Partial Optimality and Fast Lower Bounds for Weighted Correlation Clustering

Jan-Hendrik Lange [1 2] **Andreas Karrenbauer** [1] **Bjoern Andres** [1 3 4]

Abstract

Weighted correlation clustering is hard to solve and hard to approximate for general graphs. Its applications in network analysis and computer vision call for efficient algorithms. To this end, we make three contributions: We establish partial optimality conditions that can be checked efficiently, and doing so recursively solves the problem for series-parallel graphs to optimality, in linear time. We exploit the packing dual of the problem to compute a heuristic, but non-trivial lower bound faster than that of a canonical linear program relaxation. We introduce a re-weighting with the dual solution by which efficient local search algorithms converge to better feasible solutions. The effectiveness of our methods is demonstrated empirically on a number of benchmark instances.

1. Introduction

This paper is about weighted correlation clustering (Bansal et al., 2004), a combinatorial optimization problem whose feasible solutions are all clusterings of a graph, and whose objective function is a sum of weights $w^0, w^1 \colon E \to \mathbb{R}_0^+$ defined on the edges E of the graph. The weight w_e^0 is added to the sum if the nodes $\{u, v\} = e \in E$ are in the same cluster, and the weight w_e^1 is added to the sum if these nodes are in distinct clusters. The problem consists in finding a clustering of minimum weight.

Weighted correlation clustering has found applications in the fields of network analysis (Cesa-Bianchi et al., 2012) and, more recently, computer vision (Kappes et al., 2011; Keuper et al., 2015; Insafutdinov et al., 2016; Beier et al., 2017; Tang et al., 2017), partly due to its key property that the number of clusters is not fixed, constrained or penalized in the problem statement but is instead defined by the (any) solution. Weighted correlation clustering in general graphs is hard to solve exactly and hard to approximate (Demaine et al., 2006). Remarkable progress has been made toward algorithms that find feasible solutions by approximations or heuristics (cf. Section 2). Yet, the computation of lower bounds remains challenging for large instances (Swoboda & Andres, 2017).

We make the following contributions: Firstly, in order to reduce instances in size, we establish partial optimality conditions on the graph and weights that can be checked combinatorially in polynomial time and determine the values of some variables in an optimal solution. By applying these conditions recursively, we reduce an instance in size without restricting the quality of solutions. For series-parallel graphs, our algorithm solves weighted correlation clustering exactly and in linear time, as we show. For general graphs, we demonstrate its effectiveness empirically.

Secondly, in order to compute lower bounds to the optimal objective value efficiently, we define an algorithm that outputs a heuristic solution to a packing problem that is the dual of a reformulation of weighted correlation clustering. Empirically, this algorithm is shown to exhibit a run-time/tightness trade-off that is different from both the cutting plane algorithm of Kappes et al. (2015) and the message passing algorithm of Swoboda & Andres (2017), both of which solve a canonical linear program relaxation of weighted correlation clustering.

Thirdly, toward the goal of obtaining primal feasible solutions, we define a transformation of the weights w.r.t. our heuristic solution to the dual problem. This transformation is again a heuristic and is motivated by complementary slackness. Empirically, local search algorithms are shown to find feasible solutions of lower original weight when applied to instances with transformed weights.

In the supplementary material, we provide additional results that were omitted from the main paper for the sake of space. Implementations of our algorithms are provided on GitHub.

2. Related Work

Weighted correlation clustering has a long history in the field of combinatorial optimization. Grötschel & Wakabayashi (1989) state an equivalent problem for complete graphs and

[1]Max Planck Institute for Informatics, Saarbrücken, Germany [2]Saarland University, Saarbrücken, Germany [3]Bosch Center for AI, Renningen, Germany [4]University of Tübingen, Germany. Correspondence to: Jan-Hendrik Lange <jlange@mpi-inf.mpg.de>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

devise a branch-and-cut algorithm for solving this problem exactly. The polyhedral geometry of its feasible set is studied by Grötschel & Wakabayashi (1990); Deza et al. (1990; 1992), in the case of general graphs by Chopra & Rao (1993); Chopra (1994) and, for a more general problem, by Horňáková et al. (2017). For uniform absolute edge costs, Bansal et al. (2004) coined the name *correlation clustering*, established NP-hardness and the first approximation results. The connection between correlation clustering in general weighted graphs and weighted multicuts was made by Demaine et al. (2006) who thus established APX-hardness and obtained an $\mathcal{O}(\log|V|)$ approximation algorithm for the problem. Further hardness results and improved approximation algorithms for particular classes of graphs are due to Charikar et al. (2005); Chawla et al. (2006; 2015); Ailon et al. (2012); Klein et al. (2015).

Approximation algorithms are typically based on rounding the solution of a linear or semi-definite program relaxation. Due to its importance, tailored algorithms for solving the linear program relaxation more efficiently than standard methods have been proposed by Yarkony et al. (2012; 2015); Swoboda & Andres (2017). Complementary to these lower bounds, a variety of fast primal heuristics have been developed to tackle large instances (Beier et al., 2014; Pan et al., 2015; Levinkov et al., 2017). Although it has been observed that, in practice, heuristic solutions are often good, it remains difficult for large instances to determine non-trivial bounds on their optimality gap.

Partial optimality concepts have been developed and exploited successfully for node labeling problems that arise from pseudo-Boolean optimization and from maximum a-posteriori inference in Markov Random Fields, cf. (Shekhovtsov, 2014; Swoboda et al., 2016). Transferring this knowledge to weighted correlation clustering is non-trivial, due to the different nature of the problem. Two partial optimality conditions for weighted correlation clustering are established by Alush & Goldberger (2012) and are here generalized.

3. Problem Formulations

3.1. Weighted Correlation Clustering

Weighted correlation clustering is a combinatorial optimization problem whose feasible solutions are all clusterings of a graph.

