Spanning Circuits in Regular Matroids

FEDOR V. FOMIN and PETR A. GOLOVACH, Department of Informatics, University of Bergen, Norway
DANIEL LOKSHTANOV, Department of Computer Sciences, University of California Santa Barbara, USA
SAKET SAURABH, Department of Informatics, University of Bergen, Norway and The Institute of Mathematical Sciences, HBNI, India

We consider the fundamental Matroid Theory problem of finding a circuit in a matroid containing a set T of given terminal elements. For graphic matroids, this corresponds to the problem of finding a simple cycle passing through a set of given terminal edges in a graph. The algorithmic study of the problem on regular matroids, a superclass of graphic matroids, was initiated by Gavenčiak, Král', and Oum [ICALP'12], who proved that the case of the problem with $|T| = 2$ is fixed-parameter tractable (FPT) when parameterized by the length of the circuit. We extend the result of Gavenčiak, Král', and Oum by showing that for regular matroids

- the Minimum Spanning Circuit problem, deciding whether there is a circuit with at most ℓ elements containing T, is FPT parameterized by $k = \ell - |T|$;
- the Spanning Circuit problem, deciding whether there is a circuit containing T, is FPT parameterized by $|T|$.

We note that extending our algorithmic findings to binary matroids, a superclass of regular matroids, is highly unlikely: Minimum Spanning Circuit parameterized by ℓ is W[1]-hard on binary matroids even when $|T| = 1$. We also show a limit to how far our results can be strengthened by considering a smaller parameter. More precisely, we prove that Minimum Spanning Circuit parameterized by $|T|$ is W[1]-hard even on cographic matroids, a proper subclass of regular matroids.

CCS Concepts: • **Mathematics of computing → Combinatorial algorithms**; • **Theory of computation → Parameterized complexity and exact algorithms**;

A preliminary version of this paper appeared as an extended abstract in the proceedings of SODA 2017.
This work is supported by the Research Council of Norway via the projects "CLASSIS" and "MULTIVAL" and the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 267959, ERC Starting Grant n. 715744 "Pareto-Optimal Parameterized Algorithms" and the European Union's Horizon 2020 research and innovation programme (grant agreement No 819416).
Authors' addresses: F. V. Fomin and P. A. Golovach, Department of Informatics, University of Bergen, PB 7803, Bergen, 5020, Norway; emails: {fedor.fomin, petr.golovach}@uib.no; D. Lokshtanov, Department of Computer Sciences, University of California Santa Barbara, 2104 Harold Frank Hall, Santa Barbara, CA, 93106-5110, USA; email: daniello@ii.uib.no; S. Saurabh, Department of Informatics, University of Bergen, PB 7803, Bergen, 5020, Norway, The Institute of Mathematical Sciences, HBNI, 4th Cross Street, CIT Campus, Tharamani, Chennai, Tamil Nadu, 600113, India; email: saket@imsc.res.in.

1549-6325/2019/10-ART52 $15.00
https://doi.org/10.1145/3355629

Additional Key Words and Phrases: Spanning circuit, regular matroids, parameterized complexity

ACM Reference format:
Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. 2019. Spanning Circuits in Regular Matroids. *ACM Trans. Algorithms* 15, 4, Article 52 (October 2019), 38 pages.
https://doi.org/10.1145/3355629

1 INTRODUCTION

Deciding if a given graph G contains a cycle passing through a specified set T of terminal edges or vertices is a classical problem in graph theory. The study of this problem can be traced back to the fundamental theorem of Dirac from 1960s about the existence of a cycle in k-connected graph passing through a given set of k vertices [13]. According to Kawarabayashi [20] *"...cycles through a vertex set or an edge set are one of central topics in all of graph theory."* We refer to [19] for an overview on the graph-theoretical study of the problem, including the famous Lovász-Woodall Conjecture.

The algorithmic version of this question, "Is there a polynomial time algorithm deciding if a given graph contains a cycle passing through the set of terminal vertices or edges," is the problem of a fundamental importance in graph algorithms. Since the problem generalizes the classical Hamiltonian cycle problem, it is NP-complete. However, for a fixed number of terminals the problem is solvable in polynomial time. The case $|T| = 1$ with one terminal vertex or edge is trivially solved by the breadth first search. The case of $|T| = 2$ can be reduced to finding a flow of size 2 between two vertices in a graph. The case of $|T| = 3$ is already nontrivial and was shown to be solvable in linear time in [22], see also [16]. The fundamental result of Robertson and Seymour on the disjoint path problem [29] implies that the problem can be solved in time $O(n^3)$ time for a fixed number of terminals. Kawarabayashi [20] provided a quantitative improvement by showing that the problem is solvable in polynomial time for $|T| = O((\log \log n)^{1/10})$, where n is the size of the input graph.[1] Björklund et al. [3] gave a randomized algorithm solving the problem in time $2^{|T|}n^{O(1)}$. The algorithm of Björklund et al. solves also the minimization variant of the problem, where the task is to find a cycle of minimum length passing through terminal vertices. We refer to Cygan et al. [8] for an overview of different techniques in parameterized algorithms for solving problems about cycles and paths in graphs.

Matroids are combinatorial objects generalizing graphs and linear independence. The study of the structure of subsets of elements that span certain elements of a matroid is one of the central themes in matroid theory. In particular, the problem of finding a circuit containing specified elements has a long history. For graphic matroids, this problem corresponds to finding in a graph a simple cycle passing through specified edges. The classical theorem of Whitney [35] asserts that any pair of elements of a connected matroid are in a circuit. Seymour [32] obtained a characterization of binary matroids with a circuit containing a triple of elements. See also [9, 25, 28] and references there for combinatorial results about circuits containing certain elements in matroids. However, compared to graphs, the algorithmic aspects of "circuits through elements" in matroids are much less understood.

In their work on deciding first-order properties on matroids of locally bounded branch-width, Gavenčiak et al. [17] initiated the algorithmic study of the following problem.

[1]We are grateful to the anonymous reviewer who pointed to us that it is possible to improve the range when the algorithm of Kawarabayashi [20] is polynomial using the recent results of Chekuri and Chuzhoy [6] but such an improvement is a separate task that is outside of the framework of our article.

Minimum Spanning Circuit

Input:	A binary matroid M with a ground set E, a weight function $w\colon E \to \mathbb{N}$, a set of *terminals* $T \subseteq E$, and a nonnegative integer ℓ.
Task:	Decide whether there is a circuit C of M with $w(C) \leq \ell$ such that $T \subseteq C$.

Here and further we use $\mathbb{N} = \{1, 2, \ldots\}$ to denote the the set of natural numbers (without 0). We also assume that a binary matroid in the problem input is given by its representation over GF(2).

