A Benders Decomposition Approach to Correlation Clustering

Margret Keuper
University of Mannheim
Baden Wurttemberg, Germany
keuper@uni-mannheim.de

Jovita Lukasik
University of Mannheim
Baden Wurttemberg, Germany
jovita@informatik.uni-mannheim.de

Maneesh Singh
Verisk
Jersey City, New Jersey, USA
maneesh.singh@verisk.com

Julian Yarkony
Verisk
Jersey City, New Jersey, USA
julian.yarkony@verisk.com

Abstract

We tackle the problem of graph partitioning for image segmentation using correlation clustering (CC), which we treat as an integer linear program (ILP). We reformulate optimization in the ILP so as to admit efficient optimization via Benders decomposition, a classic technique from operations research. Our Benders decomposition formulation has many subproblems, each associated with a node in the CC instance's graph, which can be solved in parallel. Each Benders subproblem enforces the cycle inequalities corresponding to edges with negative (repulsive) weights attached to its corresponding node in the CC instance. We generate Magnanti-Wong Benders rows in addition to standard Benders rows to accelerate optimization. Our Benders decomposition approach provides a promising new avenue to accelerate optimization for CC, and, in contrast to previous cutting plane approaches, theoretically allows for massive parallelization.

1 Introduction

Many computer vision tasks involve partitioning (clustering) a set of observations into unique entities. A powerful formulation for such tasks is that of (weighted) correlation clustering (CC). CC is defined on a sparse graph with real valued edge weights, where nodes correspond to observations and weighted edges describe the affinity between pairs of nodes.

For example, in image segmentation (on superpixel graphs), nodes correspond to superpixels and edges indicate adjacency between superpixels. The weight of the edge between a pair of superpixels relates to the probability, as defined by a classifier, that the two superpixels belong to the same ground truth entity. This weight is positive, if the probability is greater than $\frac{1}{2}$ and negative if it is less than $\frac{1}{2}$. The magnitude of the weight is a function of the confidence of the classifier.

The CC cost function sums up the weights of the edges separating connected components (referred to as entities) in a proposed partitioning of the graph. Optimization in CC partitions the graph into entities so as to minimize the CC cost. CC is appealing, since the optimal number of entities emerges naturally as a function of the edge weights, rather than requiring an additional search over some model order parameter describing the number of clusters (entities) [37].

Optimization in CC is NP-hard for general graphs [5]. Previous methods for the optimization of CC problems such as described in Andres et al. [1, 2] and Nowozin and Jegelka [25] are based on linear programming with cutting planes. They do not scale easily to large CC problem instances and are not

Preprint. Under review.

easily parallelizable. The goal of this paper is to introduce an efficient mechanism for optimization in CC for domains, where massively parallel computation could be employed.

In this paper we apply the classic Benders decomposition from operations research [10] to CC for computer vision. Benders decomposition is commonly applied in operations research to solve mixed integer linear programs (MILP) that have a special but common block structure. Benders decomposition partitions the variables in the MILP between a master problem and a set of subproblems. The block structure requires that no row of the constraint matrix of the MILP contains variables from more than one subproblem. Variables explicitly enforced to be integral lie only in the master problem.

Optimization in Benders decomposition is achieved using a cutting plane algorithm. Optimization proceeds with the master problem solving optimization over its variables. The subsequent solution of the subproblems can be done in parallel and provides primal/dual solutions over their variables conditioned on the solution to the master problem. The dual solutions to the subproblems provide constraints to the master problem. Optimization continues until no further constraints are added to the master problem.

Benders decomposition is an exact MILP programming solver, but can be intuitively understood as a coordinate descent procedure, iterating between the master problem and the subproblems. Here, solving the subproblems not only provides a solution for their variables, but also a lower bound in the form of a hyper-plane over the master problem's variables. This lower bound is tight at the current solution to the master problem.

Benders decomposition is accelerated using the seminal operations research technique of Magnanti-Wong Benders rows (MWR) [23]. MWR are generated by solving the Benders subproblems with an alternative (often random) objective under the hard constraint of optimality (possibly within a factor) regarding the original objective of the subproblem.

Our contribution is the use of Benders decomposition with MWR to tackle optimization in CC. This allows for massive parallelization, in contrast to classic approaches to CC such as in Andres et al. [1].

2 Related Work

Correlation clustering has been successfully applied to multiple problems in computer vision including image segmentation, multi-object tracking, instance segmentation and multi-person pose estimation. The classical work of Andres et al. [1] models image segmentation as CC, where nodes correspond to superpixels. Andres et al. [1] optimize CC using an integer linear programming (ILP) branch-and-cut strategy which precludes parallel execution. Kim et al. [21] extend CC to include higher-order cost terms over sets of nodes, which they solve using an approach similar to [1]. A parallel optimization scheme for complete, unweighted graphs has been proposed by Pan et al. [26]. This approach relies on random sampling and only provides optimality bounds.

Yarkony et al. [37] tackle CC in the planar graph structured problems commonly found in computer vision. They introduce a column generation [16, 6] approach, where the pricing problem corresponds to finding the lowest reduced cost 2-colorable partition of the graph, via a reduction to minimum cost perfect matching [13, 29, 22]. This approach has been extended to hierarchical image segmentation in Yarkony and Fowlkes [35] and to specific cases of non-planar graphs in Yarkony [34], Zhang et al. [39], Andres et al. [3].

Large CC problem instances such as defined in Keuper et al. [20, 19] and Beier et al. [9] are usually addressed by primal feasible heuristics [7, 8, 18, 20, 30]. Such approaches are highly relevant in practice whenever the optimal solution is out of reach, but they do not provide any guarantees on the quality of the solution.

Tang et al. [31] tackles multi-object tracking using a formulation closely related to CC, where nodes correspond to detections of objects and edges are associated with probabilities of co-association.The work of Insafutdinov et al. [17] and Pishchulin et al. [27] build on Tang et al. [31] in order to formulate multi-person pose estimation using CC augmented with node labeling.

Our work is derived from the classical work in operations research on Benders decomposition [10, 11, 15]. Specifically, we are inspired by the fixed charge formulations of Cordeau et al. [12], which solves a mixed integer linear program over a set of fixed charge variables (opening links) and a larger set of fractional variables (flows of commodities from facilities to customers in a network) associated with constraints. Benders decomposition reformulates optimization so as to use only the integer variables and converts the fractional variables into constraints. These constraints are referred to as Benders rows. Optimization is then tackled using a cutting plane approach. Optimization is accelerated by the use of MWR [23], which are more binding than the standard Benders rows.