Let $G = (V, E)$ be a simple graph. We call a partition Π of V a *clustering* if every $S \in \Pi$ induces a connected subgraph (*cluster*) of G. For any clustering Π of G, we denote by E^0_Π the set of those edges whose nodes are in the same cluster, and by E^1_Π the (complementary) set of those edges whose nodes are in distinct clusters:

$$E^0_\Pi = \{uv \in E \mid \exists S \in \Pi : u \in S \text{ and } v \in S\}, \tag{1}$$

$$E^1_\Pi = E \setminus E^0_\Pi. \tag{2}$$

The set of edges E^1_Π is known as the *multicut* of G that corresponds to the clustering Π.

Definition 1. For any graph $G = (V, E)$ and any $w^0, w^1 : E \to \mathbb{R}^+_0$, the instance of *weighted correlation clustering* w.r.t. G, w^0 and w^1 is the optimization problem

$$\min_\Pi \quad \sum_{e \in E^0_\Pi} w^0_e + \sum_{e \in E^1_\Pi} w^1_e \ . \tag{3}$$

3.2. Minimum Cost Multicut

Weighted correlation clustering is commonly stated in the form of a binary program whose feasible solutions are the incidence vectors of the multicuts of the graph. The incidence vector $x^\Pi \in \{0,1\}^E$ corresponding to the multicut induced by Π is defined as

$$x^\Pi_e = \begin{cases} 1 & \text{if } e \in E^1_\Pi \\ 0 & \text{else.} \end{cases} \tag{4}$$

Definition 2. For any graph $G = (V, E)$ and any $c \colon E \to \mathbb{R}$, the instance of the *minimum cost multicut problem* w.r.t. G and c is the binary program

$$\min_\Pi \quad \sum_{e \in E} c_e \, x^\Pi_e \ . \tag{5}$$

The minimizers of an instance of weighted correlation clustering (Def. 1) coincide with the minimizers of the instance of minimum cost multicut (Def. 2) with $c = w^1 - w^0$, since

$$\begin{aligned} & \min_\Pi \sum_{e \in E^0_\Pi} w^0_e + \sum_{e \in E^1_\Pi} w^1_e && (6) \\ = & \min_\Pi \sum_{e \in E} \left(w^0_e \, (1 - x^\Pi_e) + w^1_e \, x^\Pi_e \right) && (7) \\ = & \underbrace{\sum_{e \in E} w^0_e}_{\text{const.}} + \min_\Pi \sum_{e \in E} \underbrace{(w^1_e - w^0_e)}_{c_e} x_e \ . && (8) \end{aligned}$$

3.3. Linear Program Relaxation

By taking the convex hull of multicut incidence vectors

$$\mathsf{MC}(G) := \operatorname{conv}\{x^\Pi \mid \Pi \text{ clustering of } G\}, \tag{9}$$

the minimum cost multicut problem (Def. 2) can be written as the integer linear programming problem

$$\min_{x \in \mathsf{MC}(G)} \quad \sum_{e \in E} c_e \, x_e. \tag{P_{MC}}$$

The set $\mathsf{MC}(G)$ is called *multicut polytope* of G (Chopra & Rao, 1993). As the minimum cost multicut problem is NP-hard, a full description of the multicut polytope in terms of its facets is impractical. For practical purposes a linear programming (LP) relaxation of P_{MC} is derived as follows.

Denote by $\mathcal{C}(G)$ the set of all simple cycles of G. For any cycle $C \in \mathcal{C}(G)$, we write E_C for the edge set of C. It is straight-forward to check the fact that any multicut incidence vector x^{Π} satisfies the system of linear inequalities

$$\forall C \in \mathcal{C}(G)\; \forall f \in E_C : \quad x_f \leq \sum_{e \in E_C \setminus \{f\}} x_e \;, \tag{10}$$

the so-called *cycle inequalities* (Chopra & Rao, 1993). Therefore, the standard linear programming relaxation is given by the program

$$\min_{x \in \mathsf{CYC}(G)} \quad \sum_{e \in E} c_e\, x_e \tag{$\mathrm{P}_{\mathsf{CYC}}$}$$

whose feasible set

$$\mathsf{CYC}(G) := \left\{ x \in [0,1]^E \;\middle|\; x \text{ satisfies (10)} \right\} \tag{11}$$

is also known as the *cycle relaxation* of $\mathsf{MC}(G)$. The problem $\mathrm{P}_{\mathsf{CYC}}$ is practical, because the cycle inequalities in (10) can be separated in polynomial time. The lower bounds thus obtained can serve to solve (small) instances of the minimum cost multicut problem by branch-and-cut because the cycle relaxation has no integer vertices except the incidence vectors of multicuts, according to Lemma 1.

Lemma 1 (Chopra & Rao (1993))**.** *For any graph $G = (V, E)$, it holds that $\mathsf{MC}(G) = \mathsf{CYC}(G) \cap \mathbb{Z}^E$.*

A reference algorithm that we use for the experiments in Section 7 further exploits the fact that a cycle inequality in (10) defines a facet of $\mathsf{MC}(G)$ iff the associated cycle is chordless.

3.4. Cycle Covering Formulation

For the presentation of this paper, we employ an alternative (integer) linear programming formulation in terms of covering cycles, which was similarly considered, e.g., by Demaine et al. (2006) for the combinatorial problem and by Charikar et al. (2005) in connection with the LP relaxation for complete graphs. We rewrite the feasible set of the general LP relaxation relative to the cost vector c. Therefore, let G and c be fixed.

We call an edge $e \in E$ *repulsive* if $c_e < 0$ and we call it *attractive* if $c_e > 0$. Note that we may w.l.o.g. remove all edges $e \in E$ with $c_e = 0$, since the choice of x_e is irrelevant to the objective. We write $E = E^+ \cup E^-$ where E^+, E^- collect all attractive and repulsive edges, respectively.

We call a cycle of G *conflicted* w.r.t. (G, c) if it contains precisely one repulsive edge. We denote by $\mathcal{C}^-(G,c) \subseteq \mathcal{C}(G)$ the set of all such cycles.

