Fixed-parameter tractability of counting small minimum \((S,T)\)-cuts

Pierre Bergé\(^1\), Benjamin Mouscadet\(^2\), Arpad Rimmel\(^2\) and Joanna Tomasik\(^2\)

\(^1\) LRI, Université Paris-Sud, Université Paris-Saclay, Orsay France
\quad Pierre.Berge@lri.fr
\(^2\) LRI, CentraleSupélec, Université Paris-Saclay, Orsay France,
\quad Arpad.Rimmel@lri.fr, Benjamin.Mouscadet@supelec.fr, Joanna.Tomasik@lri.fr

Abstract. The parameterized complexity of counting minimum cuts stands as a natural question because Ball and Provan showed its \#P-completeness. For any undirected graph \(G = (V, E)\) and two disjoint sets of its vertices \(S, T\), we design a fixed-parameter tractable algorithm which counts minimum edge \((S,T)\)-cuts parameterized by their size \(p\). Our algorithm operates on a transformed graph instance. This transformation, called drainage, reveals a collection of at most \(n = |V|\) successive minimum \((S,T)\)-cuts \(Z_i\). We prove that any minimum \((S,T)\)-cut \(X\) contains edges of at least one cut \(Z_i\). This observation, together with Menger's theorem, allows us to build the algorithm counting all minimum \((S,T)\)-cuts with running time \(2^{O(p^2)} n^{O(1)}\). Initially dedicated to counting minimum cuts, it can be modified to obtain an FPT sampling of minimum edge \((S,T)\)-cuts.

1 Introduction

The issue of counting minimum cuts in graphs has been drawing attention over the years due to its practical applications. Indeed, the number of minimum cuts is an important factor for the network reliability analysis \([2,3,4,24]\). Thereby, the probability that a stochastic graph is connected may be computed \([3]\). Furthermore, cuts on planar graphs are used for image segmentation \([8]\). An image is seen as a planar graph where vertices represent pixels and edges connect two neighboring pixels if they are similar. Counting minimum cuts provides an estimation of the number of segmentations.

We focus on the problem of counting minimum edge \((S,T)\)-cuts in undirected graphs \(G = (V, E)\), \(S, T \subseteq V\). We call it COUNTING MIN CUTS (Def. 11) as it is the counting variant of the classical problem MIN CUT, which asks for a minimum \((S,T)\)-cut in graph \(G\). Ball and Provan showed in \([26]\) that COUNTING MIN CUTS is unlikely solvable in polynomial time as it is \#P-complete. They also devised a polynomial-time algorithm for COUNTING MIN CUTS on planar graphs \([3]\). Bezáková and Friedlander \([1]\) generalized it with an \(O(n\mu + n \log n)\)-time algorithm on weighted planar graphs, where \(\mu\) is the length of the shortest \((s,t)\)-paths. For general graphs, some upper bounds on the number of minimum cuts have been given \([10]\) in function of parameters such as the radius, the
maximum degree, etc. Two fixed-parameter tractable (FPT) algorithms have been proposed for counting mincuts. Bezáková et al. [5] built an algorithm for both directed and undirected graphs with small treewidth $\lambda$; its time complexity is $O(2^{3\lambda}n\lambda)$. Moreover, Chambers et al. [3] designed an algorithm for directed graphs embedded on orientable surfaces of genus $g$: its execution time is $O(2^{g}n^{2})$. We study the fixed-parameter tractability of counting mincuts, parameterized by the size $p$ of the minimum $(S,T)$-cuts.

**Definition 1 (Counting mincuts).**

**Input:** Undirected graph $G = (V, E)$, sets of vertices $S, T \subseteq V$, $S \cap T = \emptyset$.

**Output:** The number of minimum edge $(S,T)$-cuts.

The minimum $(S,T)$-cut size for a counting mincuts instance $\mathcal{I} = (G, S, T)$ is obtained in polynomial time $[17]$. A brute force XP algorithm computes the number $C(\mathcal{I})$ of minimum $(S,T)$-cuts in time $n^{O(p)}$ by enumerating all edge sets of size $p$ and picking up those which are $(S,T)$-cuts. More efficient exponential algorithms exist, as the one of Nagamochi et al., in time $O(pm^{2} + pnC(\mathcal{I}))$, in $[24]$. Our contribution, summarized in the theorem below, is an algorithm efficient for small values of $p$.

**Theorem 1.** The counting of minimum edge $(S,T)$-cuts can be solved in time $O(2^{(p+2)pmn^{3}})$ on undirected graphs $G = (V,E)$, where $n = |V|$ and $m = |E|$.

An FPT($p$) algorithm can be deduced from the results in two articles $[5,21]$ and its execution time is $O^{*}(2^{H(p)})$ where $H(p) = \Omega\left(\frac{2^{p}}{p^{2}}\right)$. The treewidth reduction theorem established by Marx et al. in $[21]$ says that there is a linear-time reduction transforming graph $G$ into another graph $G'$ which conserves the $(s,t)$-cuts of size $p$ and such that the treewidth of $G'$, $\tau(G')$, verifies $\tau(G') = 2^{O(p)}$. After this transformation, the number of minimum $(S,T)$-cuts of $G'$ is obtained thanks to the algorithm given in $[5]$. The overall time taken with this method is $O^{*}(2^{2^{p}})$. Our result, Theorem [1] improves this exponential factor.

This result highlights a complexity gap between the counting and the enumeration, as the latter cannot be FPT parameterized by $p$. Indeed, certain instances contain a number $C(\mathcal{I}) = (\frac{n-1}{p})^{p}$ of minimum cuts, as in case of graph $G$ made of $p$ vertex-disjoint $(S,T)$-paths with $S = \{s\}$ and $T = \{t\}$.

Our algorithm is based on a cut-decomposition $\mathcal{Z}(\mathcal{I}) = (Z_{1}, \ldots, Z_{k})$ of instance $\mathcal{I}$, $1 \leq k < n$, called the **drainage** where for every $1 \leq i \leq k$, edge set $Z_{i}$ is a minimum $(S,T)$-cut. Set $R(Z_{i}, S)$ denotes the vertices which are reachable from $S$ after the removal of edges in $Z_{i}$. The reachable sets of $Z_{i}$ are included one into another: $R(Z_{1}, S) \subsetneq R(Z_{2}, S) \subsetneq \ldots \subsetneq R(Z_{k}, S)$. The drainage fulfills the following property: if $X$ is a minimum $(S,T)$-cut, some edges $B_{i}$ of a certain $Z_{i}$ belong to $X$, $B_{i} = X \cap Z_{i} \neq \emptyset$, and no other edge of $X$ has one endpoint in $R(Z_{i}, S)$. The set $B_{i}$ is called the **front dam** of cut $X$. The key idea of the recursive counting we propose is that any minimum cut $X$ is the union of its front dam with a minimum cut of a sub-instance, called **dry instance**, of $\mathcal{I}$. These techniques work as well on multigraphs, i.e. on graphs with multiple edges. After modifications, our algorithm also samples minimum edge $(S,T)$-cuts.
To design the drainage $Z(I)$, we use the concept of important cuts [20] which is the key ingredient of many FPT algorithms to solve cuts problems [7,12,14,20,22]. An $(S,T)$-cut $Y$ is important if there is no other $(S,T)$-cut $Y'$ such that $|Y'| \leq |Y|$ and $R(Y,S) \subset R(Y',S)$. There is a unique minimum important $(S,T)$-cut and it can be identified in polynomial time [20].

The second concept used in our algorithm is Menger’s theorem [23]. It states that the size of minimum edge $(S,T)$-cuts in an undirected graph is equal to the maximum number of edge-disjoint $(S,T)$-paths. As the max-flow min-cut theorem [17] generalizes Menger’s theorem, one of the largest sets of edge-disjoint $(S,T)$-paths is found in polynomial time.

To close this introductory chapter, we give a “table of contents” of our article. Section 2 introduces the notations used. Section 3 explains the construction of the drainage $Z(I) = (Z_1, \ldots, Z_k)$. In Section 4, we propose our algorithm and compute its time complexity. Finally, we conclude and give ideas about future research.

2 Definitions and notation

We summarize basic concepts of parameterized and counting complexity but also introduce the notation we will use.

**Fixed-parameter tractability.** NP-hard problems are unlikely to be solvable with polynomial time algorithms. However, solving them efficiently may become possible when parameters are associated to problem instances and the values of these parameters are small.

Referring to Downey and Fellows [15] and Niedermeier [25], a parameterized problem is said fixed-parameter tractable (FPT) if there is an algorithm solving it in time $O(f(p)P(n)) = O^*(f(p))$, where $p$ is a parameter, $n$ is the instance size, $P$ is a polynomial function, and $f$ is an arbitrary computable function. As a problem may be studied for different parameters $p_1, p_2, \ldots$, the notation “FPT” becomes ambiguous. If there is an algorithm solving a problem in time $O(f(p_1)P(n))$, then it is FPT($p_1$). In this study, the parameter $p$ of COUNTOING MINCUTS is the size of the minimum $(S,T)$-cut.

