Discrete Applied Mathematics 158 (2010) 261–276

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Most balanced minimum cuts

Paul Bonsma *

Institut für Mathematik, Sekr. MA 5-1, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany

ARTICLE INFO

Article history:
Received 12 November 2007
Received in revised form 9 September 2009
Accepted 14 September 2009
Available online 25 September 2009

Keywords:
Balanced cut
Partially ordered knapsack
Approximation algorithm

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 NP-hard. We give a PTAS for the edge cut variant and for (i). These results also hold for directed graphs. 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 this 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 algorithm 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, in directed and undirected graphs. This problem differs from the balanced cut problems that are usually studied (see e.g. [17]): 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 a $(1 - \alpha)$-fraction of all vertices (this is called an *α-balanced cut*, with $0 < \alpha \leq \frac{1}{2}$). 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 a bounded number of 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. Throughout this paper the graphs for which we study these problems are assumed to be simple and (weakly) connected. The graphs may be directed or undirected. To distinguish these, in the case of directed graphs or *digraphs* we will talk about *arcs*, which are denoted (u, v). The arc set of G is denoted by $A(G)$. Undirected graphs have *edges*, denoted by $uv = vu$, and $E(G)$ denotes the edge set of G. The directed case is strictly more general, since all results for undirected graphs will also hold for *symmetric* directed graphs, in which every arc (u, v) is paired with an arc (v, u) in the opposite direction. For a digraph G and two non-empty, disjoint sets $S \subset V(G)$ and $T \subset V(G)$, $[S, T]$ denotes the set of arcs $(u, v) \in A(G)$ with $u \in S$ and $v \in T$. When G is undirected, $[S, T]$ denotes the set of edges $uv \in E(G)$ with $u \in S$ and $v \in T$, so in this case $[S, T] = [T, S]$. We write $[S, T]_G$ if it is not obvious that the graph G is considered. A set $M \subseteq E(G)$ or $M \subseteq A(G)$ is a *cut* of G if $M = [S, \overline{S}]$ for some non-empty $S \subset V(G)$.

* Tel.: +49 0 2093 3081.
E-mail address: bonsma@math.tu-berlin.de.

0166-218X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2009.09.010

We will also consider the case with non-uniform arc weights or *capacities*; let $c : A(G) \to \mathbb{N}^+$ be a capacity function. ($\mathbb{N}^+$ denotes $\mathbb{N}\backslash\{0\}$.) $c[S, T]$ denotes the sum of capacities of arcs in $[S, T]$. The cut $[S, \overline{S}]$ is a minimum *st-cut* if $s \in S$ and $t \in \overline{S}$, and $c[S, \overline{S}]$ is minimum among all such cuts. For edge/arc cuts, most reasonable balance requirements are equivalent; we choose the objective of maximizing $\min\{|S|, |\overline{S}|\}$.

Most Balanced Minimum *st*-Cut: (MBMC)
INSTANCE: A (di)graph G with edge/arc capacities c, two distinct vertices $s, t \in V(G)$.
SOLUTION: A minimum *st*-cut $[S, \overline{S}]$.
OBJECTIVE: Maximize $\min\{|S|, |\overline{S}|\}$.

For vertex cuts we first consider undirected graphs. A vertex cut of a connected graph G is a set $C \subset V(G)$ such that $G - C$ is disconnected. It is an *st-vertex cut* for $s, t \in V(G)$ if s and t are in different components of $G - C$. An *st*-vertex cut is a *minimum st-vertex cut* if $|C|$ is minimum among all *st*-vertex 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) *(Smallest Component (SC))*
INSTANCE: A graph G, two vertices $s, t \in V(G)$.
SOLUTION: A minimum *st*-vertex cut C.
OBJECTIVE: Minimize *(maximize)* the order of the largest *(smallest)* component of $G - C$.

In the third variant below the goal is to divide the components into two parts with close to equal cardinality, in the following way: when s and t are two vertices of a graph G, an *st-cut partition* of G is a tuple (S, C, T) such that C is a minimum *st*-vertex cut of G and $\{S, C, T\}$ is a partition of $V(G)$ such that $s \in S, t \in T$ and $[S, T] = \emptyset$. Hence components of $G - C$ that do not contain s or t may be assigned to the 's-side' or the 't-side' of the cut, but all vertices of a component have to be assigned to the same side.

MBMVC - Partition (P)
INSTANCE: A graph G, two vertices $s, t \in V(G)$.
SOLUTION: An *st*-cut partition (S, C, T) of G.
OBJECTIVE: Maximize $\min\{|S|, |T|\}$.

This last variant will turn out to be most similar to MBMC. 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.

Vertex cuts can also be defined for directed graphs G as follows: a set $C \subseteq V(G)\backslash\{s, t\}$ is a minimum *st*-vertex cut if $G - C$ contains no (s, t)-path (directed path), and $|C|$ is minimum among all such cuts. In the directed case we will consider the problem MBMVC-P above. In this case a cut partition (S, C, T) is defined the same way, but note that the condition $[S, T] = \emptyset$ then refers to a directed cut, so $[T, S]$ does not have to be empty. The above problems can also be generalized to the case with vertex weights or *capacities* as follows. Consider a capacity function $c : V(G) \to \mathbb{N}^+$. For an integer or real valued function c on a set V and $C \subseteq V$, $c(C)$ denotes the sum of values of elements in C. An *st*-vertex cut is then a *minimum st-vertex cut* if $c(C)$ is minimum among all *st*-vertex cuts.

Most balanced cut problems were previously studied by Feige and Mahdian [6], who studied MBMC 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 gave a fixed parameter tractable algorithm in the vertex case, where the parameter is the number of vertices k in a minimum *st*-cut (i.e. their algorithm has complexity $n^{O(1)}2^{O(k)}$ where n is the number of vertices). In addition they sketched an NP-hardness proof for MBMC, and remarked that a similar proof yields NP-hardness of the vertex cut variant. 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 [15] or [17]), 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.

Our results. Our results yield new simple NP-hardness proofs for all four problem variants above, for undirected unweighted graphs. In addition, for MBMVC-SC we show that no approximation algorithm with ratio better than the trivial ratio $2/n$ exists unless P=NP. We also study a version of MBMC 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*-Cut (GMBMC):
INSTANCE: A (di)graph G with edge/arc capacities $c > 0$.

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]. It is not obvious that NP-hardness of GMBMC follows from the NP-hardness of MBMC; we prove this statement, even for undirected unweighted graphs.

On the positive side, we give a PTAS for MBMC (for directed and weighted graphs). Clearly this also gives a PTAS for GMBMC, by trying every combination of s and t. Similar techniques yield a PTAS for MBMVC-P for directed graphs. This can also be generalized to graphs with vertex capacities c; in this case we consider the objective of maximizing $\min\{c(S), c(T)\}$ instead of $\min\{|S|, |T|\}$. For MBMVC-LC we give a 2-approximation, also for the case with vertex capacities.

To prove these results we make heavy use of the following partial order structure of minimum st-cuts. Partial orders consist of a ground set and a transitive, irreflexive binary relation on the elements of the ground set. Hence these are actually special types of directed graphs, and we will also view them as such. A directed graph (V, A) is *transitive* if for all $(u, v) \in A$ and $(v, w) \in A$ with $u \neq w$, $(u, w) \in A$ holds. A *partial order* is then a transitive digraph without cycles. The next notion comes from partial order theory, but can be defined as well for arbitrary directed graphs: for a digraph (V, A), $I \subseteq V$ is an *ideal* if $v \in I$ and $(u, v) \in A$ imply $u \in I$, or equivalently, if $[\overline{I}, I] = \emptyset$.

Given a MBMC instance G, c, s, t, a result by Picard and Queyranne [16] allows us to construct in polynomial time a partition P of the vertices of G and a partial order relation A on the sets in this partition such that the partial order $\mathcal{P} = (P, A)$ has I as a non-trivial ideal if and only if the corresponding vertex set S of G gives a minimum st-cut $[S, \overline{S}]$. (For an ideal, *non-trivial* means non-empty and not equal to the full set.) When adding weights to the elements of the partial order equal to the number of vertices in the set, this reduces MBMC to a problem closely related to uniform partially ordered knapsack:

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

This is the uniform version of partially ordered knapsack (POK). 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. POK is hard to approximate in general, but positive results are known for special cases of the problem, e.g. when the underlying partial order is 2-dimensional. See [14] for more information. We show that the variant of the problem that we reduce to does admit a PTAS. The essential property is that the desired weight W_U is close to $w(P)/2$ in our case. In general, we show that a PTAS exists for UPOK instances (P, A), w, W_U with $w(P)/W_U \in O(\log |P|)$. Our algorithm is similar to the one given by Feige and Yahalom [7] for deciding whether directed graphs have an α-balanced oneway cut. A *oneway cut* in a directed graph is a cut with $[S, \overline{S}] = \emptyset$ (i.e. $\overline{S}$ is an ideal). They showed that the existence of an α-balanced oneway cut can be determined in time $2^{\frac{1}{1-2\alpha}} n^{O(1)}$.

To prove our results about the MBMVC variants, we show that minimum st-vertex cuts can be characterized similarly with a partial order: we again define a partition P of the vertices of G and define a partial order relation A on these sets, such that the partially ordered set $\mathcal{P} = (P, A)$ has I as a non-trivial ideal if and only if G has an st-cut partition (S, C, T) where $S \cup C$ corresponds to the ideal I. This allows the PTAS for the UPOK variant to be used for MBMVC-P. In addition we characterize the sets in P that may form extra components of a minimum st-vertex cut, i.e. components that do not contain s or t. This is essential for the 2-approximation for MBMVC-LC.

