Most Balanced Minimum Cuts and Partially Ordered Knapsack

Paul Bonsma*
Institut für Mathematik, Sekr. MA 6-1, Technische Universität Berlin,
Straße des 17. Juni 136, 10623 Berlin, Germany
bonsma@math.tu-berlin.de

November 12, 2007

Abstract

We consider the problem of finding most balanced cuts among minimum *st*-edge cuts and minimum *st*-vertex cuts, for given vertices s and t, according to different balance criteria. For edge cuts $[S, \overline{S}]$ we seek to maximize $\min\{|S|, |\overline{S}|\}$. For vertex cuts C of G we consider the objectives of (i) maximizing $\min\{|S|, |T|\}$, where $\{S, T\}$ is a partition of $V(G)\backslash C$ with $s \in S$, $t \in T$ and $[S, T] = \emptyset$, (ii) minimizing the order of the largest component of $G - C$, and (iii) maximizing the order of the smallest component of $G - C$.

All of these problems are shown to be NP-hard. We give a PTAS for the edge cut variant and for (i). We give a 2-approximation for (ii), and show that no non-trivial approximation exists for (iii) unless P=NP.

To prove these results we show that we can partition the vertices of G, and define a partial order on the subsets of the partition, such that ideals of the partial order correspond bijectively to minimum *st*-cuts of G. This shows that the problems are closely related to Uniform Partially Ordered Knapsack (UPOK), a variant of POK where element utilities are equal to element weights. Our PTAS is also a PTAS for special types of UPOK instances.

1 Introduction

We study the problem of finding *most balanced cuts* among certain sets of edge cuts and vertex cuts, for various types of balance criteria. This problem differs from the balanced cut problems that are usually studied (see e.g. [12]): in most previous research the objective is to find a cut with minimum number of edges or vertices among all cuts that satisfy a certain balance requirement, for instance the requirement that none of the resulting components should contain more than two-thirds of all vertices (this is called a 2/3-*balanced cut*). Instead, we are looking for a cut that optimizes a balance function, for instance one that minimizes the number of vertices in the largest component, among a set of edge or vertex cuts that contain at most k edges resp. vertices. In particular, we are looking for cuts that are *minimum st-cuts* for some vertex pair s and t: these are cuts that separate s from t, with minimum number of edges resp. vertices among all such cuts.

We now define the problems more formally. For further definitions, see Section 2. We assume that the input graphs for the various problems are simple and connected. For a graph

*Supported by the Graduate School "Methods for Discrete Structures" in Berlin, DFG grant GRK 1408. Part of this research was carried out when the author was working at the University of Twente.

$G = (V, E)$ and two non-empty, disjoint sets $S \subset V$ and $T \subset V$, $[S, T]$ denotes the set of edges of G with one end vertex in S and one end vertex in T. A set $M \subseteq E$ is an *edge cut* if $M = [S, \overline{S}]$ for some non-empty $S \subset V$. The cut $[S, \overline{S}]$ is a *minimum st-edge cut* if $s \in S$ and $t \in \overline{S}$, and $|[S, \overline{S}]|$ is minimum among all such cuts. For edge cuts, most reasonable balance requirements are equivalent; we choose the objective of maximizing $\min\{|S|, |\overline{S}|\}$.

Most Balanced Minimum st-Edge Cut: (MBMEC)
INSTANCE: A graph G, two distinct vertices $s, t \in V(G)$.
SOLUTION: A minimum st-edge cut $[S, \overline{S}]$.
OBJECTIVE: Maximize $\min\{|S|, |\overline{S}|\}$.

For vertex cuts the problem is defined as follows. A vertex cut of a connected graph $G = (V, E)$ is a set $C \subset V$ such that $G - C$ is disconnected. It is a *minimum st-vertex cut* for $s, t \in V$ if s and t are in different components of $G - C$, and $|C|$ is minimum among all such cuts. Since a minimum vertex cut can result in multiple components, there are different ways in which vertex cuts can be considered well-balanced. The three most natural ways are expressed by the following three variants of Most Balanced Minimum st-Vertex Cut (MBMVC). The *order* of a graph is its number of vertices.

MBMVC - Largest Component (LC)
INSTANCE: A graph G, two vertices $s, t \in V(G)$.
SOLUTION: A minimum st-vertex cut C.
OBJECTIVE: Minimize the order of the largest component of $G - C$.

MBMVC - Smallest Component (SC)
INSTANCE: A graph G, two vertices $s, t \in V(G)$.
SOLUTION: A minimum st-vertex cut C.
OBJECTIVE: Maximize the order of the smallest component of $G - C$.

MBMVC - 2-Partition (2P)
INSTANCE: A graph G, two vertices $s, t \in V(G)$.
SOLUTION: A minimum st-vertex cut C and partition $\{S, T\}$ of $V(G) \backslash C$ with $s \in S$, $t \in T$, $[S, T] = \emptyset$.
OBJECTIVE: Maximize $\min\{|S|, |T|\}$.

In the first of these variants, a solution with multiple smaller components is preferred. This is the variant that is most useful when used in the context of divide-and-conquer algorithms. The second variant on the other hand prefers solutions with few components. For the third variant, the number of components is not important, it is only important how well the components can be divided into two parts. This variant will turn out to be most similar to MBMEC. One may consider other objective functions, such as for instance minimizing the ratio between the order of the largest component and the order of the smallest component, but for many such objectives the approximability status of the resulting problem is easily deduced from our results on these three problems.

Most balanced cut problems were previously studied by Feige and Mahdian [5], who studied MBMEC and a variant closely related to MBMVC-LC (their goal was to minimize the maximum of the order of the component that contains s, and the order of the component that contains t). They proved NP-hardness, and gave a fixed parameter tractable (FPT) algorithm where the parameter is the number of edges resp. vertices in the cut. They studied this problem since it occurred as a subproblem in their method for finding small α-balanced cuts. This is one motivation for studying most balanced minimum cut problems; these problems are similar to hard cut problems such as minimum α-balanced cut or sparsest cut (see [10] or [12]), and thus may be useful in finding methods for solving (special cases of) these problems, since we will show that they are much easier to approximate. (Recent results indicating that sparsest cut is hard to approximate appear in [1, 3].)

As a second motivation for this problem, Chimani, Gutwenger and Mutzel [2] give an integer program for calculating the crossing number of a graph G, and show that edge cuts $[S, \overline{S}]$ can be used in a preprocessing step to split the instance in two. This step is correct whenever $[S, \overline{S}]$ is a minimum st-cut for some pair s and t, and the gain is larger when the cut is more balanced.

We show that most balanced minimum cut problems are closely related to the NP-hard problem *partially ordered knapsack* (POK): Both for edge cuts and vertex cuts we define a partition of the vertices of a most balanced minimum cut instance, and a partial order on the sets of this partition, such that ideals of the resulting partial order correspond bijectively to minimum st-cuts. For vertex cuts this includes the partition of the resulting components into S and T as in MBMVC-2P. Partial orders are denoted as (P, A), where P is a set of elements and A is a partial order relation on P (transitive and irreflexive). A set $I \subseteq P$ is an *ideal* of (P, A) if for all $(x, y) \in A$ and $y \in I$, it holds that $x \in I$. If we have a function $w : P \to \mathbb{R}$ assigning weights to a set P, then for any $I \subseteq P$, $w(I)$ will denote $\sum_{x \in I} w(x)$.

Uniform Partially-Ordered Knapsack (UPOK)
INSTANCE: A partial order (P, A), weights $w : P \to \mathbb{R}^+$, and an integer W_U.
SOLUTION: An ideal I of (P, A) with $w(I) \leq W_U$.
OBJECTIVE: Maximize $w(I)$.

This is the uniform version of POK, which can also be seen as a partially-ordered version of subset sum. General POK instances have both a utility function on P which should be maximized, and a cost function on P which should be bounded by W_U in a solution. For MBMEC instances G, s, t, we will show how a partial order (P, A) can be constructed such that G has a minimum st-edge cut $[S, \overline{S}]$ with $s \in S$ and $|S| = x$ if and only if (P, A) has an ideal I with $w(I) = x$. A similar construction is given for MBMVC-2P instances, and in the other direction, we also show how to construct instances of all four problems for every UPOK instance. From these relations a new, simple NP-hardness proof follows for the four problems. In addition we present a polynomial time approximation scheme (PTAS) for the special case of UPOK where $w(P) \leq cW_U$ for some constant c. This yields a PTAS for MBMEC and MBMVC-2P. For MBMVC-LC we present a 2-approximation, and for MBMVC-SC we show that there is no polynomial time approximation algorithm that is better than the trivial $2/n$-approximation (unless P=NP), where $n = |V(G)|$. We also study a version of MBMEC where the choice of the vertices s and t is not part of the instance, but may be chosen as part of the solution:

General Most Balanced Minimum st-Edge Cut (GMBMEC):
INSTANCE: A graph G.

SOLUTION: A minimum st-cut $[S, \overline{S}]$ for some vertex pair $s, t \in V(G)$.
OBJECTIVE: Maximize $\min\{|S|, |\overline{S}|\}$.

This is the version of the problem that is most relevant for the application in [2]. Clearly the PTAS for MBMEC gives a PTAS for GMBMEC, by trying every combination of s and t. It is however not obvious that NP-hardness of GMBMEC follows from the NP-hardness of MBMEC; we give a proof of this statement in this paper.