**Counting problems.** The study of #P complexity class and the counting problems it contains, started with Valiant [27]. Class #P is the set of counting problems such that their decision version is in class NP. The subclass #P-complete contains counting problems such that all problems in #P can be reduced to them with a polynomial-time counting reduction. No #P-complete problem can be solved in polynomial time unless $P=NP$. Moreover, there are decision problems such as CNF-2SAT [19] which are solvable in polynomial time but their associated counting problem is #P-complete [27]. The complexity of counting problems has been extended via the parameterized complexity framework [13,16]. A relevant question to ask about a #P-complete problem is whether there is an FPT algorithm counting all its solutions. For example, with $G$ and $H$ as input, FPT algorithms counting the number of occurrences of $H$ as a subgraph of $G$ have been intensively studied [11,18,28].
Cuts in undirected graphs. We study undirected graphs $G = (V, E)$, where $n = |V|$ and $m = |E|$. For any set of vertices $U \subseteq V$, we denote by $E[U]$ the set of edges of $G$ with two endpoints in $U$ and $G[U]$ the subgraph of $G$ induced by $U$: $G[U] = (U, E[U])$. Notation $G \setminus U$ refers to the graph deprived of vertices in $U$. For any set of edges $E' \subseteq E$, the graph $G$ deprived of edges $E'$ is denoted by $G \setminus E'$:

$$G \setminus U = G[V \setminus U] \quad \text{and} \quad G \setminus E' = (V, E \setminus E').$$

A path is a sequence of pairwise different vertices $v_1 \cdot v_2 \cdot v_3 \cdots v_i \cdot v_{i+1} \cdots$, where two successive vertices $(v_i, v_{i+1})$ are adjacent in $G$. To improve readability, we abuse notations: $v_1 \in Q$ and $(v_1, v_2) \in Q$ mean that vertex $v_1$ and edge $(v_1, v_2)$ are on path $Q$, respectively.

Cut problems usually consist in finding the smallest set of edges $X \subseteq E$ which splits the graph $G \setminus X$ into connected components meeting certain criteria. Given two sets of vertices $S$ (sources) and $T$ (targets), set $X \subseteq E$ is an $(S, T)$-cut if there is no path connecting a vertex from $S$ with a vertex from $T$ in $G \setminus X$. An $(S, T)$-cut $X$ is said to be minimum if there is no $(S, T)$-cut $X'$ such that $|X'| < |X|$. For any $(S, T)$-cut $X$, its source side $R(X, S)$ is the set of vertices that are reachable from $S$ in $G \setminus X$. Its target side $R(X, T)$ contains the vertices reachable from $T$ in $G \setminus X$. We define two sets $V^S(X)$ and $V^T(X)$:

- set $V^S(X) = \{u \in R(X, S); (u, v) \in X\}$, i.e. the vertices of $R(X, S)$ incident to cut $X$,
- set $V^T(X) = \{u \in R(X, T); (u, v) \in X\}$, i.e. the vertices of $R(X, T)$ incident to cut $X$.

Important and closest cuts. As defined in [20], an $(S, T)$-cut $X$ is important if there is no other $(S, T)$-cut $X'$ such that $|X'| \leq |X|$ and $R(X, S) \subseteq R(X', S)$. Intuitively, an important $(S, T)$-cut is such that there is no other cut smaller in size which is closer to $T$. The number of important $(S, T)$-cuts of size at most $p$ depends only on $p$ [11] and there is no more than one minimum important $(S, T)$-cut [20]. Although the proofs in [20] handle vertex cuts, an edge-to-vertex reduction preserves these properties on edge cuts [12].

Lemma 1 (Unicity of minimum important cuts [20]). For disjoint sets of vertices $S$ and $T$, there is a unique minimum important $(S, T)$-cut and it may be found in polynomial time.

On undirected graphs, we say that an important $(T, S)$-cut is a closest $(S, T)$-cut. Fig. 11 gives an example of graph $G$ with two $(S, T)$-cuts $X_1$ and $X_2$, where $S = \{s_1, s_2\}$ and $T = \{t\}$. Cut $X_1$ is not closest as the edges incident to $S$ form a cut $Z_1$ smaller than $X_1$ and $R(Z_1, S) \subseteq R(X_1, S)$. Cut $X_2$ is closest because there is no cut with at most three edges whose reachable set of vertices is contained in $R(X_2, S)$.

Definition 2. An $(S, T)$-cut $X$ is closest if there is no other $(S, T)$-cut $X'$ such that $|X'| \leq |X|$ and $R(X', S) \subseteq R(X, S)$. 
Fixed-parameter tractability of counting small minimum \((S,T)\)-cuts

As a minimum closest \((S,T)\)-cut is also a minimum important \((T,S)\)-cut on undirected graphs, there is a unique minimum closest \((S,T)\)-cut according to Lemma 1. Since the graph is uncapacitated, computing the minimum closest \((S,T)\)-cut is made in time \(O(mp)\), using \(p\) iterations of Ford-Fulkerson’s algorithm [17].

3 Framework: drainage and Menger’s paths

We introduce tools needed to design an algorithm solving counting mincuts in FPT\((p)\) time, where \(p\) is the size of any minimum \((S,T)\)-cut. We build the drainage, a collection of minimum cuts \(Z_i, i \in \{1, \ldots, k\}\), where \(k < n\), such that at least one edge of any minimum \((S,T)\)-cut \(X\) belongs to \(\bigcup_{i=1}^{k} Z_i\). Then, we highlight properties coming from Menger’s theorem.

3.1 Construction of the drainage

The drainage \(Z(I) = (Z_1, \ldots, Z_k)\) of an instance \(I = (G, S, T)\) is a collection of minimum \((S,T)\)-cuts \(Z_i, |Z_i| = p\), satisfying the following properties:

- there are less than \(n\) cuts \(Z_i\), i.e. \(1 \leq k < n\),
- the reachable sets of cuts \(Z_i\) fulfil \(R(Z_i, S) \subseteq R(Z_{i+1}, S)\) for \(i \in \{1, \ldots, k-1\}\),
- for any minimum \((S,T)\)-cut \(X\), there is at least one cut \(Z_i\) which has edges with \(X\) in common: \(X \cap Z_i \neq \emptyset\).

We construct the drainage iteratively. Let \(S_1 = S\) and \(Z_1\) be the minimum closest \((S_1,T)\)-cut. We fix \(R_1 = R(Z_1, S)\). Let \(S_2\) be the set of vertices incident to edges of \(Z_1\) inside \(R(Z_1, T)\): \(S_2 = V^T(Z_1) = \{v \notin R_1, (u,v) \in Z_1\}\).

Next, we construct \(Z_2\) which is the minimum closest \((S_2,T)\)-cut in \(G \setminus R(Z_1, S)\). If \(|Z_2| > p\), the drainage construction stops. Otherwise, if \(|Z_2| = p\), set \(R_2\) follows the same scheme as \(R_1\): \(R_2 = R(Z_2, S_2)\) in graph \(G \setminus R(Z_1, S)\). We repeat the process until no more minimum \((S_i,T)\)-cut \(Z_i\) of size \(p\) can be found. We denote by \(k\) the number of cuts \(Z_i\) produced and fix \(R_{k+1} = R(Z_k, T)\). Cuts \(Z_i\) form the minimum drainage cuts of \(I\).

Fig. 2 provides us with an example of graph \(G\) with \(S = \{s_1, s_2, s_3\}\) and \(T = \{t_1, t_2\}\) and indicates its drainage. The size of minimum \((S,T)\)-cuts is \(p = 4\). Blue, red, and green edges represent minimum drainage cuts \(Z_1, Z_2,\) and...
We study the case where no edge of \( X \) belongs to \( \emptyset \). So, some edges of \( X \) are incident to \( Z_3 \), respectively. Similarly, blue, red, green, and yellow vertices represent sets \( S_1 = S_2, S_3, \) and \( S_4 \). Reachable sets \( R_1, R_2, R_3, \) and \( R_4 \) are also appropriately colored. As the size of the minimum cut between \( S_4 \) (yellow vertices) and \( T \) in graph \( G \setminus R(Z_3, S) \) is greater than \( p \), we have \( k = 3 \).

We emphasize that set \( R_i \), which is \( R(Z_i, S_i) \) taken in \( G \setminus R(Z_{i-1}, S) \), and set \( R(Z_i, S) \) are different for \( i \neq 1 \). On the one hand, set \( R(Z_i, S) = \bigcup_{t=1}^{i} R_t \) contains the vertices reachable from \( S \) in graph \( G \) deprived of \( Z_i \). On the other hand, set \( R_i \) can be written \( R_i = R(Z_i, S) \setminus R(Z_{i-1}, S) \). Sets \( R_i \) and \( R_{i+1} \) are disjoint and nonempty, as \( S_i \subseteq R_i \) and \( S_{i+1} \subseteq R_{i+1} \). Moreover, the minimum drainage cuts are disjoint: \( Z_i \cap Z_j = \emptyset \). The number \( k \) of minimum drainage cuts is less than \( n \) and the running time needed to construct the drainage is in \( O(mnp) \). The reachable vertex sets of cuts \( Z_i \) are included one into another: \( R(Z_i, S) \subseteq R(Z_{i+1}, S) \). The following theorem shows that, for any minimum \((S, T)\)-cut \( X \), there is a cut \( Z_i \) containing edges of \( X \). Among cuts \( Z_i \) sharing edges with \( X \), we are interested in the one with the smallest index.

**Definition 3 (Front of \( X \)).** Front of \( X \), \( i(X) \), \( 1 \leq i(X) \leq k \) is the smallest index \( i \) such that \( Z_i \cap X \neq \emptyset \).

The next theorem states the properties of \( i(X) \) for any minimum \((S, T)\)-cut.

**Theorem 2.** Any minimum \((S, T)\)-cut \( X \) admits a front \( i(X) \) and \( X \cap E \left[ R(Z_{i(X)}, S) \right] = \emptyset \).

*Proof.* First, cut \( X \) cannot be entirely included in \( E[R_{k+1}] \). If it was, it would be a minimum \((S_{k+1}, T)\)-cut of size \( p \), which contradicts the drainage definition. So, some edges of \( X \) are incident to \( R(Z_k, S) \).