We mention some related results on representations of minimum cuts. It is known that all global minimum cuts in undirected graphs can be represented by a cactus graph. A *global minimum cut* is an edge cut with minimum number of edges over all edge cuts of the graph. A *cactus* is a connected graph in which every edge is part of at most one cycle. It is known that for every graph G there exists a partition P of $V(G)$, a cactus H and a mapping of sets in P to vertices of H such that the mapping gives a bijection between the *minimal* cuts of H and global minimum cuts of G. This representation was introduced by Dinits, Karzanov and Lomonosov [5], see [9] for more information. This way it can also be shown that a graph on n vertices can have at most $\binom{n}{2}$ global minimum cuts [5]. A simpler proof appears in [13]; actually, Karger [13] shows that the number of cuts with capacity at most c times the capacity of a global minimum cut is at most $\binom{n}{2c}$, hence polynomial for fixed c. Efficiently enumerating these is also possible [18]. It follows that finding a most balanced minimum st-cut is possible in polynomial time for undirected graphs, when the ratio c defined above is bounded. In contrast, without a bound on c, note that the number of minimum st-cuts can be exponential, as can be observed from the complete bipartite graph $K_{2,n}$. For directed graphs, the problem of finding most balanced minimum st-cuts $[S, \overline{S}]$ is NP-hard even when $[S, \overline{S}] = \emptyset$ [7].

The paper is structured as follows. We start by giving definitions, notations and useful theorems from the literature in Section 2. Details of the partial order structure of minimum st-edge cuts are given in Section 3. 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 MBMC. 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-P. 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.

2. Preliminaries

For graph theoretic definitions not treated here, see [4]. For definitions related to algorithms and complexity, see [11]. 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 $\epsilon > 0$.

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. $S \backslash \{x\}$. A *walk* is a sequence of vertices $v_0, \ldots, v_k$ of a graph G such that $v_i v_{i+1} \in E(G)$ when G is undirected, and $(v_i, v_{i+1}) \in A(G)$ when G is directed. So a walk between two vertices may contain edges and vertices multiple times. The walk is *closed* if $v_0 = v_k$. 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. Paths and cycles H will also be viewed as graphs, we will e.g. use $V(H)$ to denote their vertex set. We say v is *reachable* from u in G if G contains a (u, v)-path. By $R(u)$ we denote the set of all vertices that are reachable from u.

We assume all graphs in the paper to be simple, that is, without loops or multi-edges (in case of digraphs, containing both an arc (u, v) and an arc (v, u) is allowed).

Recall that a *partial order* is a transitive digraph without cycles. For a partial order $\mathcal{P}$, $v \in I \subseteq V(\mathcal{P})$ is a *minimum* (*maximum*) of I if it has no in-neighbors (out-neighbors) in I, and a minimum (maximum) of $\mathcal{P}$ when $I = V(\mathcal{P})$. 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 *antichain* if all elements of S are pairwise incomparable. The *width* of a partial order is the maximum size of an antichain, which can be determined in polynomial time. See for instance [8], where a classic algorithm using bipartite matchings is described, together with faster, more recent algorithms.

In the remainder we will use flows with a single source and sink, where capacities are given on either arcs or vertices. Let (V, A) be a digraph with a *source vertex* $s \in V$ and *sink vertex* $t \in V - s$, and a capacity function $c_A : A \to \mathbb{N}^+$ on the arcs, and/or a capacity function $c_V : V \to \mathbb{N}^+$ on the vertices. An *st-flow* is a function $f : V \times V \to \mathbb{R}$, such that

$$\begin{array}{ll} 0 \le f(u, v) \le c_A(u, v) & \text{for all } (u, v) \in A \\ f(u, v) = 0 & \text{for all } (u, v) \notin A \\ \displaystyle\sum_{u \in V} f(u, v) \le c_V(v), & \text{for all } v \neq s, t \end{array}$$

and in addition for every vertex $v \in V - s - t$ the following holds: $\sum_{w \in V} (f(v, w) - f(w, v)) = 0$ (flow conservation). Note that we may assign capacities c to s and t although these values are irrelevant for the flow. The *value* $|f|$ of an st-flow f is

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

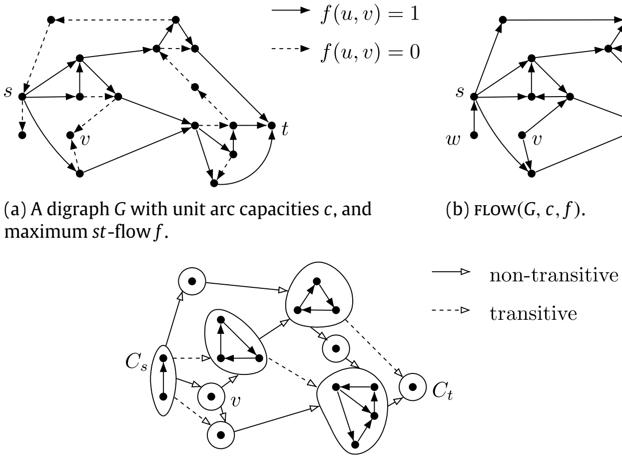
For two disjoint vertex sets $S, T \subset V$, we define $f(S, T) = \sum_{u \in S, v \in T} f(u, v)$. For a vertex $v \neq s, t$ we will write $f(v) = \sum_{u \in V} f(u, v) = \sum_{u \in V} f(v, u)$. A flow f that maximizes $|f|$ among all possible st-flows is called a *maximum st-flow*. Well-known theorems by Ford and Fulkerson [10] show that the value of a maximum st-flow in a graph with arc capacities is equal to the capacity of a minimum st-cut. In a graph with vertex capacities c, the value of a maximum st-flow is equal to $c(C)$ for a minimum st-cut C. A maximum st-flow and minimum st-cut (st-vertex cut) can be found in polynomial time. The above representation of st-flows is the *arc representation*, which will be denoted simply by the function f. A different representation is the *path representation*. In this case, a set of (s, t)-paths $\mathcal{Q}$ is given, with flow values $f(P) > 0$ for all $P \in \mathcal{Q}$. Such a flow should satisfy the arc capacities c_A and vertex capacities c_V in the following way: for all $(u, v) \in A(G)$, $\sum_{P:(u,v) \in A(P)} f(P) \le c_A(u, v)$, and for all $v \in V(G) - s - t$, $\sum_{P: v \in V(P)} f(P) \le c_V(v)$. The *value* of the flow is in this case $|f| = \sum_{P \in \mathcal{Q}} f(P)$. $\mathcal{Q}, f$ denotes a flow in path representation. It is not hard to see that if an arc representation of a flow with value v is given, a path representation of a flow with value v can be found in polynomial time, and vice versa. Moreover, we can always construct a path representation that uses at most $|A(G)|$ paths.

We will also consider flows in undirected graphs; these are defined as flows in the corresponding symmetric graph, in which every edge uv is replaced by two arcs (u, v) and (v, u) both with capacity $c(uv)$.

3. The partial order structure of minimum *st*-edge cuts

Throughout this section, G, c, s, t denotes a MBMC instance, so G is a digraph, c are non-zero integer capacities on the arcs, and s and t are two distinct vertices of G. In this section we will give a polynomial transformation from G, c, s, t to a partial order $\mathcal{P}$ 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 $\mathcal{P}$ has a non-trivial ideal $I \subset V(\mathcal{P})$ with $w(I) = x$. This reduces MBMC to the following problem.

MOST BALANCED IDEAL (MBI):
INSTANCE: A partial order $\mathcal{P} = (V, A)$, weights $w : V \to \mathbb{N}^+$.
SOLUTION: A non-trivial ideal I of $\mathcal{P}$.
OBJECTIVE: Maximize $\min\{w(I), w(\overline{I})\}$.



(a) A digraph G with unit arc capacities c, and maximum st-flow f.

(b) $\text{FLOW}(G, c, f)$.

(c) $\text{PO}(\text{FLOW}(G, c, f), s, t)$, most transitive arcs are omitted.

Fig. 1. A graph with maximum st-flow, the resulting flow digraph and partial order.

The transformation from this section is illustrated in Fig. 1; the first step is based on [16]. Let f be a maximum st-flow in G with respect to arc capacities c. The *flow (di)graph* $\text{FLOW}(G, c, f)$ of G with respect to f is defined as follows: $\text{FLOW}(G, c, f)$ has vertex set $V(G)$, and for every $u, v \in V(G)$ it contains an arc (u, v) when G contains an arc (u, v) with $f(u, v) > 0$, or when G contains an arc (v, u) with $f(v, u) < c(v, u)$. In [16] the following theorem is proved; we include the proof for completeness.

Theorem 1 (*Picard & Queyranne*). *Let f be a maximum st-flow in a digraph G with arc capacities c. Then for any $S \subset V(G)$ with $s \in S$ and $t \notin S$, $[S, \overline{S}]$ is a minimum st-cut if and only if S is an ideal of* $\text{FLOW}(G, c, f)$.