We consider the relaxation of $\mathsf{CYC}(G)$ that is constrained only by conflicted cycles. More specifically, we consider the system

$$\forall C \in \mathcal{C}^-(G,c), f \in E_C \cap E^- : \; x_f \leq \sum_{e \in E_C \setminus \{f\}} x_e \tag{12}$$

of only those linear inequalities of (10) for which the edge on the left-hand side is repulsive and all other edges are attractive. Defining

$$\mathsf{CYC}^-(G,c) := \left\{ x \in [0,1]^E \;\middle|\; x \text{ satisfies (12)} \right\} \tag{13}$$

and replacing $\mathsf{CYC}(G)$ by $\mathsf{CYC}^-(G,c)$ in $\mathrm{P}_{\mathsf{CYC}}$ has no effect on the solutions, due to the following lemma, a weaker form of which was also given by Yarkony et al. (2015).

Lemma 2. *For any $c\colon E \to \mathbb{R}$ it holds that*

$$\min_{x \in \mathsf{CYC}(G)} c^\top x = \min_{x \in \mathsf{CYC}^-(G,c)} c^\top x \tag{14}$$

and

$$\min_{x \in \mathsf{MC}(G)} c^\top x = \min_{x \in \mathsf{CYC}^-(G,c) \cap \mathbb{Z}^E} c^\top x. \tag{15}$$

Proof. Let x^* be an optimal solution to the right-hand side of (14). We show that x^* satisfies all cycle inequalities (10) by contradiction. To this end, suppose there exists a cycle $C \in \mathcal{C}(G)$ and $f \in E_C$ such that

$$x^*_f > \sum_{e \in E_C \setminus \{f\}} x^*_e.$$

If any edge $g \in E_C \setminus \{f\}$ is repulsive, then increasing x^*_g would lower the objective. Since x^* is optimal, there must be a conflicted cycle C' with $g \in E_{C'}$ such that $x^*_g = \sum_{e \in E_{C'} \setminus \{g\}} x^*_e$. Note that this means $f \notin E_{C'}$. We write $C \triangle C'$ for the cycle obtained from the symmetric difference of E_C and $E_{C'}$. Apparently, the cycle $C \triangle C'$ has one repulsive edge less and $f \in E_{C \triangle C'}$. Therefore, by repeating the argument, we may w.l.o.g. assume that all edges in $E_C \setminus \{f\}$ are attractive.
Now assume that f is attractive as well, then decreasing x^*_f would lower the objective. Therefore, since x^* is optimal, there is a conflicted cycle C' with $f \in E_{C'}$ and $g \in E_{C'} \cap E^-$ such that

$$\begin{aligned} x^*_g &= x^*_f + \sum_{e \in E_{C'} \setminus \{f,g\}} x^*_e \\ &> \sum_{e \in E_C \setminus \{f\}} x^*_e + \sum_{e \in E_{C'} \setminus \{f,g\}} x^*_e \\ &\geq \sum_{e \in E_{C \triangle C'} \setminus \{g\}} x^*_e. \end{aligned}$$

Note that $C \triangle C'$ is a conflicted cycle. Thus, we conclude that x^* violates an inequality of (12) and hence cannot be feasible. This concludes the proof of (14), the argument for (15) is analogous. $\square$

With the help of Lemma 2, we formulate P_{MC} as a set covering problem:

Definition 3. For any graph $G = (V, E)$ and any $c \in \mathbb{R}^E$, we call

$$\min_{\hat{x} \in \mathsf{SC}(G,c)} \quad \sum_{e \in E} |c_e| \, \hat{x}_e \tag{P_{SC}}$$

with $\mathsf{SC}(G, c)$ the convex hull of all $\hat{x} \in \mathbb{Z}^E$ that satisfy the system

$$\forall C \in \mathcal{C}^-(G, c): \quad \sum_{e \in E_C} \hat{x}_e \geq 1 \tag{16}$$

$$\forall e \in E: \qquad \hat{x}_e \geq 0 \tag{17}$$

the *set covering problem w.r.t. conflicted cycles*, and we call $\mathsf{SC}(G, c)$ the *set covering polyhedron w.r.t. conflicted cycles*.

Lemma 3. *For any graph $G = (V, E)$ and any $c \in \mathbb{R}^E$, we have*

$$\begin{aligned} &\min_{x \in \mathsf{CYC}^-(G,c) \cap \mathbb{Z}^E} \sum_{e \in E} c_e \, x_e \\ = \quad & L^{\mathrm{triv}} + \min_{\hat{x} \in \mathsf{SC}(G,c)} \sum_{e \in E} |c_e| \, \hat{x}_e \end{aligned} \tag{18}$$

with

$$L^{\mathrm{triv}} = \sum_{e \in E^-} c_e \tag{19}$$

the sum of negative edge costs (a trivial lower bound to the optimal value of P_{MC}*).*

Proof. We define $\hat{x}$ via $\hat{x}_e := x_e$ for any attractive edge $e \in E^+$ and $\hat{x}_e := 1 - x_e$ for any repulsive edge $e \in E^-$. Since any conflicted cycle $C \in \mathcal{C}^-(G, c)$ has precisely one repulsive edge, all conflicted cycle inequalities (12) become covering inequalities. $\square$

4. Partial Optimality

In this section, we study partial optimality for P_{MC}. More precisely, we establish conditions on an edge $e \in E$ which guarantee that x_e assumes one particular value, either 0 or 1, in at least one optimal solution (*weak persistency*). Fixations to 0 are of particular interest as they can be implemented as edge contractions (with subsequent merging of parallel edges), which effectively reduce the size of a given instance of the problem. As a corollary, we obtain an algorithm that solves weighted correlation clustering problems on series-parallel graphs in linear time.