Our results do not only contribute to the knowledge of most balanced minimum cut problems, but also to the knowledge of approximating special cases of POK. Little is known about the approximability of POK in general; positive results are known for many special cases of the problem, but not for the general case, and the current strongest negative result is that no FPTAS can exist (unless P=NP). See [9] for more information. Finally, the exhibited partial order structure of minimum st-cuts is interesting by itself, providing new insight into a fundamental concept from graph theory. The partial order structure of minimum st-edge cuts is relatively straightforward (in contrast to the structure for minimum st-vertex cuts), but to our knowledge not described in the literature.

The paper is structured as follows. We start by giving definitions, notations and useful theorems from the literature in section 2. Since the transformation from most balanced st-cut problems to POK is more straightforward in the case of edge cuts, we first state our results for edge cuts. In Section 3 we give this transformation. In Section 4 we give a PTAS for the above mentioned special case of POK, which in combination with the transformation from Section 3 gives a PTAS for MBMEC. In Section 4 we also mention two cases of underlying partial orders for which MBMEC can be solved in polynomial time. In Section 5 we prove NP-hardness of all problems mentioned above. In Section 6 we switch to vertex cuts, and construct a partial order corresponding to the minimum st-vertex cuts of a graph. Combined with the PTAS from Section 4, this gives the PTAS for MBMVC-2P, and identifies again two polynomial time solvable cases. In Section 7 we will look at the constructed partial order in more detail, and identify the elements that may cause more than two components to exist in a minimum st-vertex cut. This will yield the 2-approximation for MBMVC-LC. In Section 8 we prove that, unless P=NP, MBMVC-SC does not admit any non-trivial approximation algorithm. We end in Section 9 with a summary and open questions.

In sections where there is no confusion possible, we will simply talk about cuts instead of vertex cuts or edge cuts. An extended abstract of this paper appeared in the proceedings of CTW 2007.

2 Preliminaries

For graph theoretic definitions not treated here, see [4]. For definitions related to algorithms and complexity, see [7]. A polynomial time algorithm for a maximization (minimization) problem is called an *α-approximation algorithm* if for every instance the objective value of the returned solution is at least (at most) α times the objective value of an optimal solution to the problem. A *polynomial time approximation scheme (PTAS)* for a maximization (minimization) problem is a method for designing $(1-\epsilon)$-approximation algorithms ($(1+\epsilon)$-approximation algorithms) for every $\epsilon > 0$. For a given choice of parameter k, an algorithm is said to be *fixed parameter tractable (FPT)* if its time complexity is $g(n)f(k)$ for some polynomial g, where n is the input size of the problem, and $f(k)$ may be an arbitrary (computable) function.

We distinguish between $\subseteq$ and $\subset$, which will denote subset resp. proper subset. For a set S, we will use the notations $S + x$ and $S - x$ to denote $S \cup \{x\}$ resp. $S \setminus \{x\}$. A set S is *minimal* for a property ϕ if it satisfies ϕ, but contains no proper subset that satisfies ϕ. A set $S \subseteq X$ is *minimum* for ϕ if there is no set $T \subseteq X$ with $|T| < |S|$ that satisfies ϕ. This is defined similarly for *maximal* and *maximum*.

A *walk* between two vertices may contain edges and vertices multiple times. A walk is *closed* if it begins and ends in the same vertex. A *path* is a walk that contains no vertex twice. A path with end vertices u and v is also called a (u, v)*-path*. A *cycle* is a closed walk that contains no vertex twice other than the begin and end vertex.

We will assume that the graphs given as input to the problems are *simple*, but we will allow graph operations to introduce *loops* and *parallel edges*. We will consider directed graphs and *mixed graphs*, in which both undirected and directed edges are present. An undirected edge with end vertices u and v is denoted as uv (or vu). A directed edge from u to v is denoted as (u, v), and u called its *tail* and v its *head*. Directed edges are also called *arcs*. For an undirected or mixed graph G, $E(G)$ denotes the set of edges of G, and for a directed graph G, $A(G)$ denotes the set of arcs of G. A walk (path, cycle) in a mixed or directed graph is *directed* if all arcs in the walk occur in the same direction. An arc that is a loop is also considered a directed cycle.

A directed graph (V, A) is *transitive* if for all $(u, v) \in A$ and $(v, w) \in A$, $(u, w) \in A$ holds. Note that for a transitive graph (V, A), $(u, v) \in A$ and $(v, u) \in E$ imply that there are loops incident with u and v. A *partial order* is a transitive directed graph (V, A) without directed cycles. For a partial order (V, A), $v \in V$ is called a *minimum* (*maximum*) if it has in-degree (out-degree) zero. Vertices $u, v \in V$ are called *incomparable* if neither $(u, v) \in A$ nor $(v, u) \in A$. A subset $S \subseteq V$ is called an *anti-chain* if the elements of S are pairwise incomparable. The *width* of a partial order is the maximum size of an anti-chain, which can be determined in polynomial time [6, 11].

The next notion comes from partial order theory, but can be defined as well for arbitrary directed graphs: for a directed graph (V, A), $I \subseteq V$ is an *ideal* if $v \in I$ and $(u, v) \in A$ imply $u \in I$.

In the remainder we will need to use flows with a single source and sink, but only on undirected graphs with unit capacities on the edges, so we may define flows as follows. Let (V, E) be an undirected graph with a *source vertex* $s \in V$ and *sink vertex* $t \in V$. An *st-flow* is a function $f : V \times V \to \mathbb{R}^+$, such that

$$\begin{aligned} f(u,v) &\leq 1 && \text{if } uv \in E \\ f(u,v) &= 0 && \text{if } uv \notin E \\ f(u,v) \cdot f(v,u) &= 0 && \forall u, v \in V \times V, \end{aligned}$$

and in addition for every vertex $v \notin \{s, t\}$ the following holds:

$$\sum_{w \in V} (f(v,w) - f(w,v)) = 0.$$

These last constraints are called the flow conservation constraints. We will use the notation $f(uv) = \max\{f(u, v), f(v, u)\}$. The *value* $v(f)$ of an *st*-flow f is

$$v(f) = \sum_{w \in V} (f(s,w) - f(w,s)).$$

A flow f that maximizes $v(f)$ among all possible st-flows is called a *maximum flow*. For these types of flows, the well-known theorems by Ford and Fulkerson [6] may be stated as follows:

Theorem 1 *(MinCut-MaxFlow) Let f be a maximum st-flow, and $[S, \overline{S}]$ a minimum st-cut. Then $v(f) = |[S, \overline{S}]|$.*

Theorem 2 *A maximum st-flow can be found in polynomial time.*

Finally, the integrality theorem [6] gives the following statement for unit capacities.

Theorem 3 *Let $[S, \overline{S}]$ be a minimum st-cut with k edges. Then a set of k pairwise edge-disjoint (s, t)-paths exists.*

3 The partial order structure of minimum st-edge cuts

Throughout this section, G, s, t denotes a MBMEC instance, and when 'cuts' is written, edge cuts are meant. By k we denote the number of edges of a minimum st-cut of G. In this section we will give a polynomial transformation from G, s, t to a partial order (P, A) with weights w on the vertices such that G has a minimum st-cut $[S, \overline{S}]$ with $s \in S$ and $|S| = x$ if and only if (P, A) has an ideal I with $w(I) = x$. This reduces MBMEC to the following problem.

MOST BALANCED IDEAL (MBI):
INSTANCE: A partial order (P, A), weights $w : P \to \mathbb{N}$.
SOLUTION: An ideal I of (P, A).
OBJECTIVE: Maximize $\min\{w(I), w(P \backslash I)\}$.

The idea behind this transformation from G, s, t to (P, A) is as follows: first we identify the set of edges M that can be part of a minimum st-cut of G. The elements P of the partial order will be the components of $G - M$. We will then construct an arc set A such that ideals of (P, A) correspond bijectively to minimum st-cuts of G.

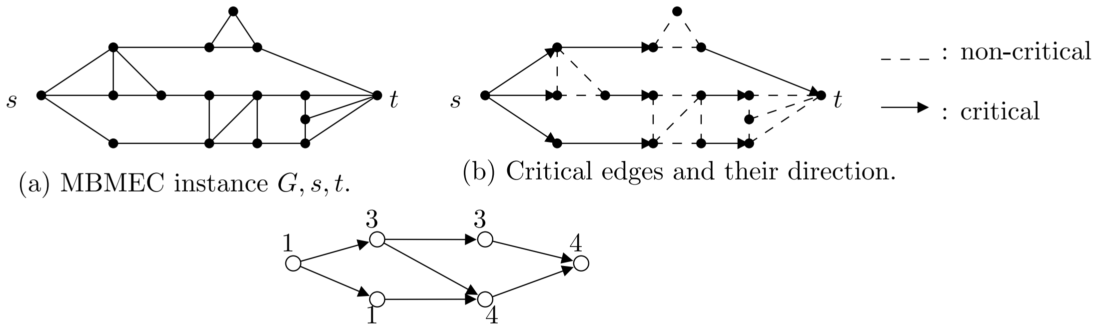
First we will consider this edge set M. A *critical edge* is an edge of G that is part of at least one minimum st-cut.