Benders decomposition has recently been introduced to computer vision (though not for CC), for the purpose of multi-person pose estimation [32, 33, 36]. In these works, multi-person pose estimation is modeled so as to admit efficient optimization, using column generation and Benders decomposition jointly. The application of Benders decomposition in our paper is distinct regarding the problem domain, the underlying integer program and the structure of the Benders subproblems.

3 Standard Correlation Clustering Formulation

In this section, we review the standard optimization formulation for CC [1], which corresponds to a graph partitioning problem w.r.t. the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. This problem is defined by the following binary edge labeling problem.

Definition 1. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $v \in \mathcal{V}$ and undirected edges $(v_i, v_j) \in \mathcal{E}$. A label $x_{v_iv_j} \in \{0, 1\}$ indicates with $x_{v_iv_j} = 1$ that the nodes v_i, v_j are in separate components and is zero otherwise. Given the edge weight $\phi_{v_iv_j} \in \mathbb{R}$, the binary edge labeling problem is to find an edge label $\mathbf{x} = (x_{v_iv_j}) \in \{0, 1\}^{|\mathcal{E}|}$, for which the total weight of the cut edges is minimized:*

$$\min_{\mathbf{x}\in\{0,1\}^{|\mathcal{E}|}} \sum_{(v_i,v_j)\in\mathcal{E}^-} -\phi_{v_iv_j}(1 - x_{v_iv_j}) + \sum_{(v_i,v_j)\in\mathcal{E}^+} \phi_{v_iv_j} x_{v_iv_j} \tag{CC$_1$}$$

$$\text{s.t.} \sum_{(v_i,v_j)\in\mathcal{E}_c^+} x_{v_iv_j} \geq x_{v_i^c v_j^c} \quad \forall c \in \mathcal{C}, \tag{1}$$

where $\mathcal{E}^-, \mathcal{E}^+$ denote the subsets of $\mathcal{E}$, for which the weight $\phi_{v_iv_j}$ is negative and non-negative, respectively, $\mathcal{C}$ is the set of undirected cycles in $\mathcal{E}$ containing exactly one member of $\mathcal{E}^-$, (v_i^c, v_j^c) is the edge in $\mathcal{E}^-$ associated with cycle c and $\mathcal{E}_c^+ \subseteq \mathcal{E}^+$ associated with cycle c.

Note that the graph $\mathcal{G}$ defined by $\mathcal{E}$ is very sparse for real problems [37]. Also we refer to an edge (v_i, v_j) with $x_{v_iv_j} = 1$ as a *cut* edge.

The objective in Eq. (CC$_1$) is to minimize the total weight of the cut edges. The constraints in Eq. (1) ensure that, within every cycle of $\mathcal{G}$, the number of cut edges can not be exactly one. This enforces the labeling $\mathbf{x}$ to decompose $\mathcal{G}$ such that cut edges are exactly those edges that straddle distinct components. We refer to the constraints in Eq. (1) as cycle inequalities.

Solving Eq. (CC$_1$) is intractable due to the large number of cycle inequalities. Andres et al. [1] generates solutions by alternating between solving the ILP over a nascent set of constraints $\hat{\mathcal{C}}$ (initialized empty) and adding new constraints from the set of currently violated cycle inequalities. Generating constraints corresponds to iterating over $(v_i, v_j) \in \mathcal{E}^-$ and identifying the shortest path between the nodes v_i, v_j in the graph with edges $\mathcal{E} \setminus (v_i, v_j)$ and weights equal to $\mathbf{x}$. If the corresponding path has total weight less than $x_{v_iv_j}$, the corresponding constraint is added to $\hat{\mathcal{C}}$. The LP relaxation of Eq. (CC$_1$)-(1) can be solved instead of the ILP in each iteration until no violated cycle inequalities exist, after which the ILP must be solved in each iteration.

We should note that earlier work in CC for computer vision did not require that cycle inequalities contain exactly one member of $\mathcal{E}^-$, which is on the right hand side of Eq. (1). It is established with Lemma(1) in Yarkony et al. [38], that the addition of cycle inequalities, that contain edges in $\mathcal{E}^-$, $\mathcal{E}^+$ on the left hand side, right hand side of Eq. (1), respectively, do not tighten the ILP in Eq. (CC$_1$)-(1) or its LP relaxation.

In this section, we reviewed the baseline approach for solving CC in the computer vision community. In the subsequent sections, we rely on the characterization of CC in Eq. (CC$_1$)-(1), though not on the specific solver of Andres et al. [1].

4 Benders Decomposition for Correlation Clustering

In this section, we introduce a novel approach to CC using Benders decomposition (referred to as BDCC). Our proposed decomposition is defined by a minimal vertex cover on $\mathcal{E}^-$ with members $\mathcal{S} \subset \mathcal{V}$ indexed by v_s. Each $s \in \mathcal{S}$ is associated with a Benders subproblem and v_s is referred to as the root of that Benders subproblem. Edges in $\mathcal{E}^-$ are partitioned arbitrarily between the subproblems, such that each $(v_i, v_j) \in \mathcal{E}^-$ is associated with either the subproblem with root v_i or the subproblem with root v_j. Here, $\mathcal{E}_s^-$ is the subset of $\mathcal{E}^-$ associated with subproblem s. The subproblem with root v_s enforces the cycle inequalities $\mathcal{C}_s$, where $\mathcal{C}_s$ is the subset of $\mathcal{C}$ containing edges in $\mathcal{E}_s^-$. We use $\mathcal{E}_s^+$ to denote the subset of $\mathcal{E}^+$ adjacent to v_s.

In this section, we assume that we are provided with $\mathcal{S}$, which can be produced greedily or using an LP/ILP solver.