4.1. Basic Conditions

A direct consequence from Lemma 3 is that we may disregard all edges that are not contained in any conflicted cycle. There are (at least) two ways this can happen: 1. An edge $e \in E$ is not contained in any cycle at all, that is, e is a *bridge*. 2. The endpoints of a repulsive edge $e = \{u, v\} \in E^-$ belong to different components of $G^+ = (V, E^+)$. In both cases, for any optimal solution x^* of P_{MC}, it holds that $x^*_e = 0$ if e is attractive, and $x^*_e = 1$ if e is repulsive. Thus, we can restrict the instance of the problem to the maximal components of G that are connected in G^+ and biconnected in G. This was also observed by Alush & Goldberger (2012).

Below, we establish more general partial optimality conditions. To this end, we need the following notation. A *cut* of G is a bipartition $B = (S_1, S_2)$ of the nodes V, i.e. $V = S_1 \dot\cup S_2$. The edge set of the cut B is denoted by $E_B = \{uv \in E \mid u \in S_1, v \in S_2\}$.

4.2. Dominant Edges

Definition 4. Let $G = (V, E)$ be any graph and let $c \in \mathbb{R}^E$. An edge $f \in E$ is called *dominant attractive* iff $c_f > 0$ and there exists a cut B with $f \in E_B$ such that

$$c_f \geq \sum_{e \in E_B \setminus \{f\}} |c_e| \ . \tag{20}$$

An edge $f \in E^-$ is called *dominant repulsive* iff $c_f < 0$ and there exists a cut B with $f \in E_B$ such that

$$|c_f| \geq \sum_{e \in E_B \cap E^+} c_e. \tag{21}$$

An edge is called *dominant* iff it is dominant attractive or dominant repulsive.

Lemma 4. *Let $G = (V, E)$ be any graph and let $c \in \mathbb{R}^E$.*

(i) If $f \in E$ is dominant attractive, then $x^*_f = 0$ in at least one optimal solution x^* of P_{MC}.

(ii) If $f \in E$ is dominant repulsive, then $x^*_f = 1$ in at least one optimal solution x^* of P_{MC}.

Proof. (i) We use the set covering formulation of P_{MC}. Suppose $f \in E^+$ is dominant and $\hat{x}^*_f = 1$ in an optimal solution $\hat{x}^*$ of P_{SC}. Every conflicted cycle that contains f also contains some edge $e \in E_B$, since B is a cut. Therefore, the vector $\hat{x} \in \{0, 1\}^E$ defined by

$$\hat{x}_e = \begin{cases} 0 & \text{if } e = f \\ 1 & \text{if } e \in E_B, e \neq f \\ \hat{x}^*_e & \text{else} \end{cases}$$

is a feasible solution to P_{SC}. It has the same objective value as $\hat{x}^*$, since f is dominant and $\hat{x}^*$ is optimal.

(ii) Suppose $f \in E^-$ is dominant and $\hat{x}^*_f = 1$ in an optimal solution $\hat{x}^*$ of P_{SC}. Every conflicted cycle that contains f also contains some edge $e \in E_B \cap E^+$, since B is a cut and every conflicted cycle contains only one repulsive edge. Then the vector $\hat{x} \in \{0,1\}^E$ defined by $\hat{x}_f = 0$, $\hat{x}_e = 1$ for all $e \in E_B \cap E^+$ and $\hat{x}_e = \hat{x}^*_e$ elsewhere is a feasible solution to P_{SC}. It has the same objective value as $\hat{x}^*$, since f is dominant and $\hat{x}^*$ is optimal. □

Lemma 4 generalizes the basic conditions discussed in Section 4.1, since each edge $f \in E$ that is not contained in any conflicted cycle is also dominant. Dominance of edges can be decided in polynomial time, by computing minimum st-cuts in G for a suitable choice of capacities. In practice, the required computational effort may be mitigated by constructing a *cut tree* of G (Gomory & Hu, 1961). The practically most relevant cuts can even be checked in linear time, which we discuss in the following section.

4.3. Two-Edge Cuts & Single-Node Cuts

In practice, it is expected that dominant edges are more likely to be found in cuts that are relatively sparse. We discuss two special cases of sparse cuts that are of particular interest, due to the following reasons. First, they can be checked in linear time, which gives rise to a fast preprocessing algorithm. Second, we show that our techniques solve P_{MC} to optimality if G is series-parallel.

Two-edge cuts. Suppose B is a two-edge cut of G, i.e. $E_B = \{e, f\}$ for two edges $e, f \in E$. Apparently, according to (20) and (21), at least one of them must be dominant. Further, it is guaranteed that we can simplify the instance by edge deletions or contractions. To see this, distinguish the following cases. If both e and f are repulsive, then both of them are dominant and we can delete them, as they are not contained in any conflicted cycle. If f is dominant attractive, we can contract f. Finally, if f is dominant repulsive and e is attractive, then we can switch the signs of their coefficients and redefine $x_f := 1 - x_f$ as well as $x_e := 1 - x_e$. Since $|E_B| = 2$, this operation does not change the set of conflicted cycles of G and thus is valid (while only adding a constant to the objective). Afterwards, the edge f is dominant attractive and we can contract f. The two-edge cuts of G can be found in linear time, by computing the 3-edge-connected components of G, cf. (Mehlhorn et al., 2017).

Single-node cuts. For any $v \in V$, let $B_v = (\{v\}, V \setminus \{v\})$ denote the cut that is induced by v. Whether E_{B_v} contains a dominant edge is easily decided by considering all edges incident to v. Moreover, if $\deg v = 2$, then B_v is also a two-edge cut and we can apply the operation described in the last paragraph. Updating the graph and applying these techniques recursively as specified in Algorithm 1 takes linear time. This has the following theoretical consequence.