Second, no edge of \( X \) belongs to \( E[R(Z_1, S)] \) as cut \( Z_1 \) is the minimum closest \((S, T)\)-cut. Therefore, there is an index \( i \geq 1 \) such that no edge of \( X \) belongs to \( E[R(Z_i, S)] \) but at least one has an endpoint in \( R_{i+1} \).

Obviously, if an edge of \( X \) belongs to \( Z_i \), the theorem holds: \( i = i(X) \). We study the case where no edge of \( X \) belongs to \( Z_i \) and there is an edge \( e \)
of $X$, $e \in E[R_{i+1}]$. According to the definition of index $i$, no edge of $X$ belongs to $E[R(Z_i, S)]$, and therefore all edges of $X$ have to be on the target side $E[R(Z_i, T)]$. Therefore, $X$ is a minimum $(S_{i+1}, T)$-cut in graph $G \setminus R(Z_i, S)$. Then, the path $X$ is a minimum closest $(S_{i+1}, T)$-cut (and we fix $Z = X$) or the minimum closest $(S_{i+1}, T)$-cut $Z$ is different than $X$ and it fulfills $R(Z, S_{i+1}) \subseteq R(X, S_{i+1})$. Since there is an edge $e \in X \cap E[R_{i+1}]$, then one of its endpoints $v \in R_{i+1}$ belongs to $R(X, T)$. Consequently, $v \notin R(Z, S_{i+1})$. This brings a contradiction: cutset $Z_{i+1}$ is the unique minimum closest $(S_{i+1}, T)$-cut and $R(Z_{i+1}, S_{i+1}) = R_{i+1}$ As vertex $v$ can be reached from $S_{i+1}$ after the removal of $Z_{i+1}$ but not after the removal of $Z$, cuts $Z$ and $Z_{i+1}$ differ. Thus, cut $Z$ cannot be the minimum closest $(S_{i+1}, T)$-cut.

In summary, there is an index $i$ such that no edge of $X$ belongs to $E[R(Z_i, S)]$ and, moreover, $X \cap Z_i \neq \emptyset$. This means that there is no index $\ell < i$ such that $X \cap Z_\ell \neq \emptyset$. Consequently, index $i$ is the front of $X$: $i = i(X)$. \hfill \Box

The reader can verify that any minimum $(S, T)$-cut of $G$ contains some edges of at least one cut $Z_1, Z_2, \text{ or } Z_3$ in Fig. 2.

### 3.2 Menger’s paths

Menger’s theorem states that the size of the minimum edge $(S, T)$-cuts is equal to the maximum number of edge-disjoint $(S, T)$-paths [18]. One of these largest sets of edge-disjoint $(S, T)$-paths can be found in polynomial time using flow techniques [17]. We denote by $Q = \{Q_1, \ldots, Q_p\}$ such a set of $p$ edge-disjoint $(S, T)$-paths, taken arbitrarily. We call paths from $Q$ Menger’s paths, to distinguish them from other paths in graph $G$.

Set $Q$ is used to identify minimum $(S, T)$-cuts. It is fixed throughout the course of the proofs in this article. The observation that edges of all minimum $(S, T)$-cuts belong to the paths from $Q$ is formulated in:

**Lemma 2.** For any minimum $(S, T)$-cut $X$, each Menger’s path $Q_j$ contains one edge of $X$. If $Q_j = v_1^{(j)} \cdot v_2^{(j)} \cdots v_{i+1}^{(j)} \cdot \dotsc \cdot v_{\ell+1}^{(j)}$ and $(v_1^{(j)}, v_{\ell+1}^{(j)}) \in X$, then $v_1^{(j)} \in R(X, S)$ and $v_{\ell+1}^{(j)} \in R(X, T)$.

**Proof of Lemma 2** If a Menger’s path $Q_j$ did not contain any edge of cut $X$, then set $X$ would not be a $(S, T)$-cut. Similarly, if path $Q_j$ contained at least two edges of cut $X$, then another Menger’s path would not contain any edge of $X$ as $|Q| = |X|$.

We suppose that $(v_1^{(j)}, v_{\ell+1}^{(j)})$ is the edge of $X$ in path $Q_j$. To prove that $v_1^{(j)} \in R(X, S)$ and $v_{\ell+1}^{(j)} \in R(X, T)$, we admit the contrary. As $Q_j$ contains one edge of $X$, the segment $v_1^{(j)} \cdot v_2^{(j)} \cdots v_{\ell}^{(j)}$ is open in $G \setminus X$ between $v_1^{(j)} \in S$ and $v_{\ell+1}^{(j)} \in R(X, T)$. Therefore, $X$ is not an $(S, T)$-cut as at least one vertex of $R(X, S)$ is connected to $R(X, T)$, a contradiction. \hfill \Box

Its consequence is that each edge of a cut $Z_i$ belongs to a Menger’s path. Fig. 3 illustrates the Menger’s paths on the instance $(G, S, T)$ already introduced...
in Fig. 2 As the minimum \((S, T)\)-cut size is four, there are four edge-disjoint \((S, T)\)-paths distinguished with colors. Menger’s paths are naturally oriented from sources to targets.

4 Counting minimum edge \((S, T)\)-cuts in undirected graphs

We describe our algorithm which counts all minimum \((S, T)\)-cuts in an undirected graph \(G\). Based on the concepts introduced in Section 3, we prove not only that it achieves its objective but also that its time complexity is FPT.

4.1 Dams and dry areas

We begin by the definition of dams which are subsets of cuts \(Z_i\) of the drainage of \(G\).

**Definition 4 (Dam).** A dam \(B_i\) is a nonempty subset of a minimum drainage cut \(Z_i\), i.e. \(B_i \subseteq Z_i\), \(B_i \neq \emptyset\).

Thanks to this definition, Theorem 2 together with the concept of the front makes us observe that any minimum \((S, T)\)-cut \(X\) contains a front dam:

**Definition 5 (Front dam).** The front dam of a minimum \((S, T)\)-cut \(X\) is \(B_{i(X)} = X \cap Z_{i(X)}\).

We know that all edges in \(X \setminus B_{i(X)}\) belong to the target side of \(Z_{i(X)}\), \(E[R(Z_{i(X)}, T)]\), and the source side of \(Z_{i(X)}\) is empty, \(X \cap E[R(Z_{i(X)}, S)] = \emptyset\). If \(X \setminus B_{i(X)} = \emptyset\), then \(X = Z_{i(X)}\). A dam \(B_i\) is characterized by:

- its level, i.e. the index \(i\) of the cut \(Z_i\) it belongs to,
- its signature \(\sigma(B_i) = \{Q_j : B_i \cap Q_j \neq \emptyset\}\), i.e. the set of Menger’s paths passing through it.
Choking graph $G$ with dam $B_i(X)$ puts in evidence a subgraph which still connects $S$ and $T$ through $X \setminus B_i(X)$. Our idea is to dam a graph gradually in order to dry it completely.

The description of the method we devised to reach this goal requires a transformation of $G$ into $G_D$ which is actually $G$ with certain edges directed ($G_D$ is a mixed graph). If edge $e$ does not belong to a Menger’s path, it stays undirected. For path $Q_j : v_{1}^{(j)} \cdot v_{2}^{(j)} \cdot v_{3}^{(j)} \cdots$, edges $(v_{i}^{(j)}, v_{i+1}^{(j)})$ become arcs $(v_{i}^{(j)}, v_{i+1}^{(j)})$, respecting the natural flow from sources to targets.

Fig. 3 illustrates graph $G_D$. Arrows indicate the arcs while bare segments represent its edges. According to Lemma 1, any minimum $(S, T)$-cut of $G$ is made up of arcs in $G_D$. Minimum drainage cuts $Z_i$ are thus composed of arcs, directed from $R_i$ to $R_{i+1}$. We insist on the fact that graph $G_D$ is only used to define the notion of dry area; we do not count minimum cuts in it.

**Definition 6 (Dry area).** The dry area of $B_i$ is the set $A^*(B_i)$ which contains the vertices of $G$ which are not reachable from $S$ in graph $G_D$ deprived of $B_i$, i.e. $G_D \setminus B_i$.

In a less formal way, set $A^*(B_i)$ keeps vertices which are dried as $B_i$ is the only means to irrigate them. The definition of the dry instance follows.

**Definition 7 (Dry instance).** The dry instance induced by a dam $B_i$ is an instance $D(I, B_i) = (G^*(B_i), S^*(B_i), T^*(B_i))$ with graph $G^*(B_i) = (V^*(B_i), E^*(B_i))$.

In particular,

- set $S^*(B_i)$ keeps vertices reachable from $S$ “just before” dam $B_i$. Formally, it contains the tails of arcs in $B_i$: $S^*(B_i) = \{u : (u, v) \in B_i\}$,
- set $T^*(B_i)$ keeps vertices placed “after” dam $B_i$ which become irrigated in $G_D \setminus B_i$. Formally, it contains the heads of arcs which have their tail either inside $S^*(B_i)$ or inside $A^*(B_i)$ and their head outside: $T^*(B_i) = \{v \notin A^*(B_i) : (u, v) \in E, u \in S^*(B_i) \cup A^*(B_i)\}$.