Below, we rewrite Eq. (CC$_1$) using an auxiliary function $Q(\phi, s, \mathbf{x})$. Here $Q(\phi, s, \mathbf{x})$ provides the cost to alter $\mathbf{x}$ to satisfy all cycle inequalities in $\mathcal{C}_s$, by increasing/decreasing $x_{v_iv_j}$ for (v_i, v_j) in $\mathcal{E}^+$/$\mathcal{E}_s^-$, respectively. Below we describe the changes of the master's problem edge labeling $\mathbf{x}$, which is based on the edge labeling of each Benders subproblem $\mathbf{x}^s = (x^s_{v_iv_j}) \in \{0,1\}^{|s|}$, where $|s|$ is the number of edges in the subproblem s.

$$(\text{CC}_1) \rightsquigarrow (\text{CC}_2)\colon \min_{\mathbf{x}\in\{0,1\}^{|\mathcal{E}|}} \sum_{(v_i,v_j)\in\mathcal{E}^-} -\phi_{v_iv_j}(1 - x_{v_iv_j}) + \sum_{(v_i,v_j)\in\mathcal{E}^+} \phi_{v_iv_j} x_{v_iv_j} + \sum_{s\in\mathcal{S}} Q(\phi, s, \mathbf{x}), \tag{CC$_2$}$$

where $Q(\phi, s, \mathbf{x})$ is defined as follows.

$$\begin{aligned} Q(\phi, s, \mathbf{x}) \quad &= \min_{\mathbf{x}^s\in\{0,1\}^{|s|}} \sum_{(v_i,v_j)\in\mathcal{E}_s^-} -\phi_{v_iv_j}(1 - x^s_{v_iv_j}) + \sum_{(v_i,v_j)\in\mathcal{E}^+} \phi_{v_iv_j} x^s_{v_iv_j} \qquad (2) \\ \text{s.t.} \quad & \sum_{(v_i,v_j)\in\mathcal{E}_c^+} x_{v_iv_j} + x^s_{v_iv_j} \geq x_{v_i^c v_j^c} - (1 - x^s_{v_i^c v_j^c}) \quad \forall c \in \mathcal{C}_s. \end{aligned}$$

We now construct a solution $\mathbf{x}^* = \{x^*_{v_iv_j}, (x^{s*}_{v_iv_j})_{s\in\mathcal{S}}\}$ for which Eq. (CC$_2$) is minimized and all cycle inequalities are satisfied. We start from a given solution $\mathbf{x} = \{x_{v_iv_j}, (x^s_{v_iv_j})_{s\in\mathcal{S}}\}$ and proceed as follows.

$$x^*_{v_iv_j} \triangleq \min(x_{v_iv_j}, x^s_{v_iv_j}) \quad \forall (v_i, v_j) \in \mathcal{E}_s^-, s \in \mathcal{S} \tag{3}$$

$$x^*_{v_iv_j} \triangleq x_{v_iv_j} + \max_{s\in\mathcal{S}} x^s_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^+. \tag{4}$$

The right hand side of Eq. (4) cannot exceed 1 at optimality because of the constraint in Eq. (2).

Given the solution $x^*_{v_iv_j}$, the optimizing solution to each Benders subproblem s is denoted $x^{s*}_{v_iv_j}$ and is defined as follows.

$$x^{s*}_{v_iv_j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in \mathcal{E}_s^- \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

In Sec. A in the supplement, we show that the cost of $\{x^*_{v_iv_j}, (x^{s*}_{v_iv_j})_{s\in\mathcal{S}}\}$ is no greater than that of $\{x_{v_iv_j}, (x^s_{v_iv_j})_{s\in\mathcal{S}}\}$, with regard to the objective in Eq. (CC$_2$) and that $Q(\phi, s, \mathbf{x}^*) = 0$ holds for all $s \in \mathcal{S}$.
It follows that there always exists an optimizing solution $\mathbf{x}$ to Eq. (CC$_2$) such that $Q(\phi, s, \mathbf{x}) = 0$ for all $s \in \mathcal{S}$.

Observe, that there exists an optimal partition $\mathbf{x}^s$ of the nodes of the graph , in Eq. (2), which is 2-colorable. This is because any partition $\mathbf{x}^s$ can be altered without increasing its cost, by merging connected components that are adjacent to one another, not including the root node v_s. Note, that merging any pair of such components, does not increase the cost, since those components are not separated by negative weight edges in subproblem s and so the result is still a partition.

Given this observation, we rewrite the optimization Eq. (CC$_2$) regarding $Q(\phi, s, \mathbf{x})$, using the node labeling formulation of min-cut, with the notation below.

We indicate with $m_v = 1$ that node $v \in \mathcal{V}$ is not in the component associated with the root of subproblem s and $m_v = 0$ otherwise. To avoid extra notation m_{v_s} is replaced by 0. Let

$$f^s_{v_iv_j} = \begin{cases} 1, & \text{for } (v_i,v_j) \in \mathcal{E}^+, \text{if } (v_i,v_j) \text{ is cut in } \mathbf{x}^s, \text{ but is not cut in } \mathbf{x} \\ 1, & \text{for } (v_i,v_j) \in \mathcal{E}^-_s, \text{if } (v_i,v_j) \text{ is not cut in } \mathbf{x}^s, \text{ but is cut in } \mathbf{x}. \end{cases} \tag{6}$$

Thus, the definition for the first/second case implies a penalty of $\phi_{v_iv_j}$/ - $\phi_{v_iv_j}$, which is added to $Q(\phi, s, \mathbf{x})$. Note moreover that $x^s_{v_iv_j} = f^s_{v_iv_j}$ for all $(v_i,v_j) \in \mathcal{E}^+$ and that $x^s_{v_iv_j} = 1 - f^s_{v_iv_j}$ for all $(v_i,v_j) \in \mathcal{E}^-_s$.

Below we write $Q(\phi, s, \mathbf{x})$ as primal/dual LP, with primal constraints associated with dual variables ψ, λ, which are noted in the primal. Given binary $\mathbf{x}$, we need only enforce that f, m are non-negative to ensure that there exists an optimizing solution for f, m which is binary. This is a consequence of the optimization being totally unimodular, given that $\mathbf{x}$ is binary. Total unimodularity is a known property of the min-cut/max flow LP [14]. The primal subproblem is therefore given by the following.

$$\begin{aligned} Q(\phi, s, \mathbf{x}) \quad &= \quad \min_{\substack{f^s_{v_iv_j} \geq 0 \\ m_v \geq 0}} \sum_{(v_i,v_j)\in\mathcal{E}^+} \phi_{v_iv_j} f^s_{v_iv_j} - \sum_{(v_s,v)\in\mathcal{E}^-_s} \phi_{v_sv} f^s_{v_sv} \\ \lambda^-_{v_iv_j} \quad &: \quad m_{v_i} - m_{v_j} \leq x_{v_iv_j} + f^s_{v_iv_j} \quad \forall (v_i,v_j) \in (\mathcal{E}^+ \setminus \mathcal{E}^+_s), \\ \lambda^+_{v_iv_j} \quad &: \quad m_{v_j} - m_{v_i} \leq x_{v_iv_j} + f^s_{v_iv_j} \quad \forall (v_i,v_j) \in (\mathcal{E}^+ \setminus \mathcal{E}^+_s), \\ \psi^-_v \quad &: \quad x_{v_sv} - f^s_{v_sv} \leq m_v \quad \forall (v_s,v) \in \mathcal{E}^-_s, \\ \psi^+_v \quad &: \quad m_v \leq x_{v_sv} + f^s_{v_sv} \quad \forall (v_s,v) \in \mathcal{E}^+_s, \end{aligned} \tag{7}$$