Algorithm 1 Single-Node Cut Preprocessing

input $G = (V, E), c\colon E \to \mathbb{R}$
Initialize objective value offset $\Delta = 0$.
Initialize a queue $Q = V$.
while $Q \neq \emptyset$ **do**
  Extract a vertex $v \in Q$.
  if $\deg v = 1$ **then**
    Get neighbor $u \in V$.
    if $c_{uv} \geq 0$ **then**
      Set $x_{uv} = 0$ and contract $uv \in E$.
    else
      Set $x_{uv} = 1$, $\Delta = \Delta + c_{uv}$ and delete $uv \in E$.
    end if
  else if $\deg v = 2$ **then**
    Get neighbors $u, w \in V$ with $|c_{uv}| \geq |c_{wv}|$.
    if $uv \in E^+$ **then**
      Set $x_{uv} = 0$ and contract $uv \in E$.
    else if $uv \in E^-$ and $wv \in E^-$ **then**
      Adjust offset $\Delta = \Delta + c_{uv} + c_{wv}$.
      Set $x_{uv} = x_{wv} = 1$ and delete $uv, wv \in E$.
    else if $uv \in E^-$ and $wv \in E^+$ **then**
      Adjust offset $\Delta = \Delta + c_{uv} + c_{wv}$.
      Redefine $x_{uv} = 1 - x_{uv}$, $x_{wv} = 1 - x_{wv}$ and $c_{uv} = -c_{uv}$, $c_{wv} = -c_{wv}$.
      Set $x_{uv} = 0$ and contract $uv \in E$.
    end if
  else if $\exists f \in B_v$ dominant attractive **then**
    Set $x_f = 0$ and contract $f \in E$.
  end if
  Add to Q all vertices $u \notin Q$ whose neighborhood was changed.
end while
return $(G, c), x, \Delta$

Corollary 1. *If G has treewidth at most 2, then Algorithm 1 can be implemented to solve P_{MC} exactly in $\mathcal{O}(|V|)$ time.*

Proof. Place the vertices of G into buckets of ascending degree and always pick a vertex of minimal degree. Every graph of treewidth 2 has a vertex v with $\deg v \leq 2$. Since Algorithm 1 only contracts or deletes edges, fixing the variables according to Lemma 4, the updated graph still has treewidth at most 2. The number of nodes decreases by 1 in every iteration, hence the algorithm terminates in time $\mathcal{O}(|E|) = \mathcal{O}(|V|)$ and outputs an optimal solution. □

5. Dual Lower Bounds

In this section, we define an algorithm for computing lower bounds for P_{MC}. This algorithm exploits the structure of

Algorithm 2 Iterative Cycle Packing (ICP)

input $G = (V, E), c\colon E \to \mathbb{R}$

Initialize $w_e = |c_e|$ for all $e \in E$ and $y = 0$, $L = L^{\text{triv}}$.
for $\ell = 3 \dots |E|$ **do**
  while $\exists C \in \mathcal{C}^-(G, c) : |E_C| \le \ell$ **do**
    Pick $C \in \mathcal{C}^-(G, c)$ such that $|E_C| \le \ell$.
    Compute $y_C = \min_{e \in E_C} w_e$.
    Redefine $w_e = \begin{cases} w_e - y_C & \text{if } e \in E_C \\ w_e & \text{else.} \end{cases}$
    Increase lower bound $L = L + y_C$.
    Remove all edges $e \in E$ with $w_e = 0$ from G.
  end while
  if $\mathcal{C}^-(G, c) = \emptyset$ **then**
    return y, L
  end if
end for

the reformulation P_{SC}. It computes a heuristic solution to the dual of its LP relaxation.

The LP relaxation (up to the constant L^{triv}) of problem P_{SC} is given by

$$\min \quad \sum_{e \in E} |c_e| \hat{x}_e \tag{22}$$

$$\text{subject to} \quad \sum_{e \in E_C} \hat{x}_e \ge 1 \qquad \forall C \in \mathcal{C}^-(G, c) \tag{23}$$

$$\hat{x}_e \ge 0 \qquad \forall e \in E \ .$$

The corresponding dual program reads

$$\max \quad \sum_{C \in \mathcal{C}^-(G,c)} y_C \tag{24}$$

$$\text{subject to} \quad \sum_{C\colon e \in E_C} y_C \le |c_e| \qquad \forall e \in E \tag{25}$$

$$y_C \ge 0 \qquad \forall C \in C^-(G, c) \ .$$

A heuristic solution of (24), and thus a lower bound for (22), is found by Algorithm 2 that we call Iterative Cycle Packing (ICP). It works as follows: Firstly, it chooses a conflicted cycle C and increases y_C as much as possible. Secondly, it decreases the weights w_e (initially $|c_e|$) of all edges $e \in E_C$ by y_C and removes all edges of zero weight. These steps are repeated until there are no conflicted cycles left.

Implementation details. The absolute running time of ICP as well as the quality of the output lower bounds depends on the choice of cycles C. We pursue the following strategy that we found to perform well empirically in both aspects: In each iteration of the main loop, we choose a repulsive edge $e = uv \in E^-$ such that u and v are in the same connected component of $G^+ = (V, E^+)$. Then, we find a conflicted cycle containing e by searching for a shortest path (in terms of hop distance) from u to v in G^+. We apply this search for conflicted cycles in rounds of increasing cycle length, using breadth-first search with an early termination criterion based on the hop distance. We also maintain and periodically update a component labeling of G^+ in order to to reduce the number of redundant shortest path searches.

Table 1. Shown below are reductions due to partial optimality, averaged across the n instances of the respective collection. We report the sizes of V and E after the reduction in terms of the original size. The set $E^\ddagger \subseteq E$ includes all remaining *non-persistent* edges. For comparison, the set $E^\dagger$ collects all non-persistent edges after exploiting only the criteria from Alush & Goldberger (2012).

	n	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert E^\ddagger\rvert$	$\lvert E^\dagger\rvert$
Image Seg.	100	63.7%	62.7%	53.5%	85.6%
Knott-3D-150	8	75.2%	88.3%	76.0%	92.7%
Knott-3D-300	8	76.7%	91.6%	82.7%	95.8%
Knott-3D-450	8	77.6%	92.4%	86.6%	96.7%
Mod. Clust.	6	92.0%	85.1%	85.1%	100%
Epinions	1	27.4%	83.2%	83.2%	86.8%
Slashdot	1	45.4%	88.9%	88.9%	92.2%

6. Re-weighting for Primal Algorithms

In this section, we exploit the dual solution in primal algorithms. The motivation is due to complementary slackness, which is made explicit in the following lemma.

