Interactive Correlation Clustering

Floris Geerts, Reuben Ndindi
Department of Mathematics and Computer Science
University of Antwerp
Email: {floris.geerts, reuben.ndindi}@uantwerpen.be

***Abstract*—Correlation clustering is to partition a set of objects into clusters such that the number of false positives and negatives is minimised. In this paper, we combine correlation clustering and user interaction. More specifically, we allow the user to control the quality of the clustering by providing error bounds on the number of false positives and negatives. If no clusterings exist that satisfy these bounds, a set of edges is returned for user inspection such that the deletion or relabelling of these edges guarantees the existence of a clustering consistent with the error bounds. However, a user may reject the deletion or relabelling of certain edges and ask for an alternative set of edges to be provided. If no such set of edges exists, a minimal change to the error bounds should be provided, after which the interactive process continues. The focus of this paper is on the algorithmic challenges involved in returning a minimal set of edges to the user. More specifically, we formalise the** INTERACTIVE CORRELATION CLUSTERING **problem and show that it is intractable. Therefore, we propose an approximation algorithm based on the well-known region growing technique. We experimentally validate the efficiency and accuracy of the approximation algorithm.**

I. INTRODUCTION

Clustering is to partition a set of given objects into clusters of similar objects. Typically, the goal is to find a clustering that minimises an objective function that measures the quality of the clustering. A wide variety of formalisations and objective functions have been considered in this context. In this paper, we focus on the formalisation of the clustering problem, known as CORRELATION CLUSTERING [1]. Intuitively, in correlation clustering the set of objects are vertices of a graph whose edges are labeled with either "+" or "−". Here, a +-edge indicates that its vertices (objects) are similar whereas a −-edge indicates the opposite. The corresponding objective function counts the number of *false positives*, i.e., −-edges whose vertices belong to the same cluster, and the number of *false negatives*, i.e., +-edges whose vertices belong two distinct clusters.

In this paper we revisit correlation clustering from an *interactive* point of view, as illustrated by the following example.

Example I.1. Consider the graph G shown in Fig. 1(a). An optimal correlation clustering of G will always have a total of three false positive and negatives. For example, the clustering shown in Fig. 1(b) has three false positives; the clustering shown in Fig. 1(c) has one false positive and two false negatives. A user may want to express that a clustering should not have any errors, or more generally, want to bound the number of false positives and negatives in a clustering. S/he can do this by specifying two *error bounds*, μ_{fp} and μ_{fn}, for the false positives and negatives, respectively. A *valid* clustering will be one in which the number of false positives and negatives does not exceed the given bounds. However, a valid clustering may not exist. Indeed, if $\mu_{fp} = \mu_{fn} = 0$ then G does not have a valid clustering (any optimal clustering has cost 3).

In *interactive correlation clustering*, we want to guide the user towards a valid clustering by allowing him/her to either minimally update the graph, to minimally change the error bounds, or combinations thereof. For example, one way to guarantee the existence of a valid clustering is to *delete* or *relabel* a set ΔE of edges. Indeed, deleting all edges or assigning all edges the same label trivially guarantees such valid clusterings. Of course, we want to minimally modify the input graph. Therefore, we envisage an interactive correlation clustering system that provides the user with a *minimal* set ΔE of edges to delete/relabel. For example, deleting or relabelling the three edges $\Delta E = \{(1, 4), (4, 6), (5, 7)\}$ corresponding to the errors in the clustering shown in Fig. 1(c) ensures that a valid clustering with no errors exists, as shown in Fig. 1(d).

In addition, when presented with the set ΔE of edges, the user may decide not to delete/relabel an edge in ΔE since s/he regards the similarity information represented by this edge as too important or trustworthy. In this case, we say that the user marks an edge as *immutable*. The immutable edges are then passed on to the interactive correlation clustering system and another set of edges $\Delta E'$ is returned, which excludes the immutable edges. For example, the user may mark the edge $(1, 4)$ as immutable. By fixing the edge $(1, 4)$, one now has to delete $\Delta E' = \{(1, 2), (3, 4), (4, 5), (5, 7)\}$ in order to obtain a valid clustering for the bounds $\mu_{fp} = \mu_{fn} = 0$. Fig. 1(e) shows a valid clustering on the updated graph. The user again inspects this set of edges and the interactive process continues until either the user is satisfied and a valid clustering does exist, or no valid clustering exists. The latter case happens when the user marked too many edges as immutable and no ΔE exists whose deletion/relabelling ensures a valid clustering.

For example, suppose now that the user marks the edge $(1, 2)$ in G as immutable. Then to satisfy the bounds, a set of edges $\Delta E'' = \{(1, 5), (2, 4), (3, 4), (5, 6)\}$ needs to be deleted. The corresponding valid clustering for $\mu_{fp} = \mu_{fn} = 0$ is shown in Fig. 1(f). Imagine that at this point the user is still not happy with $\Delta E''$ and marks the edge $(2, 4)$ as immutable. In this case, no ΔE exists that guarantees a valid clustering. Instead, the interactive correlation clustering system should inform the user as to how to minimally change the error bounds. For example, by letting $\mu_{fp} = 1$ and $\mu_{fn} = 0$ (see Fig. 1(g). The interactive process then continues. Observe that this process always terminates. In the worst case, all edges in G are marked as immutable and the bounds μ_{fp} and μ_{fn} are set such that a valid optimal correlation clustering in G exists. For example, for $\mu_{fp} = 1$ and $\mu_{fn} = 2$ the clustering shown in Fig. 1(c) is valid. ♢