Claim 4 *The critical edges of G, s, t can be found in polynomial time.*

Proof: If edge e is part of minimum st-cut, then deleting e yields a graph having an st-cut with $k - 1$ edges, so a maximum st-flow has value $k - 1$ (Theorem 1). On the other hand, after deleting a non-critical edge the graph still admits an st-flow with value k. So at most $|E(G)| + 1$ flow problems need to be solved in order to find all critical edges, which can be done in polynomial time (Theorem 2). $\square$

Similarly it follows that critical edges can alternatively be characterized as follows.

Claim 5 *An edge uv is critical if and only if either for every maximum st-flow f, $f(u, v) = 1$ holds, or for every maximum st-flow f, $f(v, u) = 1$ holds.*



(a) MBMEC instance G, s, t.

(b) Critical edges and their direction.

(c) The partial order (P, A) (without transitive arcs).
Numbers indicate $|V(C)|$.

Figure 1: The partial order structure of an MBMEC instance.

We now give the construction of the MBI instance. This construction is illustrated in Figure 1. First direct the critical edges of G in the direction of a maximum flow, giving a mixed graph (Claim 5 shows this direction is well-defined). Let M be the set of critical edges of G, s, t. Let P be the set of components of $G - M$. We define a partial order $G' = (P, A)$ as follows, using the direction assigned to the critical edges. If C_1 and C_2 are components of $G - M$, and there is a critical edge (u, v) with $u \in V(C_1)$ and $v \in V(C_2)$, then add (C_1, C_2) to A. In addition, add all transitive edges. Assign weights $w(C) = |V(C)|$ for every $C \in P$.

We first show that the constructed graph is a partial order, and then that its ideals correspond bijectively to minimum st-cuts of G.

Lemma 6 *The graph $G' = (P, A)$ as defined above is a partial order.*

Proof: Obviously G' is transitive. Now assume G' contains a directed cycle (this may be a loop). Then G' contains a directed cycle C' that contains no transitive edges. Such a cycle corresponds to a cycle C in G that contains at least one critical edge, in which all critical edges are included in the same direction along C. We will associate this direction also with C.

For every non-critical edge e, there is a maximum st-flow f of G that has $f(e) < 1$ (Claim 5). For every non-critical edge e we may consider such a flow, and choose f to be the average of all of these flows. Note that this is again a maximum st-flow, with the additional property that there is some $\epsilon > 0$ such that for every non-critical edge e, $f(e) \leq 1 - \epsilon$.

Using f we construct a new flow f' by changing the flow along the cycle C as follows. For every critical edge $(u, v) \in E(C)$, we decrease the flow by ϵ, yielding $f'(u, v) = f(u, v) - \epsilon = 1 - \epsilon$ (Claim 5). For every non-critical edge $uv \in E(C)$ we decrease the flow by ϵ if the flow is in the direction of C (if this yields $f'(u, v) = x < 0$, then instead we set $f'(v, u) = -x$ and $f'(u, v) = 0$), and increase the flow by ϵ otherwise. For all other edges e, we set $f'(e) = f(e)$. The function f' again satisfies the flow conservation constraints, since all critical edges are in the same direction along C. Since all non-critical edges started out with a flow of at most $1 - \epsilon$, there is no e with $f'(e) > 1$. So f' is again an st-flow, with same value as f, and therefore a maximum st-flow.

But now there is at least one critical edge e with $f'(e) = 1 - \epsilon$, a contradiction with Claim 5. We conclude that G' contains no directed cycles, and thus is a partial order. $\square$

From Lemma 6 we obtain the following characterization of the minimum st-cuts of G, using the defined partial order $G' = (P, A)$.

Theorem 7 *Let (P, A) be the partial order corresponding to G, s, t, as constructed above. $[S, \overline{S}]$ with $s \in S$, $t \in \overline{S}$ is a minimum st-cut of G if and only if:*

- *For all $C \in P$: $V(C) \subseteq S$, or $V(C) \subseteq \overline{S}$, and*
- *if $(C_1, C_2) \in A$ and $V(C_2) \subseteq S$, then $V(C_1) \subseteq S$.*

Proof: Let $[S, \overline{S}]$ be a minimum st-cut of G. If there is a component C of $G - M$ such that $V(C)$ is neither a subset of S nor of $\overline{S}$, then $[S, \overline{S}]$ contains one of its edges, which is a non-critical edge, a contradiction. Now suppose $(C_1, C_2) \in A$, $V(C_2) \subseteq S$ and $V(C_1) \subseteq \overline{S}$. In G', a directed path from C_1 to C_2 exists that uses only non-transitive edges. On this path, there are adjacent vertices C_1' and C_2' such that $(C_1', C_2') \in A$, $V(C_2') \subseteq S$ and $V(C_1') \subseteq \overline{S}$. Since (C_1', C_2') is a non-transitive edge, this means that there is a critical edge (u, v) in G with $u \in \overline{S}$ and $v \in S$, a contradiction with the direction assigned to the critical edges and Claim 5.

Now we prove the other direction. Suppose $S \subseteq V(G)$ satisfies the two properties. We prove that $[S, \overline{S}]$ is a minimum st-cut with $s \in S$. Consider a non-critical edge uv. Then u and v are both part of $V(C)$ for some $C \in P$, and thus by the first property, $uv \notin [S, \overline{S}]$, so $[S, \overline{S}]$ contains only critical edges.

Consider a set of k edge-disjoint (s, t)-paths (which exists by Theorem 3), and a corresponding maximum st-flow f along these paths. Let $v_0, \ldots, v_l$ be the sequence of vertices along one of these paths ($s = v_0$, $t = v_l$), so $f(v_i, v_{i+1}) = 1$ for all i. Suppose an i exists such that $v_i \in \overline{S}$, $v_{i+1} \in S$. Since this edge is in the cut, it is critical. The direction assigned is (v_i, v_{i+1}). Let $v_i \in V(C_1)$ and $v_{i+1} \in V(C_2)$, for $C_1, C_2 \in P$. Then we have $V(C_2) \subseteq S$, $V(C_1) \subseteq \overline{S}$ and $(C_1, C_2) \in A$, a contradiction with the second property. Hence every one of the chosen (s, t)-paths starts with a sequence $v_0, \ldots, v_m$ of vertices in S, and then contains a sequence of vertices $v_{m+1}, \ldots, v_l$ that are all in $\overline{S}$. Thus it contains at most one edge in $[S, \overline{S}]$.

From this it follows that $[S, \overline{S}]$ contains only k edges: Consider $e \in [S, \overline{S}]$. Recall that $[S, \overline{S}]$ contains only critical edges. So $f(e) = 1$ (Claim 5), and thus e is part of the path set. Combining this with the fact that every path contains only one edge from $[S, \overline{S}]$, it follows that $[S, \overline{S}]$ is a minimum st-cut. □

Theorem 7 shows that minimum st-cuts of G correspond bijectively to those ideals of the partial order (P, A) that are non-empty and not equal to P. Using the assigned weights $w(C) = |V(C)|$ we conclude that the MBI instance $(P, A), w$ is equivalent to the MBMEC instance G, s, t:

Corollary 8 *Let (P, A) be the partial order corresponding to G, s, t, as constructed above. G has a minimum st-cut $[S, \overline{S}]$ with $s \in S$, $|S| = x$ if and only if (P, A) has a nonempty ideal $I \subset P$ with weight $w(I) = x$.*

Note that when we remove the unique minimum and maximum of (P, A), there is a bijective correspondence between minimum st-cuts of G and *all* ideals of the resulting partial order.

The following observation bounds the width of (P, A).

Claim 9 *Let l be the minimum number of edges in any edge cut of G, and let k be the number of edges in a minimum st-edge cut of G. The width of the constructed partial order (P, A) is at most $2k/l$.*

Proof: Let w be the width of (P, A) and let X be an anti-chain of (P, A) with $|X| = w$. Consider a minimal ideal I of (P, A) containing X, and let $[S, \overline{S}]$ be the corresponding minimum *st*-cut of G. Since X is an anti-chain, every element of X is a maximum of I. Then for every $C \in X$, any critical edge of G with tail in C is part of $[S, \overline{S}]$ (Theorem 7), so there are at most k such edges. Since $|X| = w$ this means that there is some $C \in X$ such that there are at most k/w critical edges with tail in $V(C)$. It can be seen that the number of critical edges of G with head in C is the same as the number of critical edges of G with tail in C, so $|[V(C), \overline{V(C)}]| \leq 2k/w$. Using the fact that the number of edges in this cut is at least l, we obtain $w \leq 2k/l$. □

4 Algorithms for finding most balanced ideals

In the last section we transformed MBMEC to MBI. MBI is closely related to UPOK, and is also strongly NP-hard (see Section 5). The main result in this section is a PTAS for MBI, which after some minor changes is also a PTAS for UPOK instances that satisfy $w(P) \leq cW_U$ for some constant c. We end this section by mentioning some partial order types for which MBI can be solved in (pseudo-) polynomial time. The PTAS for MBI is given in Algorithm 1.

Algorithm 1 A PTAS for MBI

INPUT: A weighted partial order $(P, A), w$. (The desired approximation guarantee is $(1 - \epsilon)$.)

Let L be the set of elements $x \in P$ with $w(x) > w(P)\epsilon$.
For every $L' \subseteq L$ **do**
  If an ideal I exists with $L' \subseteq I$ and $L \backslash L' \subseteq \overline{I}$ **then**
    Let I be a minimal ideal with $L' \subseteq I$.
    While $x \in (P \backslash I) \backslash L$ exists such that $I + x$ is an ideal **do**:
      $I := I + x$.
    endwhile
  endif
endfor
Return the best solution considered throughout the algorithm.