Since graphic matroids are binary, this problem is a generalization of the problem of finding a cycle through a given set of edges in a graph. By the result of Vardy [34] about the Minimum Distance problem from coding theory, Minimum Spanning Circuit is NP-complete even when $T = \emptyset$. Gavenčiak et al. [17] observed that the hardness result of Downey et al. [15] also implies that Minimum Spanning Circuit is W[1]-hard on binary matroids with unit-weights elements when parameterized by ℓ even if $|T| = 1$. Parameterized complexity of Minimum Spanning Circuit for $T = \emptyset$ on binary matroids, i.e., the case when we ask about the existence of a circuit of length at most ℓ, is known as Even Set in parameterized complexity and is a long-standing open problem in the area. Very recently it was shown by Bhattacharyya et al. [2] that Even Set does not admit FPT algorithms under the *(randomized) Gap Exponential Time Hypothesis* (we refer to [12, 24] for the definition). The intractability of the problem changes when we restrict the input binary matroid to be regular, i.e., matroid, which has a representation by columns of a totally unimodular matrix. In particular, Gavenčiak et al. show that for $|T| = 2$, Minimum Spanning Circuit is fixed parameter tractable (FPT) being parameterized by ℓ by giving time $\ell^{\ell^{\ell^{O(\ell)}}} n^{O(1)}$ algorithm, where n is the number of elements in the input matroid. Recall that all graphic and cographic matroids are regular and thus algorithmic results for regular matroids yield algorithms on graphic and cographic matroids.

Our results. In this work, we show, and this is the main result of the article, that on regular matroids Minimum Spanning Circuit is FPT being parameterized by ℓ without any additional condition on the size of the terminal set. Actually, we obtain the algorithm for "stronger" parameterization $k = \ell - w(T)$. The running time of our algorithm is $2^{O(k^2 \log k)} \cdot n^{O(1)}$.

Our approach is based on the classic decomposition theorem of Seymour [31]. Roughly speaking, the theorem allows to decompose a regular matroid by making use of 1-, 2-, and 3-sums into graphic matroids, cographic matroids, and matroids of a fixed size (we refer to Section 3 for the precise formulation of the theorem). Thus, to solve the problem on regular matroids, one has to understand how to solve a certain extension of the problem on graphic and cographic matroids (matroids of constant size are usually trivial) and then employ Seymour's theorem to combine solutions. This is exactly the approach taken by Gavenčiak et al. [17] for solving the problem for $|T| = 2$, and this is the approach we adapt in this article. However, the details are very different. In particular, to use the general framework, we have to solve the problem on cographic matroids, which is already quite non-obvious. Gavenčiak et al. [17] adapt the method of Kawarabayashi and Thorup [21], who used it to prove that finding an edge-cut with at most s edges that separates the input graph into at least k component is FPT when parameterized by s. This approach works for $|T| = 2$ and probably may be extended for the case when the number of terminals is bounded, but we doubt that it could be applied for the parameterization by $k = \ell - w(T)$. Hence, to solve Minimum Spanning Circuit on cographic matroids, we use the recent framework of *recursive understanding* developed by Chitnis et al. [7]. We apply recursive understanding for solving the Minimal Terminal Cut problem. Here we are given a a connected graph G with a terminal set of edges $T \subseteq E(G)$ and terminal vertex sets $R_1, R_2 \subseteq V(G)$, and the task is to find a cut C of small weight satisfying the following constraints:

(a) this cut should be a minimal cut-set,
(b) it should contain all edges of T, and
(c) it should separate R_1 from R_2, meaning that $G - C$ contains distinct connected components X_1 and X_2 such that $R_i \subseteq X_i$ for $i \in \{1, 2\}$.

We believe that this problem is interesting on its own. Finally, constructing a solution by going through Seymour's matroid decomposition when $|T|$ is unbounded is also a non-trivial procedure.

With a similar approach, we also obtain an algorithm for the following decision version of the problem, where we put no constrains on the size of the circuit.

Spanning Circuit

Input:	A binary matroid M with a ground set E and a set of terminals $T \subseteq E$.
Task:	Decide whether there is a circuit C of M such that $T \subseteq C$.

We show that on regular matroids Spanning Circuit is FPT parameterized by $|T|$.

The remaining part of the article is organized as follows. In Section 2, we introduce basic notions used in the article. In Section 3, we briefly introduce the fundamental structural results of Seymour [30] about regular matroids. We also explain the refinement of the decomposition theorem of Seymour [30] given by Dinitz and Kortsarz [11] that is more convenient for the algorithmic purposes. We conclude this section by some structural results about circuits in regular matroids. Section 4 contains the algorithm for Minimal Terminal Cut . In Section 5, we give the algorithm for Minimum Spanning Circuit on regular matroids. First, we solve the extended variant of Minimum Spanning Circuit on matroids that are basic for the Seymour's decomposition [30]. Then, we explain how to obtain the general result. We follow the same scheme in Section 6 for Spanning Circuit parameterized by $|T|$. In Section 7, we provide some hardness observations and state open problems.

2 PRELIMINARIES

Parameterized Complexity. Parameterized complexity is a two-dimensional framework for studying the computational complexity of a problem. One dimension is the input size n and another one is a parameter k. It is said that a problem is *fixed parameter tractable* (or FPT) if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function f. We refer to the recent books of Cygan et al. [8] and Downey and Fellows [14] for the introduction to parameterized complexity.

It is standard for a parameterized algorithm to use *(data) reduction rules*, i.e., polynomial or FPT algorithms that either solve an instance or reduce it to another one that typically has a lesser input size and/or a lesser value of the parameter. We say that reduction rule is *safe* if it either correctly solves the problem or outputs an equivalent instance of the problem without increasing the parameter.

Graphs. We consider finite undirected (multi) graphs that can have loops or multiple edges. Throughout the article, we use n to denote the number of vertices and m the number of edges of considered graphs unless it creates confusion. For a graph G and a subset $U \subseteq V(G)$ of vertices, we write $G[U]$ to denote the subgraph of G induced by U. We write $G - U$ to denote the subgraph of G induced by $V(G) \setminus U$, and $G - u$ if $U = \{u\}$. Respectively, for $S \subseteq E(G)$, $G[S]$ denotes the graph induced by S, i.e., the graph with the set of edges S whose vertices are the vertices of G incident to the edges of S. We denote by $G - S$ the graph obtained from G by the deletion of the edges of G; for a single element set, we write $G - e$ instead of $G - \{e\}$. For $e \in E(G)$, we denote by G/e the graph obtained by the contraction of e. Since we consider multigraphs, it is assumed that if $e = uv$, then to construct G/e, we delete u and v, construct a new vertex w, and then for each $ux \in E(G)$ and each $vx \in E(G)$, where $x \in V(G) \setminus \{u, v\}$, we construct new edge wx (and possibly obtain multiple edges), and for each $e' = uv \neq e$, we add a new loop ww. For a vertex v, we denote by $N_G(v)$ the *(open) neighborhood* of v, i.e., the set of vertices that are adjacent to v in G. For a set $S \subseteq V(G)$, $N_G(S) = (\bigcup_{v\in S} N_G(v)) \setminus S$. We denote by $N_G[v] = N_G(v) \cup \{v\}$ the *closed neighborhood* of v. To vertices u and v are *true twins* if $N_G[u] = N_G[v]$, and u and v are *false twins* if $N_G(u) = N_G(v)$.