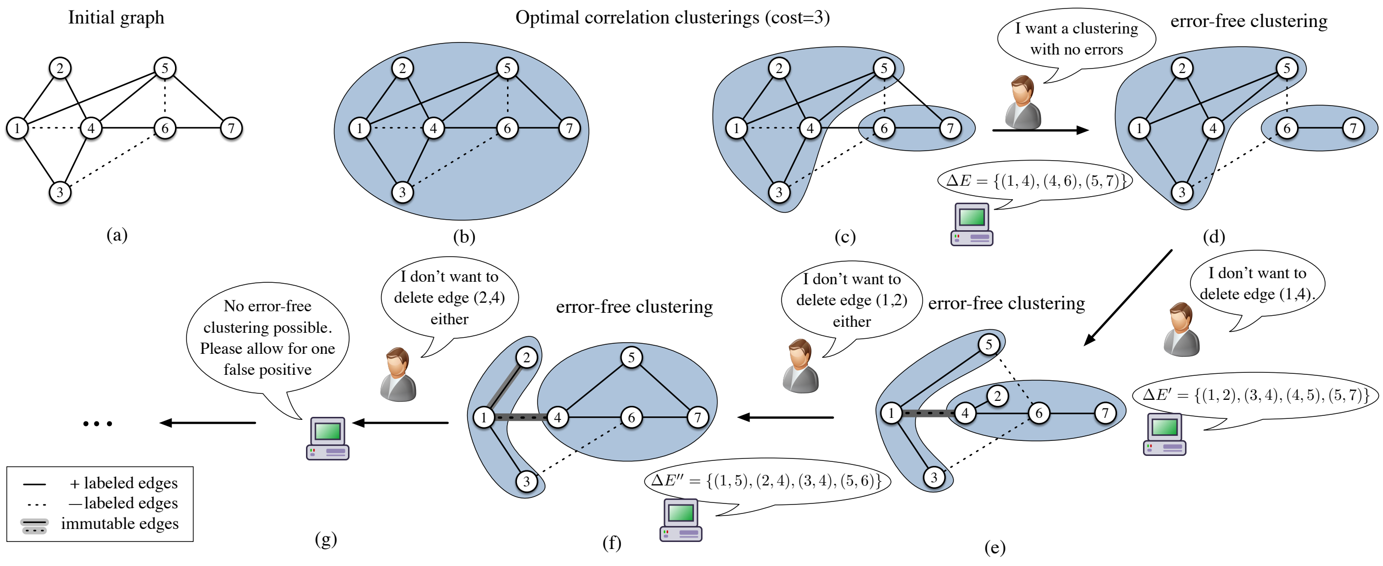

Initial graph
Optimal correlation clusterings (cost=3)
I want a clustering with no errors
error-free clustering
$\Delta E = \{(1,4),(4,6),(5,7)\}$
(a)
(b)
(c)
(d)
I don't want to delete edge (1,4)
$\Delta E' = \{(1,2),(3,4),(4,5),(5,7)\}$
I don't want to delete edge (1,2) either
error-free clustering
I don't want to delete edge (2,4) either
error-free clustering
$\Delta E'' = \{(1,5),(2,4),(3,4),(5,6)\}$
No error-free clustering possible. Please allow for one false positive
...
+ labeled edges
—labeled edges
immutable edges
(g)
(f)
(e)

Fig. 1. Illustration of interactive correlation clustering process as explained in Example I.1.

In this paper, we focus on one key algorithmic component of the interactive framework: when given an input graph G, set of immutable edges IE, and error bounds μ_{fp} and μ_{fn}, return a set ΔE of edges to the user such that the deletion/relabelling of edges in ΔE guarantees the existence of a valid clustering. More specifically, we make the following contributions:

- We formally define the INTERACTIVE CORRELATION CLUSTERING problem and we show that it is intractable.
- We present a region growing-based approximation algorithm for solving the INTERACTIVE CORRELATION CLUSTERING problem and provide a performance guarantee. More specifically, the size of the set of edges returned by the algorithm is at most a factor $O(\log(|E|^-))$ away from the optimal size. The approximation algorithm leverages a close relationship with a variant of the MULTICUT problem, called BOUNDED MULTICUT problem, which may be of interest in its own right.
- We empirically evaluate our algorithm on both synthetic and a real-life dataset. Although the approximation factor may be large in theory, we verified that in practice, the size of the returned set of edges is close to optimal.

The rest of the paper is organised as follows. In the next section, we formally define the INTERACTIVE CORRELATION CLUSTERING problem and establish its intractability. In Section III, we present our region growing-based approximation algorithm. An experimental evaluation on both synthetic and real-life data is presented in section IV. We conclude the paper with related work in Section V and outline our future research direction in section VI.

II. INTERACTIVE CORRELATION CLUSTERING

Let $G = (V, E)$ be a graph with a weight function $w : E \to \mathbb{N}$ on its edges. Assume that the set E of edges can be partitioned into two sets E^+ and E^-. An edge $e \in E^+$ carries label "+", whereas an edge $e \in E^-$ carries label "−". Intuitively, edges in E^+ represent similar objects that should be clustered together; edges in E^- represent the opposite. A clustering $\mathcal{C}$ of G is a partition of V. For a vertex $v \in V$, we denote by $\mathcal{C}(v)$ the set of vertices in the same cluster as v. In a clustering $\mathcal{C}$, we call an edge $e = (u, v)$ a false negative if $e \in E^+$ but $u \notin \mathcal{C}(v)$. Furthermore, if $e \in E^-$ and $u \in \mathcal{C}(v)$, we call $e = (u, v)$ a false positive. We denote by $w_{fn}(\mathcal{C})$ and $w_{fp}(\mathcal{C})$ the sum of the weights of false negatives and positives in $\mathcal{C}$, respectively. Similarly, for an arbitrary set E of edges we define $w(E)$ as the sum of the weights of edges in E. Finally, we define $cost(\mathcal{C}) = w_{fp}(\mathcal{C}) + w_{fn}(\mathcal{C})$. Standard CORRELATION CLUSTERING is to find a clustering $\mathcal{C}$ of minimal cost, $cost(\mathcal{C})$.