Theorem 10 *Algorithm 1 is an* $(1-\epsilon)$*-approximation algorithm for MBI with time complexity* $f(|P|)2^{1/\epsilon}$*, where* f *is a polynomial.*

Proof: It is easy to see that every step of the algorithm within the for-loop and outside of the for-loop has an implementation that is polynomial time in $|P|$. (Here we assume a computation model that is able to do e.g. additions in constant time, regardless of the size of the numbers, but in any case the time complexity is polynomial in the input size.) The number of sets L' considered is at most $2^{|L|} < 2^{1/\epsilon}$, so the total complexity of this algorithm is $f(|P|)2^{1/\epsilon}$ for some polynomial f. Thus for fixed ϵ, the algorithm runs in polynomial time.

Now we will prove that the approximation guarantee of the algorithm is $1 - \epsilon$. Define $W = w(P)$, $W_L = (1 - \epsilon)W/2$ and $W_U = (1 + \epsilon)W/2$, so $W_U - W_L = W\epsilon$. We will argue that the algorithm will find an optimal solution, or a solution between W_L and W_U. In the second case, the objective value of the returned solution is at least $(1-\epsilon)W/2$, while no solution with value higher than $W/2$ can exist, which proves the approximation ratio. Call elements in L *large*, and all other elements of P *small*.

Let I_O be an optimal ideal, and let L_O be the set of large elements in I_O. In one of the iterations of the for-loop, L_O will be considered. Let $I_{\min}$ be the (unique) minimal ideal of (P, A) that contains L_O, and let $I_{\max}$ be the maximal ideal that contains L_O, but does not contain any element from $L\backslash L_O$.

If $w(I_{\max}) \leq W_L$, then $I_{\max}$ is the ideal with the best objective value among all ideals containing exactly L_O as large elements, so $I_{\max} = I_O$. The ideal $I_{\max}$ is considered in the algorithm, since in the iteration where $L' = L_O$ is considered, the while loop ends with a maximal ideal that does not contain any element from $L\backslash L_O$, which by uniqueness is $I_{\max}$. So in this case, the algorithm finds the optimum solution. Similarly, if $w(I_{\min}) \geq W_U$, then $I_{\min}$ must be the optimum solution, which is considered in the algorithm. Finally, suppose that $w(I_{\min}) \leq W_U$ and $w(I_{\max}) \geq W_L$. In this case, the algorithm will consider a solution with value between W_L and W_U. This is because the while loop starts with $I_{\min}$, ends with $I_{\max}$, and in between these solutions only adds small elements, such that the weight is incremented with small steps, which are smaller than $W\epsilon = W_U - W_L$. This concludes the proof. $\square$

Thus we have a $(1-\epsilon)$-approximation algorithm for MBI for every $\epsilon > 0$. Combining this with the polynomial transformation from MBMEC to MBI of the previous section, a PTAS is found for MBMEC.

Corollary 11 *A PTAS exists for MBMEC.*

With minor changes Algorithm 1 is also a PTAS for special instances of UPOK: the set of large elements L needs to be defined as the elements x with $w(x) > W_U\epsilon$. In order for $|L|$ to be bounded by a constant for fixed ϵ, it is necessary that $w(P) \leq cW_U$ for some constant c. When this condition is fulfilled, the modified algorithm is a PTAS for UPOK. Note that in the last line, the notion of 'best solution' should be slightly different in the case of UPOK, and that no ideals I have to be considered with $w(I) > W_U$, but these changes are not necessary for the algorithm to be a correct PTAS.

Theorem 12 *A PTAS exists for UPOK instances $(P, A), w, W_U$ with $w(P) \leq cW_U$ for some constant c.*

We now consider types of partial orders for which MBI can be solved in (pseudo-) polynomial time. Considering the transformation from MBMEC to MBI, it follows that also pseudopolynomial time algorithms for special cases of MBI yield polynomial time algorithms for the corresponding MBMEC instances.

In [9], a pseudopolynomial time algorithm for POK is given for the case when the partial order is 2-dimensional. A partial order is *2-dimensional* if it is the intersection of two linear orders. Such partial orders can be recognized in polynomial time. The algorithm from [9] is based on dynamic programming. This also gives a pseudopolynomial time algorithm for MBI on such instances, and a polynomial time algorithm for MBMEC for corresponding instances.

We now give a simple polynomial time algorithm to solve POK and MBI when the partial order has bounded width.

Claim 13 *For partial orders (P, A) with width at most w, POK and MBI can be solved in time $O(|P|^{w+O(1)})$.*

Proof: The algorithm is as follows. When the width of (P, A) is w, then consider all subsets $S \subseteq P$ with $|S| \leq w$. For every such set, in polynomial time one can construct a minimal ideal I that contains S. Return the ideal with the best objective value, among all ideals that are considered. The number of sets S considered is $O(|P|^w)$, hence for bounded w this is a polynomial time algorithm.

This algorithm finds an optimal solution: Let I_O be the ideal of (P, A) that maximizes the objective value. The maxima of I_O are pairwise incomparable, hence I_O contains at most w maxima. When choosing S to be the set of these maxima, then the minimal ideal that contains S is exactly I_O, hence I_O is considered in the algorithm. □

Together with Claim 9, it follows that if the ratio between the number of edges in a minimum st-cut and the minimum number of edges over all cuts of G is bounded by a constant, MBMEC can be solved in polynomial time. In particular, if minimum st-cuts are also minimum edge cuts of G, the problem can be solved in polynomial time. This statement is stronger than the statement in [5] that MBMEC can be solved in polynomial time whenever the number of edges in a minimum st-cut is bounded. However we do not know if there is also an FPT algorithm for POK or MBI when the width is chosen as the parameter, that is, we do not know if w can be removed from the exponent of the time complexity, even if we allow an otherwise pseudopolynomial time algorithm.

5 NP-hardness proofs for most balanced cut problems

In this section we prove the NP-completeness of the decision variant of MBMEC, which has an additional parameter l and asks whether the instance G, s, t has a minimum st-edge cut $[S, \overline{S}]$ with $\min\{|S|, |\overline{S}|\} \geq l$. The transformation is a straightforward transformation from the decision variant of UPOK, which is nearly the reverse of the transformation in Section 3. In this section 'cuts' will mean edge cuts.

Theorem 14 *The decision version of MBMEC is NP-complete.*

Proof: An instance of the decision version of UPOK consists of a partial order (P, A) with weights w, and in addition to the upper bound W_U, a lower bound W_L. We may take the weights to be non-zero natural numbers. The question is whether there is an ideal I with $W_L \leq w(I) \leq W_U$. This problem is known to be *strongly* NP-complete [8], that is, even if the weights are encoded in unary and therefore the instance size is $\Omega(w(P))$, the problem is NP-complete. As a first step, we scale all weights w and the bounds W_L and W_U with a factor $|A| + 2$, so for all $u \in P$ we may now assume $w(u) \geq |A| + 2$.

We transform this instance to a MBMEC instance G, s, t, l. Choose weights w_s and w_t such that $w_s - w_t = w(P) - W_L - W_U$ and $\min\{w_s, w_t\} = |A| + 2$. Introduce a complete graph C_s on w_s vertices, and a complete graph C_t on w_t vertices. In addition, for every $v \in P$ introduce a complete graph C_v on $w(v)$ vertices. Note that for this step of the transformation to be polynomial, we need that the weights are encoded in unary. These complete graphs will be called the *blocks* of G. Label one of the vertices of C_s as s, and one of the vertices of C_t as t.

For every arc $(u, v) \in A$ we introduce an undirected edge in G between arbitrary vertices in the blocks C_u and C_v. Since every such block has at least $|A| + 2$ vertices, this can be

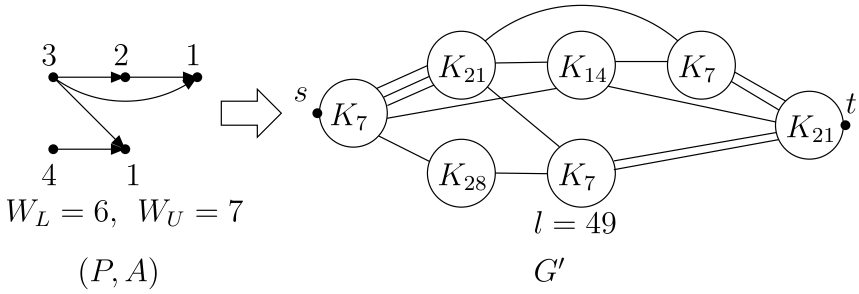


Figure 2: The transformation from UPOK to MBMEC.

done without introducing parallel edges. In addition, for every $u \in P$, k edges are introduced between C_s and C_u, where k is the number of arcs in A that have u as tail. l edges are introduced between C_u and C_t where l is the number of arcs in A that have u as head. This completes the construction.

Note that G has an st-cut with $|A|$ edges; consider for instance the cut that separates C_u from the rest of the graph. We can now also construct a set $\mathcal{P}$ of $|A|$ edge disjoint (s,t)-paths in G, such that every path in $\mathcal{P}$ contains exactly three edges between different blocks, and every such edge is included in a path of $\mathcal{P}$. It follows that minimum st-cuts in G contain exactly $|A|$ edges.