This yields to the corresponding dual subproblem.

$$\max_{\substack{\lambda \geq 0 \\ \psi \geq 0}} - \sum_{(v_i,v_j)\in(\mathcal{E}^+ \setminus \mathcal{E}^+_s)} (\lambda^-_{v_iv_j} + \lambda^+_{v_iv_j}) x_{v_iv_j} + \sum_{(v_s,v)\in\mathcal{E}^-_s} \psi^-_v x_{v_sv} - \sum_{(v_s,v)\in\mathcal{E}^+_s} \psi^+_v x_{v_sv} \tag{8}$$

$$\begin{aligned} \text{s.t.} \quad & \psi^+_{v_i} \, \mathbb{1}_{\mathcal{E}^+_s}(v_s, v_i) - \psi^-_{v_i} \, \mathbb{1}_{\mathcal{E}^-_s}(v_s, v_i) + \\ & \sum_{\substack{v_j \\ (v_i,v_j)\in(\mathcal{E}^+ \setminus \mathcal{E}^+_s)}} (\lambda^-_{v_iv_j} - \lambda^+_{v_iv_j}) + \sum_{\substack{v_j \\ (v_j,v_i)\in(\mathcal{E}^+ \setminus \mathcal{E}^+_s)}} (\lambda^+_{v_jv_i} - \lambda^-_{v_jv_i}) \geq 0 \qquad \forall v_i \in \mathcal{V} - v_s \end{aligned}$$

$$\begin{aligned} -\phi_{v_sv} - \psi^-_v \quad &\geq \quad 0 \qquad \forall (v_s, v) \in \mathcal{E}^-_s \\ \phi_{v_sv} - \psi^+_v \quad &\geq \quad 0 \qquad \forall (v_s, v) \in \mathcal{E}^+_s \\ \phi_{v_iv_j} - (\lambda^-_{v_iv_j} + \lambda^+_{v_iv_j}) \quad &\geq \quad 0 \qquad \forall (v_i,v_j) \in (\mathcal{E}^+ \setminus \mathcal{E}^+_s). \end{aligned}$$

In Eq. (8) and subsequently $\mathbb{1}_\Lambda(x)$ denotes the binary indicator function for some set Λ, which returns one if $(x \in \Lambda)$ and zero otherwise. We now consider the constraint that $Q(\phi, s, \mathbf{x}) = 0$. Note that any dual feasible solution for the dual problem (8) describes an affine function of $\mathbf{x}$, which is a tight lower bound on $Q(\phi, s, \mathbf{x})$. We compact the terms λ, ψ into ω^z, where $\omega^z_{v_iv_j}$ is associated with the $x_{v_iv_j}$ term.

$$\omega^z_{v_iv_j} = \begin{cases} -(\lambda^-_{v_iv_j} + \lambda^+_{v_iv_j}), & \text{if } (v_i,v_j) \in (\mathcal{E}^+ \setminus \mathcal{E}^+_s) \\ -\psi^+_{v_j}, & \text{if } (v_i,v_j) \in \mathcal{E}^+_s \\ \psi^-_{v_j}, & \text{if } (v_i,v_j) \in \mathcal{E}^-_s \\ 0, & \text{if } (v_i,v_j) \in (\mathcal{E}^- \setminus \mathcal{E}^-_s). \end{cases}$$

We denote the set of all dual feasible solutions across $s \in \mathcal{S}$ as $\mathcal{Z}$, with $z \in \mathcal{Z}$. Observe, that to enforce that $Q(\phi, s, \mathbf{x}) = 0$, it is sufficient to require that $\sum_{(v_i,v_j)\in\mathcal{E}} x_{v_iv_j} \omega^z_{v_iv_j} \leq 0$, for all $z \in \mathcal{Z}$. We formulate CC as optimization using $\mathcal{Z}$ below.

$$(\text{CC}_2) \rightsquigarrow (\text{CC}_3) = \min_{\mathbf{x}\in\{0,1\}^{|\mathcal{E}|}} \sum_{(v_i,v_j)\in\mathcal{E}^+} \phi_{v_iv_j} x_{v_iv_j} - \sum_{(v_i,v_j)\in\mathcal{E}^-} (1 - x_{v_iv_j}) \phi_{v_iv_j} \tag{CC$_3$}$$

$$\text{s.t.} \sum_{(v_i,v_j)\in\mathcal{E}} x_{v_iv_j} \omega^z_{v_iv_j} \leq 0 \qquad \forall z \in \mathcal{Z}$$

Algorithm 1 Benders Decomposition for CC (BDCC)

```
$\hat{\mathcal{Z}} = \{\}$
done_LP = False
repeat
    $\mathbf{x}$ = Solve Eq. (CC$_3$) over $\hat{\mathcal{Z}}$ enforcing integrality if and only if done_LP=True
    did_add = False
    for $s \in \mathcal{S}$ do
        if $\exists (v_i, v_j) \in \mathcal{E}_s^-$ s.t. $d(v_i, v_j) < x_{v_i v_j}$ then
            $z_1$ = Get Benders row via Eq (8).
            $z_2$ = Get MWR via Sec. 5.
            $\hat{\mathcal{Z}} = \hat{\mathcal{Z}} \cup z_1 \cup z_2$
            did_add = True
        end if
    end for
    if did_add=False then
        done_LP = True
    end if
until did_add=False AND $x_{v_i v_j} \in \{0, 1\} \; \forall (v_i, v_j) \in \mathcal{E}$
Return $\mathbf{x}$
```

4.1 Cutting Plane Optimization

Optimization in Eq. (CC$_3$) is intractable since $|\mathcal{Z}|$ equals the number of dual feasible solutions across subproblems, which is infinite. Since we cannot consider the entire set $\mathcal{Z}$, we use a cutting plane approach to construct a set $\hat{\mathcal{Z}} \subset \mathcal{Z}$, that is sufficient to solve Eq. (CC$_3$) exactly. We initialize $\hat{\mathcal{Z}}$ as the empty set. We iterate between solving the LP relaxation of Eq. (CC$_3$) over $\hat{\mathcal{Z}}$ (referred to as the master problem) and generating new Benders rows until no violated constraints exist.

This ensures that no violated cycle inequalities exist but may not ensure that $\mathbf{x}$ is integral. To enforce integrality, we iterate between solving the ILP in Eq. (CC$_3$) over $\hat{\mathcal{Z}}$ and adding Benders rows to $\hat{\mathcal{Z}}$. By solving the LP relaxation first, we avoid unnecessary and expensive calls to the ILP solver.