Let μ_{fp} and μ_{fn} be two natural numbers. We regard a clustering $\mathcal{C}$ as being *valid* provided that $w_{fp}(\mathcal{C})$ and $w_{fn}(\mathcal{C})$ are below these thresholds. We have seen that such valid clusterings do not always exist, however, but the existence can be guaranteed when sufficiently many edges are deleted from the input graph. Clearly, we want to delete as less edges as possible. The determination of the set of edges to delete constitutes the following problem:

Problem 1 (INTERACTIVE CORRELATION CLUSTERING)**.** *Given a graph $G = (V, E)$ and natural numbers μ_{fp} and μ_{fn}, find a set ΔE of edges, such that $w(\Delta E)$ is minimal and such that there exists a clustering $\mathcal{C}$ of $G' = (V, E \setminus \Delta E)$ for which $w_{fn}(\mathcal{C}) \leqslant \mu_{fn}$ and $w_{fp}(\mathcal{C}) \leqslant \mu_{fp}$ holds.* ■

An equivalent formulation would consist of finding the set ΔE of edges to be relabelled instead of deleted. Here, by relabelling we mean that a +-labeled edge becomes a −-labeled edge, and vice versa. In this paper, we consider the deletion variant of the problem as stated earlier.

Not surprisingly, the INTERACTIVE CORRELATION CLUSTERING problem is computationally infeasible. Indeed, its decision version that is to determine given $G = (V, E)$, μ_{fp}, μ_{fn}, and integer $L \geq 0$ whether or not there exists a set ΔE of edges such that $w(\Delta E) \leqslant L$ and such that after deleting ΔE from G, the updated graph $G' = (V, E \setminus \Delta E)$ has a clustering

$\mathcal{C}$ such that $w_{fp}(\mathcal{C}) \leqslant \mu_{fp}$ and $w_{fn}(\mathcal{C}) \leqslant \mu_{fn}$, is NP-complete.

Proposition 1. *The decision version of* INTERACTIVE CORRELATION CLUSTERING *is NP-complete for both weighted and unweighted graphs.*

Proof: For the lower bound, we prove that the decision version of INTERACTIVE CORRELATION CLUSTERING is NP-hard by reducing it from the decision version of CORRELATION CLUSTERING. The latter decision version is to determine given an input graph $H = (W, F)$ and integer $K \geq 0$, whether or not there exists a clustering $\mathcal{C}$ of H such that $cost(\mathcal{C}) \leqslant K$. This problem was proven to be NP-hard in [1] for both weighted and unweighted graphs.

The reduction is as follows. Let $H = (W, F)$ and $K \geq 0$ be an instance of CORRELATION CLUSTERING. We define the corresponding instance of INTERACTIVE CORRELATION CLUSTERING by letting $G = H$, $L = K$, $\mu_{fn} = 0$ and $\mu_{fp} = 0$. For the correctness of the reduction, consider a clustering $\mathcal{C}$ of H such that $cost(\mathcal{C}) \leqslant K$. If we delete all edges corresponding to the false positives and negatives in $\mathcal{C}$ from H, then the clustering induced by $\mathcal{C}$ on the updated graph has no false positives and negatives. Hence, by letting ΔE be the set of edges corresponding to the false positives and negatives in $\mathcal{C}$ we obtain a solution for INTERACTIVE CORRELATION CLUSTERING with $|\Delta E| \leqslant L = K$, $\mu_{fn}(\mathcal{C}) = 0$ and $\mu_{fp}(\mathcal{C}) = 0$. Conversely, suppose that by deleting edges in ΔE from G with $|\Delta E| \leqslant K$, we have that there is a clustering $\mathcal{C}$ of $G' = (V, E \setminus \Delta E)$ with no false positives and negatives. Then, $\mathcal{C}$ is a clustering of H such that $cost(\mathcal{C}) = w(\Delta E) \leqslant K = L$. Hence, solutions of CORRELATION CLUSTERING correspond to solutions of INTERACTIVE CORRELATION CLUSTERING with $\mu_{fn} = 0$ and $\mu_{fp} = 0$, and vice versa.

For the upper bound, consider the following NP-algorithm: (1) Guess (a) a set ΔE of at most L edges; and (b) a clustering $\mathcal{C}$ of the updated graph $G' = (V, E \setminus \Delta E)$. (2) Verify (in PTIME) whether $w_{fn}(\mathcal{C}) \leqslant \mu_{fn}$ and $w_{fp}(\mathcal{C}) \leqslant \mu_{fp}$ hold. If so, accept the guess and return "yes"; otherwise reject the guess. Clearly, this algorithm correctly decides the (decision variant of) INTERACTIVE CORRELATION CLUSTERING. ■

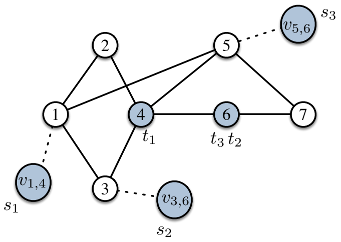
In view of this intractability result, we next develop an approximation algorithm for the INTERACTIVE CORRELATION CLUSTERING problem.

III. APPROXIMATION ALGORITHM

We next present a $\mathrm{O}(\log |E^-|)$-approximation algorithm for the INTERACTIVE CORRELATION CLUSTERING problem. The approximation algorithm is obtained by following a similar strategy as is used for the approximation algorithm for the CORRELATION CLUSTERING problem given in [2]. More specifically, we first establish a relationship between the INTERACTIVE CORRELATION CLUSTERING problem and variant of the MULTICUT problem, called the BOUNDED MULTICUT problem. Leveraging this relationship, we modify the region growing approximation algorithm for the MULTICUT problem [3] to an approximation algorithm for the BOUNDED MULTICUT problem. All combined, this results in an approximation algorithm for INTERACTIVE CORRELATION CLUSTERING.