Cuts. Let G be a graph. A *cut* (A, B) of a graph G is a partition of $V(G)$ into two disjoint sets A and B. A set $S \subseteq E(G)$ is an *(edge) cut-set* if the deletion of S increases the number of components. A cut-set S is *(inclusion) minimal* if any proper subset of S is not a cut-set. A *bridge* is a cut-set of size one. For two disjoint vertex sets of vertices A and B of a graph G, $E(A, B) = \{uv \in E(G) \mid u \in A, v \in B\}$. Clearly, $E(A, B)$ is an edge cut-set, and for any cut-set $S \subseteq E(G)$, there is a cut (A, B) with $S = E(A, B)$. Notice also that $E(A, B)$ is a minimal cut-set of a connected graph G if and only if $G[A]$ and $G[B]$ are connected.

Matroids. We refer to the book of Oxley [27] for the detailed introduction to matroid theory. Recall that a matroid M is a pair $(E, \mathcal{I})$, where E is a finite *ground* set of M and $\mathcal{I} \subseteq 2^E$ is a collection of *independent* sets that satisfy the following three axioms:

I1. $\emptyset \in \mathcal{I}$,
I2. if $X \in \mathcal{I}$ and $Y \subseteq X$, then $Y \in \mathcal{I}$,
I3. if $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then there is $e \in Y \setminus X$ such that $X \cup \{e\} \in \mathcal{I}$.

We denote the ground set of M by $E(M)$ and the set of independent set by $\mathcal{I}(M)$ or simply by E and $\mathcal{I}$ if it does not create confusion. If a set $X \subseteq E$ is not independent, then X is *dependent*. Inclusion maximal independent sets are called *bases* of M. We denote the set of bases by $\mathcal{B}(M)$ (or simply by $\mathcal{B}$). The matroid M^* with the ground set $E(M)$ such that $\mathcal{B}(M^*) = \mathcal{B}^*(M) = \{E \setminus B \mid B \in \mathcal{B}(M)\}$ is *dual* to M.

An (inclusion) minimal dependent set is called a *circuit* of M. We denote the set of all circuits of M by $C(M)$ or simply C if it does not create a confusion. The circuits satisfy the following conditions (*circuit axioms*):

C1. $\emptyset \notin C$,
C2. if $C_1, C_2 \in C$ and $C_1 \subseteq C_2$, then $C_1 = C_2$,
C3. if $C_1, C_2 \in C, C_1 \neq C_2$, and $e \in C_1 \cap C_2$, then there is $C_3 \in C$ such that $C_3 \subseteq (C_1 \cup C_2) \setminus \{e\}$.

An one-element circuit is called *loop*, and if $\{e_1, e_2\}$ is a two-element circuit, then it is said that e_1 and e_2 are *parallel*. An element e is *coloop* if e is a loop of M^* or, equivalently, $e \in B$ for every $B \in \mathcal{B}$. A *circuit* of M^* is called *cocircuit* of M. A set $X \subseteq E$ is a *cycle* of M if X either empty or X is a disjoint union of circuits. By $\mathcal{S}(M)$ (or $\mathcal{S}$) we denote the set of all cycles of M. The sets of circuits and cycles completely define matroid. Indeed, a set is independent if and only if it does not contain a circuit, and the circuits are exactly inclusion minimal nonempty cycles.

Let M be a matroid, $e \in E(M)$. The matroid $M' = M - e$ is obtained by *deleting* e if $E(M') = E(M) \setminus \{e\}$ and $I(M') = \{X \in \mathcal{I}(M) \mid e \notin X\}$. We say that M' is obtained from M by *adding a parallel to e element* if $E(M') = E(M) \cup \{e'\}$, where e' is a new element, and $\mathcal{I}(M') = \mathcal{I}(M) \cup \{(X \setminus \{e\}) \cup \{e'\} \mid X \in \mathcal{I}(M) \text{ and } e \in X\}$. It is straightforward to verify that $\mathcal{I}(M')$ satisfies the axioms I.1-3, i.e., M' is a matroid with the ground set $E(M) \cup \{e'\}$. It is also easy to see that $\{e, e'\}$ is a circuit, that is, e and e' are parallel elements of M'.

We can observe the following.

Observation 2.1. *Let $\{e_1, e_2\}, C \in C$ for a matroid M. If $e_1 \in C$ and $e_2 \notin C$, then $C' = (C \setminus \{e_1\}) \cup \{e_2\}$ is a circuit.*

Proof. By the axiom C3, $(\{e_1, e_2\} \cup C) \setminus \{e_1\} = (C \setminus \{e_1\}) \cup \{e_2\} = C'$ contains a circuit C''. Suppose that $C'' \neq C'$. Notice that because $C \setminus \{e_1\}$ contains no circuit, $e_2 \in C''$. As $e_1 \notin C''$, we obtain that $(\{e_1, e_2\} \cup C'') \setminus \{e_2\}$ contains a circuit, but $(\{e_1, e_2\} \cup C'') \setminus \{e_2\}$ is a proper subset of C; a contradiction. Hence, $C'' = C'$, i.e., C' is a circuit. □

Matroids associated with graphs. Let G be a graph. The *cycle* matroid $M(G)$ has the ground set $E(G)$ and a set $X \subseteq E(G)$ is independent if $X = \emptyset$ or $G[X]$ has no cycles. Notice that C is a circuit of $M(G)$ if and only if C induces a cycle of G. The *bond* matroid $M^*(G)$ with the ground set $E(G)$ is dual to $M(G)$, and X is a circuit of $M^*(G)$ if and only if X is a minimal cut-set of G. Respectively, Minimum Spanning Circuit for a cycle matroid $M(G)$ is to decide whether G has a cycle C of weight at most ℓ that goes through the edges of T, and for a bond matroid $M^*(G)$ it is to decide whether G has a minimal cut-set C of weight at most ℓ that contains T. We say that M is a *graphic* matroid if M is isomorphic to $M(G)$ for some graph G. Respectively, M is *cographic* if there is graph G such that M is isomorphic to $M^*(G)$. Notice that $e \in E$ is a loop of a cycle matroid $M(G)$ if and only if e is a loop of G, and e is a loop of $M^*(G)$ if and only if e is a bridge of G.