Proof. Observe that $[S, \overline{S}]$ with $s \in S$ and $t \in \overline{S}$ is a minimum st-cut if and only if both $f(S, \overline{S}) = c[S, \overline{S}]$ and $f(\overline{S}, S) = 0$ hold (here $f(S, \overline{S}) - f(\overline{S}, S) = |f|$ follows from flow conservation, $|f| = c[S, \overline{S}]$ from the min-cut max-flow theorem, and $f(S, \overline{S}) \leq c[S, \overline{S}]$ from the capacity constraints). So if $[S, \overline{S}]$ is a minimum st-cut of G, $[S, \overline{S}]_G$ does not contain an arc (u, v) with $f(u, v) < c(u, v)$, and $[\overline{S}, S]_G$ does not contain an arc (u, v) with $f(u, v) > 0$. So by definition of $G' = \text{FLOW}(G, c, f)$, $[\overline{S}, S]_{G'} = \emptyset$. To prove the other direction, if S is an ideal of G' with $s \in S$ and $t \in \overline{S}$ then for every arc $(u, v) \in [S, \overline{S}]_G$ $f(u, v) = c(u, v)$ must hold, and for every arc $(u, v) \in [\overline{S}, S]_G$ $f(u, v) = 0$ holds. Hence $[S, \overline{S}]$ is a minimum st-cut. □

We remark that in [16] the graph $\text{FLOW}(G, c, f)$ is defined such that all arcs appear in the opposite direction, which gives the well-known *residual graph* for the flow f, but for our purposes it is more convenient to use this arc direction.

Picard and Queyranne also observed that from $\text{FLOW}(G, c, f)$, a partial order can be obtained by first contracting all strong components of $\text{FLOW}(G, c, f)$ into single vertices and then adding all transitive arcs to the graph. This way, all minimum st-cuts of G still correspond to ideals of the resulting partial order. The reverse implication is not true, but elements of the partial order can be deleted to ensure that there is a bijection between partial order ideals and minimum st-cuts. For instance, the two strong components given by the single vertices v and w in Fig. 1(b) yield non-empty ideals that do not separate s from t. Deleting v and w gives a bijection between *non-trivial* ideals and minimum st-cuts; deleting s and t as well gives a bijection from *all* ideals. However, since we need that the ground set of the partial order is a partition of the vertices of the original graph, we must define the partial order differently. This is illustrated in Fig. 1(c). Elements of the partial order are subsets of the vertices of $G' = \text{FLOW}(G, c, f)$, which are indicated in the figure by showing the subgraph of G' they induce. Note that all elements of the partial order induce strong components of G', except possibly C_s and C_t. C_s and C_t are the unique minimum and maximum, respectively. To obtain this last property, an arc is added from C_s to the set containing v, although v is not reachable from s in G'. Recall that $R(u)$ denotes the set of vertices that are reachable from u. The next definition and lemma will be used later with $\text{FLOW}(G, c, f)$ in the role of D.

Definition 2. Let D be a digraph with vertices $s, t \in V(D)$ such that $t \in R(s)$ and $s \notin R(t)$. Then $\text{PO}(D, s, t) = \mathcal{P}$ is defined as follows:

- Let $C_s \subset V(D)$ be the set of vertices v with $s \in R(v)$, and let $C_t \subset V(D)$ be the set of vertices v with $v \in R(t)$.
- $\mathcal{P}$ has as vertices the sets C_s, C_t, and all sets $V(H)$ where H is a strong component of D that contains no vertices of C_s or C_t.
- For distinct $C_1, C_2 \in V(\mathcal{P})$, $(C_1, C_2) \in A(\mathcal{P})$ if and only if (i) $C_1 = C_s$, (ii) $C_2 = C_t$, or (iii) C_2 is reachable from C_1 in D.

Elements of $V(\mathrm{PO}(D,s,t))$ will also be called *blocks* of D. The next lemma shows that $\mathcal{P}=\mathrm{PO}(D,s,t)$ is indeed a partial order, of which the non-trivial ideals correspond bijectively to ideals of D that separate s from t. For $I\subseteq V(\mathcal{P})$, let $V(I)$ denote $\cup_{C\in I} C$, so $V(I)\subseteq V(D)$. Throughout, for partial orders $\mathrm{PO}(D,s,t)$, we will use C_s and C_t to denote the sets that contain s and t.

Lemma 3. *Let $\mathcal{P}=\mathrm{PO}(D,s,t)$. Then*

1. *$V(\mathcal{P})$ is a partition of $V(D)$,*
2. *$\mathcal{P}$ is a partial order with unique minimum C_s and unique maximum C_t.*
3. *$S\subset V(G)$ is an ideal of D with $s\in S$, $t\notin S$ if and only if $\mathcal{P}$ has a non-trivial ideal I with $V(I)=S$.*

Proof. $\mathrm{PO}(D,s,t)$ is only defined when $s\notin R(t)$, which guarantees that $C_s\cap C_t=\emptyset$. Strong components of D are either contained in C_s or C_t, or disjoint from them, so $V(\mathcal{P})$ is a partition of $V(D)$.

All vertices from which C_s can be reached are part of C_s, so C_s has in-degree zero, and similarly C_t has out-degree zero. All other elements of $V(\mathcal{P})$ are out-neighbors (in-neighbors) of C_s (C_t), so this is the unique minimum (maximum). To check transitivity, consider arcs (C_1,C_2) and (C_2,C_3). If $C_1=C_s$ or $C_3=C_t$ then clearly $(C_1,C_3)\in A(\mathcal{P})$. Otherwise, C_2 is reachable from C_1, C_3 is reachable from C_2 and C_2 is a strong component, so $(C_1,C_3)\in A(\mathcal{P})$. Note that $C_1=C_3$ is not possible since then a larger strongly connected subgraph would exist. It follows that $\mathcal{P}$ is a partial order.

Now we prove the third property. Consider an ideal S of D with $s\in S$, $t\notin S$. It follows that $C_s\subseteq S$ and $C_t\notin S$. For every strong component H of D that does not contain vertices of C_s or C_t, either $V(H)\subseteq S$ or $V(H)\cap S=\emptyset$. So there exists a non-empty $I\subset V(\mathcal{P})$ with $V(I)=S$. Since S is an ideal, I is an ideal of $\mathcal{P}$. For the other direction, consider a non-trivial ideal I of $\mathcal{P}$. I then contains the unique minimum C_s and does not contain the unique maximum C_t. So $S=V(I)$ contains s but not t. Since no $C_1\in I$ can be reached from a $C_2\notin I$, it follows that $V(I)$ is an ideal of D. $\square$

Theorem 1 gives a bijection between minimum st-cuts of G and ideals of $G'=\mathrm{FLOW}(G,c,f)$ that separate s from t. Lemma 3 subsequently gives a bijection between such ideals of G' and non-trivial ideals of $\mathrm{PO}(G',s,t)$, hence we have the following theorem.

Theorem 4. *Let f be a maximum st-flow in a digraph G with arc capacities c, and let $\mathcal{P}=\mathrm{PO}(\mathrm{FLOW}(G,c,f),s,t)$. G has $[S,\overline{S}]$ as a minimum st-cut if and only if $\mathcal{P}$ has a non-trivial ideal $I\subset V(\mathcal{P})$ with $V(I)=S$.*

If $\mathcal{P}_1$ and $\mathcal{P}_2$ are both partial orders such that $V(\mathcal{P}_1)$ and $V(\mathcal{P}_2)$ are partitions of G, and $\mathcal{P}_1$ has an ideal I_1 with $V(I_1)=S$ if and only if $\mathcal{P}_2$ has an ideal I_2 with $V(I_2)=S$, then it is easily seen that $\mathcal{P}_1=\mathcal{P}_2$ (both the partition and the arc set are the same). Hence from Theorem 4 and Lemma 3 it follows that $\mathrm{PO}(\mathrm{FLOW}(G,c,f))$ is uniquely determined by the choice of G,c,s,t, and does not depend on the chosen flow f. Therefore we will also denote this partial order as $\mathrm{PO}_A(G,c,s,t)$, ignoring the flow in the notation (the subscript A indicates that c denotes *arc* capacities).

For $\mathcal{P}=\mathrm{PO}_A(G,c,s,t)$, we can assign weights $w(C)=|C|$ for all $C\in V(\mathcal{P})$. This way we obtain a MBI instance $\mathcal{P}$, w that is equivalent to the original MBMC instance, in the sense that any solution to one problem immediately yields a solution to the other problem with the same objective value. Note also that all steps in the construction of $\mathrm{PO}_A(G,c,s,t)$ can be done in polynomial time.

This transformation can also easily be done when G is undirected: simply replace every edge uv by two arcs (u,v) and (v,u), both with capacity $c(uv)$. Then $[S,\overline{S}]$ is a minimum st-cut in G if and only if it is a minimum st-cut in the resulting directed graph, for which we can then construct the partial order as shown above.

4. Algorithms for finding the most balanced ideals

In Section 9 we transformed MBMC 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 (P,A), w, W_U with $w(P)/W_U\in O(\log|P|)$. The PTAS for MBI is given in Algorithm 1.

Feige and Yahalom [7] described an algorithm that is very similar to Algorithm 1, as was pointed out to us by a referee. They showed that the algorithm can decide in time $2^{\frac{1}{1-2\alpha}}n^{O(1)}$ whether P has an *α-balanced ideal*, that is, an ideal I with $\alpha|P|\le w(I)\le(1-\alpha)|P|$ $(0<\alpha<1/2)$. We analyze the algorithm from an approximation viewpoint, and show that it yields a PTAS for the optimization problem. An important observation used in the proof below is that for a partial order (P,A) and any set $S\subseteq P$, there is a *unique* maximal ideal that is disjoint from S, and an analog statement holds for minimal ideals containing a set $S\subseteq P$. Indeed, if two different minimal ideals I_1 and I_2 exist that contain S, then w.l.o.g. $I_1\backslash I_2\neq\emptyset$. Then $I_1\backslash I_2$ must also contain a maximum of I_1 which is not in S, which contradicts the minimality of I_1.