**Fig. 4.** An example of dam $B_2$ and its dry instance $D(I, B_2) = (G^*(B_2), S^*(B_2), T^*(B_2))$
Proof. Let \( h \) be a closest dam of \( i \), then it is its unique closest dam. Indeed, if a vertex is unreachable from \( S \) in \( G_D \setminus B_i \), then it also is unreachable from \( S \) in \( G_D \setminus B_h \) according to Def. 8.

\[ E^*(B_i) \setminus \bigcup_{j \in B_i} E^*(B_j) \]

**Lemma 3.** For any dam \( B_i \), there is no arc \((u, v)\) in \( G_D \) such that \( u \notin A^*(B_i) \) and \( v \notin A^*(B_i) \), except for arcs in \( B_i \). Moreover, there is no undirected edge with exactly one endpoint in \( A^*(B_i) \).

**Proof.** Suppose that such an arc \((u, v) \notin B_i \) exists. As \( u \notin A^*(B_i) \), it is reachable from \( S \) in \( G_D \setminus B_i \). Therefore, \( v \) can be reached too: this contradicts \( v \notin A^*(B_i) \). For the same reason, there is no undirected edge \((u, v)\) with only endpoint in \( A^*(B_i) \) as the existence of this edge makes both its endpoints be reachable from \( S \) in \( G_D \setminus B_i \).

In Theorem 3 we provide a characterization of any minimum \((S, T)\)-cut which is based on dry instances and on closest dams. We start by:

**Definition 8.** A dam \( B_h \) is closer than dam \( B_i \) if (i) \( h < i \), (ii) \( \sigma(B_h) = \sigma(B_i) \), and (iii) edges in \( B_i \) are the only edges of level \( i \) inside the dry instance of \( B_h \): \( E^*(B_h) \cap Z_i = B_i \).

As a consequence, the dry area of \( B_i \) is included in the dry area of \( B_h \) when \( B_h \) is closer than \( B_i \): \( A^*(B_i) \subseteq A^*(B_h) \). Indeed, if a vertex is unreachable from \( S \) in \( G_D \setminus B_i \), then it also is unreachable from \( S \) in \( G_D \setminus B_h \) as arcs of \( B_i \) cannot be attained from \( S \) in \( G_D \setminus B_h \) according to Def. 8.

**Definition 9 (Closest dam).** Dam \( B_i \) is a closest dam if no dam \( B_h, h < i \) is closer than \( B_i \).

For any dam \( B_i \), either \( B_i \) is a closest dam or there is a closest dam \( B_h \neq B_i \), closer than \( B_i \). Each dam \( B_i \) admits a closest dam (itself or \( B_h \)) which is unique.

**Lemma 4.** Any dam \( B_i \) has a unique closest dam.

**Proof.** If \( B_i \) is already a closest dam, then it is its own unique closest dam.

Now, we suppose that there are two closest dams of \( B_i \), denoted by \( B_{h_1} \) and \( B_{h_2} \). Necessarily, \( h_1 \neq h_2 \), otherwise \( B_{h_1} = B_{h_2} \) as, according to Def. 8 they have the same signature.
We prove that under the hypothesis \( h_1 < h_2 \), dam \( B_{h_1} \) is closer than \( B_{h_2} \). Suppose, towards a contradiction, that there is an arc \( e_2 = (u_2, v_2) \in E_{h_2} \) which does not belong to set \( E^*(B_{h_1}) \). As \( e_2 \notin E^*(B_{h_1}) \), vertex \( u_2 \) is not inside \( A^*(B_{h_1}) \), otherwise, according to Def. 3, \( e_2 \) would belong to \( E^*(B_{h_1}) \). So, there is a path \( \hat{Q} \) connecting sources from \( S \) with \( u_2 \), which avoids arcs in \( B_{h_1} \). Arc \( e_2 \) belongs to a Menger’s path \( Q_j \). A section of this path, denoted by \( \hat{Q}_j \), connects \( u_2 \) with a tail \( u_3 \) of an arc \( (u_3, v_3) \) in \( B_i \); because \( Q_j \) passes through \( B_i \); \( \sigma(B_i) = \sigma(B_{h_2}) \). The concatenated path \( \hat{Q} \cdot \hat{Q}_j \) connects \( S \) with \( B_i \) while avoiding \( B_{h_1} \); this is a contradiction as \( B_i \subseteq E^*(B_{h_1}) \). Consequently, \( B_{h_2} \subseteq E^*(B_{h_1}) \cap Z_{h_2} \).

The equality \( B_{h_2} = E^*(B_{h_1}) \cap Z_{h_2} \) comes from the fact that \( B_{h_1} \) and \( B_{h_2} \) have the same signature \( \sigma(B_i) \). Arcs in \( E^*(B_{h_1}) \cap Z_{h_2} \) belong to different Menger’s paths. If \( B_{h_2} \neq E^*(B_{h_1}) \cap Z_{h_2} \), then we have: \( |\sigma(B_{h_2})| < |\sigma(E^*(B_{h_1}) \cap Z_{h_2})| \leq |\sigma(B_{h_1})| \). Therefore, \( B_{h_2} = E^*(B_{h_1}) \cap Z_{h_2} \), so \( B_{h_1} \) is closer than \( B_{h_2} \) which is contradictory to our assumption that \( B_{h_2} \) is a closest dam. \( \square \)

Moreover, if \( B_h \) is a closest dam then its complement \( \overline{B}_h = Z_h \setminus B_h \) is also a closest dam. This property will be used to prove the fixed-parameter tractability of our algorithm.

**Lemma 5.** If \( B_h \) is a closest dam, then \( \overline{B}_h = Z_h \setminus B_h \) is also a closest dam.

**Proof.** Suppose that \( B_h \) is closest and \( \overline{B}_h \) is not; let \( \overline{B}_\alpha \) denote the closest dam of \( \overline{B}_h \), \( \alpha < h \).

First, we focus on the dry instance of \( \overline{B}_\alpha \) between levels \( \alpha \) and \( h \). We prove that no edge/arc \( (u, v) \), with the exception of arcs from dam \( \overline{B}_\alpha \), has one endpoint inside the dry instance of \( \overline{B}_\alpha \) before level \( h \) (i.e. in the source side of cut \( Z_h \)) and one outside. We distinguish two cases:

- **Case 1:** If arc \( (u, v) \) is such that \( u \notin A^*(\overline{B}_\alpha) \) and \( v \in A^*(\overline{B}_\alpha) \), then Lemma 3 brings the contradiction. This argument also holds when \( (u, v) \) is undirected.
- **Case 2:** If arc \( (u, v) \) is such that \( u \in A^*(\overline{B}_\alpha) \) is before level \( h \) and \( v \notin A^*(\overline{B}_\alpha) \), then a Menger’s path \( Q_j \) leaves the dry instance of \( \overline{B}_\alpha \) through this arc. However, dams \( \overline{B}_\alpha \) and \( \overline{B}_h \) have the same signature, so path \( Q_j \) also contains an arc of \( \overline{B}_h \) placed after arc \( (u, v) \). Consequently, there exists an arc \( (u', v') \), \( u' \notin A^*(\overline{B}_\alpha) \) and \( v' \in A^*(\overline{B}_\alpha) \), to make path \( Q_j \) go back inside \( D(I, \overline{B}_\alpha) \). This contradicts Case 1. Fig. 3 illustrates the explanations given in Case 2 on a graph \( G \) with dams \( B_h, \overline{B}_h, B_\alpha, \) and \( \overline{B}_\alpha \). In this example, vertices \( u' \) and \( v \) are identical.

Second, we show that any vertex of \( V^S(B_h) \) is unreachable from \( S \) in \( G_D \setminus B_\alpha \), where \( B_\alpha = Z\alpha \setminus \overline{B}_\alpha \). We suppose that a path \( Q \) inside graph \( G_D \setminus B_\alpha \) connects a source \( s \in S \) with a vertex \( w \in V^S(B_h) \). Path \( Q \) necessarily traverses level \( \alpha \), so it contains an arc \( (u, v) \) of \( \overline{B}_\alpha \), as the complement dam \( B_\alpha \) has been removed. As dam \( \overline{B}_\alpha \) is closer than \( \overline{B}_h \), vertices of \( V^T(\overline{B}_\alpha) \) form a subset of \( A^*(\overline{B}_\alpha) \), otherwise one vertex of \( V^T(\overline{B}_\alpha) \) is reachable in \( G_D \setminus B_\alpha \) and Menger’s paths make a vertex in \( V^S(\overline{B}_h) \) be reachable too, which is impossible. For this
reason, path $Q$ must contain a vertex $v \in A^*(B_\alpha)$. Therefore, it connects a vertex $v$ inside the dry instance of $B_\alpha$ with vertex $w$ which is outside. As a consequence, there is an edge/arc leaving the dry instance of $B_\alpha$ on path $Q$, which is a contradiction with our reasoning in Case 2.

Eventually, all vertices of $V^S(B_h)$ are unreachable from $S$ in $G_{D \setminus B_\alpha}$, so arcs of $B_h$ belong to the dry instance of $B_\alpha$. Conversely, arcs of $B_h$ does not belong to $E^*(B_\alpha)$, as the Menger’s paths containing arcs of $B_\alpha$ connect $S$ with $B_h$ despite the removal of $B_\alpha$. Therefore, $E^*(B_\alpha) \cap Z_h = B_h$. Moreover, $\sigma(B_\alpha) = \sigma(B_h)$ as their complement dams have also the same signature: $\sigma(B_\alpha) = \sigma(B_h)$. Dam $B_\alpha$ is thus closer than $B_h$, which is a contradiction because $B_h$ is supposed to be a closest dam.

Observe that the dry areas of a dam $B_i$ and its complement, $A^*(B_i)$ and $A^*(\overline{B_i})$ respectively, are disjoint because any vertex is reachable from $S$ either in $G\setminus B_i$ or in $G\setminus \overline{B_i}$ or in both of them.

