, we prove the statements made relating In_i to Out_i:

Proof. Let $X = \{p^{-1}(v) \mid v \in \text{In}_i\}$. If $v \in \text{In}_i$, then $\mathcal{C}^2(v) = i \neq \mathcal{C}(v)$ and v is not in a group-contained orbit. Let $u = p^{-1}(v)$, then $u \in X$, and $\mathcal{C}(u) = \mathcal{D}^2(p(u)) = \mathcal{D}^2(v) = \mathcal{C}^2(v) = i$, which also implies $u \neq v$. On the other hand, $\mathcal{C}^2(u) \neq \mathcal{C}^2(v) = i$, by the small loops property, so $u \in \text{Out}_i$. Consequently, $X \subseteq \text{Out}_i$. A similar argument in the other direction shows that $\text{Out}_i \subseteq X$.

We break the sketch up into a few parts:

REMARK 6.

$$\delta(C_i^2) - \delta(C_i) \leq 2\left[\delta(C_i, \text{In}_i) - \delta(C_i, \text{Out}_i)\right] + O(\epsilon)c^* .$$

Proof. Expanding $\delta(C_i^2)$ gives

$$\delta(C_i^2) = \delta(C_i) + 2\left[\delta(C_i, \text{In}_i) - \delta(C_i, \text{Out}_i)\right]$$
$$+ \left[\delta(\text{In}_i) - \delta(\text{In}_i, \text{Out}_i)\right]$$
$$+ \left[\delta(\text{Out}_i) - \delta(\text{Out}_i, \text{In}_i)\right] .$$

Consider the second-last term,

$$\delta(\text{In}_i) - \delta(\text{In}_i, \text{Out}_i) = \sum_{u \in \text{In}_i} \left[\delta(u, \text{In}_i) - \delta(u, \text{Out}_i)\right]$$
$$= \sum_{\substack{u \in \text{In}_i \\ v \in \text{Out}_i}} \left[\delta(u, p(v)) - \delta(u, v)\right]$$
$$\leq \sum_{\substack{u \in \text{In}_i \\ v \in \text{Out}_i}} \delta(v, p(v))$$
(triangle inequality)
$$\leq |V_1| \sum_{v \in V_1} \delta(v, p(v))$$
($\text{In}_i, \text{Out}_i \subseteq V_1$)
$$\leq O(\epsilon)c^* .$$
(Remark 2 and Lemma 5)

We can bound the final term in the same way, which completes the proof.

REMARK 7.

$$\sum_{i \in [k]} \delta(C_i, \text{In}_i) - \delta(C_i, \text{Out}_i)$$
$$= \sum_{v \in V_1} \left[\delta_{\mathcal{C}^2(v)}(v) - \delta_{\mathcal{C}(v)}(v)\right]$$

Proof. From the statements relating In_i and Out_i:

$$\sum_{i \in [k]} \delta(C_i, \text{In}_i) - \delta(C_i, \text{Out}_i)$$
$$= \sum_{i \in [k]} \sum_{v \in \text{In}_i} \left[\delta_i(v) - \delta_i(p^{-1}(v))\right]$$
$$= \sum_{i \in [k]} \sum_{v \in \text{In}_i} \left[\delta_{\mathcal{C}^2(v)}(v) - \delta_{\mathcal{C}(p^{-1}(v))}(p^{-1}(v))\right]$$

For a vertex u in V_1 that is not in some In_i, $p(u) = u$, so the corresponding term inside the brackets above is zero. Therefore the right hand side is, after separating the terms,

$$\sum_{v \in V_1} \delta_{\mathcal{C}^2(v)}(v) - \sum_{v \in V_1} \delta_{\mathcal{C}(p^{-1}(v))}(p^{-1}(v))$$
$$= \sum_{v \in V_1} \delta_{\mathcal{C}^2(v)}(v) - \sum_{v \in V_1} \delta_{\mathcal{C}(v)}(v) ,$$

as $v \in V_1$ implies $p^{-1}(v) \in V_1$.