A. Relationship between Correlation Clustering and the Multicut problem

We recall from [2] how the CORRELATION CLUSTERING and MULTICUT problems are related. An instance of the MULTICUT problem consists of an edge-weighted graph $G = (V, E)$ together with a set $S = \{(s_i, t_i) \mid i \in [1, k]\}$ of source-sink pairs, and is to find a set T of minimal weight (i.e., a *multicut*) such that the removal of the edges in T from G disconnects all pairs in S. For our purpose, its suffices to consider how a MULTICUT instance can be constructed from an instance of CORRELATION CLUSTERING. Let $G = (V, E)$ be a graph with a weight function $w : E \rightarrow \mathbb{N}$ on its edges, and let μ_{fn} and μ_{fp} two integers. Recall that $E = E^+ \cup E^-$. We transform G as follows: (1) For every edge $(u, v) \in E^-$, we introduce a new vertex v_{uv} and a new $-$-labeled edge (v_{uv}, u) with same weight as (u, v) and a source-sink pair (v_{uv}, v); (2) the edges in E^+ remain the same. Note that this transformation keeps the sizes of $|E^+|$ and $|E^-|$ intact, and adds at most new $|E^-|$ vertices. Let S be the set of all the source-sink pairs (v_{uv}, v) and let G_{mc} be the new graph obtained from G by the above transformation. The resulting MULTICUT instance is then given by (G_{mc}, S). As an example, the MULTICUT instance obtained from the graph G shown in Fig 1(a) is given by



Here, the shaded vertices indicate the three source/sink pairs corresponding to the negative edges. Given a clustering $\mathcal{C}$ of G with $cost(\mathcal{C}) = W$ we obtain a multicut T of G_{mc} as follows: Let T' be the set of edges in E corresponding to the false positive and negatives in $\mathcal{C}$. Then $T = (T' \cap E^+) \cup \{(v_{u,v}, u) \mid (u, v) \in T' \cap E^-\}$ is a multicut for (G_{mc}, S) such that $w(T) = W$. Conversely, given a multicut T or (G_{mc}, S) such that $w(T) = W$, we define T' as the set T in which all newly added $-$-labeled edges are replaced by their corresponding $-$-labeled edges in G. The corresponding clustering $\mathcal{C}$ is obtained by taking every connected component in $E^+ \setminus T'$ as a cluster. It can be shown that $cost(\mathcal{C}) = W$ [2].

B. Relationship between Interactive Correlation Clustering and the Bounded Multicut problem

We next establish a similar relationship for the INTERACTIVE CORRELATION CLUSTERING problem. For this, we need the following variant of the MULTICUT problem.

Problem 2 (BOUNDED MULTICUT)**.** *Given a graph $G = (V, E)$ with $E = E^+ \cup E^-$, a weight function $w : E \rightarrow \mathbb{N}$ on its edges, and a collection S of pairs of distinct vertices (s_i, t_i) of G, find a set of edges ΔE of minimal weight such that there is a multicut T in $G = (V, E \setminus \Delta E)$ such that $w_{fp}(T) \leqslant \mu_{fp}$ and $w_{fn}(T) \leqslant \mu_{fn}$.* ■

Using the same construction as above, we can see the INTERACTIVE CORRELATION CLUSTERING problem as an instance of the Bounded Multicut problem. Indeed, let ΔE be a solution of Interactive Correlation Clustering for the graph $G = (V, E)$. That is, there exists a clustering $\mathcal{C}$ in $G' = (V, E \setminus \Delta E)$ such that $w_{fp}(\mathcal{C}) \leq \mu_{fp}$ and $w_{fn}(\mathcal{C}) \leq \mu_{fn}$. As shown above, this implies that there is a multicut T of (G'_{mc}, S_p) such that $w_{fp}(T) \leqslant \mu_{fp}$ and $w_{fn}(T) \leqslant \mu_{fn}$. Conversely, given a solution ΔE for the Bounded Multicut problem for $(G_{mc} = (V', E'), S)$, we know that that there exists a multicut T in $G = (V', E' \setminus \Delta E')$ such that $w_{fp}(T) \leqslant \mu_{fp}$ and $w_{fn}(T) \leqslant \mu_{fn}$. Then, by letting ΔE to consist of the positive edges in $\Delta E'$ and the negative edges in G correspond to the new $-$-labeled edges in $\Delta E'$, it is easy to see that ΔE is a solution of Interactive Correlation Clustering, where the clustering $\mathcal{C}$ in $G = (V, E \setminus \Delta E)$ is obtained from the multicut T, as described previously. This implies the following.

Observation. *Any (approximation) algorithm for the* Bounded Multicut *problem results in an (approximation) for the* Interactive Correlation Clustering *problem.*

In the remainder of this section we develop an approximation algorithm for Bounded Multicut.

C. Solving the Bounded Multicut problem exactly

We show that the Bounded Multicut problem can be solved by means of the following integer program, IP_{BMC}, which is a modification of the standard program for solving Multicut [4]. Let $G = (V, E)$ be a graph with $E = E^{+} \cup E^{-}$ and a weight function $w : E \rightarrow \mathbb{N}$ on its edges. Furthermore, let S be a set of k pairs of distinct vertices (s_i, t_i). Let μ_{fp} and μ_{fn} be two non-negative integers. We denote by w_{uv} the weight of edge (u, v).

$$\boxed{\begin{array}{lll} \text{IP}_{\text{BMC}}\text{: minimize } \sum_{(u,v)\in E} w_{uv}(x_{uv} - y_{uv}) & & \\ \qquad \text{subject to} & & \\ \qquad\qquad \sum_{(u,v)\in p_i} x_{uv} \geq 1, p_i \in \mathcal{P}_i, 1 \leq i \leq k & & \text{(i)} \\ \qquad\qquad \sum_{(u,v)\in E^{+}} w_{uv} y_{uv} \leq \mu_{fn} & & \text{(ii)} \\ \qquad\qquad \sum_{(u,v)\in E^{-}} w_{uv} y_{uv} \leq \mu_{fp} & & \text{(iii)} \\ \qquad\qquad x_{uv} \geq y_{uv} & & \text{(iv)} \\ \qquad\qquad x_{uv}, y_{uv} \in \{0, 1\} & & \text{(v)} \end{array}}$$

Here, $\mathcal{P}_i$ denotes the set of all paths from s_i to t_i. Observe that this integer program has exponentially many constraints but, similarly as in the standard Multicut case, it can be converted into one of polynomial size. For completeness, we provide this conversion below. Observe that the integer program for Multicut, IP_{MC}, can be obtained by setting μ_{fp} and μ_{fn} to zero, i.e., by ignoring the y_{uv} variables.

We first verify the correctness of the integer program IP_{BMC}.