Theorem 5. *Algorithm 1 is an $(1-\epsilon)$-approximation algorithm for MBI with time complexity $n^{O(1)}2^{1/\epsilon}$, where n is the input size.*

Proof. It is easy to see that every step of the algorithm within the for-loop and outside of the for-loop has a complexity that 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 $n^{O(1)}2^{1/\epsilon}$.

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 $W_U-W_L=\epsilon W$. We will argue that the algorithm will find an optimal solution, or a solution between

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 $L \backslash L' \subseteq \bar{I}$ **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.

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 (unique) 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 $I_{\max}$. (Observe that if an ideal is not maximal for this property, there is always a single element that can be added, and hence the algorithm finds a maximal ideal.) 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 $\epsilon W = W_U - W_L$. This concludes the proof. □

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

Corollary 6. *A PTAS exists for MBMC.*

With minor changes Algorithm 1 is also a PTAS for special instances (P, A), w, W_U of UPOK: the set of large elements L needs to be defined as the elements x with $w(x) > \epsilon W_U$. 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 either way the algorithm returns a solution with weight at least $(1-\epsilon)$ times the optimal weight. Now we consider the time complexity. Let $W = w(P)$ and let $n = |P|$. The complexity is bounded by $n^{O(1)}2^{|L|} < n^{O(1)}2^{W/(\epsilon W_U)}$. So for the class of instances with $W/W_U \in O(\log n)$, this is bounded by $n^{O(1/\epsilon)}$.

Theorem 7. *A PTAS exists for classes of UPOK instances (P, A), w, W_U with $w(P)/W_U \in O(\log |P|)$.*

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

In [14], 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 [14] is based on dynamic programming. This also gives a pseudopolynomial time algorithm for MBI on such instances, and a polynomial time algorithm for MBMC for corresponding instances.

Finally we remark that MBI can easily be solved in polynomial time when the partial order has bounded width w by simply enumerating all $O\left(\binom{n}{w}\right)$ ideals. The width can be shown to be bounded in the case of undirected graphs when the ratio between the capacities of minimum st-cuts and minimum global cuts is bounded. So this gives an alternative algorithm for this case (see also Section 1).

5. NP-hardness proofs for the most balanced cut problems

In this section we prove that MBMC is NP-hard, also when restricted to undirected instances with unit edge capacities. More precisely, we prove the NP-completeness of the decision variant of MBMC, which has an additional parameter l and asks

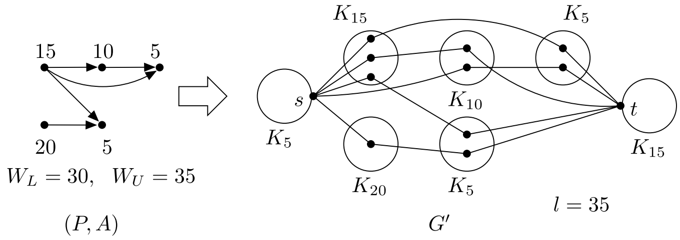


Fig. 2. The transformation from UPOK to MBMC.

whether the instance G, s, t has a minimum st-edge cut $[S, \bar{S}]$ with $\min\{|S|, |\bar{S}|\} \geq l$. The transformation is a straightforward transformation from the decision variant of UPOK, which is nearly the reverse of the transformation in Section 3. Feige and Mahdian [6] gave a reduction from Max Clique to prove the NP-completeness of MBMC. We give a different construction, and a detailed correctness proof using Theorem 1.

Theorem 8. *The decision version of MBMC is NP-complete when restricted to undirected graphs with unit edge capacities.*

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 integers. 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 [12], that is, even if the weights are encoded in unary and therefore the instance size is $\Omega(w(P))$, the problem is NP-complete. W.l.o.g. assume that every vertex in P is incident with at least one arc. As a first step, we scale all weights w and the bounds W_L and W_U with a factor $|A|$, so for all $u \in P$ we may now assume $w(u) \geq |A|$.

We transform this instance to an undirected MBMC instance G, s, t, l with unit arc capacities as illustrated in Fig. 2. Choose weights w_s and w_t such that $w_s + W_L = w_t + (w(P) - W_U)$ and $\min\{w_s, w_t\} = |A|$ (in the example this gives $w_s = 5$ and $w_t = 15$). 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 choose vertices $x_1 \in V(C_u)$ and $x_2 \in V(C_v)$, and add the edges sx_1, x_1x_2 and x_2t to G. Since every block has at least $|A|$ vertices, we can always choose x_1 to be different from previous choices of x_1, and choose x_2 to be different from previous choices of x_2. This way, no parallel edges are introduced. All edges of G have a capacity of 1. This completes the construction.

Now consider the following flow f (this is actually a flow in the corresponding symmetric digraph). For every $(u, v) \in (P, A)$, we send a flow of 1 along the path s, x_1, x_2, t in G (using the vertices x_1 and x_2 that were chosen for (u, v)). This is a maximum flow that saturates all edges between different blocks. Hence if we consider the flow graph $G' = \textsc{flow}(G, c, f)$ defined in Section 3, there is an arc from $V(C_u)$ to $V(C_v)$ in G' if and only if $(u, v) \in A$, $u = s$ or $v = t$. Furthermore, there is no flow between different vertices of a block C_v, so all of these blocks correspond to strongly connected components in G' (with arcs in both directions for every edge). This shows that ideals of (P, A) correspond bijectively to non-trivial ideals $S \subset V(G')$ of G' as shown in Section 3. So by Theorem 1, (P, A) has a non-trivial ideal with weight x if and only if G has an st-cut $[S, \bar{S}]$ with $|S| = w_s + x$.

Consider an ideal I of (P, A) and corresponding st-cut $[S, \bar{S}]$ 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

$$\begin{aligned}
&W_L \leq w(I) \leq W_U \Leftrightarrow W_L + w_s \leq |S| \leq W_U + w_s \Leftrightarrow \\
&|S| \geq W_L + w_s \wedge |\bar{S}| \geq 2w_s + W_L + W_U - (W_U + w_s) \Leftrightarrow \\
&|S| \geq W_L + w_s \wedge |\bar{S}| \geq w_s + W_L.
\end{aligned}$$

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 8 proves NP-completeness, which is shown schematically in Fig. 3. An edge between a vertex v and a complete graph K_k in this figure means that v is adjacent to all vertices in K_k. In this case we first need to scale the weights and W_L and W_U by a factor $2|A|$. Let $d^-(v)$ and $d^+(v)$ denote the in- and out-degree of vertices $v \in P$. For every $v \in P$ we introduce a complete graph C_v on $w(v) - d^-(v) - d^+(v)$ vertices, and we introduce complete graphs C_s and C_t on w_s and w_t vertices (w_s and w_t are defined the similar as in the above proof, but now $\min\{w_s, w_t\} = 2|A|$ should hold). The vertices s and t are chosen in C_s and C_t again. For every arc $(u, v) \in A$, we do the following: add a vertex x_1, and join it to all vertices from C_s and C_u. Add a vertex x_2, and join it to all vertices from C_u and C_v. Finally, add a vertex x_3, and join it to all vertices from C_v and C_t. All vertex capacities are set to one. This completes the construction of the MBMVC instance. Similar to the above proof, it can be checked that any minimum

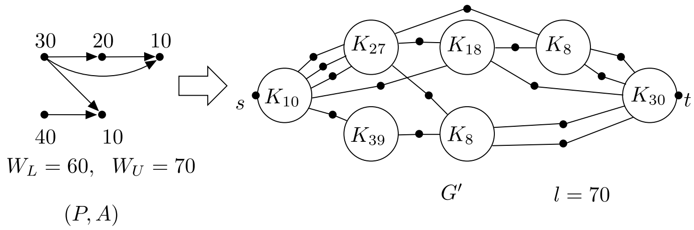


Fig. 3. The transformation from UPOK to MBMVC.

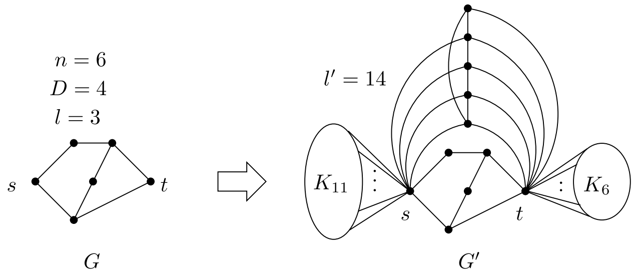


Fig. 4. The transformation from MBMC to GMBMC.

st-vertex cut contains $|A|$ vertices, and separates the graph G into two components $G[S]$ and $G[T]$ with $s \in S$, $t \in T$. Finally, such a cut with $|S| = w_s + x$ and $|T| = w_t + w(P) - x$ exists if and only if (P, A) has an ideal I with $w(I) = x$. (This can formally be proved using the theory from Section 6.) Since in every case a minimum st-vertex cut yields only two components, this proves the NP-completeness of all three variants of the problem.