Notice also that by the addition of an element parallel to $e \in E$ for $M(G)$ we obtain $M(G')$ for the graph G' obtained by adding a new edge with the same end vertices as e. Respectively, by adding of an element parallel to $e \in E$ for $M^*(G)$ we obtain $M^*(G')$ for the graph G' obtained by subdividing e. Hence, adding or deleting a parallel element of graphic or cographic matroid does not put it outside the corresponding class.

Matroid representations. Let M be a matroid and let F be a field. An $n \times m$-matrix A over F is a *representation of M over F* if there is one-to-one correspondence f between E and the set of columns of A such that for any $X \subseteq E$, $X \in \mathcal{I}$ if and only if the columns $f(X)$ are linearly independent (as vectors of F^n); if M has such a representation, then it is said that M has a *representation over F*. In other words, A is a representation of M if M is isomorphic to the *column matroid* of A, i.e., the matroid whose ground set is the set of columns of A and a set of columns is independent if and only if these columns are linearly independent. A matroid is *binary* if it can be represented over GF(2). A matroid is *regular* if it can be represented over any field. In particular, it is well-known that graphic and cographic matroids are regular [27]. Equivalently, a matroid is regular if it can be represented by a *totally unimodular* matrix over reals [27].

As we are working with binary matroids, we assume throughout the article that for an input matroid, we are given its representation over GF(2). Then it can be checked in polynomial time whether a subset of the ground set is independent by checking the linear independence of the corresponding columns.

3 STRUCTURE OF REGULAR MATROIDS

Our results for regular matroids use the structural decomposition for regular matroids given by Seymour [30]. Recall that, for two set X and Y, $X \triangle Y = (X \setminus Y) \cup (Y \setminus X)$ denotes the *symmetric difference* of X and Y. For our purpose we also need the following proposition.

Proposition 3.1 (see [27]). *Let C_1 and C_2 be circuits (cycles) of a binary matroid M. Then $C_1 \triangle C_2$ is a cycle of M.*

To describe the decomposition of matroids we need the notion of "r-sums" of matroids. However, for our purpose it is sufficient that we restrict ourselves to binary matroids and up to 3-sums. We refer to [33, Chapter 8] for a more detailed introduction to matroid sums. Let M_1 and M_2 be binary matroids. The *sum* of M_1 and M_2, denoted by $M_1 \triangle M_2$, is the matroid M with the ground set $E(M_1) \triangle E(M_2)$. The cycles of M are all subsets $C \subseteq E(M_1) \triangle E(M_2)$ of the form $C_1 \triangle C_2$, where C_1 is a cycle of M_1 and C_2 is a cycle of M_2. This does indeed define a binary matroid [30] as can be seen from Proposition 3.1, in which the circuits are the minimal nonempty cycles and the independent sets are (as always) the sets that do not contain any circuit. For our purpose the following special cases of matroid sums are sufficient.

Definition 3.1.

(1) If $E(M_1) \cap E(M_2) = \emptyset$ and $E(M_1), E(M_2) \neq \emptyset$, then M is the *1-sum* of M_1 and M_2 and we write $M = M_1 \oplus_1 M_2$.
(2) If $|E(M_1) \cap E(M_2)| = 1$, the unique $e \in E(M_1) \cap E(M_2)$ is not a loop or coloop of M_1 or M_2, and $|E(M_1)|, |E(M_2)| \geq 3$, then M is the *2-sum* of M_1 and M_2, and we write $M = M_1 \oplus_2 M_2$.
(3) If $|E(M_1) \cap E(M_2)| = 3$, the 3-element set $Z = E(M_1) \cap E(M_2)$ is a circuit of M_1 and M_2, Z does not contain a cocircuit of M_1 or M_2, and $|E(M_1)|, |E(M_2)| \geq 7$, then M is the *3-sum* of M_1 and M_2 and we write $M = M_1 \oplus_3 M_2$.

If $M = M_1 \oplus_r M_2$ for some $r \in \{1, 2, 3\}$, then we write $M = M_1 \oplus M_2$.

Definition 3.2. An $\{1, 2, 3\}$*-decomposition* of a matroid M is a collection of matroids $\mathcal{M}$, called the *basic matroids* and a rooted binary tree T in which M is the root and the elements of $\mathcal{M}$ are the leaves such that any internal node is 1-, 2-, or 3-sum of its children.

We also need the special binary matroid R_{10} to be able to define the decomposition theorem for regular matroids. It is represented over GF(2) by the 5×10-matrix whose columns are formed by vectors that have exactly three non-zero entries (or rather three ones) and no two columns are identical:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Now we are ready to give the decomposition theorem for regular matroids due to Seymour [30].

Theorem 1 ([30]). *Every regular matroid M has an $\{1, 2, 3\}$-decomposition in which every basic matroid is either graphic, cographic, or isomorphic to R_{10}. Moreover, such a decomposition (together with the graphs whose cycle and bond matroids are isomorphic to the corresponding basic graphic and cographic matroids) can be found in time polynomial in $|E(M)|$.*

For our algorithmic purposes, we will not use Theorem 1 but rather a modification proved by Dinitz and Kortsarz [11]. Dinitz and Kortsarz [11] observed that some restrictions in the definitions of 2- and 3-sums are not important for the algorithmic purposes. In particular, in the definition of the 2-sum, the unique $e \in E(M_1) \cap E(M_2)$ is not a loop or coloop of M_1 or M_2, and $|E(M_1)|, |E(M_2)| \geq 3$ could be dropped. Similarly, in the definition of 3-sum the conditions that $Z = E(M_1) \cap E(M_2)$ does not contain a cocircuit of M_1 or M_2, and $|E(M_1)|, |E(M_2)| \geq 7$ could be dropped. We define *extended* 1-, 2- and 3-sums by omitting these restrictions. Clearly, Theorem 1 holds if we replace sums by extended sums in the definition of the $\{1, 2, 3\}$-decomposition. To simplify notations, we use $\oplus_1, \oplus_2, \oplus_3$, and $\oplus$ to denote these extended sums. Finally, we also need the notion of a conflict graph associated with an $\{1, 2, 3\}$*-decomposition* of a matroid M given by Dinitz and Kortsarz [11].

Definition 3.3 ([11]). Let $(T, \mathcal{M})$ be a $\{1, 2, 3\}$-decomposition of a matroid M. The *intersection* (or *conflict*) graph of $(T, \mathcal{M})$ is the graph G_T with the vertex set $\mathcal{M}$ such that distinct $M_1, M_2 \in \mathcal{M}$ are adjacent in G_T if and only if $E(M_1) \cap E(M_2) \neq \emptyset$.