Proposition 2. *A solution of IP_{BMC} corresponds to a solution of the* Bounded Multicut *problem, and vice versa.*

Proof: Let ΔE be a solution of Bounded Multicut and let T be a multicut in $G = (V, E \setminus \Delta E)$ such that $w_{fp}(T) \leqslant \mu_{fp}$ and $w_{fn}(T) \leqslant \mu_{fn}$. Based on this, we define the following valuation ν:

$$\nu(x_{uv}) = \begin{cases} 1 & \text{if } (u, v) \in T \cup \Delta E \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\nu(y_{uv}) = \begin{cases} 1 & \text{if } (u, v) \in T \\ 0 & \text{otherwise.} \end{cases}$$

We claim that this valuation satisfies the conditions (i) - (v) of the integer program IP_{BMC}. Clearly, (iv) and (v) are satisfied by the definition. Note that $T \cup \Delta E$ is a multicut for the original graph $G = (V, E)$. It is known that condition (i) expresses that every source-sink pair is disconnected. Hence, condition (i) is satisfied since $\nu(x_{uv}) = 1$ for all $(u, v) \in T \cup \Delta E$. Clearly, (ii) and (iii) are satisfied since $\nu(y_{uv}) = 1$ for all $(u, v) \in T$. Finally we also remark that the objective function corresponds to $w(\Delta E) = \sum_{e \in \Delta E} w_e$. Indeed, $\nu(x_{uv}) - \nu(y_{uv}) = 1$ for all $(u, v) \in \Delta E$, and $\nu(x_{uv}) - \nu(y_{uv}) = 0$ for all other edges.

For the converse, let ν be a valuation that satisfies conditions (i) - (v). Consider the set of edges $\Delta E = \{(u, v) \mid \nu(x_{uv}) - \nu(y_{uv}) = 1\}$ and let $T = \{(u, v) \mid \nu(y_{uv}) = 1\}$. It can be readily verified that this results in a solution of Bounded Multicut. As before, $w(\Delta E) = \sum_{e \in \Delta E} w_e$. ■

We next describe a standard procedure to turn IP_{BMC} into an equivalent integer program of polynomial size. Let S be the set of k source-sink pairs. We introduce binary variables z^i_u, one for each vertex u in the graph and each $(s_i, t_i) \in S$. We then replace the constraint (i) in IP_{BMC} with the following two constraints:

$$\begin{array}{ll} z^i_u - z^i_v \leqslant x_{uv}, \text{ for all } (u, v) \in E,\ 1 \leq i \leq k & \text{(i')} \\ z^i_{s_i} - z^i_{t_i} \geq 1, \text{ for all } (s_i, t_i) \in S. & \text{(i'')} \end{array}$$

We show that constraint (i) is equivalent to the constraints (i') & (i"). Consider a source-sink pair (s_i, t_i) in S and assume that we have a path p between this pair consist of the following edges $(s_i, v_1), (v_1, v_2), \ldots, (v_n, t_i)$. Using constraint (i") we have that

$$1 \leqslant z^i_{s_i} - z^i_{t_i} = z^i_{s_i} - z^i_{v_1} + z^i_{v_1} - z^i_{v_2} + \cdots - z^i_{v_n} + z^i_{v_n} - z^i_{t_i}$$

and using constraint (i'), we have

$$z^i_{s_i} - z^i_{v_1} \leqslant x_{s_i v_1}, z^i_{v_1} - z^i_{v_2} \leqslant x_{v_1 v_2}, \cdots, z^i_{v_n} - z^i_{t_i} \leqslant x_{v_n t_i}.$$

Hence,

$$1 \leqslant z^i_{s_i} - z^i_{t_i} \leqslant \sum_{(u,v)\in p} x_{uv}.$$

Note that this holds for any path p between any source-sink pairs. Hence, condition (i) is satisfied.

For the converse, assume that constraint (i) is satisfied. For each $(s_i, t_i) \in S$ we set the variables z^i_u as follows: $z^i_{s_i} = 1$ and $z^i_{t_i} = 0$ hence satisfying condition (i"). Furthermore, for each path p from s_i to t_i we identify the first edge (u, v) such that $x_{uv} = 1$. Note that such an edge must exist since (i) is satisfied. Denote by T_i the initial vertices u of these edges for all paths from s_i to t_i. Let $\text{Pre}(T_i)$ be the set of vertices on paths from s_i to any vertex in T_i, including T_i; and $\text{Post}(T_i)$ the vertices on paths from any vertex in T_i to t_i, excluding T_i. We then define $z^i_u = 1$ if $u \in \text{Pre}(T_i)$; $z^i_u = 0$ if $u \in \text{Post}(T_i)$. It is readily verified that condition (i') is satisfied. Hence replacing constraint (i) with the constraints (i') & (i") results in an equivalent integer program formulation of Bounded Multicut, of polynomial size.

D. Approximation algorithm

We are finally ready to present our region growing-based approximation algorithm for the BOUNDED MULTICUT problem. It is a modification of the standard region-growing algorithm for the MULTICUT problem as given in [3]. In particular, (a) we use the linear relaxation of IP_{BMC} for BOUNDED MULTICUT rather than the relaxation of the integer program IP_{MC} for MULTICUT; and (b) we postprocess the output of the algorithm to obtain a set ΔE and multicut in the updated graph $G = (V, E \setminus \Delta E)$.