To generate Benders rows given $\mathbf{x}$, we iterate over $\mathcal{S}$ and generate one Benders row using Eq. (8), if s is associated with a violated cycle inequality, which we determine as follows. Given $s, \mathbf{x}$ we iterate over $(v_i, v_j) \in \mathcal{E}_s^-$. We find the shortest path from v_i to v_j on graph $\mathcal{G}$ with edges $\mathcal{E}$, with weights equal to the vector $\mathbf{x}$. If the length of this path, denoted as $d(v_i, v_j)$, is strictly less than $x_{v_i v_j}$, then we have identified a violated cycle inequality associated with s.

We describe our cutting plane approach in Alg. 1, with line by line description in Sec. B in the supplementary material. To accelerate optimization, we add MWR in addition to standard Benders rows, which we describe in the following Sec. 5.

Prior to termination of Alg. 1, one can produce a feasible integer solution $\mathbf{x}^*$ from any solution $\mathbf{x}$, provided by the master problem, as follows. First, for each $(v_i, v_j) \in \mathcal{E}$, set $x^{**}_{v_i v_j} = 1$, if $x_{v_i v_j} > \frac{1}{2}$ and otherwise set $x^{**}_{v_i v_j} = 0$. Second, for each $(v_i, v_j) \in \mathcal{E}$, set $x^{*}_{v_i v_j} = 1$, if v_i, v_j are in separate connected components of the solution described by $\mathbf{x}^{**}$ and otherwise set $x^{*}_{v_i v_j} = 0$. The cost of the feasible integer solution $\mathbf{x}^*$ provides an upper bound on the cost of the optimal solution. In Sec. C (supplementary material), we provide a more involved approach to produce feasible integer solutions.

In this section, we characterized CC using Benders decomposition and provided a cutting plane algorithm to solve the corresponding optimization.

5 Magnanti-Wong Benders Rows

We accelerate Benders decomposition (see Sec. 4) using the classic operations research technique of Magnanti-Wong Benders Rows (MWR) [23]. The Benders row, given in Eq. (8), provides a tight bound at $\mathbf{x}^*$, where $\mathbf{x}^*$ is the master problem solution used to generate the Benders row. However, ideally, we want our Benders row to provide good lower bounds for a large set of $\mathbf{x}$ different from $\mathbf{x}^*$,

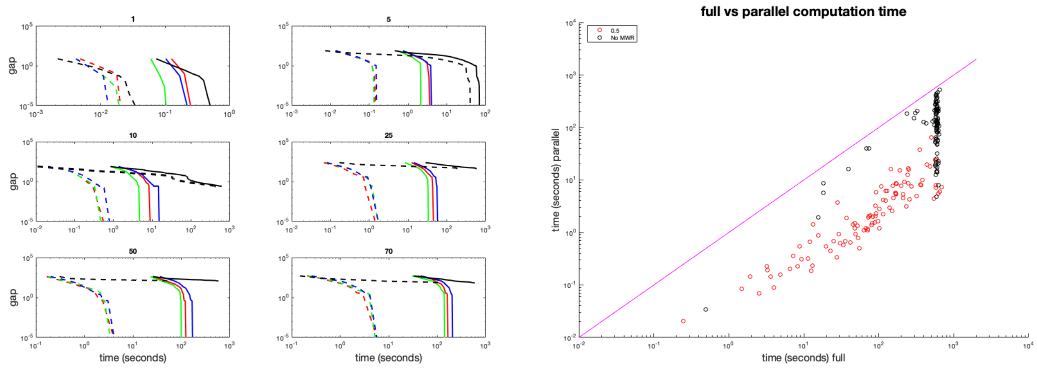

Figure 1: **Left:** We plot the gap between the upper and lower bounds as a function of time for various values of τ on selected problem instances. We use red,green,blue for $\tau = [0.5, 0.99, .01]$ respectively, and black for not using Magnanti-Wong rows. We show both the computation time with and without exploiting parallelization of subproblems with dotted and solid lines, respectively. We use titles to indicate the approximate difficulty of the problem as ranked by input file size of 100 files.
Right: We compare the benefits of parallelization and MWR across our data set. We scatter plot the total running time versus the total running time when solving each subproblem is done on its own CPU across problem instances. We use red to indicate $\tau = 0.5$ and black to indicate that MWR are not used. We draw a line with slope=1 in magenta to better enable appreciation of the red and black points. NOTE: The time spent generating Benders rows, in a given iteration of BDCC when using parallel processing, is the maximum time spent to solve any sub-problem for that iteration.

while being tight (or perhaps very active) at $\mathbf{x}^*$. To achieve this, we use a modified version of Eq. (8), where we replace the objective and add one additional constraint.

We follow the tradition of the operations research literature and use a random negative valued vector (with unit norm) in place of the objective Eq. (8). This random vector is unique each time a Benders subproblem is solved. We experimented with using as an objective $\frac{-1}{.0001+|\phi_{v_i v_j}|}$, which encourages the cutting of edges with large positive weight, but it works as well as the random negative objective. Here .0001 is a tiny positive number. It prevents the terms in the objective from becoming infinite.

Below, we enforce the new Benders row to be active at $\mathbf{x}^*$, by requiring that the dual cost is within a tolerance $\tau \in (0, 1)$ of the optimum w.r.t. the objective in Eq. (8).

$$\tau Q(\phi, s, \mathbf{x}) \leq \; - \sum_{(v_i,v_j)\in(\mathcal{E}^+ \setminus \mathcal{E}_s^+)} (\lambda^-_{v_i v_j} + \lambda^+_{v_i v_j}) x_{v_i v_j} + \sum_{(v_s,v)\in\mathcal{E}_s^-} \psi_v^- x_{v_s v} - \sum_{(v_s,v)\in\mathcal{E}_s^+} \psi_v^+ x_{v_s v} \tag{9}$$

Here, $\tau = 1$ requires optimality w.r.t. the objective in Eq. (8), while $\tau = 0$ ignores optimality. In our experiments, we found that $\tau = \frac{1}{2}$ provides strong performance.

6 Experiments: Image Segmentation

In this section, we demonstrate the value of our algorithm BDCC on CC problem instances for image segmentation on the benchmark Berkeley Segmentation Data Set (BSDS) [24]. Our experiments demonstrate the following three findings. (1) BDCC solves CC instances for image segmentation; (2) BDCC successfully exploits parallelization; (3) the use of MWR dramatically accelerates optimization.