Furthermore the following set of cuts $[S, \overline{S}]$ shows that every edge e of G with end vertices in different blocks C_u and C_v is part of some minimum st-cut. Let I be a minimal ideal I of (P, A) with $u \in I$. Since (P, A) is a partial order and I is minimal, $v \notin I$. Let $S = (\cup_{u \in I} V(C_u)) \cup V(C_s)$. this cut contains exactly one edge of every path in $\mathcal{P}$, and contains only edges that are part of the paths in $\mathcal{P}$. It follows that every edge between blocks is critical. All other edges are easily seen not to be critical, because every block has at least $|A| + 2$ vertices.

Now if we construct the partial order corresponding to G, s, t as is done in Section 3, we obtain exactly the weighted partial order $(P, A), w$, extended with two elements corresponding to C_s and C_t. Then Corollary 8 shows that G has a minimum st-cut $[S, \overline{S}]$ with $|S| = w_s + x$ if and only if (P, A) has an ideal $I \subseteq P$ with $w(I) = x$.

Consider an ideal I of (P, A) and corresponding st-cut $[S, \overline{S}]$ of G with $s \in S$. Using the fact that the total number of vertices of G is $w(P) + w_s + w_t = 2w_s + W_L + W_U$, we see that

$$W_L \leq w(I) \leq W_U \Leftrightarrow W_L + w_s \leq |S| \leq W_U + w_s \Leftrightarrow$$

$$|S| \geq W_L + w_s \ \wedge \ |\overline{S}| \geq 2w_s + W_L + W_U - (W_U + w_s) \Leftrightarrow$$

$$|S| \geq W_L + w_s \ \wedge \ |\overline{S}| \geq w_s + W_L.$$

This shows that if we choose $l = w_s + W_L$, the instances are equivalent. This transformation is polynomial in $w(P)$. Since we assumed that the UPOK instance was encoded in unary, and had instance size $\Omega(w(P))$, the transformation is therefore polynomial. □

For the three given variants of MBMVC, a construction similar to the one in Theorem 14 proves NP-completeness; only minor changes are needed in the construction to ensure that there is a bijective correspondence between the ideals of the UPOK instance and the minimum st-vertex cuts of the constructed graph. In addition, the construction can be made such that

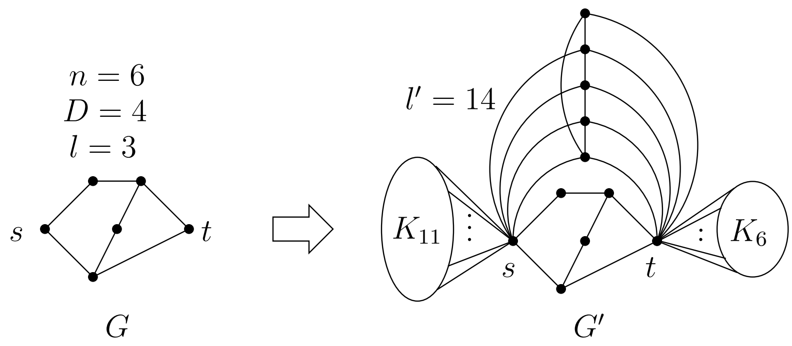


Figure 3: The transformation from MBMEC to GMBMEC.

every minimum st-vertex cut will only result in two components, so the chosen objective value does not matter.

Theorem 15 *The decision versions of MBMVC-2P, MBMVC-SC and MBMVC-LC are NP-complete.*

We now consider the problem GMBMEC. The NP-hardness of MBMEC allows us to easily prove the NP-hardness of GMBMEC.

Theorem 16 *The decision version of GMBMEC is NP-complete.*

Proof: Let G, s, t, l be an instance for the decision variant of MBMEC, where a minimum st-cut of G contains $k \geq 1$ edges. Let $D = \max\{4, \Delta(G)\}$, and let $n = |V(G)|$. This instance will be transformed into an instance G', l' of the decision version of GMBMEC, which asks whether there is a solution with objective value at least l'. The construction is illustrated in Figure 3. Note that the instance G, s, t, l in this figure is a NO-instance for MBMEC, but G, l is a YES-instance for GMBMEC, which can be seen by choosing s and t differently, namely as the two vertices of degree three.

The construction is as follows. Start with G. Introduce two large complete graphs K_s and K_t on $n + D + 1$ and n vertices respectively, and join all vertices of K_s to s, and all vertices of K_t to t. In addition introduce a cycle C on $D + 1$ vertices. Join all vertices of C to t, and join all vertices of C except for one with s. This completes the construction of G'.

It can be checked that a minimum st-cut in G' contains $k+D$ edges, and that any minimum st-cut $[S, \overline{S}]$ with $s \in S$ has $V(C) \subset \overline{S}$, $V(K_s) \subset S$ and $V(K_t) \subset \overline{S}$. So such a cut has at least $n + D + 2$ vertices on both sides.

Note also that any minimal cut that does not separate s and t has all of its edges incident with vertices of the same component of $G' - s - t$, so the smallest side of such a cut contains at most $n + D + 1$ vertices. It follows that a cut that is an optimal GMBMEC solution for G' separates s from t. Now let $[S, \overline{S}]$ be such an optimal GMBMEC solution for G', which is a minimum xy-cut.

If $x \in V(G) - s - t$ or $x \in V(C)$ then $d(x) \leq D$, so a minimum xy-cut contains at most D edges and therefore does not separate s from t, a contradiction. The same holds for y. If x and y are both part of $V(K_s) + s$ or both part of $V(K_t) + t$, then the cut also does not separate s from t. We conclude w.l.o.g. that $x \in V(K_s) + s$ and $y \in V(K_t) + t$. Together with the fact that $[S, \overline{S}]$ separates s from t, it follows that $[S, \overline{S}]$ is a minimum st-cut.

We have proved that every cut that is an optimal GMBMEC solution for G' is a minimum st-cut. From this it follows that G' has minimum xy-cut for some x and y, with at least $l' = l + n + D + 1$ vertices on both sides, if and only if G has a minimum st-cut with at least l vertices on both sides. This completes the NP-completeness proof. □

6 The partial order structure of minimum st-vertex cuts

From now on we will consider vertex cuts, and 'cut' will be used for vertex cuts. For every MBMVC variation the instance consists of a graph G with designated vertices s and t. Let k be the number of vertices in a minimum st-cut of G. In this section we will construct a partial order such that there is a bijection between ideals and minimum st-cuts of G, similar to Theorem 7. This construction can be combined with the PTAS from Section 4 to yield a PTAS for MBMVC-2P. To be precise, we will construct a partial order (P, A) where elements in P correspond to a subset of the vertices of G, such that together these subsets partition $V(G)$. An ideal I of (P, A) will then correspond to a vertex set $(S \cup C) \subset V(G)$ with $s \in S$, such that C is a minimum st-vertex cut, and $[S, \overline{S \cup C}] = \emptyset$. The definitions and constructions in this section are illustrated in Figure 4.

Definition 17 *Let G be a graph with $s, t \in V(G)$. A vertex is a* critical vertex *of G, s, t if it is part of some minimal st-vertex cut.*

Definition 18 *Consider a graph G with $s, t \in V(G)$, and a corresponding set of critical vertices.*

Consider the following relation on edges $e, f \in E(G)$: $e \sim f$ if and only if there is a (possibly closed) walk in G containing e and f that does not have critical vertices among its internal vertices.

The blocks *of G, s, t are the equivalence classes of '$\sim$'.*

Note that the relation '$\sim$' defined above is indeed an equivalence relation, and that all edges incident with a non-critical vertex v are part of the same block B.

Let p be the number of critical vertices of G, s, t, and label the critical vertices with $c_1, \dots, c_p$. We fix a set of k internally vertex disjoint (s,t)-paths $Q_1, \dots, Q_k$. If two distinct critical vertices c_i and c_j lie on the same path Q_i, with c_i closer to s, measured along Q_i, then c_i will be called a *path predecessor* of c_j, and c_j a *path successor* of c_i.

p $c_1, \dots, c_p$ $Q_1, \dots, Q_k$

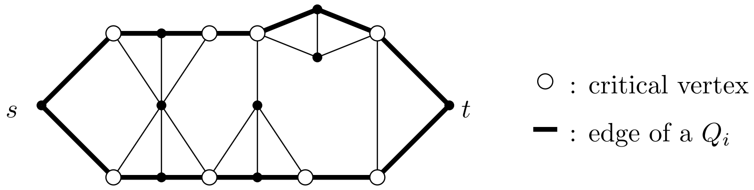
Let q be the number of blocks of the instance G, s, t. It will be convenient to label the blocks with the numbers $p+1, \dots, p+q$. For $i \in \{p+1, \dots, p+q\}$, Let E_i denote the edge set that constitutes of block i, so $\{E_{p+1}, \dots, E_{p+q}\}$ is a partition of $E(G)$. Let V_i be the set of non-critical vertices of G that are incident with (at least one edge of) E_i. We may assume that $s \in V_{p+1}$ and $t \in V_{p+q}$. In addition, for $i = 1, \dots, p$, define $V_i = \{c_i\}$. So $\{V_1, \dots, V_{p+q}\}$ is a partition of $V(G)$.

q E_i V_i

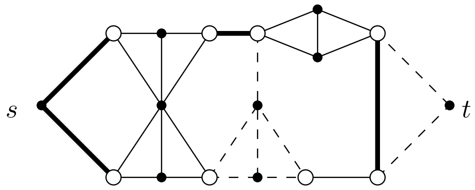
Determining p and q and constructing the above paths, edge sets and vertex sets can all be done in polynomial time: finding the critical vertices can be done using a standard transformation where all vertices are expanded to edges [6], and then using the approach from Claim 4.