Pseudo-code of the algorithm is shown in the Fig. 2. We start from the integer program IP_{BMC} for BOUNDED MULTICUT and relax it to its corresponding linear program by replacing the constraint (v) in the IP_{BMC} with $x_{uv}, y_{uv} \in [0, 1]$ (line 1). We can obtain a solution for the relaxation of IP_{BMC} in PTIME by using its equivalent polynomially sized linear program, as discussed earlier. Denote by d_e the returned valuation for variable x_e. We ignore the valuations returned for the y_e variables. Let $F = \sum_{(u,v)\in E} w_e d_e$. We grow regions (lines 4–15), starting from one source vertex at a time. Let s be a source vertex. We add vertices to a region around s in the order determined by their distance to s as given by the values d_e (line 6). Whenever we grow a region (line 8), we update the *volume* $\mathcal{V}$(region) of the region such that $\mathcal{V} = F/k + \sum_{e\cap\text{region}\neq\emptyset} w_e d_e$ (line 9). That is, we add $w_e d_e$ to the volume for every edge that has at least one vertex in the current region. At the same time, we update the *cost* of a region c(region) such that it equals $\sum_{e\in\delta(\text{region})} w_e$, where δ(region) consists of all edges that have a single vertex in the current region (line 9). We do this until a stopping condition is satisfied (line 10). We then update the graph by removing all vertices (and their incident edges) in region from the current graph (line 12) and add δ(region) to the result set T (line 13). It is well-known that the stopping condition guarantees that T is a multicut and furthermore, that $w(T) \leqslant 4\ln(k+1)F$ [3]. Finally, we split T into a set ΔE and multicut C of $G = (V, E \setminus \Delta E)$ and S, such that $w_{\mathit{fp}}(C) \leqslant \mu_{\mathit{fp}}$ and $w_{\mathit{fn}}(C) \leqslant \mu_{\mathit{fn}}$ (line 16). This is done by removing edges from T and putting those in ΔE until $C = T \setminus \Delta E$ satisfies the bounds. Finally, ΔE is returned (line 17). Clearly, this results in a solution for BOUNDED MULTICUT.

It remains to identify the approximation factor of the algorithm. For this, it suffices to observe that

$$\Big(\sum_{(u,v)\in E} w_{uv} d_{uv}\Big) - (\mu_{\mathit{fp}} + \mu_{\mathit{fn}}) \leqslant \Big(\sum_{(u,v)\in E} w_{uv} d_{uv}\Big) - \Big(\sum_{(u,v)\in E} w_{uv} y_{uv}\Big) \leqslant \sum_{(u,v)\in E} w_{uv} d_{uv}$$

and thus

$$F = \sum_{(u,v)\in E} w_{uv} d_{uv} \leqslant \Big(\sum_{(u,v)\in E} w_{uv}(d_{uv} - y_{uv})\Big) + (\mu_{\mathit{fp}} + \mu_{\mathit{fn}}) = |\Delta E_{\text{lp}}| + (\mu_{\mathit{fp}} + \mu_{\mathit{fn}}),$$

where $|\Delta E_{\text{lp}}|$ denotes the objective value of the relaxation of IP_{BMC}. Hence,

$$w(T) = w(\Delta E) + w(C) \leqslant 4\ln(k+1)F \leqslant 4\ln(k+1)(|\Delta E_{\text{opt}}| + (\mu_{\mathit{fp}} + \mu_{\mathit{fn}})).$$

since $|\Delta E_{\text{lp}}| \leqslant |\Delta E_{\text{opt}}|$ where $|\Delta E_{\text{opt}}|$ is the size of the optimal solution as given by the integer program IP_{BMC}. From this, we may conclude that

$$w(\Delta E) \leqslant 4\ln(k+1)F \leqslant 4\ln(k+1)(|\Delta E_{opt}| + (\mu_{\mathit{fp}} + \mu_{\mathit{fn}})).$$

We thus have indeed obtained a $O(\log k)$-approximation algorithm for BOUNDED MULTICUT. Since the k source-sink pairs correspond to the negative edges (recall that the transformation from INTERACTIVE CORRELATION CLUSTERING to BOUNDED MULTICUT), this algorithm is a $O(\log(|E^-|))$-approximation algorithm for INTERACTIVE CORRELATION CLUSTERING.

```
BMulticut ($G = (V, E), S = \{(s_i, t_i) \mid i \in [1, k]\}$, $\mu_{fp}$, $\mu_{fn}$)
1.  Find an optimal fractional solution of LP obtaining in this way
    distance labels $d_e$ on the edges and the value $F = \sum_{(u,v)\in E} w_e d_e$;
2.  Let $k := |S|$, $\epsilon := 2\ln(k+1)$;
3.  Initialize $H := G$, $T := \emptyset$ and $\Delta E := \emptyset$; Let grow = true;
4.  while($|S| > 0$) /* assume that all pairs in $S$ are connected */
5.      pick a source-sink pair $(s, t)$ from $S$ and let region := $\emptyset$;
6.      Let $L$ be the list of vertices in $H$, sorted by their
        increasing distance to $s$; Assume that $s$ is the first
        element $L[0]$ in this list and let $L = L[0]$;
7.      while(grow)
8.          region = region $\cup$ $L$;
9.          update volume $\mathcal{V}$(region) and cost $c$(region);
10.         if $c$(region) $\leqslant \epsilon\mathcal{V}$(region) then grow = false,
            else let $L = L$.next;
11.     end
12.     $H := H \setminus$ region;
13.     $T = T \cup \delta$(region);
14.     remove all pairs in $S$ that are disconnected in $H$;
15. end
16. Remove edges from $T$ and put these in $\Delta E$ until for $C = T \setminus \Delta E$
    we have that $w_{fp}(C) \leqslant \mu_{fp}$ and $w_{fn}(C) \leqslant \mu_{fn}$;
17. return $\Delta E$.
```

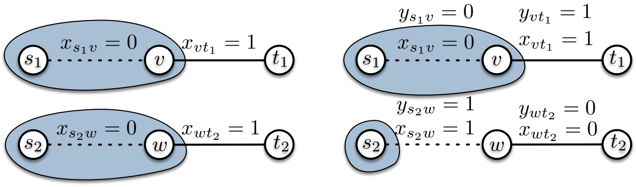
Fig. 2. Approximation algorithm for BOUNDED MULTICUT.

E. Discussion

The region growing algorithm uses the relaxation of IP_{BMC} to determine, among other things, the order in which vertices are added to the region. However, the algorithm only uses the valuations d_{uv} for the variables x_{uv} and does not explicitly leverages the availability of valuations c_{uv} for y_{uv}. As an immediate consequence, the weight of the obtained multicut is related to $F = \sum_{(u,v)\in E} w_e d_e$ rather than the objective function of IP_{BMC} ($\sum_{(u,v)\in E} w_e(d_e - c_e)$). We thus get an overly pessimistic upper bound on the quality of the approximation. Nevertheless, we will see in the experimental section that we get ΔE's that are close to optimal.