4.2 A characterization of minimum cuts with dry instances

Theorem \[\text{3}\] provides us with the keystone to build our FPT($p$) algorithm. It combines the concepts of dry instance and closest dam: given a minimum $(S,T)$-cut $X$ and its front dam $B_{i(X)}$, either $X \setminus B_{i(X)} = \emptyset$ and $X = Z_{i(X)}$ or edges in $X \setminus B_h(X) \neq \emptyset$ belong to the dry instance of the dam $\overline{B_h(X)} = Z_h(X) \setminus B_h(X)$, where $B_h(X)$ is the closest dam of $B_i(X)$.

**Theorem 3.** If $X \neq Z_{i(X)}$ is a minimum cut for $I$, $B_{i(X)}$ its front dam, and $B_{h(X)}$ the closest dam of $B_{i(X)}$, then set $X \setminus B_{i(X)}$ is a minimum cut for the dry instance of $\overline{B_h(X)} = Z_h(X) \setminus B_h(X)$, i.e. $D(I,\overline{B_h(X)})$.

**Proof.** We prove that all edges in $X \setminus B_{i(X)}$ belong to the dry instance of $\overline{B_h(X)}$. Let us suppose ad absurdum that an edge $e = (u, v) \in X \setminus B_{i(X)}$ is reachable from $S$ in graph $G_D$ deprived of the dam $\overline{B_h(X)}$. Edge $e$ is an arc $(u, v)$ in $G_D$. 

---

**Fig. 5.** Illustration of the contradiction we arose for Case 2 in the proof of Lemma $5$.
We denote by $P_e$ a path in $G_D$ starting from a source $s \in S$ and terminating with $(u, v)$, deprived of arcs of $\overline{B_{i(X)}}$, $P_e : s \cdots u \cdot v$.

Due to the iterative construction of cuts $Z_i$, path $P_e$ necessarily contains one edge $e_h = (u_h, v_h)$ of level $h(X)$ when it arrives at a level greater than $i(X) \geq h(X)$. We know that this edge $e_h$ does not belong to $\overline{B_{i(X)}}$, so $e_h \in B_{h(X)}$ and $P_e : s \cdots u_h \cdot v_h \cdots u \cdot v$. From now on, we focus on the segment of path $P_e$, denoted by $P_e^{(h)}$, which contains all edges between $e_h$ and $e$, i.e. $P_e^{(h)} : u_h \cdot v_h \cdots u \cdot v$.

The proof goes in two steps:

Step 1: Path $P_e^{(h)}$ contains an arc of $B_{i(X)}$.

Step 2: The existence of a path in $G_D$ containing both an arc of $B_{i(X)}$ (proven in Step 1) and arc $e$ contradicts the definition of cut $X$.

For Step 1, let us suppose that the path $P_e^{(h)}$ does not contain arcs of $B_{i(X)}$. This means that path $P_e^{(h)}$ passes by the dry instance of $B_{h(X)}$ between levels $h(X)$ and $i(X)$, otherwise it would necessarily contain an arc of $B_{i(X)}$. So, there is an edge/arc of this path, $\tilde{e} = (\tilde{u}, \tilde{v})$, where $\tilde{u}$ is in the dry area of $B_{h(X)}$ but not in the dry area of $B_{i(X)}$, and $\tilde{v}$ lies outside the dry area of $B_{h(X)}$. In brief, $\tilde{u} \in A^*(B_{h(X)}) \setminus A^*(B_{i(X)})$ and $\tilde{v} \notin A^*(B_{h(X)})$. Edge $\tilde{e}$ must be an arc $(\tilde{u}, \tilde{v})$ in $G_D$ according to Lemma 3. Arc $\tilde{e}$ belongs to a Menger’s path $Q_{j(\tilde{e})}$ passing through $B_{h(X)}$, i.e. $Q_{j(\tilde{e})} \subseteq \sigma(B_{h(X)})$. Path $Q_{j(\tilde{e})}$ must traverse an arc of $B_{i(X)}$ as $\sigma(B_{i(X)}) = \sigma(B_{h(X)})$. As a consequence, path $Q_{j(\tilde{e})}$ connects a vertex $\tilde{v}$ outside the dry area of $B_{h(X)}$ with the tail $u'_i$ of an arc $e'_i = (u'_i, v'_i)$ of $B_{i(X)}$. This is a contradiction, as vertex $u'_i$ is supposed not to be reachable from $S$ in $G_D$ deprived of cut $B_{h(X)}$. Path $P_e^{(h)}$ thus contains an arc of $B_{i(X)}$ denoted by $e_i = (u_i, v_i)$.

For Step 2, let $\tilde{e} = (\tilde{u}, \tilde{v}) \neq e_i$ be the first arc of cut $X$ in path $P_e^{(h)}$ which arrives after $e_i$ on this path (Fig. 6). By this way, we ensure that no edge of $X$ lies on path $P_e^{(h)}$ between vertices $v_i$ and $\hat{u}$. Arc $\tilde{e}$ exists as $e$ is a potential candidate to be one.

As vertex $v_i$ is the head of $e_i$ in graph $G_D$, $v_i \in R(X, T)$ according to Lemma 2. For the same reason, vertex $\hat{u} \in R(X, S)$ as it is the tail of arc $\tilde{e} \in X$. We know that path $P_e^{(h)}$ in $G_D$ connects these two vertices and there is no arc of $X$ on the segment of $P_e^{(h)}$ connecting them. Let us go back now to the initial undirected graph $G$. In graph $G \setminus X$, vertices $v_i \in R(X, T)$ and $\hat{u} \in R(X, S)$ are connected whereas they must be separated by $X$. The presence of an edge $(u, v)$ of $X$ outside $E^*(\overline{B_{i(X)}})$ yields a contradiction.

In summary, all edges in $X \setminus B_{i(X)}$ belong to the dry instance of $\overline{B_{h(X)}}$. They necessarily form a cut in instance $D(\mathcal{I}, \overline{B_{h(X)}})$, otherwise $X$ would not separate $S$ and $T$ in $\mathcal{I}$. The $p - |B_{h(X)}|$ Menger’s paths in signature $\sigma(\overline{B_{h(X)}})$ are edge-disjoint inside $D(\mathcal{I}, \overline{B_{h(X)}})$, so the minimum cut size of this instance is greater than $p - |B_{h(X)}|$. As $X \setminus B_{i(X)}$ contains $p - |B_{i(X)}| = p - |B_{h(X)}|$ edges, we conclude that it is a minimum cut of $D(\mathcal{I}, \overline{B_{h(X)}})$.

Therefore, any minimum $(S, T)$-cut $X$, which is not a minimum drainage cut $Z_{i(X)}$ itself, can be partitioned into two sets, $B_{i(X)}$ and $X \setminus B_{i(X)}$, such that:
- \( B_i(\mathcal{X}) \) is a minimum cut of instance \( \mathcal{D}(\mathcal{I}, B_h(\mathcal{X})) \) and a dam of \( \mathcal{I} \),
- \( X \setminus B_i(\mathcal{X}) \) is a minimum cut of instance \( \mathcal{D}(\mathcal{I}, B_{\mathcal{h}}(\mathcal{X})) \) and all its edges belong to the target side of \( Z_i(\mathcal{X}) \), \( E[R(Z_i(\mathcal{X}), T)] \).

Conversely, given a closest dam \( B_h \) and its complement \( \overline{B_h} = Z_h \setminus B_h \), the union \( B_i \cup X_{\overline{B_h}} \), where the closest dam of \( B_i \) is \( B_h \) and \( X_{\overline{B_h}} \) is a minimum cut of \( \mathcal{D}(\mathcal{I}, \overline{B_h}) \), separates \( S \) from \( T \).

**Theorem 4.** Let \( B_h \) be a closest dam of \( \mathcal{Z}(\mathcal{I}) \) and \( \overline{B_h} = Z_h \setminus B_h \). Let \( B_i \) be a dam such that \( B_h \) is closer than \( B_i \) and \( X_{\overline{B_h}} \) a minimum cut of \( \mathcal{D}(\mathcal{I}, \overline{B_h}) \). Then, \( B_i \cup X_{\overline{B_h}} \) is a minimum \((S, T)\)-cut for instance \( \mathcal{I} \).

**Proof.** As dam \( B_h \) is closer than \( B_i \), the edges of \( B_i \) form a minimum cut of \( \mathcal{D}(\mathcal{I}, B_h) \). Indeed, they are the edges of level \( i \) inside \( \mathcal{D}(\mathcal{I}, B_h) \), so they separate \( S^*(B_h) \) from \( T^*(B_h) \). Moreover, we know there is a set of \( |\sigma(B_h)| \) edge-disjoint paths from \( S^*(B_h) \) to \( T^*(B_h) \) in \( \mathcal{D}(\mathcal{I}, B_h) \). As \( |\sigma(B_h)| = |\sigma(B_i)| = |B_i| \), set \( B_i \) is a minimum cut of \( \mathcal{D}(\mathcal{I}, B_h) \).

We suppose that there is an open \((S, T)\)-path \( Q \) in undirected graph \( G \) deprived of edges \( B_i \cup X_{\overline{B_h}} \). Path \( Q \) cannot avoid level \( h \) of the drainage and passes through one edge of \( Z_h \). As \( B_h \cup \overline{B_h} = Z_h \), some edges of path \( Q \) belong either to the dry instance of \( B_h \) or to the dry instance of \( \overline{B_h} \), or to both of them.

First, from Lemma 3 we know that no edge of graph \( H \) has one endpoint in the dry area \( A^*(B_h) \) of \( B_h \) and the another one in the dry area \( A^*(\overline{B_h}) \) of \( \overline{B_h} \).