To benchmark performance, we employ cost terms provided by the OPENGM2 dataset [4] for BSDS. This allows for a direct comparison of our results to the ones from Andres et al. [1]. We use the random unit norm negative valued objective when generating MWR. We use CPLEX to solve all linear and integer linear programming problems considered during the course of optimization. We use a maximum total CPU time of 600 seconds, for each problem instance (regardless of parallelization).

Table 1: We show the percentage of problems solved that have a duality gap of up to tolerance ϵ, within a certain amount of time (10,50,100,300) seconds, with and without MWR/parallelization. We use par =1 to indicate the use of parallelization and par=0 otherwise. Here $\tau = 0$ means that no MWR are generated.

ϵ=0.1	τ	par	10	50	100	300
	0.5	0	0.149	0.372	0.585	0.894
	0	0	0.0106	0.0532	0.0745	0.106
	0.5	1	0.266	0.777	0.904	0.968
	0	1	0.0426	0.0745	0.0745	0.138
ϵ=1	τ	par	10	50	100	300
	0.5	0	0.149	0.394	0.606	0.904
	0	0	0.0106	0.0638	0.0745	0.16
	0.5	1	0.319	0.819	0.947	0.979
	0	1	0.0532	0.0745	0.106	0.17
ϵ=10	τ	par	10	50	100	300
	0.5	0	0.202	0.426	0.628	0.915
	0	0	0.0532	0.0957	0.128	0.223
	0.5	1	0.447	0.936	0.979	0.989
	0	1	0.0638	0.128	0.181	0.287

We formulate the selection of $\mathcal{S}$, as a minimum vertex cover problem, where for every edge $(v_i, v_j) \in \mathcal{E}^-$, at least one of v_i, v_j is in $\mathcal{S}$. We solve for the minimum vertex cover exactly as an ILP. Given $\mathcal{S}$, we assign edges in $\mathcal{E}^-$ to a connected selected node in $\mathcal{S}$ arbitrarily. We found experimentally that solving for the minimum vertex cover consumed negligible CPU time for our dataset. We attribute this fact to the structure of our problem domain, since the minimum vertex cover is an NP-hard problem. For problem instances where solving for the minimum vertex cover exactly is difficult, the minimum vertex cover problem can be solved approximately or greedily.

In Fig. 1 (left) we demonstrate the effectiveness of BDCC with various τ for different problem difficulties. We observe that the presence of MWR dramatically accelerates optimization. However, the exact value of τ does not effect the speed of optimization dramatically. We show performance with and without relying on parallel processing. Our parallel processing times assume that we have one CPU for each subproblem. For the problem instances in our application the number of subproblems is under one thousand, each of which are very easy to solve. The parallel and non-parallel time comparisons share only the time to solve the master problem. We observe large benefits of parallelization for all settings of τ. However, when MWR are not used, we observe diminished improvement, since the master problem consumes a larger proportion of total CPU time.

In Fig. 1(right), we demonstrate the speed up induced by the use of parallelization. For most problem instances, the total CPU time required when using no MWR was prohibitively large, which is not the case when MWR are employed. Thus most problem instances solved without MWR terminated early.

In Tab. 1, we consider the convergence of the bounds for $\tau = \{0, \frac{1}{2}\}$; ($\tau = 0$ means that no MWR are generated). We consider a set of tolerances on convergence regarding the duality gap, which is the difference between the anytime solution (upper bound) and the lower bound on the objective. For each such tolerance ϵ, we compute the percentage of instances, for which the duality gap is less than ϵ, after various amounts of time. We observe that the performance of optimization without MWR, but exploiting parallelization performs worse than using MWR, but without paralleliziation. This demonstrates that, across the dataset, MWR are of greater importance than parallelization.

7 Conclusions

We present a novel methodology for finding optimal correlation clustering in arbitrary graphs. Our method exploits the Benders decomposition to avoid the enumeration of a large number of cycle inequalities. This offers a new technique in the toolkit of linear programming relaxations, that we expect will find further use in the application of combinatorial optimization to problems in computer vision.

The exploitation of results from the domain of operations research may lead to improved variants of BDCC. For example, one can intelligently select the subproblems to solve instead of solving all subproblems in each iteration. This strategy is referred to as partial pricing in the operations research literature. Similarly one can devote a minimum amount of time in each iteration to solve the master problem so as to enforce integrality on a subset of the variables of the master problem.

References

[1] B. Andres, J. H. Kappes, T. Beier, U. Kothe, and F. A. Hamprecht. Probabilistic image segmentation with closedness constraints. In *Proceedings of the Fifth International Conference on Computer Vision (ICCV-11)*, pages 2611–2618, 2011.

[2] B. Andres, T. Kroger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Kothe, and F. A. Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *Proceedings of the Twelveth International Conference on Computer Vision (ECCV-12)*, 2012.

[3] B. Andres, J. Yarkony, B. S. Manjunath, S. Kirchhoff, E. Turetken, C. Fowlkes, and H. Pfister. Segmenting planar superpixel adjacency graphs w.r.t. non-planar superpixel affinity graphs. In *Proceedings of the Ninth Conference on Energy Minimization in Computer Vision and Pattern Recognition (EMMCVPR-13)*, 2013.

[4] B. Andres, T. Beier, and J. H. Kappes. Opengm2, 2014.

[5] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In *Journal of Machine Learning*, pages 238–247, 2002.

[6] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1996.

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

[8] T. Beier, F. A. Hamprecht, and J. H. Kappes. Fusion moves for correlation clustering. In *CVPR*, 2015.

[9] T. Beier, B. Andres, K. Ullrich, and F. A. Hamprecht. An efficient fusion move algorithm for the minimum cost lifted multicut problem. volume LNCS 9906, pages 715–730. Springer, 2016. doi: 10.1007/978-3-319-46475-6_44.

[10] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

[11] J. R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations research*, 33(5):989–1007, 1985.

[12] J.-F. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation science*, 35(4):375–388, 2001.

[13] M. E. Fisher. On the dimer solution of planar ising models. *Journal of Mathematical Physics*, 7(10): 1776–1781, 1966.

[14] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3): 399–404, 1956.

[15] A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by benders decomposition. *Management science*, 20(5):822–844, 1974.

[16] P. Gilmore and R. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research (volume 9)*, 1961.

[17] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[18] A. Kardoost and M. Keuper. Solving minimum cost lifted multicut problems by node agglomeration. In *ACCV 2018, 14th Asian Conference on Computer Vision*, Perth, Australia, 2018.

[19] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015.

[20] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *ICCV*, 2015.