Next we will show how to construct a directed graph G' which has vertex set $V(G') = \{1, \dots, p+q\}$, and a transitive arc set. This graph G' will not yet be a partial order since it

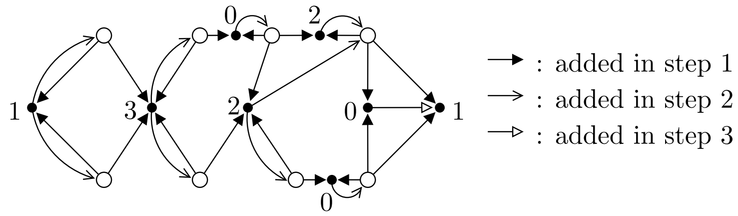
G'



(a) The critical vertices of G, and the two paths Q_1 and Q_2.



(b) A partition of the edges into eight blocks.



(c) The resulting graph G', most transitive arcs are omitted. Numbers indicate $|V_i|$.

Figure 4: The construction of G'.

contains directed cycles, but contracting all of these will give the desired partial order. The arcs of G' are constructed using the following four *arc addition steps*:

1. For all critical vertices c_i incident with a block E_j, add an arc (i, j).

2. For all critical vertices c_i and blocks E_j: if E_j is incident with s or with a path predecessor of c_i, then add an arc (j, i).

3. For all $j \in \{p+1, \ldots, p+q-1\}$, add an arc $(j, p+q)$.

4. Add all transitive arcs.

Figure 4 shows an example of the construction of G' for an instance G, s, t. Most of the transitive arcs are omitted in the drawing of G', for clarity. This example also shows that G' will have directed cycles. Therefore, in the next claim we will need to use the fact that ideals are defined for general directed graphs.

Claim 19 *Let $I \subseteq V(G')$ be an ideal of G'. If $I \neq \emptyset$, then $p+1 \in I$. If $I \neq V(G')$, then $p+q \notin I$.*

Proof: Every block is incident with at least one critical vertex. So if I contains a block, it contains a critical vertex by the first arc addition step, since I is an ideal. So every non-empty ideal I contains some critical vertex c_i. By the second arc addition step it then follows that $p+1 \in I$ since E_{p+1} is incident with s.

For proving the second statement, assume that $p+q \in I$. By the third arc addition step, for every $j \in \{p+1,\dots,p+q-1\}$, $j \in I$. Then by the first arc addition step, $i \in I$ for every $i \in \{1,\dots,p\}$. Hence $I = V(G')$. □

Theorem 20 *Let G, s, t be a MBMVC instance and let G' be the corresponding graph as constructed above. Let X be subset of $V(G)$ with $s \in X$, $t \notin X$. The graph G' has an ideal $I \subset V(G')$ with $\cup_{i\in I} V_i = X$ if and only if X can be partitioned into $\{C, S\}$ such that C is a minimum st-cut, and $[S, \overline{X}] = \emptyset$.*

Proof: Let $I \subseteq V(G')$ be an ideal of G' with $\cup_{i\in I} V_i = X$, so I is non-empty and not equal to $V(G')$. Let $V(I)$ denote $\cup_{i\in I} V_i$, and let $T = V(G)\backslash V(I)$. We will first show that $V(I) = S \cup C$, for some minimum st-cut C, and some set S such that $[S, T] = \emptyset$. $V(I)$

Let $C \subseteq V(I)$ be the set of critical vertices in $V(I)$ that do not have a path successor in $V(I)$, and let $S = V(I)\backslash C$.

First we show that $s \in S$ and $t \in T$: I is non-empty, so $p+1 \in I$ (Claim 19). So $s \in V_{p+1} \subseteq V(I) = S \cup C$, and since s is not a critical vertex, $s \in S$. Since $I \neq V(G')$, $p+q \notin I$ (Claim 19). Since $t \in V_{p+q}$, we have $t \in T$.

Next we show that $|C| = k$. We know that $p+1 \in I$. Block E_{p+1} is incident with critical vertices from every path from the path set $Q_1, \dots, Q_k$, so by the first arc addition step, at least one critical vertex from every path Q_i is in $V(I)$. By the definition of C, at most one critical vertex from every path is in C, so $|C| = k$.

Finally we prove that $[S, T] = \emptyset$. Suppose there is an edge $uv \in E(G)$ with $u \in S$ and $v \in T$. First consider the case that u and v both are critical vertices, so $v = c_x$ for some x. Then $E_j = \{uv\}$ for some j. Since $u \in S$, we know that a path successor c_i of u is in $V(I)$, so $(j, i) \in E(G')$ by arc addition step 2. Also, $(x, j) \in E(G')$ by step 1. But $x \notin I$ and $i \in I$, which contradicts that I is an ideal. Next, assume that u is a non-critical vertex, so $u \in V_j$ for some $j \in \{p+1,\dots,p+q\}$, and that $v = c_i$ is a critical vertex. Vertex v is incident with E_j, so by step 1, $(i, j) \in E(G')$. This is again a contradiction with I being an ideal. Now let $u = c_i$ be a critical vertex and $v \in V_j$ for some $j \in \{p+1,\dots,p+q\}$. Since $u \notin C$, some path successor c_x of u is in $V(I)$, and $(j, x) \in E(G')$ by step 2, again a contradiction. Finally, if u and v are both non-critical and therefore in the vertex set of a block, then they must be part of the same block. So they are both in $V(I)$ or both in T, a contradiction.

This proves that the constructed C is a minimum st-cut separating S from T.

To prove the other direction, let C be a minimum st-cut, and let $\{S, T\}$ be a partition of the remaining vertices so that $[S, T] = \emptyset$, $s \in S$ and $t \in T$. We will show that G' has an ideal I with $V(I) = C \cup S$.

The set I is constructed according to the following two rules.

1. For $x \in V(G')$ with $V_x \neq \emptyset$: add x to I if and only if $V_x \subseteq C \cup S$.
2. Now consider $x \in V(G')$ with $V_x = \emptyset$. Then $E_x = \{c_i c_j\}$ for a pair of critical vertices c_i and c_j. In this case add x to I if and only if $c_i \in S$ or $c_j \in S$.

First we show that $V(I) = S \cup C$, by showing that for every $j \in \{p+1,\dots,p+q\}$, either $V_j \subseteq S$ or $V_j \subseteq T$. The set V_j contains no critical vertices, so every vertex $v \in V_j$ is either in S or in T. For every pair of vertices $u, v \in V_j$, a (u, v)-walk exists in G that only uses vertices of V_j, by the definition of blocks. So if $u \in S$ and $v \in T$, then $[S, T] \neq \emptyset$, a contradiction.

Next we show that I is an ideal. We consider the arcs added in the first three arc addition steps. If these do not contradict I being an ideal, then the same holds for the transitive arcs added in the fourth step.

Consider an arc (j, i) that was added in the second arc addition step, with $i \in I$. We prove that $j \in I$. Block E_j is incident with a path predecessor c_x of c_i. Since $i \in I$, we know that $c_i \in C \cup S$, and thus it follows that $c_x \in S$. So if $V_j \neq \emptyset$, then $V_j \subseteq S$, and thus $j \in I$ by the first rule above. If $V_j = \emptyset$, then $j \in I$ by the second rule.

Now consider an arc (i, j) added in the first arc addition step, and suppose $j \in I$. We prove that $i \in I$. Block E_j is incident with c_i. If $V_j \neq \emptyset$, then the first rule above shows that $V_j \subseteq S$. So $c_i \in S \cup C$, and thus $i \in I$. If $V_j = \emptyset$ then the second rule shows that the unique edge in E_j is incident with a vertex $u \in S$. Either $u = c_i$ holds, or u is adjacent to c_i. In both cases, $c_i \in S \cup C$, and thus $i \in I$.

Since $t \in T$ and $t \in V_{p+q}$, we have $p + q \notin I$, so the arcs added in the third arc addition step also do not prevent I from being an ideal. This concludes the proof that I is an ideal.□

From Theorem 20 we obtain the following corollary.