Theorem 9. *The decision versions of MBMVC-P, MBMVC-SC and MBMVC-LC are NP-complete when restricted to unit vertex capacities.*

We now consider the problem GMBMC. The NP-hardness of MBMC allows us to prove the NP-hardness of GMBMC.

Theorem 10. *The decision version of GMBMC is NP-complete for undirected graphs with unit edge capacities.*

Proof. Let G, s, t, l be an instance for the decision variant of MBMC, where a minimum st-cut of G contains $k \geq 1$ edges. By $\Delta(G)$ we denote the maximum degree of G. 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 GMBMC, which asks whether there is a solution with objective value at least l'. The construction is illustrated in Fig. 4. Note that the instance G, s, t, l in this figure is a NO-instance for MBMC, but G, l is a YES-instance for GMBMC, 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 H_s and H_t on $n + D + 1$ and n vertices respectively, and join all vertices of H_s to s, and all vertices of H_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 $V(C) \subset \overline{S}$, $V(H_s) \subset S$ and $V(H_t) \subset \overline{S}$. (The techniques from Section 3 can for instance be used for this.) 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 GMBMC solution for G' separates s from t. Now let $[S, \overline{S}]$ be such an optimal GMBMC 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 $V(H_s) + s$ or both part of $V(H_t) + t$, then the cut also does not separate s from t. We conclude w.l.o.g. that $x \in V(H_s) + s$ and $y \in V(H_t) + t$. 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 GMBMC 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. For every MBMVC variation, the instance consists of a (possibly directed) graph G and two vertices s and t. All vertices $v \neq s, t$ are assigned integer capacities $c(v) > 0$. Let k_{st} be the capacity of a minimum st-vertex 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 4. This construction can be combined with the PTAS from Section 4 to yield a PTAS for MBMVC-P. To be precise, we will again construct a partial order $\mathcal{P}$ where elements in $V(\mathcal{P})$ correspond to subsets of the vertices of G, such that together these subsets partition $V(G)$. An ideal I of $\mathcal{P}$ will then correspond to an st-cut partition (S, C, T) of G with $V(I) = S \cup C$. The definitions and constructions in this section are illustrated in Fig. 5. Throughout this section, G is a directed graph.

We consider a path representation $\mathcal{Q}, f$ of a maximum st-flow, subject to the vertex capacities c. So we choose a set of (s, t)-paths $\mathcal{Q}$, together with flow values f, such that $\sum_{P\in\mathcal{Q}} f(P) = k_{st}$, and for every vertex $v \in V(G) - s - t$, $\sum_{P:v\in V(P)} f(P) \leq c(v)$. As we mentioned in Section 2, such a flow and path representation can be found in polynomial time. A vertex u that lies before a vertex v on one of these paths $P \in \mathcal{Q}$ is called a *path predecessor* of v, and v is called a *path successor* of u. The *flow through a vertex* v is $f(v) = \sum_{P:v\in V(P)} f(P)$. A vertex $v \neq s, t$ with $f(v) = c(v)$ is called *saturated* (since the capacities of s and t are irrelevant for st-flows, they are never considered saturated). We have the following simple observation. The case where C is a minimum st-cut and f is a maximum st-flow is one case where the next statement can be applied.

Claim 11. *Let G be a digraph with vertex capacities c, let $\mathcal{Q}, f$ be an st-flow in G, and let $C \subseteq V(G) - s - t$ be a vertex set that contains at least one vertex of every path in $\mathcal{Q}$. Then $c(C) = |f|$ if and only if all vertices of C are saturated by f, and every path in $\mathcal{Q}$ contains exactly one vertex of C.*

Proof. Let α_P be the number of vertices of C that lie on a path $P \in \mathcal{Q}$. By our assumption, $\alpha_P \geq 1$ for all P.

$$c(C) = \sum_{v\in C} c(v) \geq \sum_{v\in C} f(v) = \sum_{P\in\mathcal{Q}} f(P) \cdot \alpha_P \geq \sum_{P\in\mathcal{Q}} f(P) = |f|.$$

From this it follows that $c(C) = |f|$ if and only if both inequalities above are equalities, which proves the statement. □

Let G be a digraph for which an st-flow $\mathcal{Q}, f$ is given. For two vertices u and v of G, if G contains a (u, v)-path that contains no saturated vertices as *internal* vertices, then we say *v is non-saturated reachable from u* (with respect to $\mathcal{Q}, f$). The following useful observations follow easily from Claim 11.

Claim 12. *Let (S, C, T) be an st-cut partition of a digraph G and let $\mathcal{Q}, f$ be a maximum st-flow of G.*

1. *If $v \in S \cup C$, then all path predecessors of v in $\mathcal{Q}$ are in S.*
2. *C is exactly the set of vertices in $S \cup C$ that have no saturated path successor in $S \cup C$.*
3. *For every vertex $v \in S$, all vertices that are non-saturated reachable from v with respect to $\mathcal{Q}, f$ are in $S \cup C$.*

Proof. By Claim 11, every path $P \in \mathcal{Q}$ contains exactly one vertex of C, which is saturated. Since $[S, T] = \emptyset$, every path $P \in \mathcal{Q}$ then consists of a sequence of vertices in S (since the first vertex $s \in S$), a single vertex in C, and a sequence of vertices in T (since $t \in T$). This proves the first statement, and shows that C is exactly the set of vertices in $S \cup C$ with no path successor in $S \cup C$. Since all vertices in C are saturated, C can then also be characterized as in the second statement.

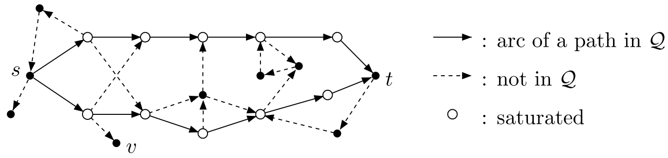
To prove the third statement, consider $u \in S$, and suppose a vertex v is non-saturated reachable with a (u, v)-path P'. Since none of the internal vertices of P' are saturated, they are not in C. Since $[S, T] = \emptyset$, all internal vertices of P' are therefore in S, and the end vertex v may be in S or C, but not in T. □

For a digraph G with vertex capacities c and maximum st-flow $\mathcal{Q}, f$, the *vertex flow digraph* $G' = \textsc{flow}_V(G, c, \mathcal{Q}, f)$ is now constructed as follows (see also Fig. 5(b); note that many transitive arcs are omitted). G' has the same vertex set as G. For a saturated vertex v, add an arc (u, v) for any vertex u that is non-saturated reachable from a path predecessor of v. (In particular, for every path predecessor u of v, $(u, v) \in A(G')$.) For a non-saturated vertex v, add an arc (u, v) for any vertex u that is non-saturated reachable from v.

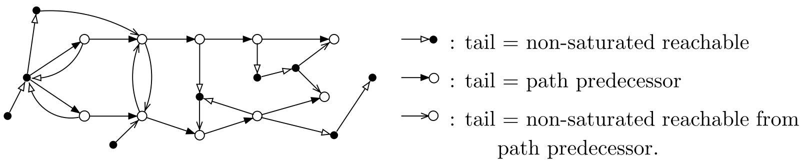
Note that the structure of $\textsc{flow}_V(G, c, \mathcal{Q}, f)$ clearly depends on the choice of the flow $\mathcal{Q}, f$. In a different maximum st-flow, a vertex may have different path predecessors and different non-saturated reachable vertices. This is illustrated by Fig. 6 which shows a different flow for the graph from Fig. 5 (most transitive arcs are omitted again in the flow graph). However, it can be seen that the ideals of the vertex flow digraphs in both cases are exactly the same. In the next theorem we show that this holds in general, provided that the chosen flow is a maximum st-flow. This theorem is the analog of Theorem 1 for vertex cuts. The correspondence between ideals and cut partitions is illustrated in Fig. 7 (recall that $[T, S]$ does not have to be empty).

Theorem 13. *Let $\mathcal{Q}, f$ be a maximum st-flow in a digraph G with vertex capacities c. The graph $\textsc{flow}_V(G, c, \mathcal{Q}, f)$ has I as an ideal with $s \in I$, $t \notin I$ if and only if G has an st-cut partition (S, C, T) such that $S \cup C = I$.*

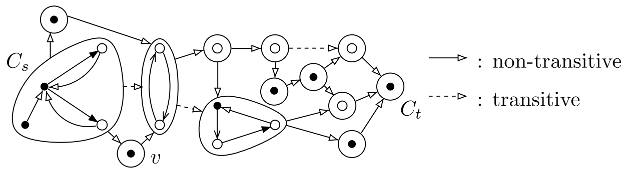
Proof. Let $G' = \textsc{flow}_V(G, c, \mathcal{Q}, f)$ and let (S, C, T) be an st-cut partition of G. We prove that $I = S \cup C$ is an ideal of G'. Consider an arc $(u, v) \in A(G')$ with $v \in I$. We show that $u \in I$. If v is a saturated vertex, then by construction of G', u is non-saturated reachable from some path predecessor w of v. By Claim 12, $w \in S$. Then by the third statement in Claim 12, $u \in S \cup C = I$. On the other hand, if v is not saturated, then $v \notin C$ (Claim 11) so $v \in S$. By construction of G', u is non-saturated reachable from v. Then by Claim 12 again, $u \in S \cup C = I$. This shows that I is an ideal.



(a) Graph G with unit vertex capacities and two (s, t)-paths of a maximum flow $\mathcal{Q}, f$.