Dinitz and Kortsarz [11] showed how to modify a given decomposition to make the conflict graph a forest. In fact, they proved a slightly stronger condition that for any 3-sum (which by definition is summed along a circuit of size 3), the circuit in the intersection is contained entirely in two of the lowest-level matroids. In other words, while the process of summing matroids might create new circuits that contain elements that started out in different matroids, any circuit that is used as the intersection of a sum existed from the very beginning.

This allows us to choose in which order to perform 1-, 2-, or 3-sums. More formally, let $\mathcal{T}$ be a decomposition tree whose nodes are matroids from $\mathcal{M}$ such that the following holds. For any two distinct matroids $M_1, M_2 \in \mathcal{M}$ with a nonempty intersection

(i) M_1 and M_2 are adjacent in $\mathcal{T}$, and
(ii) $|E(M_1) \cap E(M_2)| \in \{1, 3\}$.

For every subtree $\mathcal{T}'$ of $\mathcal{T}$, we define $M_{\mathcal{T}'}$ associated with $\mathcal{T}'$ as follows. If $\mathcal{T}'$ is an isolated node, then $M_{\mathcal{T}'}$ is the corresponding matroid of $\mathcal{M}$. Otherwise, we select an arbitrary edge $e = M_1M_2 \in E(\mathcal{T}')$ and consider the subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ of $\mathcal{T}' - e$. We construct $M_{\mathcal{T}_1}$ and $M_{\mathcal{T}_2}$ and define $M_{\mathcal{T}'} = M_{\mathcal{T}_1} \oplus_i M_{\mathcal{T}_2}$ for $i = |E(M_1) \cap E(M_2)| + 1$ if $|E(M_1) \cap E(M_2)| \leq 1$ and $i = 3$ otherwise. This way we construct $M_{\mathcal{T}}$. Such a decomposition is not uniquely defined as the choice of edges for constructing the decomposition was arbitrary. Nevertheless, the conditions (i) and (ii) guarantee that the construction can be performed as described and, moreover, that the matroid $M_{\mathcal{T}}$ is unique.

We state the result of [11] in the following form that is convenient for us.

Theorem 2 ([11]). *For a given regular matroid M, there is a (conflict) tree $\mathcal{T}$, whose set of nodes is a set of matroids $\mathcal{M}$, where each element of $\mathcal{M}$ is a graphic or cographic matroid, or a matroid obtained from R_{10} by (possible) deleting some elements and adding parallel elements, that has the following properties:*

(i) if two distinct matroids $M_1, M_2 \in \mathcal{M}$ have nonempty intersection, then M_1 and M_2 are adjacent in $\mathcal{T}$,
(ii) for any distinct $M_1, M_2 \in \mathcal{M}$, $|E(M_1) \cap E(M_2)| = 0, 1$ or 3,
(iii) $M = M_{\mathcal{T}}$.

Moreover, $\mathcal{T}$ can be constructed in a polynomial time.

If $\mathcal{T}$ is a conflict tree for a matroid M, then we say that $M = M_{\mathcal{T}}$ is defined by $\mathcal{T}$.

In our algorithms we are working with rooted conflict trees. Fixing a *root* r in $\mathcal{T}$ defines the natural parent–child, descendant, and ancestor relationships on the nodes of $\mathcal{T}$. Our algorithms are based on performing *bottom-up* traversal of the tree $\mathcal{T}$. We say that a node M_ℓ of $\mathcal{T}$ is a *leaf* if it has no children, and M_s is a *sub-leaf* if it has at least one child and the children of M_s are leaves. Let M_ℓ be a leaf and let M_s be its adjacent sub-leaf. We say that M_ℓ is an *h-leaf* for $h \in \{1, 2, 3\}$ if the edge between M_s and M_ℓ corresponds to the extended h-sum.

As in Minimum Spanning Circuit and Spanning Circuit we are looking for circuits containing terminals, we need some results about the structure of circuits of matroids and matroid sums.

Lemma 3.1. *Let $Z = \{e_1, e_2, e_3\}$ be a circuit of a binary matroid M. Let also C be a circuit of M such that $C \cap Z = \{e_3\}$. If $C' = C \triangle Z$ is not a circuit, then C' is a disjoint union of two circuits C_1 and C_2 containing e_1 and e_2, respectively, and $C_1 \triangle Z$ and $C_2 \triangle Z$ are circuits.*

Proof. By Proposition 3.1, C' is a cycle of M. If C' is not a circuit, then C' is a disjoint union of circuits of M. If C' contains a circuit C'' such that $C'' \cap Z = \emptyset$, then $C'' \subset C$ contradicting the condition that C is a minimal dependent set. Hence, each circuit of C' contains an element of

Z. Since $Z \cap C' = \{e_1, e_2\}$, C' is a disjoint union of two circuits C_1 and C_2 containing e_1 and e_2, respectively.

Suppose that, say, $C_1 \triangle Z$, is not a circuit. Then by the above, $C_1 \triangle Z$ is a disjoint union of two circuits C_2' and C_3' containing e_2 and e_3, respectively. But then $C'' = C_2 \triangle C_2'$ is a cycle and $C'' \subset C$ contradicting that C is a circuit. Hence, $C_1 \triangle Z$ and $C_2 \triangle Z$ are circuits. □

Lemma 3.2. *Let $Z = \{e_1, e_2, e_3\}$ be a circuit of a binary matroid M. Let also C be a circuit of M such that $C \cap Z = \{e_1, e_2\}$. Then $C' = C \triangle Z$ is a circuit of M.*

Proof. By Proposition 3.1, C' is a cycle of M. Because $e_3 \in C'$, there is a circuit $C'' \subseteq C'$ containing e_3. If $C'' \neq C'$, then the cycle $C'' \triangle Z \subset C$ contradicting the fact that C is a circuit. Hence, $C' = C''$, i.e., C' is a circuit. □

Lemma 3.3. *Let $M = M_1 \oplus_r M_2$ for $r \in \{1, 2, 3\}$, where M_1 and M_2 are binary matroids, and $Z = E(M_1) \cap E(M_2)$.*

(i) If $r = 1$, then $\mathcal{C}(M) = \mathcal{C}(M_1) \cup \mathcal{C}(M_2)$.
(ii) If $r = 2$ and $Z = \{e\}$, then

$$\mathcal{C}(M) = \{C \in \mathcal{C}(M_1) \mid e \notin C\} \cup \{C \in \mathcal{C}(M_2) \mid e \notin C\} \cup \{C_1 \triangle C_2 \mid C_1 \in \mathcal{C}(M_1), C_2 \in \mathcal{C}(M_2), e \in C_1, e \in C_2\}.$$