Second, we show that the existence of path \( Q \) yields a contradiction with the definition of the dry instance. As sets \( A^*(B_h) \) and \( A^*(\overline{B_h}) \) cannot be connected by an edge of \( G \), path \( Q \) “traverses” completely at least one of the dry instances \( \mathcal{D}(\mathcal{I}, B_h) \) or \( \mathcal{D}(\mathcal{I}, \overline{B_h}) \), with no loss of generality we say \( \mathcal{D}(\mathcal{I}, B_h) \). In other words, a segment of \( Q \) connects \( S^*(B_h) \) and \( T^*(B_h) \). This is not possible because dam \( B_i \) separates these two sets of vertices. With the dry instance \( \mathcal{D}(\mathcal{I}, \overline{B_h}) \), we
obtain the same contradiction as \(X_{\overline{B}_h}\) separates \(S^*(\overline{B}_h)\) and \(T^*(\overline{B}_h)\). Therefore, \(B_i \cup X_{\overline{B}_h}\) separates \(S\) from \(T\) with \(p\) edges, as \(|X_{\overline{B}_h}| = |\sigma(\overline{B}_h)| = p - |B_i|\). \(\square\)

We now prove a stronger result for set \(X \setminus B_i(X)\). In fact, edges of set \(X \setminus B_i(X)\) lie in the target side of level \(i(X) - h(X) + 1\) in the drainage of instance \(\mathcal{D}(I, \overline{B}_h(X))\). This statement is formulated in the theorem below.

**Theorem 5.** Let \(X\) be a minimum \((S, T)\)-cut of \(G\) and let \((Z_1', \ldots, Z_k')\) be the drainage of instance \(\mathcal{D}(I, \overline{B}_h(X))\). Then, set \(Z_{i(X) - h(X) + 1}'\) is equal to \(\overline{B}_i(X) = Z_{i(X)}\setminus B_i(X)\) and edges \(X \setminus B_i(X)\) belong to the target side of \(Z_{i(X) - h(X) + 1}'\) inside instance \(\mathcal{D}(I, \overline{B}_h(X))\).

**Proof.** According to Theorem 3 we know that edges of \(X \setminus B_i(X)\) belong to the dry instance of \(\overline{B}_h(X)\), i.e. \(E^*(\overline{B}_h(X))\). Moreover, they are also in the target side of \(Z_{i(X)}\), as \(B_i(X) \subseteq Z_{i(X)}\) is the front dam of \(X\). We denote by \(\overline{B}_i(X)\) the complement of \(B_i(X)\) in \(Z_{i(X)}\), \(\overline{B}_i(X) = Z_{i(X)}\setminus B_i(X)\). We want to prove that dam \(\overline{B}_i(X)\) is the minimum drainage cut of level \(i(X) - h(X) + 1\) in instance \(\mathcal{D}(I, \overline{B}_h(X))\). For this purpose, we first prove that \(\overline{B}_i(X)\) is closer than \(\overline{B}_i(X)\).

Dam \(\overline{B}_i(X)\) has the same signature as \(\overline{B}_h(X)\) because their respective complement fulfil \(\sigma(\overline{B}_i(X)) = \sigma(\overline{B}_h(X))\). Moreover, we prove that arcs of \(\overline{B}_i(X)\) are in the dry instance of \(\overline{B}_h(X)\). Suppose an arc \(e_i = (u_i, v_i)\) of \(\overline{B}_i(X)\) does not belong to \(E^*(\overline{B}_h(X))\). Let \(Q_j\) be the Menger’s path containing arc \(e_i\). Then, no arc of \(Q_j\) after \(e_i\) is inside instance \(\mathcal{D}(I, \overline{B}_h(X))\). This is a contradiction as path \(Q_j\) must contain an edge of cut \(X\). Indeed, path \(Q_j\) contains neither an arc of \(B_i(X)\) as \(\sigma(B_i(X)) \cap \sigma(\overline{B}_i(X)) = \emptyset\) nor an arc of \(X \setminus B_i(X)\) as all its arcs after level \(i(X)\) do not belong to \(\mathcal{D}(I, \overline{B}_h(X))\) (contradiction with Theorem 3).

In summary, dam \(\overline{B}_i(X)\) is closer than \(\overline{B}_i(X)\).

All dams \(\overline{B}_\ell\) such that \(\sigma(\overline{B}_h(X)) = \sigma(\overline{B}_\ell)\) and \(h(X) < \ell < i(X)\) have a common closest dam: \(\overline{B}_\ell(X)\). Indeed, if it is not the case for a dam \(\overline{B}_\ell\), there is an arc \((u_\ell, v_\ell)\) of \(\overline{B}_\ell\) where \(u_\ell \notin A^*(\overline{B}_h(X))\). As a consequence, dam \(\overline{B}_i(X)\) is not contained in \(E^*(\overline{B}_h(X))\), as arc \((u_\ell, v_\ell)\) belongs to a Menger’s path containing an arc of \(\overline{B}_i(X)\).

We focus now on the drainage of instance \(\mathcal{D}(I, \overline{B}_h(X))\). The minimum drainage cut of level one in this instance is \(\overline{B}_h(X)\) itself, as it is the minimum closest \((S^*(\overline{B}_h(X)), T^*(\overline{B}_h(X)))\)-cut.

Let \(\overline{B}_h(X)+1\) be the dam of level \(h(X)+1\) in \(I\) fulfilling \(\sigma(\overline{B}_h(X)) = \sigma(\overline{B}_h(X)+1)\) and \(\overline{B}_h(X)+1\) its complement. We prove that \(\overline{B}_h(X)+1\) is the minimum drainage cut of level two in \(\mathcal{D}(I, \overline{B}_h(X))\), i.e. it is the minimum closest cut between \(V^T(\overline{B}_h(X))\) and \(T^*(\overline{B}_h(X))\) in graph \(G^*(\overline{B}_h(X))\) deprived of \(R(\overline{B}_h(X), S^*(\overline{B}_h(X)))\). Suppose that there is another minimum closest cut \(Z_{\overline{B}_h(X)} \neq \overline{B}_h(X)+1\). Set \(X_{\overline{B}_h(X)} = X_{\overline{B}_h(X)} \cup B_h(X)+1\) is a minimum \((S, T)\)-cut of instance \(I\) according to Theorem 4. As its edges belong to the target side of \(Z_{h(X)}\), it is a minimum \((S_{h(X)}+1, T)\)-cut. Based on the definition of \(Z_{\overline{B}_h(X)}\), the reachable set of \(X_{\overline{B}_h(X)}\)
is necessarily included into the reachable set of $Z_{h(X)}+1 = B_{h(X)}+1 \cup \overline{B}_{h(X)}+1$ in graph $G \setminus R(Z_{h(X)}, S)$. This is a contradiction to the construction of the drainage, as $Z_{h(X)}+1$ is the unique minimum closest $(S_{h(X)}+1, T)$-cut in graph $G \setminus R(Z_{h(X)}, S)$. Consequently, $\overline{B}_{h(X)}+1$ is the minimum drainage cut of level two inside $D(I, B_{h})$.

We can iterate these arguments on dams $\overline{B}_{h}+2, \overline{B}_{h}+3$, etc. For example, dam $\overline{B}_{h}+2, \sigma(\overline{B}_{h}+2) = \sigma(\overline{B}_{h}(X))$, is the minimum closest $(V^{T}(\overline{B}_{h}(X)+1), T)$-cut when graph $G^{*}(\overline{B}_{h}(X))$ is deprived of $R(\overline{B}_{h}(X)+1, S^{*}(\overline{B}_{h}(X)))$. Otherwise, it would imply that $Z_{h(X)}+2$ is not the minimum closest $(S_{h(X)}+2, T)$-cut in graph $G \setminus R(Z_{h(X)}+1, S)$, which contradicts the construction of the drainage. Eventually, dam $\overline{B}_{h}+3$ is the minimum drainage cut of level three inside $D(I, \overline{B}_{h}(X))$, dam $\overline{B}_{h}+3$ of level four, etc. Then, dam $\overline{B}_{i}(X)$ is the minimum drainage cut of level $i(X) - h(X) + 1$ in instance $D(I, \overline{B}_{h}(X))$.

Coming back to Theorem 3 the edges of $X \setminus B_{i}(X)$ belong to both $E^{*}(\overline{B}_{i}(X))$ and $E[R(Z_{i}(X), T)]$. So, they are in the target side of dam $\overline{B}_{i}(X)$ inside instance $D(I, \overline{B}_{h}(X))$.

4.3 Description of the algorithm

Our algorithm starts by computing the drainage $Z(I)$ and the Menger’s paths of input instance $I$. For all dams $B_{i}$, it counts the minimum cuts of size $p$ in $I$ which admit the front dam $B_{i}$. If $B_{i} \neq Z_{i}$, it does this recursively by counting the minimum cuts in instance $D(I, \overline{B}_{h})$ which only contains edges from the target side of the internal level $i-h+1$ of $D(I, \overline{B}_{h})$, where $B_{h}$ is the closest dam of $B_{i}$. The minimum cut size in $D(I, \overline{B}_{h})$ is at most $p-1$.  