Ideally, we would like to grow regions using $d_{uv} - c_{uv}$ rather than d_{uv}. This does not work, however, since constraint (i) in IP_{BMC} only refers to the x_{uv} variables and this constraint, together with the stop condition for growing regions (line 10 in the algorithm), ensures that the result is indeed a multicut. Further investigation is required as how to make better use of the available valuations of the y_{uv} variables.

However, we remark that the additional constraints (ii), (iii), and (iv) in IP_{BMC} may indeed help to guide the region growing process towards a better solution than when ignoring these constraints and using the relaxation of IP$_{\text{MC}}$ for the standard MULTICUT problem. Indeed, consider the following simple instance of BOUNDED MULITCUT for $\mu_{fp} = 1$ and $\mu_{fn} = 1$, i.e., we allow for one positive and one negative edge to belong to the multicut. As before, dashed edges represent edges in E^-, solid edges correspond to edges in E^+. We adorned the edges with valuations for x_{uv} obtained by relaxing IP$_{\text{MC}}$ (left) and with valuations for x_{uv} and y_{uv} obtained by relaxing IP$_{\text{BMC}}$ (right). In this example, we get integer solutions.



As can be seen, by growing regions based on IP$_{\text{MC}}$ we get a multicut consisting of two positive edges (the regions are gray-shaded). After post processing, we put one of these in ΔE and thus $|\Delta E| = 1$. However, by growing regions based on IP$_{\text{BMC}}$ we immediately obtain a multicut that is valid, i.e., it consists of one positive and one negative edge. As a consequence, an empty ΔE will be returned by the algorithm. This shows the advantage of growing regions based on the relaxation of IP$_{\text{BMC}}$.

IV. EXPERIMENTAL EVALUATION

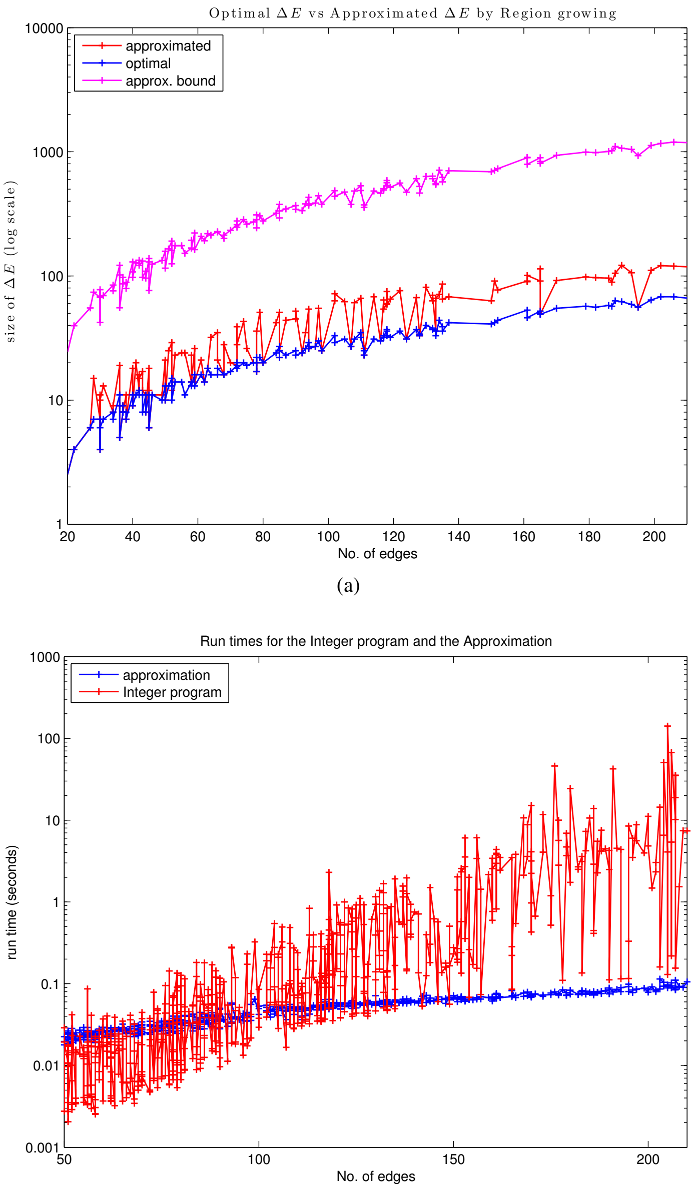
In this section, we describe the empirical evaluation of our approximation algorithm on synthetic and real datasets. For solving the integer program IP$_{\text{BMC}}$ and its relaxation, we use the IBM Cplex Optimizer [5]. The region growing algorithm itself is implemented in Java. The experiments were conducted on a GNU/Linux machine with Intel(R) Xeon(R) CPU 2.90GHz (16 cores) and 32GB memory.

A. Synthetic data

We generate our synthetic graph data along the same lines as in [6]. That is, starting with an initial number of vertices and number of clusters, we randomly add "+" and "-" labelled edges to the graph using parameters to control the number of false positive and negatives. We report averaged results over 20 runs.

Quality of Approximation. We first investigate the quality of our results by comparing it against the optimal solution, obtained by solving the integer program IP$_{\text{BMC}}$. To guarantee the feasibility of solving the integer program, we only use small graph datasets (up to 210 edges). For these datasets we varied the user-defined bounds μ_{fp} and μ_{fn} between 0 and 20 and investigated the relationship with the returned ΔE. Not surprisingly, as we increase the bounds the size of ΔE dropped. We do not show the plot here due to lack of space. Figure 3 (a) shows the size of the optimal ΔE, the size of set of edges returned by the approximation algorithm, and the theoretical upper bound on the approximation guarantee, in terms of the number of edges in the input graph. One can see that the approximation algorithm obtains solutions that are consistently close to the optimal; much better than predicted by the theoretical approximation guarantee.