(iii) If $r = 3$, then

$$\begin{aligned}\mathcal{C}(M) =&\{C \in \mathcal{C}(M_1) \mid Z \cap C = \emptyset\} \cup \{C \in \mathcal{C}(M_2) \mid Z \cap C = \emptyset\}\\ &\cup\{C_1 \triangle C_2 \mid C_1 \in \mathcal{C}(M_1), C_2 \in \mathcal{C}(M_2), C_1 \cap Z = \{e\} \text{ and } C_2 \cap Z = \{e\}\\ &\text{for some } e \in Z, \text{ and } C_1 \triangle Z \in \mathcal{C}(M_1) \text{ or } C_2 \triangle Z \in \mathcal{C}(M_2)\}.\end{aligned}$$

Proof. The claims (i) and (ii) follow directly from Definition 3.1. Hence, we have to prove only (iii). Recall that Z is a circuit of M_1 and M_2 in the case of the extended 3-sum. Notice that the structure of $\mathcal{C}(M)$ is more complicated in this case. In particular, if $C_1 \in \mathcal{C}(M_1)$, $C_2 \in \mathcal{C}(M_2)$, $C_1, C_2 \neq Z$, and $C_1 \cap Z = C_2 \cap Z \neq \emptyset$, then $C = C_1 \triangle C_2$ is a cycle of M by the definition, but C is not necessarily a circuit. In fact, it may happen that C is a disjoint union of two circuits. We show that this happens if and only if $C_1 \triangle Z$ and $C_2 \triangle Z$ are circuits of M_1 and M_2, respectively. Now we proceed with the formal proof.

Let C be a circuit of M. If $C \subseteq E(M_i)$ for $i \in \{1, 2\}$, then C is a cycle of M_i and, by minimality, C is a circuit of M_i. Assume that $C \setminus E(M_i) \neq \emptyset$ for each $i \in \{1, 2\}$. By definition, $C = C_1 \triangle C_2$ and $C_1 \cap Z = C_2 \cap Z$, where C_1 and C_2 are cycles of M_1 and M_2, respectively.

If $Z \subseteq E(C_1)$, then by Proposition 3.1, $C' = C_1 \triangle Z \subseteq C$ is a cycle of M_1. Hence, C' is a cycle of M contradicting that C is a minimal dependent set. If $C_1 \cap Z = \emptyset$, then $C_1 \subset C$ is a circuit of M and this contradicts the minimality of C. Therefore $1 \leq |C_1 \cap Z| \leq 2$. Suppose that $|C_1 \cap Z| = 2$. Consider $C_1' = C_1 \triangle Z$ and $C_2' = C_2 \triangle Z$. By Proposition 3.1, C_i' is a cycle of M_i, $i \in \{1, 2\}$. Clearly, $C = C_1' \triangle C_2'$, but now $|C_1' \cap Z| = |C_2' \cap Z| = 1$. It means that we always can assume that $C = C_1 \triangle C_2$, where $C_1 \cap Z = \{e\}$ and $C_2 \cap Z = \{e\}$ for some $e \in Z$.

Suppose that one of the cycles C_1 and C_2, say, C_1, is not a circuit. Then C_1 is a disjoint union of circuits of M_1. This union contains a circuit C_1' with $e \in C_1'$. Then $C' = C_1' \triangle C_2 \subset C$ is a cycle of M contradicting the minimality of C. Hence, C_1 and C_2 are circuits of M_1 and M_2, respectively.

Suppose that $C_1' = C_1 \triangle Z$ and $C_2' = C_2 \triangle Z$ are not circuits of M_1 and M_2, respectively. By Lemma 3.1, for $i \in \{1, 2\}$, C_i' is a disjoint union of two circuits C_i^1 and C_i^2 of M_i containing e_1 and e_2, respectively, for distinct $e_1, e_2 \in Z \setminus \{e\}$. Then $C' = C_1^1 \triangle C_2^1$ is a cycle of M contradicting the minimality of C. Hence, for at least one $i \in \{1, 2\}$, C_i' is a circuit of M_i.

In the opposite direction, if C is a circuit of M_1 or M_2 such that $C \cap Z = \emptyset$, then C is a circuit of M. Suppose now that $C = C_1 \triangle C_2$, where C_1 and C_2 are circuits of M_1 and M_2, respectively, $C_1 \cap Z = \{e\}$ and $C_2 \cap Z = \{e\}$ for some $e \in Z$, and $C_1 \triangle Z$ or $C_2 \triangle Z$ is a circuit of M_1 or M_2, respectively. We show that C is a circuit of M.

To obtain a contradiction, assume that C is not a circuit. By Proposition 3.1, C is a cycle of M. Therefore, there is a circuit $C' \subset C$. If $C' \subseteq E(M_1)$ or $C' \subseteq E(M_2)$, then $C' \subset C_1$ or $C' \subset C_2$, but this contradicts the condition that C_1 and C_2 are circuits of M_1 and M_2, respectively. Hence, $C' \setminus E(M_1) \neq \emptyset$ and $C' \setminus E(M_2) \neq \emptyset$. As we already proved above, $C' = C_1' \triangle C_2'$, where C_i' is a circuit of M_i, $i \in \{1, 2\}$, and $C_i' \cap Z = \{e'\}$ and $C_2' \cap Z = \{e'\}$ for some $e' \in Z$. Clearly, $C_1' \setminus \{e'\} \subseteq C_1 \setminus \{e\}$ and $C_2' \setminus \{e'\} \subseteq C_2 \setminus \{e\}$ and at least one of the inclusions is proper. If $e' = e$, then $C_1' \subseteq C_1$ and $C_2' \subseteq C_2$ and at least one of the inclusions is proper contradicting the fact that C_1 and C_2 are circuits of M_1 and M_2, respectively. Hence, $e' \neq e$. If $C_1' \setminus \{e'\} = C_1 \setminus \{e\}$, then $\{e, e'\} = C_1' \triangle C_1$. This contradicts the condition that Z is a circuit. Hence, $C_1' \setminus \{e'\} \subset C_1 \setminus \{e\}$. But then $C_1' \subset C_1 \triangle Z$, and therefore $C_1 \triangle Z$ is not a circuit of M_1. Symmetrically, $C_2 \triangle Z$ is not a circuit of M_2; a contradiction. Hence, C is a circuit of M. □