Lemma 5. *Assume the primal LP* (22) *is tight, i.e., its optimal solution* $\hat{x}^*$ *also solves* P_{SC}*, and the solution output by ICP solves the dual* (24) *optimally. Then, for every* $e \in E$ *with positive residual weight* $w_e > 0$*, it holds that* $\hat{x}^*_e = 0$.

Proof. If $w_e > 0$, the constraint (25) at $e \in E$ is inactive at the optimal dual solution. Thus, $\hat{x}^*_e = 0$ in the optimal primal solution, by complementary slackness. □

Of course, the assumption of Lemma 5 is too strong for practical purposes. However, the intuition is that if the LP relaxation is fairly tight and the obtained dual solution is close to optimal, it can still provide useful information about the primal problem. More specifically, the weights w_e output by ICP can be interpreted as an indication of how likely the primal variable $\hat{x}_e$ is zero in an optimal solution. In order to make use of this information, we propose to shift the weights of the primal problem to a convex combination $\lambda|c_e| + (1 - \lambda)w_e$ of the original and residual weights, for a suitable choice of $\lambda \in (0, 1)$. Experiments in Section 7 show that this shift can guide primal heuristics toward better feasible solutions to the original problem.

7. Experiments

In this section, we study partial optimality, dual lower bounds and re-weightings empirically, for all instances of

Table 2. Reported below are lower bounds found by linear programming (LP), message passing (MPC) and iterative cycle packing (ICP), relative to the objective value of the best known feasible solution, as well as average total running times until convergence.

	Image Seg.		*Knott-3D-150*		*Knott-3D-300*		*Knott-3D-450*		*Mod. Clust.*		*Epinions*		*Slashdot*	
	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap
LP	1.31	0.03%	1.16	0.01%	148.54	0.00%	2174.95	0.01%	0.98	5.59%	–	–	–	–
MPC	7.42	0.03%	0.38	0.01%	17.34	0.00%	1236.28	0.01%	1.13	9.52%	36012	0.10%	7419	7.54%
ICP	0.10	0.21%	0.04	0.14%	0.54	0.16%	3.31	0.18%	0.03	11.08%	65	1.04%	24	8.60%

the weighted correlation clustering problem from Kappes et al. (2015) and Leskovec et al. (2010).

Instances. From Kappes et al. (2015), we consider all three collections of instances: *Image Segmentation* contains instances w.r.t. planar superpixel adjacency graphs of photographs. *Knott-3D* contains instances w.r.t. non-planar supervoxel adjacency graphs of volume images taken by a serial sectioning electron microscope. *Modularity Clustering* contains instances w.r.t. complete graphs. In all three collections, the edge costs c_e are fractional and non-uniform. For all these instances, except one in the collection *Modularity Clustering*, optimal solutions are accessible and are computed here as a reference. From Leskovec et al. (2010), we consider directed graphs of the social networks *Epinions* and *Slashdot*, each with more than half a million edges labeled either $+1$ or -1. Instances of the minimum cost multicut problem are defined here by removing the orientation of edges, by deleting all self-loops, and by replacing parallel edges by a single edge with the sum of their costs[1].

7.1. Partial Optimality

In order to study the partial optimality conditions of Section 4 empirically, we process the above instances as follows: First, we remove all edges of cost 0, all bridges, as well as all repulsive edges whose nodes belong to distinct connected components of G^+. Second, we check for every $v \in V$ whether the cut $B_v = (\{v\}, V \setminus \{v\})$ induces dominant edges. If we find dominant attractive edges or vertices of degree ≤ 2, we perform contractions and deletions according to Alg. 1. Both steps are repeated until no further edges can be removed or contracted.

After the main reduction step, which takes linear time and is thus very fast, we further check all remaining edges $uv \in E$ for dominance in any (general) uv-cut. To this end, we construct a cut tree of G with the help of Gusfield's algorithm (Gusfield, 1990), which takes $|V| - 1$ max-flow computations. Despite the increased computational effort, we only found a small number of additional dominant attractive edges and thus could only perform few further contractions. However, we found a significant number of additional dominant repulsive edges.

The effect of our method in the total number of nodes and edges is shown in Table 1. We also report the number of remaining edges that are not dominant repulsive. It can be seen from this table that the numbers are effectively reduced. This is explained, firstly, by the sparsity of the graphs and, secondly, by the non-uniformity of the costs. From the comparison to the number of remaining non-persistent variables when only the criteria of Alush & Goldberger (2012) are applied, it can be seen that our more general criteria reveal considerably more persistency.

It may be expected that optimization methods benefit in terms of runtime from the reduced size of the instances. On the instances of Kappes et al. (2015), we found the effect to be insignificant due to their small original size. On *Epinions* and *Slashdot*, however, the runtime of the local search algorithm GAEC+KLj (cf. Section 7.3) decreased by more than 70%. For completeness, we provide the numbers in the supplements.

7.2. Dual Lower Bounds

In order to put into perspective the dual lower bounds output by Iterative Cycle Packing (ICP) as described in Section 5, we compare this algorithm, firstly, to the cutting plane algorithm for $\mathrm{P}_{\mathsf{CYC}}$ of Kappes et al. (2015), with Gurobi for solving the LPs (denoted here by LP) and, secondly, to the message passing algorithm of Swoboda & Andres (2017), applied to $\mathrm{P}_{\mathsf{CYC}}$, with code and parameter settings kindly provided by the authors (denoted here by MPC).