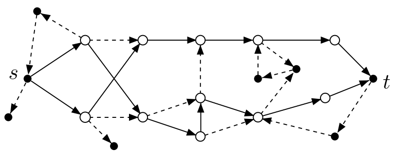


(b) $\text{FLOW}_V(G, c, \mathcal{Q}, f)$, most transitive arcs are omitted.

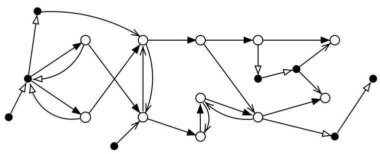


(c) $\text{PO}(\text{FLOW}_V(G, c, \mathcal{Q}, f), s, t)$, most transitive arcs are omitted.

Fig. 5. A flow $\mathcal{Q}, f$ in G, the resulting graph $\text{FLOW}_V(G, c, \mathcal{Q}, f)$ and partial order.

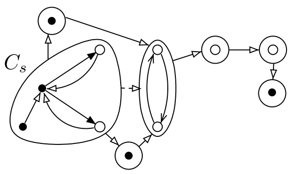


(a) An alternative maximum st-flow $\mathcal{Q}, f$.

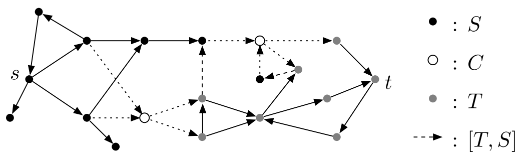


(b) $\text{FLOW}_V(G, c, \mathcal{Q}, f)$ (few transitive arcs).

Fig. 6. An alternative flow $\mathcal{Q}', f'$ in G, and the resulting graph $\text{FLOW}_V(G, c, \mathcal{Q}', f')$.



(a) An ideal of $\text{PO}_V(G, c, s, t)$.



(b) The corresponding cut partition (S, C, T).

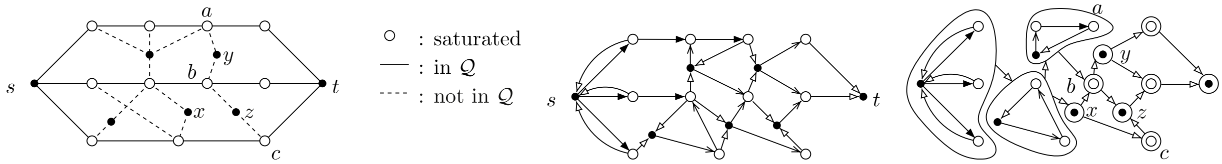
Fig. 7. An ideal and the corresponding cut partition.

To prove the other direction, consider an ideal I of G' with $s \in I$, $t \notin I$. Let $T = \bar{I}$. We construct S and C as follows. For every path $P \in \mathcal{Q}$, the last vertex of P that is part of I is included in C, and $S = I \backslash C$.

We now prove that $c(C) = |f|$. For this, we first show that every vertex in C is saturated. Consider a non-saturated vertex u that lies on a path $P \in \mathcal{Q}$. The next vertex v on P is non-saturated reachable from u, so $(v, u) \in A(G')$. This shows that u can never be the last vertex of P that is part of the ideal I. Next, we show that every path in $\mathcal{Q}$ contains exactly one vertex of C. Suppose this is not true for the path $Q_1 \in \mathcal{Q}$. Let v be the last I-vertex of Q_1, and let $u \in V(Q_1) - v$ be the last I-vertex of a different path $Q_2 \in \mathcal{Q}$. So u is a path predecessor of v. Let w be the next vertex of Q_2, so $w \notin I$. But w is non-saturated reachable from u in G, so by construction of G', $(w, v) \in A(G')$. This contradicts that I is an ideal of G'. Using the facts that every vertex in C is saturated and every path contains exactly one C-vertex, we obtain that $c(C) = |f|$ (Claim 11).

Since C contains only saturated vertices, $s \in S$ and $t \in T$. It only remains to show that in G, $[S, T] = \emptyset$. Suppose that $(u, v) \in [S, T]_G$. If u is not saturated, then $(v, u) \in A(G')$, a contradiction. If u is saturated, then since it is not part of C, it has a saturated path successor w that is in I. But then $(v, w) \in A(G')$, a contradiction. This concludes the proof that (S, C, T) is an st-cut partition. $\square$

Similar to Section 3, Combining Lemma 3 with Theorem 13 gives the following theorem (see Fig. 5(c)).



(a) An undirected graph G with unit vertex capacities, and an st-flow Q, f ($\forall P \in Q : f(P) = 1$).

(b) $\text{FLOW}_V(G, c, Q, f)$; few transitive arcs.

(c) $\text{PO}_V(G, c, s, t)$; no transitive arcs.

Fig. 8. $\text{FLOW}_V(G, c, \mathcal{Q}, f)$ and partial order for an undirected graph.

Theorem 14. *Let $\mathcal{Q}, f$ be a maximum st-flow in a digraph G with vertex capacities c, and let $\mathcal{P} = \text{PO}(\text{FLOW}_V(G, c, \mathcal{Q}, f), s, t)$. There exists an st-cut partition (S, C, T) of G with $S \cup C = X$ if and only if $\mathcal{P}$ has a non-trivial ideal $I \subset V(\mathcal{P})$ with $V(I) = X$.*

Also similar to before (see the text below Theorem 4), it follows that $\text{PO}(\text{FLOW}_V(G, c, \mathcal{Q}, f), s, t)$ is uniquely determined by the choice of G, c, s, t, and does not depend on the chosen flow, so we may also denote this graph by $\text{PO}_V(G, c, s, t)$ (the superscript V indicates that c are *vertex* capacities)[1].

Let $\mathcal{P} = \text{PO}_V(G, c, s, t)$, and let $k_{st} = c(C)$ for a minimum st-vertex cut C. C_s and C_t denote the vertices of $\mathcal{P}$ that contain s and t respectively. Assign weights w to vertices $C \in V(\mathcal{P})$ as follows: $w(C) = c(C)$ for all $C \neq C_t$, and $w(C_t) = c(C_t) + k_{st}$, so $w(V(\mathcal{P})) = c(V(G)) + k_{st}$. This yields a MBI instance $\mathcal{P}, w$. Any non-trivial ideal I of $\mathcal{P}$ contains C_s but not C_t (Lemma 3), so $\mathcal{P}$ has a non-trivial ideal I with $\min\{w(I), w(V(\mathcal{P})\backslash I)\} \geq k_{st} + x$ if and only if $w(I) \geq k_{st} + x$ and $c(V(G)) - w(I) \geq x$. By Theorem 14, this in turn holds if and only if G, c, s, t has an st-cut partition (S, C, T) with $c(S) \geq x$ and $c(T) \geq x$. Now the PTAS from Section 4 can be used for the equivalent MBI instance $\mathcal{P}, w$. We remark that $c(s)$ and $c(t)$ can be chosen arbitrarily since these values do not influence minimum st-cuts. In particular, this method works when $c(v) = 1$ for all vertices v, which approximates the unweighted variant of MBMVC-P. Note that every construction in this section can be done in polynomial time, and therefore we have a PTAS for MBMVC-P.

Theorem 15. *A PTAS exists for the problem of finding an st-cut partition (S, C, T) that maximizes $\min\{c(S), c(T)\}$ in a directed graph with vertex capacities c.*

7. Minimum vertex cuts with more than two components

In Section 9 we identified the partial order structure of minimum st-vertex cuts in directed graphs, and mapped all cuts C plus partitions of the remaining vertices into S and T to ideals of the partial order. Obviously these constructions also work for undirected graphs, after first replacing edges with arcs in both directions. In this section we will restrict ourselves to undirected graphs, and study in which situations a minimum st-vertex cut C may result in more than two components, hence the situations in which there is not a unique choice for choosing S and T, when C is given. Throughout this section we will often use Theorem 14, sometimes implicitly.

Definition 16. Let C be an st-vertex cut of an undirected graph 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.

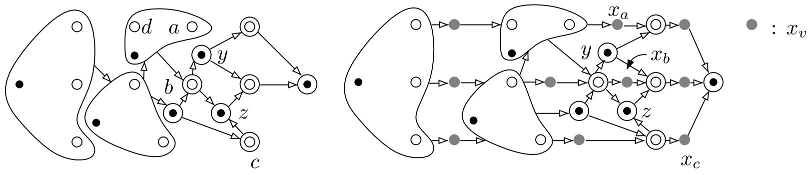
Definition 17. Let $\mathcal{Q}, f$ be a maximum st-flow in an undirected graph G with vertex capacities c. A set $C \in V(\text{PO}_V(G, c, s, t))$ is a *separable block* of $G, \mathcal{Q}, f$ if C contains no saturated vertices of the flow $\mathcal{Q}, f$ in G, and C does not contain s or t.

We remark that it will follow later that separable blocks actually do not contain any vertex used in a maximum st-flow. These definitions are illustrated in Fig. 8. This figure uses the same conventions as Fig. 5 (for instance regarding arc types). For clarity, most transitive arcs are omitted again. In this example the three separable blocks consist of the single vertices x, y and z. Observe that a minimum st-vertex cut exists that has both y and z as extra component (this cut contains the vertices a, b and c), but x cannot occur as extra component together with y or z. Similar to what we observed for previous notions, it will turn out that the choice of the maximum flow does not matter for the characterization of separable blocks. Separable blocks are determined by the choice of G, c, s and t, which will follow from the next two lemmas. We will already anticipate this fact by speaking about the *separable blocks of* G, c, s, t, disregarding the chosen flow in this expression.