Proof. [Proof of Lemma 6.] From Remarks 6 and 7,

$$\sum_i \delta(C_i^2) \leq \sum_i \delta(C_i) + O(\epsilon)c^*$$
$$+ \sum_{v \in V_1} \left[\delta_{\mathcal{C}^2(v)}(v) - \delta_{\mathcal{C}(v)}(v)\right]$$

Now,

$$\sum_{\substack{v \in V_1 \\ g(\mathcal{C}^2(v)) \neq 0}} \delta_{\mathcal{C}^2(v)}(v) - \sum_{\substack{v \in V_1 \\ g(\mathcal{C}(v)) \neq 0}} \delta_{\mathcal{C}(v)}(v)$$
$$\leq \sum_{\substack{v \in V_1 \\ g(\mathcal{C}^2(v)) \neq 0}} \tilde{\delta}_{\mathcal{C}^2(v)}(v) - \sum_{\substack{v \in V_1 \\ g(\mathcal{C}(v)) \neq 0}} \tilde{\delta}_{\mathcal{C}(v)}(v)$$
$$+ |V_1|\frac{4c^*}{m} \in O(\epsilon c^*) ,$$

from Lemma 2, Lemma 4, and Remark 2. In contrast,

$$\sum_{\substack{v \in V_1 \\ g(\mathcal{C}^2(v)) = 0}} \delta_{\mathcal{C}^2(v)}(v) - \sum_{\substack{v \in V_1 \\ g(\mathcal{C}(v)) = 0}} \delta_{\mathcal{C}(v)}(v)$$
$$= \sum_{\substack{v \in V_1 \\ g(\mathcal{C}(v)) = 0}} \delta_{\mathcal{C}(v)}(p(v)) - \sum_{\substack{v \in V_1 \\ g(\mathcal{C}(v)) = 0}} \delta_{\mathcal{C}(v)}(v)$$

Now, an application of the triangle inequality shows that $\delta_{\mathcal{C}(v)}(p(v)) - \delta_{\mathcal{C}(v)}(v) \leq |C_{\mathcal{C}(v)}|\delta(v, p(v))$, so

$$\sum_{v \in V_1, g(\mathcal{C}(v)) = 0} \left[\delta_{\mathcal{C}(v)}(p(v)) - \delta_{\mathcal{C}(v)}(v)\right]$$
$$\leq \sum_{v \in V_1, g(\mathcal{C}(v)) = 0} |C_{\mathcal{C}(v)}|\delta(v, p(v))$$
$$\leq O(1)\frac{s}{m}c^* + \leq O(\epsilon)c^* ,$$

where the final inequality follows from Lemma 5 and the facts that $|C_{\mathcal{C}(v)}| \leq s$ and $s \leq m\epsilon$.

Proof of Theorem 2 The theorem is an application of the following Lemma:

LEMMA 10. *If $f(\epsilon)$ satisfies* MIN-k-UNCUT$(\mathcal{C}^2) \leq (1 + f(\epsilon))$ MIN-k-UNCUT$(\mathcal{C})$, *then*

$$\text{MIN-}k\text{-CC}(\mathcal{C}^2) \leq (1 + 2f(\epsilon)) \text{ MIN-}k\text{-CC}(\mathcal{C}).$$

Proof. Note that, for any clustering $\mathcal{X}$:

$$\begin{aligned}
&\text{MIN-}k\text{-CC}(\mathcal{X}) \\
&\quad = \text{MIN-}k\text{-UNCUT}(\mathcal{X}) + e_c(\mathcal{X}) - \text{MAX-}k\text{-CUT}(\mathcal{X}) \\
&\quad = 2\,\text{MIN-}k\text{-UNCUT}(\mathcal{X}) + e_c(\mathcal{X}) - \Delta.
\end{aligned}$$