Fig. 7. Illustration of Theorem 3 for any minimum cut with the front dam $B_{i}$, the tails of arcs in $X \setminus B_{i}$ belong to the yellow zone.
We denote by $C_0(I) = C(I)$ the total number of minimum $(S,T)$-cuts of instance $I$. We define $C_{\ell}(I)$ as the number of minimum cuts of instance $I$ which are composed of edges from $E[R(Z_\ell,T)]$ only. For example, $C_2(I)$ gives the number of minimum $(S,T)$-cuts in instance $I$ without edges of $Z_1 \cup Z_2$. Value $C_{\ell}(I)$, $0 \leq \ell \leq k - 1$, can be written:

$$C_{\ell}(I) = k - \ell + \sum_{\text{Closest dam } B_i \subseteq Z_i} \sum_{i : i > \ell, \exists B_i : B_i \text{ closer than } B_i} C_{i-h+1} \left( D \left( I, \overline{B}_h \right) \right).$$  \hspace{1cm} (1)

The first $k - \ell$ cuts are the minimum drainage cuts of $I$ with level greater than $\ell$, i.e. cuts $Z_{\ell+1}, \ldots, Z_k$. The second term counts cuts taking edges only from $E[R(Z_\ell,T)]$ and admitting a front dam $B_{i(X)} \neq Z_{i(X)}$. Theorems 2 and 3 guarantee that any of these minimum $(S,T)$-cuts is counted at least once. Indeed, for any front dam $B_i$ and its closest dam $B_{\ell}$, we compute the number of cuts in instance $D \left( I, \overline{B}_h \right)$ such that all their edges belong to the target side of $\overline{B}_i$, which is the internal level $i - h + 1$ in $D \left( I, \overline{B}_h \right)$. In the event that the drainage of $D \left( I, \overline{B}_h \right)$ has less than $i - h + 1$ levels, then $C_{i-h+1} \left( D \left( I, \overline{B}_h \right) \right) = 0$, as it means no minimum cut of $I$ has the front dam $B_i$.

Conversely, the unicity of a closest dam ensures us that each minimum cut is counted exactly once. A minimum $(S,T)$-cut $X \neq Z_{i(X)}$ has a unique front dam $B_{i(X)}$ and the closest dam $B_{h(X)}$ of $B_{i(X)}$ is unique (Lemma 4). Finally, Theorem 5 secures that all cuts counted with Eq. 1 are minimum $(S,T)$-cuts.

Value $C_0(I)$ is computed thanks to recursive calls on multiple instances $D \left( I, \overline{B}_h \right)$. From now on, we distinguish the input instance $I$ (for which we want to compute $C_0(I)$) with other instances (denoted by $J$ later on) of the recursive tree. The base cases of the recursion, i.e. the leaves of the recursive tree, are the computation of values $C_{\ell}(J)$ either in instances $J$ where the minimum cut size is one or in instances where no minimum cut admits a front dam $B_i \neq Z_i$, $i > \ell$. In both cases, the only minimum cuts of $J$ are its minimum drainage cuts. Each recursive call of the algorithm makes the minimum cut size decrease: for example, if the minimum cut size of $J$ is $q$, then it is $|\overline{B}_h| < q$ for an instance $D \left( J, \overline{B}_h \right)$. Therefore, the recursive tree is not deeper than $p - 1$.

Fig. 8 illustrates the recursive scheme of our algorithm with a tree describing the relationship between the instances. Three instances $I$, $J$, and $J'$ of the recursive tree are represented. For example, instance $J = D \left( I, \overline{B}_h \right)$ is the son of instance $I$ in the tree as it is one of its dry instances.

In parallel, another graph (black dashed arcs in Fig. 8) contains arcs with endpoints $C_{\ell}(J)$. An arc connects two “compartments” $C_{\ell}(J)$ and $C_{\ell'}(J')$ when the computation of $C_{\ell}(J)$ depends on $C_{\ell'}(J')$. The minimum $(S,T)$-cut size of $J'$ is smaller than the one of $J$. This is why the graph made of arcs between compartments is a DAG.

Then, we present the proof of Theorem 6 which allows us to declare the fixed-parameter tractability of counting mincuts.

**Theorem 6.** There are at most $2^p^2^m$ instances in the recursive tree.
Proof of Theorem 6. The depth of instance $I$ in the recursive tree is zero, we say $\Delta (I) = 0$. For any closest dam $B_h$ of $Z(I)$, the depth of the dry instance of $B_h$ is one: $\Delta (D(I, B_h)) = 1$. More generally, if $J$ is an instance of depth $d \geq 0$ and $B_h$ a closest dam of $Z(J)$, then instance $D(J, B_h)$ is at depth $d + 1$.

We prove that, for any edge $e$ in graph $G$, there are at most $2^{pd}$ instances $J$ of depth $d$ such that edge $e$ belongs to the graph of $J$. This fact makes the total number of instances in the recursive tree be upper-bounded by $2^{pd} m$.

We proceed by induction. There is one instance defined for depth $d = 0$: it is $I$ and it obviously contains edge $e$, so the number of instances with depth $d = 0$ containing $e$ is thus $2^{pd} = 1$. Let $d \geq 1$ and $J'$ be an instance of depth $d$ containing edge $e$: there is an instance $J$ of depth $d - 1$ and one of its closest dam $B_h$ such that $J' = D(J, B_h)$. As the graph of instance $J'$ is a subgraph of those of $J$, the latter contains edge $e$.

Using the induction hypothesis, there are at most $2^{p(d-1)}$ instances $J$ of depth $d - 1$ containing edge $e$. Now, given an instance $J$ with $\Delta (J) = d - 1$, we bound the number of dams $B_h$ (they are closest dams according to Lemma 5) of $J$ such that $D(J, B_h)$ contains edge $e$. We distinguish two cases:

- Case 1: edge $e$ belongs to a minimum drainage cut $Z_i$ of instance $J$. We focus on the dams $B_i$ of level $i$ containing edge $e$. Their cardinality is bounded by $2^p$. The edges of level $i$ belonging to the dry instance of $B_h$, $D(J, B_h)$, form one of these dams $B_i$. As each dam $B_i$ admits a unique closest dam (Lemma 4), there cannot be more than $2^p$ closest dams $B_h$ such that $D(J, B_h)$ contains edge $e$.

- Case 2: edge $e$ is located between two minimum drainage cuts $Z_i$ and $Z_{i+1}$, $e \in R_{i+1}$. Consequently, the level of any closest dam $B_h$ such that $D(J, B_h)$ contains $e$ is less than $i$: $i \geq h$. Therefore, the edges of level $i$ belonging to
the dry instance of $\overline{B}_h$ form a dam $B_i$. Thus, the argument used in Case 1 arises the same conclusion: there cannot be more than $2^p$ closest dams such that $D(J, \overline{B}_h)$ contains edge $e$.

Finally, the number of instances written as $J' = D(J, \overline{B}_h)$ where $\Delta(J) = d - 1$ and $J'$ contains $e$, is upper-bounded by $2^{p(d-1)2^p} = 2^{pd}$. We conclude that there are less than $2^{pd}$ instances of depth $d$ containing edge $e$. The total number of instances is thus smaller than $\sum_{d=0}^{p-1}2^{pd}m \leq 2^{p^2}m$.

For any instance $J$ of the recursive tree, the algorithm computes its drainage $Z(J)$, its Menger’s paths and all instances $D(J, B_h)$ where $B_h$ is a closest dam of $Z(J)$. This third operation is done by enumerating all dams $B_i$ of $Z(J)$, verifying whether there is another dam $B_h$ which is closer than $B_i$, and (if $B_i$ is a closest dam) identifying the vertices/edges of $D(J, B_i)$ thanks to a depth-first search in $G_D \setminus B_i$. As there are at most $2^p n$ dams in $Z(J)$, its execution time is $O(2^{2p}n^3)$. The overall complexity is $O \left(2^{p^2} m (m n p + 2^{2p} n^3)\right) = O \left(2^{p^2+2} p mn^3\right)$.

5 Sampling minimum edge $(S,T)$-cuts in undirected graphs

We sketch the algorithm which produces one of the minimum $(S,T)$-cuts according to the uniform distribution over all minimum $(S,T)$-cuts. We run our counting algorithm (Section 4.3) and execute a post-processing, described below.

As in Section 4.3, we distinguish the input instance $\mathcal{I}$ from the other instances $J$ of the recursive tree. Our method to sample minimum cuts consists in searching in the recursive tree, already filled out with values $C_\ell(J)$ during the counting. A minimum cut of $\mathcal{I}$ is extracted thanks to a randomly driven descent in the recursive tree.

We start at root $C_0(\mathcal{I})$. With probability $\frac{k}{C_0(\mathcal{I})}$, the sampling algorithm returns one of the minimum drainage cuts of $Z(\mathcal{I})$ taken uniformly over them. Said differently, each cut $Z_i$ has probability $\frac{1}{C_0(\mathcal{I})}$ to be produced. With probability $1 - \frac{k}{C_0(\mathcal{I})}$, we will go one step down the tree. Concretely, for any dam $B_i$ of $Z(\mathcal{I})$ and its closest dam $B_h$, we visit node $C_{i-h+1}(D(\mathcal{I}, \overline{B}_h))$ of depth 1 with probability $\frac{C_{i-h+1}(D(\mathcal{I}, \overline{B}_h))}{C_0(\mathcal{I})}$. The sampling algorithm returns the union of $B_i$ with the cut obtained by a recursive call on $C_{i-h+1}(D(\mathcal{I}, \overline{B}_h))$. The algorithm applied on $C_{i-h+1}(D(\mathcal{I}, \overline{B}_h))$ either selects a minimum drainage cut of level greater than $i-h+1$ in $D(\mathcal{I}, \overline{B}_h)$ (uniform selection among these cuts) or visits a node at depth 2, etc.