Corollary 21 *Let G, s, t be a MBMVC-2P instance and let G' be the corresponding graph as constructed above. Let k be the number of vertices in a minimum st-vertex cut of G. A MBMVC-2P solution C, S, T exists for G with $\min\{|S|, |T|\} \geq l$ if and only if G' has an ideal I with $l + k \leq |V(I)| \leq |V(G)| - l$.*

This corollary is very similar to Corollary 8, and it seems we can then use the PTAS from Section 4 to solve the resulting problem on G'. However, G' contains directed cycles, and thus is not a partial order. Fortunately this is easily solved by contracting all strong components C of G' into a single vertex x. This process is illustrated in Figure 5. For x we then assign V_x to be the union of all sets V_i with $i \in V(C)$. Numbers in Figure 5(b) indicate $|V_x|$. Some arcs become transitive, which are drawn as dashed arcs. Doing this for all strong components yields a partial order (P, A). Since for any ideal I, the vertices of a strong component are (P, A)
either all in I, or all not in I, (P, A) and G' have the same ideal structure:

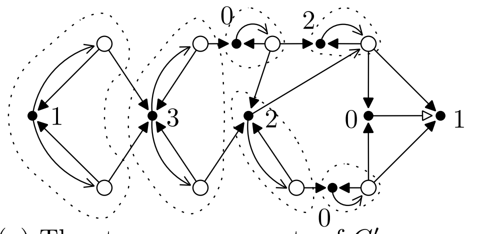
Claim 22 *Let G' and (P, A) be as constructed above. G' has an ideal I with $|V(I)| = x$ if and only if (P, A) has an ideal I' with $|V(I')| = x$.*

Assign weights w to vertices $i \in P$ as follows: if $t \notin V_i$ then $w(i) = |V_i|$, and if $t \in V_i$ then $w(i) = |V_i| + k$. This yields a MBI instance $(P, A), w$ that has an ideal $I \subset P$ with $\min\{w(I), w(P \backslash I)\} \geq k + l$ if and only if G, s, t has a MBMVC-2P solution C, S, T with $\min\{|S|, |T|\} \geq l$. Now the PTAS from Section 4 can be used for the MBI instance. Note that every construction in this section can be done in polynomial time, so:

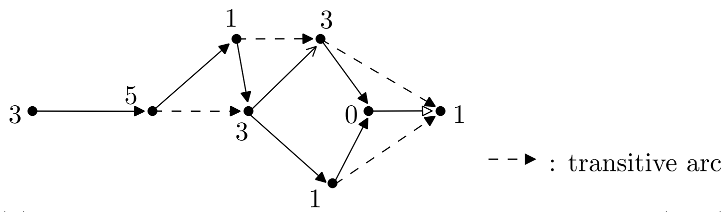
Corollary 23 *A PTAS exists for MBMVC-2P.*

7 Minimum vertex cuts with more than two components

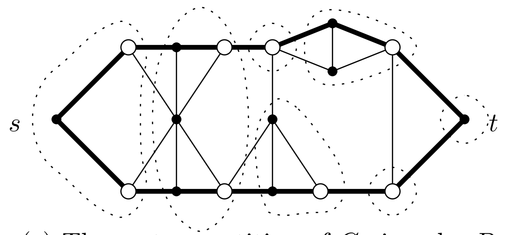
In the last section we identified the partial order structure of minimum *st*-vertex cuts, and mapped all cuts plus partitions of the remaining vertices into S and T to ideals of the partial order. In this section we will study in which situations a minimum *st*-vertex cut may result in more than two components, hence the situations in which this partition into S and T is not unique. For the MBMVC instance G, s, t we will again use the definitions and notations introduced in Section 6, such as the paths Q_i, blocks E_j, graph G' etc.



(a) The strong components of G'.



(b) Contracting strong components gives a partial order (P, A).



(c) The vertex partition of G given by P.

Figure 5: Contracting strong components of G' gives the partial order (P, A).

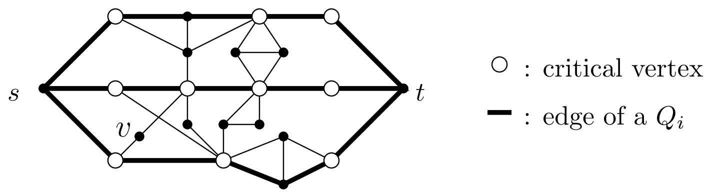
Definition 24 *Let C be an st-cut of G. The components of $G - C$ that contain s and t respectively are called the* s-component *and the* t-component. *All other components are called* extra components *of the cut.*

Definition 25 *A block E_j of G, s, t is called a* separable block *if*

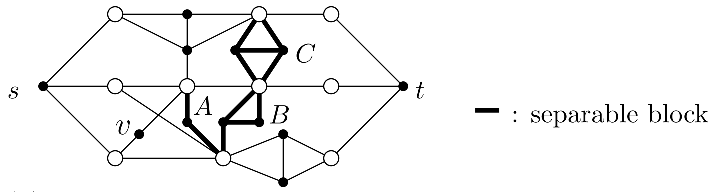
1. *$s, t \notin V_j$, and*
2. *E_j is not incident with a pair of critical vertices u and v such that u is a path predecessor of v, and*
3. *j is not part of a directed cycle of G'.*

These definitions are illustrated in Figure 6. This instance has three separable blocks, A, B and C. In the corresponding partial order, B and C form an antichain, but A is not part of an antichain together with another separable block. Note that there is for instance a minimum st-cut that has extra components corresponding to B and C. Note also that there is no minimum st-cut having the vertex v in an extra component. This explains the directed cycle condition in the definition of separable blocks.

Lemma 26 *Let C be a minimum st-cut of G. If H is an extra component of C, then $V(H) = V_j$ for some separable block E_j of G, s, t.*



(a) The critical vertices of G, and the three (s,t)-paths.



(b) The three separable blocks of G, s, t.

Figure 6: An MBMVC instance with three separable blocks.

Proof: Suppose $V(H)$ contains a critical vertex v. Vertex v lies on an (s,t)-path Q_i. Since C is a minimum st-cut, it contains exactly one vertex of Q_i. It follows that in $G - C$, a path exists from v to s or from v to t, a contradiction with the fact that H is an extra component. So $V(H)$ contains no critical vertices. Then $V(H)$ cannot contain vertices from different blocks since H is connected, and we have $V(H) \subseteq V_j$ for some block E_j. The cut C contains only critical vertices so $V(H) = V_j$.

Now we prove that E_j is a separable block. Clearly $s, t \notin V_j$. Consider the second property of separable blocks. If E_j is incident with at least two vertices u and v from the same (s,t)-path Q_i, then one of them is not in C, and thus is part of the component H, a contradiction with $V(H) = V_j$.

Finally we prove that the third property of separable blocks holds for E_j. Let H_s be the s-component of C. Choosing $S = V(H_s)$ and $T = (V(G) \backslash C) \backslash S$ gives $[S,T] = \emptyset$, $s \in S$ and $t \in T$, so G' has an ideal I with $V(I) = S \cup C$ (Theorem 20). But similarly, choosing $S' = V(H_s) \cup V_j$ and $T' = V(G) \backslash C \backslash S'$ satisfies the conditions of Theorem 20, and thus corresponds to an ideal I' of G'. We have $I' \backslash I = \{j\}$, so j is not part of a directed cycle of G'.

It follows that E_j is a separable block. □

Lemma 27 *Let $E_{\sigma_1}, \ldots, E_{\sigma_l}$ be a number of separable blocks of G, s, t. There is a minimum st-cut C with $G[V_{\sigma_1}], \ldots, G[V_{\sigma_l}]$ as extra components if and only if every pair σ_i and σ_j is incomparable in G'.*

Proof: Let C be a minimum st-cut where $E_{\sigma_1}, \ldots, E_{\sigma_l}$ correspond to the extra components of C. Consider an arbitrary pair E_{σ_i} and E_{σ_j}, and let H_s be the s-component of $G - C$. There are no edges from $V(H_s) \cup V_{\sigma_i}$ to the other components of $G - C$, so by Theorem 20, $V(H_s) \cup V_{\sigma_i} \cup C = V(I)$ for some ideal I of G', and therefore $(\sigma_j, \sigma_i) \notin A(G')$. Similarly, $(\sigma_i, \sigma_j) \notin A(G')$ follows, so σ_i and σ_j are incomparable.

To prove the other direction, suppose $\sigma_1, \ldots, \sigma_l$ are all pairwise incomparable in G', and let $I \subseteq V(G')$ be the minimum ideal containing all of these elements. By Theorem 20, I corresponds to a minimum st-cut C of G, and vertex set $S \subset V(G)$ with $V_{\sigma_i} \subseteq S$ for all $i \in \{1, \ldots, l\}$. Since $\sigma_1, \ldots, \sigma_l$ are pairwise incomparable, and they are not part of a directed cycle of G', all of these elements are maximal elements of I. So if $(\sigma_i, j) \in A(G')$ for a critical vertex c_j, then $c_j \notin I$. We have $(\sigma_i, j) \in A(G')$ when E_{σ_i} is incident with a path predecessor of c_j. It follows that, for any critical vertex u incident with E_{σ_i}, all path successors of u are not in $S \cup C$, but u is. Hence all critical vertices incident with E_{σ_i} are in C, and $G[V_{\sigma_i}]$ indeed is an extra component of C. □

Algorithm 2 gives an overview of the 2-approximation algorithm for MBMVC-LC. In the proof of Theorem 28 we give more details on the implementation of the steps of the algorithm.

Algorithm 2 A 2-approximation for MBMVC-LC

INPUT: An MBMVC instance G, s, t.

1. Identify the blocks and separable blocks of G, s, t, and construct graph G' as in Section 6.
2. Let $\mathcal{B} \subset V(G')$ be the set of indices of all separable blocks of G, s, t. Let $P = G'[\mathcal{B}]$.
3. For every $j \in \mathcal{B}$ of G, s, t, assign a weight $w(j) = |V_j|$.
4. Find a maximum weight anti-chain Q of P, with respect to the weights w.
5. Output a minimum st-cut C that has $G[V_i]$ as extra component for all $i \in Q$.