[21] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo. Higher-order correlation clustering for image segmentation. In *Advances in Neural Information Processing Systems,25*, pages 1530–1538, 2011.

[22] V. Kolmogorov. Blossom v: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation*, 1(1):43–67, 2009.

[23] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations research*, 29(3):464–484, 1981.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth International Conference on Computer Vision (ICCV-01)*, pages 416–423, 2001.

[25] S. Nowozin and S. Jegelka. Solution stability in linear programming relaxations: Graph partitioning and unsupervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 769–776. ACM, 2009.

[26] X. Pan, D. Papailiopoulos, S. Oymak, B. Recht, K. Ramchandran, and M. I. Jordan. Parallel correlation clustering on big graphs. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 82–90, Cambridge, MA, USA, 2015. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2969239.2969249`.

[27] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4929–4937, 2016.

[28] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, june 2007.

[29] W.-K. Shih, S. Wu, and Y. Kuo. Unifying maximum cut and minimum cut of a planar graph. *Computers, IEEE Transactions on*, 39(5):694–697, May 1990.

[30] P. Swoboda and B. Andres. A message passing algorithm for the minimum cost multicut problem. In *CVPR*, 2017.

[31] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Subgraph decomposition for multi-target tracking. In *CVPR*, 2015.

[32] S. Wang, K. Kording, and J. Yarkony. Exploiting skeletal structure in computer vision annotation with benders decomposition. *arXiv preprint arXiv:1709.04411*, 2017.

[33] S. Wang, A. Ihler, K. Kording, and J. Yarkony. Accelerating dynamic programs via nested benders decomposition with application to multi-person pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–666, 2018.

[34] J. Yarkony. Next generation multicuts for semi-planar graphs. In *Proceedings of the Neural Information Processing Systems Optimization in Machine Learning Workshop (OPT-ML)*, 2015.

[35] J. Yarkony and C. Fowlkes. Planar ultrametrics for image segmentation. In *Neural Information Processing Systems*, 2015.

[36] J. Yarkony and S. Wang. Accelerating message passing for map with benders decomposition. *arXiv preprint arXiv:1805.04958*, 2018.

[37] J. Yarkony, A. Ihler, and C. Fowlkes. Fast planar correlation clustering for image segmentation. In *Proceedings of the 12th European Conference on Computer Vision(ECCV 2012)*, 2012.

[38] J. Yarkony, T. Beier, P. Baldi, and F. A. Hamprecht. Parallel multicut segmentation via dual decomposition. In *International Workshop on New Frontiers in Mining Complex Patterns*, pages 56–68. Springer, 2014.

[39] C. Zhang, F. Huber, M. Knop, and F. Hamprecht. Yeast cell detection and segmentation in bright field microscopy. In *ISBI*, 2014.

A APPENDIX: $Q(\phi, s, \mathbf{x}^*) = 0$ at Optimality

In this section, we demonstrate that there exists an $\mathbf{x}^*$, that minimizes Eq. (CC$_2$), for which $Q(\phi, s, \mathbf{x}^*) = 0$. Given an arbitrary solution $\{x_{v_iv_j}, (x^s_{v_iv_j})_{s\in\mathcal{S}}\}$ another solution $\{x^*_{v_iv_j}, (x^{s*}_{v_iv_j})_{s\in\mathcal{S}}\}$ is constructed, for which $Q(\phi, s, \mathbf{x}^*) = 0$ holds, without increasing the objective in Eq. (CC$_2$). We write the updates below in terms of $\mathbf{x}^s$.

$$\begin{aligned} x^*_{v_iv_j} &\triangleq x_{v_iv_j} + \max_{s\in\mathcal{S}} x^s_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^+ \\ x^*_{v_iv_j} &\triangleq x_{v_iv_j} + x^s_{v_iv_j} - 1 \quad \forall (v_i, v_j) \in \mathcal{E}^-_s, s \in \mathcal{S} \\ x^{s*}_{v_iv_j} &\triangleq 0 \quad \forall (v_i, v_j) \in \mathcal{E}^+ \\ x^{s*}_{v_iv_j} &\triangleq 1 \quad \forall (v_i, v_j) \in \mathcal{E}^-_s, s \in \mathcal{S}. \end{aligned} \tag{10}$$

The updates in Eq. (10) are equivalent to the following updates using f^s,f^{s*}. Here f^s, f^{s*} correspond to the optimizing solution for f in subproblem s, given $\mathbf{x}, \mathbf{x}^*$ respectively.

$$\begin{aligned} x^*_{v_iv_j} &= x_{v_iv_j} + \max_{s\in\mathcal{S}} f^s_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^+ \\ x^*_{v_iv_j} &= x_{v_iv_j} - f^s_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^-_s, s \in \mathcal{S} \\ f^{s*}_{v_iv_j} &= 0 \quad \forall (v_i, v_j) \in \mathcal{E}^+ \\ f^{s*}_{v_iv_j} &= 0 \quad \forall (v_i, v_j) \in \mathcal{E}^-_s \end{aligned} \tag{11}$$

These updates in Eq. (10) and Eq. (11) preserve the feasibility of the primal LP in Eq. (7). Also notice, that since f^{s*} is a zero valued vector for all $s \in \mathcal{S}$, then $Q(\phi, s, x^*) = 0$ for all $s \in \mathcal{S}$.

We now consider, the total change in Eq. (CC$_2$) corresponding to edge $(v_i, v_j) \in \mathcal{E}^+$, induced by Eq. (10), which is non-positive. The objective of the master problem increases by $\phi_{v_iv_j} \max_{s\in\mathcal{S}} x^s_{v_iv_j}$, while the total decrease in the objectives of the subproblems is $\phi_{v_iv_j} \sum_{s\in\mathcal{S}} x^s_{v_iv_j}$. Since the latter value is greater than the former value, the total change in problem (CC$_2$) decreases more than it increases. Considering on the other hand the total change of Eq. (CC$_2$) corresponding to edge $(v_i, v_j) \in \mathcal{E}^-$, induced by Eq. (10), which is zero, yields in an increase of the objective of the master problem by $-\phi_{v_iv_j}(1 - x^n_{v_iv_j})$, while the objective of subproblem s decreases by $-\phi_{v_iv_j}(1 - x^s_{v_iv_j})$. This shows that the objective of Eq. (CC$_2$) is minimized for $\mathbf{x}^*$.

B Line by Line Description of BDCC

We provide the line by line description of Alg. 1.