In this way, we ensure that the minimum $(S,T)$-cuts are sampled uniformly. Indeed, a cut with front dam $B_i$ is chosen with probability $\frac{C_{i-h+1}(D(\mathcal{I}, \overline{B}_h))}{C_0(\mathcal{I})}$ which is the ratio of the number $C_{i-h+1}(D(\mathcal{I}, \overline{B}_h))$ of minimum cuts with front dam $B_i$ by the total number $C_0(\mathcal{I})$ of minimum cuts in instance $\mathcal{I}$. 

6 Conclusion

In this study, we were interested in the parameterized complexity of counting the minimum \((S, T)\)-cuts in undirected graphs. The conclusion is that an algorithm running in \(O(2^{p(p+2)}pmn^3)\) was devised. Our algorithm starts by “draining” the graph: the drainage is made of \(k < n\) minimum cuts \(Z_i\). For any minimum cut of the instance, at least one of the minimum drainage cuts \(Z_i\) contains edges of \(X\). For this reason, we believe that the drainage could be used on other cut problems. We already used it to sample minimum edge \((S, T)\)-cuts.

Our work gives rise to questions concerning the counting of minimum edge \((S, T)\)-cuts in undirected graphs. These questions are:

1. Is there an \(\text{FPT}(p)\) algorithm solving \textsc{counting mincuts} with smaller polynomial factors? In particular, for dense graphs, our algorithm complexity is \(O(n^3)\) if we neglect the function of \(p\). At first sight, it seems difficult to avoid the use of a maximum flow algorithm \[17\], which is \(O(mp)\). Is it possible to identify an algorithm with running time \(O(f(p)n^2)\), \(O(f(p)n^3)\), \(O(f(p)n^4)\)?

2. Is there an algorithm solving \textsc{counting mincuts} in time \(2^{o(p^2)}n^{O(1)}\)? In other words, we wonder whether it is possible to lower the factor \(O(p^2)\) in the exponential and to find an algorithm with running time \(2^{O(p)n^{O(1)}}\) or \(2^{O(p \log p)n^{O(1)}}\), for example.

Our algorithm can be generalized to undirected graphs with positive integer weights. The idea is to transform these undirected weighted graphs into undirected multigraphs: if the weight of edge \(e = (u, v)\) is larger than \(p + 1\), then we replace \(e\) by \(p + 1\) edges \((u, v)\) in parallel. Otherwise, if its weight is \(w \leq p\), we replace it by \(w\) edges \((u, v)\). By this method, the number of edges in the transformed graph is upper-bounded by \(m(p + 1)\) and it preserves the minimum cuts of the input graph. Moreover, the techniques used in this article to compute the number of minimum \((S, T)\)-cuts, such as flow algorithms or closest cuts, work on multigraphs.

We can modify slightly our algorithm to enumerate minimum \((S, T)\)-cuts. This simply consists in stacking in the recursive tree the set of cuts counted with \(C_{\ell}(I)\) instead of value \(C_{\ell}(I)\) itself. The number of instances stays \(\text{FPT}(p)\) but the sets of cuts put inside the recursive tree may contain a number \(\Omega(n^p)\) of elements.

We believe that the techniques proposed in this study could be used to count minimum vertex \((S, T)\)-cuts in undirected graphs and minimum \((S, T)\)-cuts in directed graphs. However, we explain below why major changes of our algorithm are needed to make it work on these applications.

Let us focus on minimum vertex \((S, T)\)-cuts. The drainage can be extended to vertex cuts, as the unicity of the minimum closest cut is preserved. Fig. \ref{fig:draining} illustrates how the drainage could be defined for minimum vertex \((S, T)\)-cuts: cut \(Z_1\) is the minimum closest \((S, T)\)-cut, set \(R_1\) is equal to \(R(Z_1, S_1)\) with \(S_1 = S\). Then, \(S_2 = Z_1\), cut \(Z_2\) is the minimum closest \((S_2, T)\)-cut in graph \(G \setminus R(Z_1, S)\), and so on.
Fig. 9. An example of drainage when cuts are composed of vertices.

With this definition, the drainage fulfills the properties given in Section 3 for the edge version, the most important of them being that any minimum vertex \((S, T)\)-cut \(X\) admits a front dam \(B_i(X)\). Moreover, there is a vertex version of the Menger’s theorem, stating that the size of the minimum \((S, T)\)-cut is equal to the maximum number of vertex-disjoint \((S, T)\)-paths. The largest set of maximum vertex-disjoint \((S, T)\)-paths is computed in polynomial time and, consequently, the definitions of the dry area and the dry instance can be naturally extended.

Fig. 10. Illustration of the impossibility to prove Theorem 3 for vertex cuts

However, Theorem 3 does not hold anymore: the set \(X \setminus B_i(X)\) of a minimum \((S, T)\)-cut \(X\) is not necessarily included in the dry instance of \(\overline{B}_h(X)\). We give an example in Fig. 10. Set \(X = \{v_1, v_2\}\) is a minimum vertex \((S, T)\)-cut and its two vertices are drawn in red. One of its vertices \(v_1\) belongs to cut \(Z_1\), so \(B_i(X) = B_h(X) = \{v_1\}\). We observe that vertex \(v_2\) is reachable from \(S\) in graph \(G_{D} \setminus \overline{B}_h(X)\) because the dry area of \(\overline{B}_h(X) = \{u_1\}\) does not contain any vertex different than \(u_1\) itself. To pursue the work presented in this article, our intention is to modify the definitions of the dry area and the dry instance in order to make Theorem 3 be true for vertex cuts.

Acknowledgement

We would like to thank Saket Saurabh who pointed out the related work [21] during the WG’19 conference.
References

1. Arvind, V., Raman, V.: Approximation algorithms for some parameterized counting problems. In: Proc. of ISAAC. pp. 453–464 (2002)
2. Ball, M.O., Colbourn, C.J., Provan, J.S.: Network reliability. Handbooks in Operations Research and Management Science, vol. 7, pp. 673 – 762. Elsevier (1995)
3. Ball, M.O., Provan, J.S.: Calculating bounds on reachability and connectedness in stochastic networks. Networks 13(2), 253–278 (1983)
4. Ball, M.O., Provan, J.S.: Computing network reliability in time polynomial in the number of cuts. Operations Research 32(3), 516–526 (1984)
5. Bezákova, I., Chambers, E.W., Fox, K.: Integrating and sampling cuts in bounded treewidth graphs. In: Advances in the Math. Sciences. pp. 401–415 (2016)
6. Bezáková, I., Friedlander, A.J.: Counting and sampling minimum (s,t)-cuts in weighted planar graphs in polynomial time. Theor. Comput. Sci. 417, 2–11 (2012)
7. Bousquet, N., Daligault, J., Thomassé, S.: Multicut is FPT. In: Proc. of STOC. pp. 459–468 (2011)
8. Boykov, Y., Veksler, O.: Graph cuts in vision and graphics: Theories and applications. In: Handbook of Math. Models in Computer Vision, pp. 79–96 (2006)
9. Chambers, E.W., Fox, K., Nayyeri, A.: Counting and sampling minimum cuts in genus $g$ graphs. In: Proc. of SoCG. pp. 249–258 (2013)
10. Chandran, L.S., Ram, L.S.: On the number of minimum cuts in a graph. In: Proc. of COCOON. pp. 220–229 (2002)
11. Chen, J., Liu, Y., Lu, S.: An improved parameterized algorithm for the minimum node multiway cut problem. Algorithmica 55(1), 1–13 (2009)
12. Chitnis, R.H., Hajiaghayi, M., Marx, D.: Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. SIAM J. Comput. 42(4), 1674–1696 (2013)
13. Curticapean, R.: Counting problems in parameterized complexity. In: Proc. of IPEC. pp. 1–18 (2018)
14. Cygan, M., Lokshtanov, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Minimum bisection is fixed parameter tractable. In: Proc. of STOC. pp. 323–332 (2014)
15. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science, Springer (1999)
16. Flum, J., Grohe, M.: The parameterized complexity of counting problems. SIAM J. Comput. 33(4), 892–922 (2004)
17. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canad. J. Math. 8, 399–404 (1956)
18. Guillemot, S., Sikora, F.: Finding and counting vertex-colored subtrees. Algorithmica 65(4), 828–844 (2013)
19. Krom, M.R.: The decision problem for a class of firstorder formulas in which all disjunctions are binary. Mathematical Logic Quarterly 13(12), 15–20 (1967)
20. Marx, D.: Parameterized graph separation problems. Theor. Comput. Sci. 351(3), 394–406 (2006)
21. Marx, D., O’Sullivan, B., Razgon, I.: Finding small separators in linear time via treewidth reduction. ACM Trans. Algorithms 9(4), 30:1–30:35 (2013)
22. Marx, D., Razgon, I.: Fixed-parameter tractability of multicut parameterized by the size of the cutset. In: Proc. of STOC. pp. 469–478 (2011)
23. Menger, K.: Zur allgemeinen Kurventheorie. Fundamenta Mathematicae 10(1), 96–115 (1927)
24. Nagamochi, H., Sun, Z., Ibaraki, T.: Counting the number of minimum cuts in undirected multigraphs. IEEE Trans. Reliab. 40, 610–614 (1991)
25. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Series in Math. and Its Applications, OUP Oxford (2006)
26. Provan, J.S., Ball, M.O.: The complexity of counting cuts and of computing the probability that a graph is connected. SIAM J. Comput. 12(4), 777–788 (1983)
27. Valiant, L.G.: The complexity of counting the permanent. Theor. Comput. Sci. 8, 189–201 (1979)
28. Williams, V.V., Williams, R.: Finding, minimizing, and counting weighted subgraphs. SIAM J. Comput. 42(3), 831–854 (2013)