Theorem 28 *Algorithm 2 is a 2-approximation algorithm for MBMVC-LC.*

Proof: We first show that the algorithm is well-defined. Separable blocks are by definition not part of directed cycles of G', so the subgraph of G' induced by $\mathcal{B}$ contains no directed cycles. However transitivity is maintained, so $P = G'[\mathcal{B}]$ is a partial order. That shows that in Step 4 we can speak about anti-chains of P. Since Q is an anti-chain of P, Lemma 27 shows that indeed a cut C exists in G that has extra components corresponding to each $i \in Q$.

Next we show how the algorithm can be implemented in polynomial time. We observed earlier that in polynomial time, the blocks of G etc. can be identified, and G' can be constructed. Similarly, the separable blocks can be identified, and the weighted partial order $P = G'[\mathcal{B}]$ can be constructed. For Step 4 it is now necessary to find a maximum weight anti-chain in a partial order. This problem can be solved in polynomial time by translating it to a minimum flow problem, see [11]. Finally, note that the construction of a cut C in Lemma 27 can be done in polynomial time.

Finally we show that the returned solution is a 2-approximation. Let a minimum st-cut C_O of G be an optimal solution for MBMVC-LC, and let C_A be the minimum st-cut given by the algorithm. By k we denote again the number of vertices in a minimum st-cut of G. Let W_O respectively W_A be the total number of vertices in extra components of C_O and C_A. Both W_O and W_A are equal to the total weight of an antichain of the partial order P (Lemma 26, Lemma 27), so by choice of W_A, $W_O \leq W_A$.

First consider the case that the largest component L of $G - C_A$ is the s-component or t-component. Then $|V(L)| < |V(G)| - W_A - k \leq |V(G)| - W_O - k$. Either the s-component or the t-component of $G - C_O$ has size at least $(|V(G)| - W_O - k)/2$, so in this case we have proved the 2-approximation.

Now suppose the largest component L of $G - C_A$ is an extra component. By Lemma 26, $L = G[V_j]$ for some separable block E_j. Since C_O contains only critical vertices, all vertices

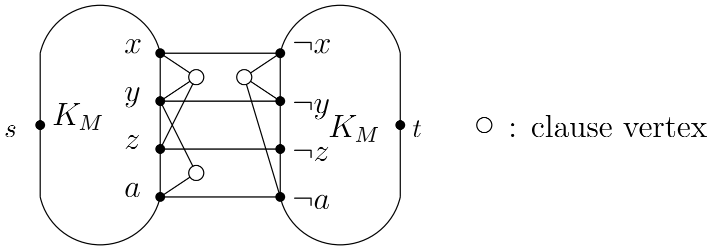


Figure 7: An MBMVC-SC instance corresponding to $(\neg x \vee \neg y \vee \neg z) \wedge (\neg y \vee \neg a) \wedge (x \vee y \vee a)$.

of V_j are part of the same component of $G - C_O$, and thus the optimal cut has a component that is just as large as L, in which case C_A is optimal. □

8 The inapproximability of MBMVC-SC

Any algorithm that returns some minimum st-cut of G trivially is a $2/n$-approximation for MBMVC-SC. The next theorem shows that it is impossible to do any better (unless P=NP).

Theorem 29 *No $1/(\alpha n)$-approximation algorithm with $\alpha < 1/2$ exists for MBMVC-MSC unless P=NP, where n is the number of vertices of the input graph.*

Proof: Let $\alpha < 1/2$. We give a reduction from monotone satisfiability (MSAT). A SAT instance consist of a set of boolean variables U and a set of clauses $\mathcal{C}$ over these variables. By x and $\neg x$ we denote the positive resp. negative literal corresponding to a variable $x \in U$. A clause is a set of literals over U. Given such an instance, the question is whether there is a truth assignment for the variables such that every clause contains at least one true literal. This decision problem is NP-complete even when restricted to instances where every clause contains either only positive literals, or only negative literals [7]. Such instances are called *monotone*. (The NP-completeness of this problem is easily deduced using a transformation from SAT: for every clause with both positive and negative literals, introduce a new variable and introduce two new clauses. For instance, $(x \vee y \vee \neg z)$ becomes $(c \vee x \vee y) \wedge (\neg c \vee \neg z)$. This yields an equivalent monotone instance.)

For any MSAT instance $U, \mathcal{C}$, we will show how to construct in polynomial time a MBMVC-SC instance G, s, t such that if $U, \mathcal{C}$ is a NO-instance, every minimum st-cut yields an isolated vertex, and if $U, \mathcal{C}$ is a YES-instance, a minimum st-cut exists that yields two components that contain more than $\alpha|V(G)|$ vertices. Giving this instance as input to a hypothetical $1/(\alpha n)$-approximation algorithm for MBMVC-SC would therefore solve MSAT in polynomial time, proving the statement.

We now explain the transformation, which is illustrated in Figure 7. Start with two copies of a large complete graph K_M (the exact value of M will be determined later), and call these graphs G_s and G_t. Choose one vertex in G_s to be s, and one vertex in G_t to be t. For every variable $x \in U$, do the following: choose a vertex in G_s, and label it x, and choose a vertex in G_t and label it $\neg x$. These are the *literal vertices* for the variable x. Add an edge between x and $\neg x$. Do this such that no vertex receives two different labels (M will be chosen large enough for this). For every clause introduce a *clause vertex*, and join it to the vertices corresponding to the *negations* of the literals in this clause.

It is easy to see that a minimum st-cut contains exactly $|U|$ vertices; one of the two literal vertices for every variable is in a minimum st-cut. It follows that any minimum st-cut corresponds to a truth assignment of the variables (a variable is made true if and only if its positive literal is in the cut). In addition, for every possible truth assignment of the variables, selecting the vertices of G that correspond to true literals gives a minimum st-cut: for this it is essential that we started with a *monotone* SAT instance such that no clause vertex has both a neighbor in G_s and in G_t. A clause vertex forms an isolated vertex in a minimum st-cut if and only if the corresponding truth assignment makes the clause false. On the other hand, in an assignment where every clause is true, the corresponding cut has two large components, both with at least $M - |U|$ vertices. The total number of vertices of G is $n = 2M + |\mathcal{C}|$. Now if we choose $M > (\alpha|\mathcal{C}| + |U|)/(1 - 2\alpha)$, then we have

$$M(1 - 2\alpha) > \alpha|\mathcal{C}| + |U| \Leftrightarrow M - |U| > \alpha(2M + |\mathcal{C}|) = \alpha n.$$

This shows that a $1/(\alpha n)$-approximation algorithm would be able to distinguish between the two cases, and hence answer the MSAT problem correctly. Note that since $1-2\alpha$ is a constant, this choice of M yields a polynomial transformation. This concludes the proof. □

9 Conclusions

In this paper we gave a number of initial results on most balanced minimum cut problems, which have received little study until now. We considered edge cuts and three natural vertex cut variants. All variants turned out to be NP-hard. We identified polynomial time solvable cases and gave approximation algorithms for all problems, except for MBMVC-SC which was shown to be inapproximable in a strong sense. Our results are based on the partial order structure of minimum st-cuts, which is interesting by itself. One of our algorithms is also a PTAS for a special type of UPOK problems. The following questions indicate possible directions for future research.

1. We gave a 2-approximation for MBMVC-LC, but the strongest negative result for this problem is only that it is NP-hard. Can this approximation ratio be improved?

2. For many special cases of POK, good approximation algorithms are known [9]. For general cases of POK and UPOK, little is known about the approximability. It would be useful to find better positive or negative results for these problems.

3. We showed that MBMEC and MBMVC-2P can be solved in polynomial time if the corresponding partial order has bounded width, however the width appears in the exponent of the algorithm. Is it possible to give a polynomial time algorithm for these problems without this property, i.e. an FPT algorithm where the width is the parameter? See [5] for FPT algorithms for a slightly smaller instance class.

4. MBMEC and MBMVC-2P can be solved in polynomial time when the underlying partial order is 2-dimensional. For which other instance classes can the problem be solved in polynomial time?

5. We studied the problem of finding most balanced cuts among the set of all minimum st-cuts. The problem of finding most balanced cuts among all cuts with bounded number of edges or vertices is interesting as well.

References

[1] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Comput. Complexity*, 15(2):94–114, 2006.

[2] M. Chimani, C. Gutwenger, and P. Mutzel. On the minimum cut of planarizations. Technical Report TR06-1-003, University of Dortmund, Germany, 2005.

[3] J. Chuzhoy and S. Khanna. Hardness of cut problems in directed graphs. In *Proceedings of the 38th STOC*, pages 527–536, New York, 2006. ACM.

[4] R. Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.

[5] U. Feige and M. Mahdian. Finding small balanced separators. In *Proceedings of the 38th STOC*, pages 375–384, New York, 2006. ACM.

[6] L. R. Ford, Jr. and D. R. Fulkerson. *Flows in networks*. Princeton University Press, Princeton, N.J., 1962.

[7] M. R. Garey and D. S. Johnson. *Computers and intractability*. Freeman, San Francisco, 1979.

[8] D. S. Johnson and K. A. Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Math. Oper. Res.*, 8(1):1–14, 1983.

[9] S. G. Kolliopoulos and G. Steiner. Partially ordered knapsack and applications to scheduling. *Discrete Appl. Math.*, 155(8):889–897, 2007.

[10] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[11] R. H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. In *Graphs and Order*, pages 41–101. Reidel, Dordrecht, 1985.

[12] D. B. Shmoys. Cut problems and their application to divide-and-conquer. In D.S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 192–235. PWS Publishing Company, Boston, 1997.