Lemma 3.3 gives an idea how to solve Minimum Spanning Circuit and Spanning Circuit using Theorem 2 and, simultaneously, indicates the main technical difficulties. First, we should be able to solve the problems on basic matroids. Then we "glue" circuits together using Lemma 3.3. Assume that $M = M_1 \oplus M_2$, $T_1 = T \cap E(M_1) \neq \emptyset$, and $T_2 = T \cap E(M_2)$. Suppose that we are able to solve, say, Minimum Spanning Circuit for M_1 and M_2. If $M = M_1 \oplus_1 M_2$, then it is trivial to solve Minimum Spanning Circuit on M: If $T_2 \neq \emptyset$, then we have a no-instance, and, otherwise, we should solve the problem on M_1. The case $M = M_1 \oplus_2 M_2$ is more complicated but still straightforward. Let $\{e\} = E(M_1) \cap E(M_2)$. We have two subcases: $T_2 = \emptyset$ and $T_2 \neq \emptyset$. In the first subcase, we find a circuit of minimum weight ω in M_2 containing e by solving the auxiliary instance of Minimum Spanning Circuit on M_2 with the unique terminal e assuming that $w(e) = 0$. Then we assign the weight ω to e in M_1 and solve the instance of the problem on M_1. In the second subcase, when $T_2 \neq \emptyset$, we find a solution of minimum weight ω for Minimum Spanning Circuit on M_2 with the set of terminals $T_2 \cup \{e\}$ assuming that $w(e) = 0$. Then we assign the weight ω to e in M_1 and solve the problem on M_1 with the set of terminals $T_1 \cup \{e\}$. For $M = M_1 \oplus_3 M_3$, we follow the same strategy, but here the situation is more difficult as demonstrated by Lemma 3.3 (iii). In particular, if $T_2 \neq \emptyset$, we obtain six cases that should be analyzed. Let $Z = \{e_1, e_2, e_3\} = E(M_1) \cap E(M_2)$. Then a (hypothetical) solution C for Minimum Spanning Circuit on M can be represented as $C = C_1 \triangle C_2$ for circuits C_1 and C_2 of M_1 and M_2, respectively, such that $C_1 \cap C_2 = \{e_i\}$ and $C_1 \cap Z = C_2 \cap Z = \{e_i\}$ for some $i \in \{1, 2, 3\}$, and this gives us three possibilities. Then, for each $i \in \{1, 2, 3\}$, we either demand $Z \triangle C_1$ be a circuit of M_1 or, symmetrically, demand $Z \triangle C_2$ be a circuit of M_2. To be able to go through these cases, we have to switch from the original Minimum Spanning Circuit to the special version of the problem tailored for this analysis that is formally defined in Section 5. For Spanning Circuit, the situation is slightly more simple as we have no weights, but we still have to overcome the same difficulties.

We conclude this section by the following lemma about circuits in graphic and cographic matroids. We need this lemma to be able to impose the conditions that, given a circuit Z of size three, $C \triangle Z$ is a circuit of a graphic (cographic) matroid for a circuit C.

Lemma 3.4. *Let $Z = \{e_1, e_2, e_3\}$ be a circuit of a binary matroid M. Let also C be a circuit of M such that $C \cap Z = \{e_3\}$. Then the following holds:*

(i) If $M = M(G)$ for a graph G, then $C' = C \triangle Z$ is a circuit of M if and only if C induces a cycle of $G - v$, where v is the vertex of G incident with e_1 and e_2.

(ii) *If $M = M^*(G)$ for a connected graph G, then $C' = C \triangle Z$ is a circuit of M if and only if $C = E(A,B)$ for a cut (A,B) of G such that $G[A]$ and $G[B]$ are connected graphs and $e_1, e_2 \in E(G[A])$ or $e_1, e_2 \in E(G[B])$.*

Proof. The first claim is straightforward. To show (ii), recall that C is a minimal cut-set of G. Hence, there is a cut (A,B) of G such that $C = E(A,B)$ and $G[A]$ and $G[B]$ are connected.

Assume that $e_1 \in E(G[A])$ and $e_2 \in E(G[B])$. Since Z is a minimal cut-set of G, we have that e_1 and e_2 are bridges of $G[A]$ and $G[B]$, respectively. Then $C \triangle Z$ is a cut-set separating G into three components. Hence C' is not a minimal cut-set, which is a contradiction. Therefore, $e_1, e_2 \in E(G[A])$ or $e_1, e_2 \in E(G[B])$.

Suppose now that $C = E(A,B)$ for a cut (A,B) of G such that $G[A]$ and $G[B]$ are connected and $e_1, e_2 \in E(G[A])$. Because Z is a minimal cut-set, $\{e_1, e_2\}$ is a minimal cut-set of $G[A]$. Let (A_1, A_2) be a cut of $G[A]$ such that $E(A_1, A_2) = \{e_1, e_2\}$. Assume that the end-vertex of e_3 in A is in A_1. Since Z is a minimal cut-set, the edges of $C \setminus \{e_3\}$ join A_2 with B. It implies that $C \triangle Z$ is a minimal cut-set that separates A_2 and $A_1 \cup B$. □

4 MINIMAL CUT WITH SPECIFIED EDGES

To construct an algorithm for Minimum Spanning Circuit for regular matroids, we need an algorithm for cographic matroids. Let G be a connected graph, and let $T \subseteq E(G)$ be a set of terminal edges. For sets $R_1, R_2 \subseteq V(G)$, we say that $C \subseteq E(G)$ is (R_1, R_2)-*terminal cut-set* if C is (a) a minimal cut-set; (b) $C \supseteq T$; and (c) $G - C$ contains distinct connected components X_1 and X_2 such that $R_i \subseteq X_i$ for $i \in \{1, 2\}$.

We will need solve the following auxiliary parameterized problem.

Minimal Terminal Cut parameterized by k

Input:	A connected graph G, a weight function $w\colon E(G) \to \mathbb{N}$, a set of terminals $T \subseteq E(G)$, sets $R_1, R_2 \subseteq V(G)$, and a positive integer k.
Task:	Decide whether G contains an (R_1, R_2)-terminal cut-set C such that $w(C) - w(T) \leq k$.

We say that an (R_1, R_2)-terminal cut-set C with the required weight is a *solution* of Minimal Terminal Cut. Observe that if in the instance of Minimal Terminal Cut we have $R_1 \cap R_2 \neq \emptyset$, then the problem does not have a solution and this is a no-instance.

Notice that in the special case when $R_1 = R_2 = \emptyset$, Minimal Terminal Cut asks whether G has a minimal cut-set C containing T such that $w(C) - w(T) \leq k$. This means that Minimal Terminal Cut parameterized by k is equivalent to Minimum Spanning Circuit parameterized by $k = \ell - w(T)$ on $M^*(G)$. Nevertheless, we have to consider nonempty sets R_1 and R_2 to be able to impose the following additional condition on solution C: Set $C \triangle Z$ should be a minimal cut-set (a circuit of $M^*(G)$) for a given cut-set Z of size three as it is explained in Lemma 3.4 (ii). We will need this to invoke Theorem 2. For our purposes, it is sufficient to consider the cases when R_1 and R_2 are either empty or contain the end-vertices of two edges of the cut-set Z, but it is more convenient to solve Minimal Terminal Cut for arbitrary R_1 and R_2. We also believe that Minimal Terminal Cut is interesting in its own. In what follows, we prove that Minimal Terminal Cut is FPT.