Results are shown in Figure 1 and Table 2. It can be seen from the figure and the table that, for the large and hard instances *Epinions* and *Slashdot*, ICP converges at under 10^2 seconds, outputting lower bounds that are matched and exceeded by MPC at around 10^3 seconds. It can be seen from Table 2 that the situation is similar for the smaller instances: The lower bounds output by ICP are a bit worse than those output by LP or MPC (here compared to the best optimal solution known) but are obtained faster (by as much as three orders of magnitude for *Knott-3D-450*).

It is known from Kappes et al. (2015) that their instances can be solved faster than their LP relaxations by means of branch-and-cut, separating only integer infeasible points

[1] This results in 2703 edges of cost 0 for *Epinions*, and 1949 such edges for *Slashdot*.

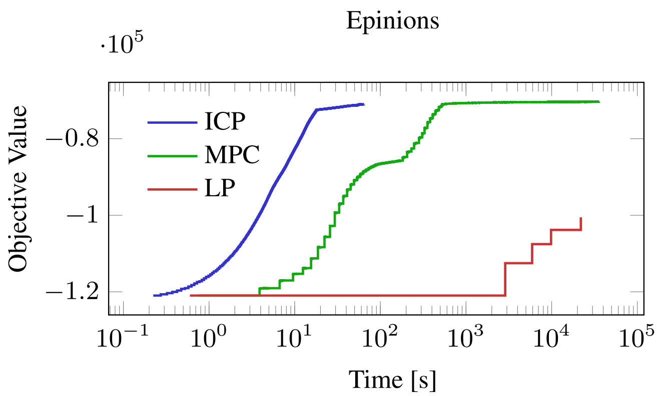

Figure 1. Shown above is the convergence of lower bounds found by linear programming (LP), message passing (MPC) and iterative cycle packing (ICP), for the large and hard instances *Epinions* and *Slashdot*.

Table 3. Reported below is the effect of re-weighting instances of weighted correlation clustering by ICP. Feasible solutions found by the heuristic algorithms GAEC and GAEC+KLj for original instances (no prefix) are compared to feasible solutions found by these algorithms for reweighted instances (prefix ICP). All gaps are w.r.t. the original cost function obtained by relation to the objective value of the best known lower bound.

	Image Seg.		*Knott-3D-150*		*Knott-3D-300*		*Knott-3D-450*		*Mod. Clust.*		*Epinions*		*Slashdot*	
	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap
GAEC	0.01	0.53%	0.01	0.28%	0.06	0.47%	0.21	0.25%	0.01	8.09%	2.19	1.75%	1.52	7.76%
ICP+GAEC	0.09	0.14%	0.04	0.02%	0.55	0.24%	3.61	0.08%	0.03	3.80%	68.53	0.56%	25.82	7.06%
GAEC+KLj	0.03	0.34%	0.02	0.11%	0.42	0.15%	6.46	0.07%	0.01	1.28%	8324	0.18%	9217	6.69%
ICP+GAEC+KLj	0.11	0.10%	0.05	0.00%	0.81	0.03%	8.71	0.05%	0.04	1.14%	7336	0.12%	4451	6.58%

by cycle inequalities using BFS (instead of Dijkstra's algorithm), and resorting to the strong (undisclosed) cuts of Gurobi for cutting off fractional solutions. We restrict our comparison here to algorithms that seek to solve the LP relaxation $\mathrm{P}_{\mathrm{CYC}}$. This is justified by the fact that size ultimately renders integer linear programming intractable. We conclude that ICP is capable of computing non-trivial lower bounds fast.

7.3. Re-weighting

In order to study the re-weighting described in Section 6, we measure its effect on heuristic algorithms for finding feasible solutions. To this end, we employ the implementations of Levinkov et al. (2017) of *Greedy Additive Edge Contraction* (GAEC), an algorithm that starts from singleton clusters and greedily contracts attractive edges with maximum non-negative cost, and of KLj, the well-known Kernighan-Lin heuristic for graph partitioning that recursively improves an initial clustering by splitting, merging or exchanging nodes between neighboring clusters.

A comparison between the feasible solutions found by applying the heuristics GAEC and GAEC+KLj to original instances, on the one hand, and to instances re-weighted by ICP with $\lambda = \frac{1}{2}$, on the other hand, can be found in Table 3. Note that we only re-weight the input to GAEC and let KLj run with original weights, starting from the solution returned by GAEC, as we found this approach to be advantageous. It can be seen from Table 3 that our re-weighting consistently improves the gap. On average, it is slightly less effective than the reparameterization with the more accurate dual solutions obtained from MPC, as proposed by Swoboda & Andres (2017). A more detailed comparison is provided in the supplements.

8. Conclusion

We have established partial optimality conditions, a heuristic lower bound and a heuristic re-weighting for instances of the weighted correlation clustering problem. We have shown advantages of each of these constructions empirically. Checking a subset of our partial optimality conditions recursively gives a fast combinatorial algorithm that efficiently reduces the size of problem instances. Conceptually, it solves the problem for series-parallel graphs to optimality, in linear time. Our dual heuristic algorithm provides non-trivial lower bounds and valuable dual information fast. For future work, it is relevant to examine if more sophisticated dual solvers such as MPC benefit from a "warm-start" that transforms and exploits the heuristic dual solution.

References

Ailon, N., Avigdor-Elgrabli, N., Liberty, E., and van Zuylen, A. Improved approximation algorithms for bipartite correlation clustering. *SIAM Journal on Computing*, 41(5): 1110–1121, 2012. doi: 10.1137/110848712.

Alush, A. and Goldberger, J. Ensemble segmentation using efficient integer linear programming. *TPAMI*, 34(10): 1966–1977, 2012. doi: 10.1109/TPAMI.2011.280.

Bansal, N., Blum, A., and Chawla, S. Correlation clustering. *Machine Learning*, 56(1–3):89–113, 2004. doi: 10.1023/B:MACH.0000033116.57574.95.