Consider

$$\begin{aligned}
&\text{MIN-}k\text{-CC}(\mathcal{C}^2) - \text{MIN-}k\text{-CC}(\mathcal{C}) \\
&\quad = 2\left(\text{MIN-}k\text{-UNCUT}(\mathcal{C}^2) - \text{MIN-}k\text{-UNCUT}(\mathcal{C})\right) \\
&\qquad + e_c(\mathcal{C}^2) - e_c(\mathcal{C}) \\
&\quad \leq 2f(\epsilon)\,\text{MIN-}k\text{-UNCUT}(\mathcal{C})) \\
&\quad = f(\epsilon)\left[\text{MIN-}k\text{-CC}(\mathcal{C}) - (e_c(\mathcal{C}) - \Delta)\right] \\
&\quad \leq 2f(\epsilon)\,\text{MIN-}k\text{-CC}(\mathcal{C}).
\end{aligned}$$

The last inequality above follows from this reasoning

$$\begin{aligned}
&\Delta - e_c(\mathcal{X}) \\
&\quad = \text{MIN-}k\text{-UNCUT}(\mathcal{X}) + \text{MAX-}k\text{-CUT}(\mathcal{X}) - e_c(\mathcal{X}) \\
&\quad = \text{MIN-}k\text{-UNCUT}(\mathcal{X}) + e_c(\mathcal{X}) - \text{MAX-}k\text{-CUT}(\mathcal{X}) \\
&\qquad - 2(e_c(\mathcal{X}) - \text{MAX-}k\text{-CUT}(\mathcal{X})) \\
&\quad \leq \text{MIN-}k\text{-CC}(\mathcal{X}),
\end{aligned}$$

which is a consequence of

$$\begin{aligned}
&\text{MAX-}k\text{-CUT}(\mathcal{X}) \\
&\quad = \sum_{e \in E_c(\mathcal{X})} \delta(e) \leq \sum_{e \in E_c(\mathcal{X})} 1 = e_c(\mathcal{X}).
\end{aligned}$$

A.3 Proofs from Phase Three

Proof of Lemma 8

Proof. For convenience, let

$$\begin{aligned}
X &= \text{MAX-}k\text{-CUT}(\mathcal{C}) & W &= e_c(\mathcal{C}) - X \\
Y &= \text{MIN-}k\text{-UNCUT}(\mathcal{C}) & Z &= e_u(\mathcal{C}) - Y,
\end{aligned}$$

all positive quantities. Also, let f stand for $f(\epsilon)$ and g for $g(\epsilon)$. Then

$$\begin{aligned}
X + W = e_c(\mathcal{C}) &= \sum_{i<j} |C_i||C_j| \geq \sum_{i<j} f^2 M^2 \\
&\geq f^2 \frac{M^2 k}{2},
\end{aligned}$$

And

$$Y + Z = e_u(\mathcal{C}) = \sum_i \binom{|C_i|}{2} \leq \sum_i \binom{M}{2} = k\binom{M}{2}$$

$$\leq \frac{M^2 k}{2}.$$

Combining these, we have,

$$(\text{A.1}) \qquad X + W \geq f^2(Y + Z) \geq f^2 Z$$

Consider the following:

$$\begin{aligned}
&f^2 g(X + Z) - 2(Y + W) \\
&\quad = g(f^2 Z) + f^2 g X - 2(Y + W) \\
&\quad \leq g(X + W) + f^2 g X - (f^2 + 1)Y - 2W \\
&\qquad \text{(by (A.1), and } f \leq 1) \\
&\quad = (f^2 + 1)(gX - Y) + (g - 2)W < 0 \\
&\qquad \text{(as } gX < Y, \text{ and } g \leq 1),
\end{aligned}$$

which completes the proof.

Proof of Lemma 9

Proof. Note that

$$\begin{aligned}
\text{MIN-}k\text{-CC}(\mathcal{X}) &= \sum_{j \in [\gamma]} \text{MIN-}m_j\text{-CC}(\mathcal{X}|_j) \\
&\quad + \sum_{g(\mathcal{X}(u)) \neq g(\mathcal{X}(v))} (1 - \delta(u,v)),
\end{aligned}$$

and similarly for $\mathcal{Y}$. The g-equivalence of $\mathcal{X}$ and $\mathcal{Y}$ tells us that the rightmost summation is common to both clusterings, leading to the statement of the lemma.