Secondly, we investigate how large the ΔE's returned by our algorithm are, compared to number of edges in the graph.



(a)

(b)

Fig. 3. Experimental evaluation on synthetic data

Since ΔE is returned to the user for inspection, we want its size to be reasonable. Since there is no need for solving an integer program in this experiment (we only need to solve its linear relaxation), we report our findings in Table I for larger graphs. As can be seen, only a small fraction of edges are returned for user inspection.

No. Vertices $\|V\|$	No. Edges $\|E\|$	$\|\Delta E\|/\|E\|$
250	634	0.23
270	1250	0.13
290	2094	0.07
300	2566	0.06
310	3119	0.06
330	4389	0.04
450	5739	0.05

TABLE I. RATIO OF $|\Delta E|$ VS $|E|$.

Scalability. As a sanity check, in Fig. 3(b) we compare the running times of solving the integer program IP_{BMC} against the running time of our approximation algorithm. Not surprisingly, solving IP_{BMC} becomes quickly infeasible for large input graphs. Instead, the approximation algorithm returns results within reasonable time. We remark that for small graphs, solving IP_{BMC} exactly is sometimes faster than running the approximation algorithm. Note, however, that these differences are very small (log scale) and probably due to background processes.

We also investigated the impact of solving the linear relaxation of IP_{BMC} on the overall running time of our approximation algorithm. Table II shows that solving the linear program constitutes the dominant factor in the whole region growing process.

No. Vertices $\lvert V\rvert$	No. All Edges $\lvert E\rvert$	LP Runtime (s)	Region growing runtime (s)
250	634	15.56	0.57
270	1250	89.93	0.45
290	2094	189.69	0.89
300	2566	172.18	0.99
310	3119	153.32	1.04
330	4389	196.48	1.51
450	5739	1269.41	2.28

TABLE II. LP and Region growing run times

B. Real data

We used the Epinions social network dataset from [7]. It is a directed graph depicting a who-trust-whom network from a general consumer site epinions.com. For each pair of nodes in the graph (users), which had both directions, we randomly picked one direction to obtain an undirected graph. We sampled 10 subgraphs with sizes ranging from 1647 to 7808 in number of vertices and 1000 to 5500 in number of edges. We varied the false negative and positive bounds between 0 and 10. After 10 runs, for all the datasets we obtained an average ΔE of size 6, with most datasets returning 0 even when both false negative and positive bounds were set to 0. The largest value of ΔE was 22 returned from the 5500-edge graph. This reflects a real-life situation in which in a properly clustered social network, one would expect ΔE to be small.

V. Related Work

The Correlation Clustering problem was introduced in [1] and approximation algorithms have been reported in [1], [8], [9], [10] with the goal of minimising disagreements or maximising agreements. Since then, a number of variations of the correlation clustering problem have been considered: by fixing the number of clusters [11]; by allowing overlapping clusters [12]; and for generally labeled edges [6]. None of these works consider correlation clustering in the presence of user-defined error bounds, however.

Similarly, the Multicut problem has received ample attention, see e.g., [13] for a survey. Most relevant to our work is the region-growing approximation algorithm presented in [3], [4]. To our knowledge, the Bounded Multicut problem has not been studied so far.

Most closely related to this paper is the $O(\log n)$-approximation algorithm for Correlation Clustering presented in [2]. In that work, the region growing algorithm for Multicut [3] is used to obtain an approximation algorithm for Correlation Clustering. We follow a similar strategy in this paper, albeit in the presence of error bounds, as explained in Section III.

VI. Conclusion

We have formulated an interactive correlation clustering framework and provided an approximation algorithm for one of its main building blocks, i.e., the identification of a minimal set of edges to delete in order to guarantee the existence of valid clustering relative to the user-defined error bounds. The algorithm is experimentally validated and despite being an approximation, the returned set of edges is in practice close to optimal. As part of future work, we aim to investigate the techniques used in [14] to obtain an alternative approximation algorithm, as well as to build a prototype system that fully supports the interactive features mentioned in the Introduction. Furthermore, a more extensive qualitative analysis of the algorithms will be carried out. In addition, we are currently investigating interactive correlation clustering when clusters can overlap and when edges can carry labels from an arbitrary set, i.e., not only $+$ and $-$ labels.

References

[1] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[2] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica, "Correlation clustering in general weighted graphs," *Theoretical Computer Science*, vol. 361, no. 2, pp. 172–187, 2006.

[3] N. Garg, V. V. Vazirani, and M. Yannakakis, "Approximate max-flow min-(multi) cut theorems and their applications," *SIAM Journal on Computing*, vol. 25, no. 2, pp. 235–251, 1996.

[4] V. V. Vazirani, *Approximation algorithms*. Springer, 2001.

[5] IBM, "Cplex optimzer," http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, accessed: 2013-11-27.

[6] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen, "Chromatic correlation clustering," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1321–1329.

[7] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 1361–1370.

[8] M. Charikar, V. Guruswami, and A. Wirth, "Clustering with qualitative information," in *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*, 2003, pp. 524–533.

[9] E. D. Demaine and N. Immorlica, "Correlation clustering with partial information," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2003, pp. 1–13.

[10] C. Swamy, "Correlation clustering: maximizing agreements via semidefinite programming," in *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, 2004, pp. 526–527.

[11] I. Giotis and V. Guruswami, "Correlation clustering with a fixed number of clusters," in *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithm*, 2006, pp. 1167–1176.

[12] F. Bonchi, A. Gionis, and A. Ukkonen, "Overlapping correlation clustering," in *Proceedings of the 11th IEEE International Conference on Data Mining*, 2011, pp. 51–60.

[13] M.-C. Costa, L. Létocart, and F. Roupin, "Minimal multicut and maximal integer multiflow: A survey," *European Journal of Operational Research*, vol. 162, no. 1, pp. 55 – 69, 2005.

[14] S. Barman and S. Chawla, "Region growing for multi-route cuts," in *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms*, 2010, pp. 404–418.