Lemma 18. *Let C be a minimum st-vertex cut of an undirected graph G with vertex capacities c. If H is an extra component of C, then $V(H)$ is a separable block of G, c, s, t.*

Proof. Let $B = V(H)$, and let $\mathcal{Q}, f$ be a maximum st-flow. Since H is an extra component, $s \notin B$ and $t \notin B$. Suppose B contains a saturated vertex v. Then there is an (s, t)-path $P \in \mathcal{Q}$ that contains v. C contains exactly one vertex of P (Claim 11).

[1] We remark that $\text{PO}_V(G, c, s, t)$ can be defined differently, by transforming the digraph G with vertex capacities c to a digraph G' with arc capacities c' in the following standard way [10]: replace vertices $v \in V(G)$ by a vertex pair v_1, v_2 and arc (v_1, v_2) with capacity $c(v)$, and replace arcs $(u, v) \in A(G)$ by arcs (u_2, v_1) with infinite capacity. $\text{PO}_V(G, c, s, t)$ can then be defined as a modification of $\text{PO}_A(G', c', s_2, t_1)$, which allows for a different proof of Theorem 14 using the results from Section 3. The approach we have chosen here is however necessary to prove the results in Section 7.



(a) $\text{PO}_V(G, c, s, t)$, without transitive arcs.

(b) $\text{PO}^*_V(G, c, s, t)$, without transitive arcs.

Fig. 9. Constructing $\text{PO}^*_V(G, c, s, t)$ from $\text{PO}_V(G, c, s, t)$.

It follows that in $G - C$, a path exists from v to s or from v to t, contradicting that H is an extra component. So B contains no saturated vertices.

By Theorem 14, there is an ideal I of $\mathcal{P} = \text{PO}_V(G, c, s, t)$ such that $V(I)$ is the union of C and the vertex set of the s-component of C, and there is an ideal I' with $V(I') = V(I) \cup B$. In addition, since H is connected and contains no saturated vertices, H is part of some component of $G - C'$ for any minimum st-vertex cut C'. Then Theorem 14 shows that every ideal J of $\mathcal{P}$ either has $B \in V(J)$, or $B \cap V(J) = \emptyset$. Combining these observations gives $B \in V(\mathcal{P})$, hence B is a separable block with respect to $\mathcal{Q}, f$. $\square$

Lemma 19. *Let G be an undirected graph with vertex capacities c, and let (S, C, T) be an st-cut partition of G with $S \cup C = V(I)$ for $I \subseteq V(\text{PO}_V(G, c, s, t))$. If B is a separable block and a maximum of I, then $G[B]$ is an extra component of C.*

Proof. Let $\mathcal{Q}, f$ be the maximum st-flow for which B is a separable block. Since B is a maximum of I, $I' = I - B$ is also an ideal of $\mathcal{P} = \text{PO}_V(G, c, s, t)$. Let (S', C', T') be the corresponding st-cut partition. By the second statement from Claim 12, C' (C) contains exactly those saturated vertices in $S' \cup C'$ $(S \cup C)$ of which no saturated path successor is contained in $S' \cup C'$ $(S \cup C)$. But the difference B contains no saturated vertices of $\mathcal{Q}, f$ since it is a separable block, so $C' = C$. This shows that both (S, C, T) and (S', C, T') are st-cut partitions for the same cut, hence B must induce an extra component of the cut C. (Note that $G[B]$ is connected: if not, an st-cut partition can be constructed that corresponds to an ideal I of $\mathcal{P}$ where $V(I)$ only contains a strict subset of B, contradicting $B \in V(\mathcal{P})$.) $\square$

From these two lemmas it follows that the separable blocks are independent of the choice of maximum st-flow: if B is a separable block with respect to one flow, it induces an extra component of some cut by Lemma 19, and then by Lemma 18, B is a separable block for any maximum st-flow.

We now construct a weighted partial order $\mathcal{P}'$ as follows, see Fig. 9. This example continues on the example from Fig. 8, and uses the same st-flow for the construction. Let $\mathcal{P} = \text{PO}_V(G, c, s, t)$, and start with $\mathcal{P}' = \mathcal{P}$. Fix a maximum st-flow $\mathcal{Q}, f$. For every saturated vertex $v \in V(G)$ such that there exists a $B_v \in V(\mathcal{P})$ with $v \in B_v$ but where B_v contains no path successors of v[2], add an element x_v to $V(\mathcal{P}')$, and the arc (B_v, x_v). In addition, add arcs (x_v, B) for every $B \in V(\mathcal{P})$ that contains a path successor of v. Add all transitive arcs. In Fig. 9, note that we add a vertex x_a but not a vertex x_d because a is a path successor of d in the chosen flow. Observe that $\mathcal{P}'$ is still a partial order: if the arcs (B_v, x_v) and (x_v, B) are added, then B_v contains v, which is a path predecessor of some vertex in B, so $(B_v, B) \in A(\mathcal{P})$. This shows that adding the element x_v and these incident arcs cannot introduce cycles. Observe also that this implies that all transitive arcs that are added are incident with the new elements x_v, so the subgraph of $\mathcal{P}'$ induced by $V(\mathcal{P})$ is exactly $\mathcal{P}$. Hence we have:

Claim 20. *The graph $\mathcal{P}'$ as constructed above from $\mathcal{P} = \text{PO}_V(G, c, s, t)$ is a partial order, and $\mathcal{P}'[V(\mathcal{P})] = \mathcal{P}$.*

In addition assign the following weights to vertices of $\mathcal{P}'$: For $B \in V(\mathcal{P})$, let $w(B) = |B|$ if B is a separable block of G, and $w(B) = 0$ otherwise. For all x_v, set $w(x_v) = 1$.

The goal of the next three lemmas is to show that this weighted partial order $\mathcal{P}', w$ has an antichain with weight x if and only if G has a minimum st-vertex cut C such that x is the sum of $|C|$ and the number of vertices in extra components of C. From this it will follow again that the choice of maximum flow does not matter, so we denote this partial order $\mathcal{P}'$ as $\text{PO}^*_V(G, c, s, t)$, omitting the chosen flow. Note that in Fig. 9(b) there is a unique maximum antichain, which contains x_a, x_b, x_c and the two blocks containing y and z, which corresponds to the cut containing vertices a, b and c, with extra components y and z (see Fig. 8).

Lemma 21. *Let X be an antichain in $\text{PO}^*_V(G, c, s, t)$. Then G has a minimum st-vertex cut C such that*

- *for every $x_v \in X$, $v \in C$, and*
- *for every separable block $B \in X$, $G[B]$ is an extra component of C.*

Proof. Consider a minimum ideal I' of $\mathcal{P}' = \text{PO}^*_V(G, c, s, t)$ containing X, so every element in X is a maximum of I'. Then $I = I' \cap V(\mathcal{P})$ is an ideal of $\mathcal{P}$ (Claim 20). So G has an st-cut partition (S, C, T) with $S \cup C = V(I)$ (Theorem 14). Every

2 Note that these are exactly the vertices that are in some minimum st-vertex cut (Theorem 14, Claim 12).

separable block in X is a maximum of I', and thus of I, and therefore an extra component of C (Lemma 19). For every $x_v \in X$, by construction of $\mathcal{P}'$ it holds that I' contains no block B that contains a path successor of v, since x_v is a maximum of I'. But I' does contain the block $B_v \in V(\mathcal{P})$ with $v \in B_v$, because I' is an ideal. Hence v is a saturated vertex in $S \cup C$ with no saturated path successors in $S \cup C$, and thus $v \in C$ (Claim 12). $\square$

Next we will show that the converse of Lemma 21 also holds, but for this we first need the following lemma.

Lemma 22. *Let $\mathcal{P}' = \mathrm{PO}_V^*(G, c, s, t)$, and let (S, C, T) be an st-cut partition of G where I is the ideal of $\mathcal{P} = \mathrm{PO}_V(G, c, s, t)$ with $V(I) = S \cup C$. Then $I' \subset V(\mathcal{P}')$ with $I' \cap V(\mathcal{P}) = I$ is an ideal of $\mathcal{P}'$ if*

- *I' contains all x_v with $v \in S$, and*
- *I' contains no x_v with $v \in T$.*

Proof. We show that a set I' with the above properties is an ideal of $\mathcal{P}'$. If not, then w.l.o.g. there is a non-transitive arc $(u, v) \in A(\mathcal{P}')$ with $u \notin I'$, $v \in I'$. Since $I' \cap V(\mathcal{P})$ is an ideal of $\mathcal{P}$, w.l.o.g. this arc is of one of the two non-transitive arc types that were added in the construction of $\mathcal{P}'$. We consider these two types, to show that such an arc (u, v) cannot exist.

Consider an arc (B_v, x_v), with $v \in B_v \in V(\mathcal{P})$. Since $x_v \in I'$, we have $v \notin T$, so $v \in S \cup C$. This means that $B_v \in I$, so $B_v \in I'$, which concludes the proof of this case.