- Line 1: Initialize the nascent set of Benders rows $\hat{\mathcal{Z}}$ to the empty set.
- Line 2: Indicate that we have not solved the LP relaxation yet.
- Line 3-17: Alternate between solving the master problem and generating Benders rows, until a feasible integral solution is produced.
    1. Line 4: Solve the master problem providing a solution $\mathbf{x}$, which may not satisfy all cycle inequalities. We enforce integrality if we have finished solving the LP relaxation, which is indicated by done_lp=True.
    2. Line 5: Indicate that we have not yet added any Benders rows to this iteration.
    3. Line 6-13: Add Benders rows by iterating over subproblems and adding Benders rows corresponding to subproblems, associated with violated cycle inequalities.
        - Line 7: Check if there exists a violated cycle inequality associated with $\mathcal{E}^-_s$. This is done by iterating over $(v_i, v_j) \in \mathcal{E}^-_s$ and checking if the shortest path from v_i to v_j is less than $x_{v_iv_j}$. This distance is defined on the graph's edges $\mathcal{E}$ with weights equal to $\mathbf{x}$.
        - Lines 8-10: Generate Benders rows associated with subproblem s and add them to nascent set $\hat{\mathcal{Z}}$.
        - Line 11: Indicate that a Benders row was added this iteration.
    4. Lines 14-16: If no Benders rows were added to this iteration, we enforce integrality on $\mathbf{x}$, when solving the master problem for the remainder of the algorithm.
- Line 18 Return solution $\mathbf{x}$.

C Generating Feasible Integer Solutions Prior to Convergence

Prior to the termination of optimization, it is valuable to provide feasible integer solutions on demand. This is so that a practitioner can terminate optimization, when the gap between the objectives of the integral solution and the relaxation is small. In this section we consider the production of feasible integer solutions, given the current solution $\mathbf{x}^*$ to the master problem, which may neither obey cycle inequalities or be integral. We refer to this procedure as rounding.

Rounding is a coordinate descent approach defined on the graph $\mathcal{G}$ and its edges $\mathcal{E}$ with weights κ, determined using $\mathbf{x}^*$ below.

$$\begin{aligned} \kappa_{v_iv_j} &= \phi_{v_iv_j}(1 - x^*_{v_iv_j}) \quad \forall (v_i, v_j) \in \mathcal{E}^+ \\ \kappa_{v_iv_j} &= \phi_{v_iv_j} x^*_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^- \end{aligned} \tag{12}$$

Consider that $\mathbf{x}^*$ is integral and feasible (where feasibility indicates that $\mathbf{x}^*$ satisfies all cycle inequalities). Let $\mathbf{x}^{s*}$ define the boundaries in partition $\mathbf{x}^*$, of the connected component containing s. Here $x^{s*}_{v_iv_j} = 1$ if exactly one of v_i, v_j is in the connected component containing s under cut $\mathbf{x}^*$. Observe, that $Q(\kappa, s, \mathbf{x}^{0s}) = 0$, where $x^{0s}_{v_iv_j} = \mathbb{1}_{\mathcal{E}^-_s}(v_i, v_j)$, is achieved using $\mathbf{x}^{s*}$ as the solution to Eq. (7). Thus $\mathbf{x}^{s*}$ is the minimizer of Eq. (7). The union of the edges cut in $\mathbf{x}^{s*}$ across $s \in \mathcal{S}$ is identical to $\mathbf{x}^*$. Note that when $\mathbf{x}^*$ is integral and feasible then the solution produced below has cost equal to that of $\mathbf{x}^*$.

$$\begin{aligned} \mathbf{x}^{s*} &\triangleq \text{ minimizer of } Q(\kappa, s, \mathbf{x}^{0s}) \quad \forall s \in \mathcal{S} \\ x^+_{v_iv_j} &\triangleq \max_{s \in \mathcal{S}} x^{s*}_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^+ \\ x^+_{v_iv_j} &\triangleq x^{s*}_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^-_s, s \in \mathcal{S} \end{aligned} \tag{13}$$

The procedure of Eq. (13) can be used regardless of whether $\mathbf{x}^*$ is integral or feasible. Note that if $\mathbf{x}^*$ is close to integral and close to feasible, then Eq. (13) is biased to produce a solution that is similar to $\mathbf{x}^*$ by design of κ.

We now consider a serial version of Eq. (13), which may provide improved results. We construct a partition $\mathbf{x}^+$ by iterating over $s \in \mathcal{S}$, producing component partitions as in Eq. (13). We alter κ by allowing for the cutting of edges previously cut with cost zero. We formally describe this serial rounding procedure below in Alg. 2.

Algorithm 2 Generating an Integral and Feasible Solution Given Infeasible and or Non-Integral Input $\mathbf{x}^*$)

```
1: $x^+_{v_iv_j} = 0 \quad \forall (v_i, v_j) \in \mathcal{E}$
2: $\kappa_{v_iv_j} = \phi_{v_iv_j} x^*_{v_iv_j} \quad \forall (v_i, v_j) \in \mathcal{E}^-$
3: $\kappa_{v_iv_j} = \phi_{v_iv_j}(1 - x^*_{v_iv_j}) \quad \forall (v_i, v_j) \in \mathcal{E}^+$
4: for $s \in \mathcal{S}$ do
5:     $\mathbf{x}^s =$ minimizer for $Q(\kappa, s, \mathbf{x}^{0s})$ given fixed $\kappa, s$.
6:     $x^+_{v_iv_j} = \max(x^+_{v_iv_j}, x^s_{v_iv_j}) \quad \forall (v_i, v_j) \in \mathcal{E}$
7:     $\kappa_{v_iv_j} = \kappa_{v_iv_j}(1 - x^+_{v_iv_j}) \quad \forall (v_i, v_j) \in \mathcal{E}$
8: end for
9: Return $\mathbf{x}^+$
```

- Line 1: Initialize $\mathbf{x}^+$ as the zero vector.
- Line 2-3: Set κ according to Eq. (12)
- Line 4-8: Iterate over $s \in \mathcal{S}$ to construct $\mathbf{x}^+$ by cutting edges cut in the subproblem.
    1. Line 5: Produce the lowest cost cut $\mathbf{x}^s$ given altered edge weights κ for subproblem s.
    2. Line 6: Cut edges in $\mathbf{x}^+$ that are cut in $\mathbf{x}^s$.
    3. Line 7: Set $\phi_{v_iv_j}$ to zero for cut edges in $\mathbf{x}^+$.
- Line 9: Return the solution $\mathbf{x}^+$

When solving for the fast minimizer of $Q(\kappa, s, \mathbf{x}^{0n})$, we rely on the network flow solver of Rother et al. [28], though we do not exploit its capacity to tackle non-submodular problems.