Beier, T., Kroeger, T., Kappes, J. H., Köthe, U., and Hamprecht, F. A. Cut, glue & cut: A fast, approximate solver for multicut partitioning. In *CVPR*, 2014.

Beier, T., Pape, C., Rahaman, N., Prange, T., Berg, S., Bock, D. D., Cardona, A., Knott, G. W., Plaza, S. M., Scheffer, L. K., Koethe, U., Kreshuk, A., and Hamprecht, F. A. Multicut brings automated neurite segmentation closer to human performance. *Nature Methods*, 14(2):101–102, 2017. doi: 10.1038/nmeth.4151.

Cesa-Bianchi, N., Gentile, C., Vitale, F., and Zappella, G. A correlation clustering approach to link classification in signed networks. In *COLT*, 2012.

Charikar, M., Guruswami, V., and Wirth, A. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005. doi: 10.1016/j.jcss.2004.10.012.

Chawla, S., Krauthgamer, R., Kumar, R., Rabani, Y., and Sivakumar, D. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006. doi: 10.1007/s00037-006-0210-9.

Chawla, S., Makarychev, K., Schramm, T., and Yaroslavtsev, G. Near optimal LP rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *STOC*, 2015. doi: 10.1145/2746539.2746604.

Chopra, S. The graph partitioning polytope on series-parallel and 4-wheel free graphs. *SIAM Journal on Discrete Mathematics*, 7(1):16–31, 1994. doi: 10.1137/S0895480191199415.

Chopra, S. and Rao, M. The partition problem. *Mathematical Programming*, 59(1–3):87–115, 1993. doi: 10.1007/BF01581239.

Demaine, E. D., Emanuel, D., Fiat, A., and Immorlica, N. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2–3):172–187, 2006. doi: 10.1016/j.tcs.2006.05.008.

Deza, M., Grötschel, M., and Laurent, M. Complete descriptions of small multicut polytopes. In *Applied Geometry And Discrete Mathematics, Proceedings of a DIMACS Workshop*, pp. 221–252, 1990.

Deza, M. M., Grötschel, M., and Laurent, M. Clique-web facets for multicut polytopes. *Mathematics of Operations Research*, 17(4):981–1000, 1992. doi: 10.1287/moor.17.4.981.

Gomory, R. E. and Hu, T. C. Multi-Terminal Network Flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961. ISSN 03684245. doi: 10.2307/2098881.

Grötschel, M. and Wakabayashi, Y. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1):59–96, 1989. doi: 10.1007/BF01589097.

Grötschel, M. and Wakabayashi, Y. Facets of the clique partitioning polytope. *Mathematical Programming*, 47: 367–387, 1990. doi: 10.1007/BF01580870.

Gusfield, D. Very simple methods for all pairs network flow analysis. *SIAM J. Comput.*, 19(1):143–155, 1990. doi: 10.1137/0219009.

Horňáková, A., Lange, J.-H., and Andres, B. Analysis and optimization of graph decompositions by lifted multicuts. In *ICML*, 2017.

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, pp. 34–50, 2016. doi: 10.1007/978-3-319-46466-4_3.

Kappes, J. H., Speth, M., Andres, B., Reinelt, G., and Schnörr, C. Globally optimal image partitioning by multicuts. In *EMMCVPR*, 2011.

Kappes, J. H., Andres, B., Hamprecht, F. A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B. X., Kröger, T., Lellmann, J., Komodakis, N., Savchynskyy, B., and Rother, C. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*, 115 (2):155–184, 2015. doi: 10.1007/s11263-015-0809-x.

Keuper, M., Andres, B., and Brox, T. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015. doi: 10.1109/ICCV.2015.374.

Klein, P. N., Mathieu, C., and Zhou, H. Correlation clustering and two-edge-connected augmentation for planar graphs. In *STACS*, 2015. doi: 10.4230/LIPIcs.STACS.2015.554.

Leskovec, J., Huttenlocher, D. P., and Kleinberg, J. M. Signed networks in social media. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI*, pp. 1361–1370, 2010. doi: 10.1145/1753326.1753532.

Levinkov, E., Kirillov, A., and Andres, B. A comparative study of local search algorithms for correlation clustering. In *GCPR*, 2017. doi: 10.1007/978-3-319-66709-6_9.

Mehlhorn, K., Neumann, A., and Schmidt, J. M. Certifying 3-edge-connectivity. *Algorithmica*, 77(2):309–335, Feb 2017. doi: 10.1007/s00453-015-0075-x.

Pan, X., Papailiopoulos, D. S., Oymak, S., Recht, B., Ramchandran, K., and Jordan, M. I. Parallel correlation clustering on big graphs. In *NIPS*, 2015.

Shekhovtsov, A. Maximum persistency in energy minimization. In *CVPR*, pp. 1162–1169, 2014. doi: 10.1109/CVPR.2014.152.

Swoboda, P. and Andres, B. A message passing algorithm for the minimum cost multicut problem. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.530.

Swoboda, P., Shekhovtsov, A., Kappes, J. H., Schnörr, C., and Savchynskyy, B. Partial optimality by pruning for MAP-inference with general graphical models. *TPAMI*, 38(7):1370–1382, 7 2016. doi: 10.1109/TPAMI.2015.2484327.

Tang, S., Andriluka, M., Andres, B., and Schiele, B. Multiple people tracking by lifted multicut and person reidentification. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.394.

Yarkony, J., Ihler, A. T., and Fowlkes, C. C. Fast planar correlation clustering for image segmentation. In *ECCV 2012*, pp. 568–581, 2012. doi: 10.1007/978-3-642-33783-3_41.

Yarkony, J., Beier, T., Baldi, P., and Hamprecht, F. A. Parallel multicut segmentation via dual decomposition. In Appice, A., Ceci, M., Loglisci, C., Manco, G., Masciari, E., and Ras, Z. W. (eds.), *New Frontiers in Mining Complex Patterns*, pp. 56–68, Cham, 2015. Springer International Publishing. doi: 10.1007/978-3-319-17876-9_4.