Now consider an arc (x_v, B) where $B \in V(\mathcal{P})$ contains a path successor w of v (with respect to the flow used to construct $\mathcal{P}'$). If $B \in I'$, then $w \in S \cup C$. So its path predecessor v is part of S (Claim 12). This gives $x_v \in I'$, and thus a problematic arc can also not be of this form, which rules out all cases. $\square$

Lemma 23. *Let C be a minimum st-vertex cut of G. Then $\mathrm{PO}_V^*(G, c, s, t)$ has an antichain X such that*

- *for all $v \in C$, $x_v \in X$, and*
- *for every extra component $G[B]$ of $G - C$, $B \in X$.*

Proof. Let $\mathcal{P}' = \mathrm{PO}_V^*(G, c, s, t)$. We prove that a set $X \subseteq V(\mathcal{P}')$ that contains the elements stated above is an antichain of $\mathcal{P}'$. Let S be the vertex set of the s-component of C. $\mathcal{P} = \mathrm{PO}_V(G, c, s, t)$ has an ideal I with $V(I) = C \cup S$ (Theorem 14). Adding x_v for all $v \in S$ gives an ideal I' of $\mathcal{P}'$ (Lemma 22). Note that this ideal I' contains no elements of X.

Consider an extra component $G[B]$ of the cut C, and let $I'' = I' + B$. $I'' \cap V(\mathcal{P})$ is again an ideal of $\mathcal{P}$ (Theorem 14). Since B is a separable block (Lemma 18), it contains no saturated vertices, so it contains no vertex v with $x_v \in V(\mathcal{P}')$. Then I'' is again an ideal of $\mathcal{P}'$ (Lemma 22). Hence this is an ideal of $\mathcal{P}'$ that contains B but no other elements of X.

Similarly, for any $v \in C$, $\mathcal{P}$ has an ideal that contains x_v, that contains no other elements of X. This ideal is simply $I'' = I' + x_v$, which is again an ideal of $\mathcal{P}'$ by Lemma 22.

In conclusion, we can add all elements of X independently to I' while maintaining an ideal, so they are all pairwise incomparable, and X is an antichain. $\square$

The above constructions are combined in Algorithm 2 for a 2-approximation algorithm for MBMVC-LC.

Algorithm 2 A 2-approximation for MBMVC-LC

INPUT: An undirected graph G with vertex capacities c and distinct vertices s, t.

1. Construct the partial order $\mathcal{P}' = \mathrm{PO}_V^*(G, c, s, t)$, with weights $w(x_v) = 1$ for all x_v, $w(B) = |B|$ for all separable blocks, and $w(B) = 0$ for all other blocks.
2. Find a maximum weight antichain X of $\mathcal{P}'$, with respect to the weights w.
3. Output a minimum st-cut C that has $G[B]$ as extra component for all separable blocks $B \in X$, and such that $v \in C$ for all $x_v \in X$.

Theorem 24. *Algorithm 2 is a 2-approximation algorithm for MBMVC-LC.*

Proof. We first show that all steps have a polynomial time implementation. In Step 1, constructing $\mathcal{P}' = \mathrm{PO}_V^*(G, c, s, t)$ and identifying the separable blocks of G, c, s, t can be done using an arbitrary maximum st-flow as shown above, and such a flow can be found in polynomial time. For Step 2, a maximum weight antichain X of a partial order can be found in polynomial time [8]. Lemma 21 then shows that G has a minimum st-vertex cut as stated in Step 3, and the proof shows that this cut can easily be found in polynomial time.

We now show that the returned solution is a 2-approximation. Let $n = |V(G)|$. Let a minimum st-vertex cut C_O of G be an optimal solution for MBMVC-LC, and let C_A be the minimum st-vertex cut given by the algorithm. W_O and W_A respectively denote the total number of vertices in extra components of C_O and C_A. By choice of C_A, $|C_A| + W_A \geq |C_O| + W_O$, since there is an antichain in $\mathcal{P}'$ with weight $|C_O| + W_O$ (Lemma 23).

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

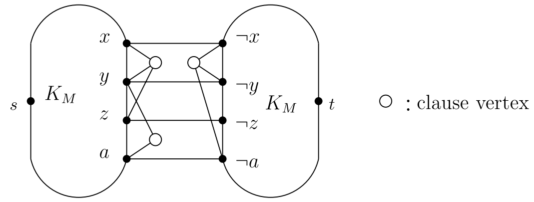


Fig. 10. An MBMVC-SC instance corresponding to $(\neg x \vee \neg y \vee \neg z) \wedge (\neg y \vee \neg a) \wedge (x \vee y \vee a)$.

Now suppose the largest component L of $G - C_A$ is an extra component. By Lemma 18, $L = G[B]$ for some separable block $B \in V(\mathcal{P})$. Since $G[B]$ is connected, C_O contains only saturated vertices (Claim 11) and B does not contain any, it follows that B is part of a component of $G - C_O$. Therefore the optimal cut C_O has a component that is just as large as L, in which case C_A is optimal. □

We remark that in the above proof, the new partial order $\text{PO}_V^*(G, c, s, t)$ is only needed when vertex capacities are non-uniform; in the uniform case every minimum st-vertex cut contains the same number of vertices, so proofs similar to those above show that it suffices to find a maximum antichain in $\text{PO}_V(G, c, s, t)$, where only separable blocks are assigned non-zero weights equal to their capacity sums.

Secondly we remark that the algorithm, lemmas and definitions above also apply to directed graphs. However in the directed case, the objective value that is maximized is not so natural, so we preferred to treat only undirected graphs.

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 in polynomial time (unless P=NP).

Theorem 25. *No $1/(\alpha n)$-approximation algorithm with $\alpha < 1/2$ exists for MBMVC-SC 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 [11]. 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 on n vertices 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 exactly two components, which both contain more than αn 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.

The transformation is illustrated in Fig. 10. 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 results on most balanced minimum cut problems, which have received little study until now. We considered edge cuts and three natural vertex cut variants. 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. In the edge cut case, the partial order structure we used is similar to the one described in [16]. In the vertex cut case, the partial order representation of all minimum *st*-cuts is new. 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. We gave a PTAS for a special type of UPOK. How well can this problem be approximated in general? Are there other special cases of UPOK/POK that can be solved/approximated efficiently?
3. In the introduction we remarked that MBMC can be solved in polynomial time for undirected graphs if the ratio between the capacity of a minimum *st*-cut and the capacity of a global minimum cut is bounded by a constant c. Related to this, in Section 4 we remarked that MBI and UPOK can be solved in polynomial time if the width of the partial order is bounded by a constant c. However in both cases the parameter c appears in the exponent of the complexity. Is it possible to give a polynomial time algorithm for these problems without this property, i.e. a fixed parameter tractable (FPT) algorithm for parameter c? This is closely related to the result in [6], where an FPT algorithm is given in the case where the number of edges in a minimum *st*-cut is the parameter.
4. MBMC and MBMVC-P 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?

Acknowledgements

The author thanks the anonymous referees for their many helpful suggestions. This work was 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.

References

[1] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, D. Sivakumar, On the hardness of approximating multicut and sparsest-cut, Comput. Complexity 15 (2) (2006) 94–114.
[2] M. Chimani, C. Gutwenger, P. Mutzel, On the minimum cut of planarizations, Tech. Rep. TR06-1-003, University of Dortmund, Germany, 2005.
[3] J. Chuzhoy, S. Khanna, Hardness of cut problems in directed graphs, in: Proceedings of the 38th STOC, ACM, New York, 2006, pp. 527–536.
[4] R. Diestel, Graph Theory, 3rd edition, in: Graduate Texts in Mathematics, vol. 173, Springer-Verlag, Berlin, 2005.
[5] E. Dinits, A. Karzanov, M. Lomonosov, On the structure of a family of minimal weighted cuts in a graph, in: Studies in Discrete Optimization, Nauka, Moscow, 1976, pp. 290–306 (in Russian).
[6] U. Feige, M. Mahdian, Finding small balanced separators, in: Proceedings of the 38th STOC, ACM, New York, 2006, pp. 375–384.
[7] U. Feige, O. Yahalom, On the complexity of finding balanced oneway cuts, Inf. Process. Lett. 87 (1) (2003) 1–5.
[8] S. Felsner, V. Raghavan, J. Spinrad, Recognition algorithms for orders of small width and graphs of small Dilworth number, Order 20 (4) (2003) 351–364. 2004.
[9] L. Fleischer, Building chain and cactus representations of all minimum cuts from Hao-Orlin in the same asymptotic run time, J. Algorithms 33 (1) (1999) 51–72.
[10] L.R. Ford Jr., D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.
[11] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, San Francisco, 1979.
[12] D.S. Johnson, K.A. Niemi, On knapsacks, partitions, and a new dynamic programming technique for trees, Math. Oper. Res. 8 (1) (1983) 1–14.
[13] D.R. Karger, Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm, in: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (Austin, TX, 1993), ACM, New York, 1993, pp. 21–30.
[14] S.G. Kolliopoulos, G. Steiner, Partially ordered knapsack and applications to scheduling, Discrete Appl. Math. 155 (8) (2007) 889–897.
[15] T. Leighton, S. Rao, Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms, J. ACM 46 (6) (1999) 787–832.
[16] J. Picard, M. Queyranne, On the structure of all minimum cuts in a network and applications, in: Combinatorial Optimization, II (Proc. Conf., Univ. East Anglia, Norwich, 1979), Math. Program. Stud. 13 (1980) 8–16.
[17] D.B. Shmoys, Cut problems and their application to divide-and-conquer, in: D. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishing Company, Boston, 1997, pp. 192–235.
[18] V.V. Vazirani, M. Yannakakis, Suboptimal cuts: Their enumeration, weight and number (extended abstract), in: ICALP '92: Proceedings of the 19th International Colloquium on Automata, Languages and Programming, Springer-Verlag, London, UK, 1992, pp. 366–377.