Theorem 3. *Minimal Terminal Cut is solvable in time $2^{O(k^2 \log k)} \cdot n^{O(1)}$.*

The proof of Theorem 3 is technical and is given in the remaining part of the section. It is based on a (non-trivial) application of the recent algorithmic technique of *recursive understanding* introduced by Chitnis et al. [7].

4.1 Preliminaries

First, we introduce some notions required for the proof of Theorem 3.

Let G be a graph, $X \subseteq V(G)$. We say that G' is obtained from G by the *contraction of* X, if we get G' by deleting the vertices of X and replacing by a vertex x, and then each edge $uv \in E(G)$ with $u, v \in X$ is replaced by a loop xx, and each edge $uv \in E(G)$ with $u \in X$ and $v \notin X$ is replaced by xv. Notice that while contracting, we do not reduce the number of edges and that we can obtain loops and multiple edges by this operation. For an edge weighted graph, we assume that every new edge has the same weight as the edge it replaces. To simplify notations, throughout this section we also assume that if the contraction is done for some set $X \subseteq V(G)$ in an instance (G, w, T, R_1, R_2, k) of Minimal Terminal Cut, then if $X \cap R_i \neq \emptyset$ for $i \in \{1, 2\}$, then the vertex obtained from X is in R_i, and if a terminal edge is replaced, then the obtained edge is included in T.

For a set X, we denote by $\mathcal{P}(X)$ the set of all partitions of X. We assume that $\mathcal{P}(X) = \emptyset$ if $X = \emptyset$.

The main idea behind the recursive understanding technique [7] is the following. We try to find a minimal cut-set of bounded size that separates an input graph into two sufficiently big parts. If such a cut-set exists, then we solve the problem recursively for one of the parts and replace this part by an equivalent graph of bounded size; the equivalence here means that the replacement keeps all essential solutions of the original part. In our case, the replacement is obtained by contracting some edges. This way, we obtain a graph of smaller size. If the input graph has no cut-set with the required properties, then it either has a bounded size or has high connectivity. In the case of the bounded size graph we can apply brute force, and if the graph is highly connected, then we can exploit this property to solve the problem. To define formally what we mean by high connectivity, we need the following definition.

Definition 4.1 ([7]). Let G be a connected graph and let p, q be positive integers. A cut (A, B) of G is called a (q, p)*-good edge separation* if

(i) $|A|, |B| > q$,
(ii) $|E(A, B)| \leq p$,
(iii) $G[A]$ and $G[B]$ are connected.

Chitnis et al. proved the following lemma [7].

Lemma 4.1 ([7]). *There exists a deterministic algorithm that, given a connected graph G along with integers p and q, in time $2^{O(\min\{p,q\} \log(p+q))} \cdot n^3 \log n$ either finds a (q, p)-good edge separation or correctly concludes that no such separation exists.*

Let G be a connected graph and let p, q be positive integers. We say that G is (q, p)-unbreakable if there is no cut (A, B) of G such that

(i) $|A|, |B| > q$, and
(ii) $|E(A, B)| \leq p$,

Notice that in this definition, it is not required that $G[A]$ and $G[B]$ should be connected. It particular, this means that every (q, p)-unbreakable graph has no (q, p)-good edge separation but not the other way around. We use Lemma 4.1 to show the following.

Lemma 4.2. *There exists a deterministic algorithm that, given a connected graph G along with integers p and q, in time $2^{O(\min\{p,q\} \log(p+q))} \cdot n^3 \log n$ either finds a (q, p)-good edge separation or correctly concludes that G is (pq, p)-unbreakable.*

Proof. We use Lemma 4.1 to find a (q, p)-good edge separation. If the algorithm returns a (q, p)-good edge separation, then we return it. Assume that the algorithm reported that no such separation exists. We claim that G is (pq, p)-unbreakable. To obtain a contradiction, assume that (A, B)

is a cut of G such that $|A|, |B| > pq$ and $|E(A,B)| \leq p$. Consider $G[A]$. Because G is connected and $|E(A,B)| \leq p$, $G[A]$ has at most p components. Hence, G has a component H_A with at least $q+1$ vertices. Symmetrically, we obtain that $G[B]$ has a components H_B with at least $q+1$ vertices. Let C be a minimum cut-set in G that separates $V(H_A)$ and $V(H_B)$. Clearly, $|C| \leq p$. Let (A', B') be the cut of G with $V(H_A) \subseteq A'$, $V(H_B) \subseteq B'$ and $E(A', B') = C$. We have that (A', B') is a (q,p)-good separation, but it contradicts the assumption that the algorithm reported that there is no such a separation. □

We use Lemma 4.2 to find a (q,p)-good edge separation for appropriate p and q. If such a cut (A,B) exists, then we solve the problem recursively for one of the parts, say, for $G[A]$. But to be able to obtain a solution for the original instance, we should combine solutions for the both parts. We use the fact that $G[A]$ is separated from the remaining part of the graph by a small number of vertices that are the end-vertices of the edges of the cut-set that are called *border terminals.* (In fact, we keep $2p$ border terminals to execute the recursive step.) As we want to find all essential solutions for $G[A]$ to replace this graph by a graph of bounded size, we have to take into account all possibilities for the part of a solution in B to separate the border terminals.

This leads us to the following definition. Let (G, w, T, R_1, R_2, k) be an instance of Minimal Terminal Cut given together with a set $X \subseteq V(G)$ of border terminals of G. We say that an instance $(\hat{G}, w, T, \hat{R}_1, \hat{R}_2, \hat{k})$ of Minimal Terminal Cut is obtained from (G, w, T, R_1, R_2, k) by *border contraction* if $\hat{k} \leq k$ and there is a partition $(X_1, \ldots, X_t) \in \mathcal{P}(X)$ and partition (I_1, I_2) of $\{1, \ldots, t\}$, where I_i can be empty, such that $\hat{G}$ is obtained by consecutively contracting $X_1, \ldots, X_t$, and setting $\hat{R}_i = R_i \cup \{x_j \mid j \in I_i\}$ for $i \in \{1,2\}$, where each x_j is the vertex obtained from X_j by contraction. Let us note that the total number of different border contractions of a given instance depends only on the size of X and k and is $k \cdot |X|^{O(|X|)}$.

It leads us to the following auxiliary problem. In this problem, we have to output a solution (if there is any) for each of the instances of Minimal Terminal Cut obtained by all possible border contractions of a given instance. Notice that this is not a decision problem.

Border Contractions parameterized by k

Input:	A connected graph G, a weight function $w\colon E(G) \to \mathbb{N}$, a set of terminals $T \subseteq E(G)$, sets $R_1, R_2 \subseteq V(G)$, a positive integer k, and a sets of border terminals $X \subseteq V(G)$ with $|X| \leq 4k$.
Task:	Output for each possible instance of Minimal Terminal Cut which can be obtained from (G, w, T, R_1, R_2, k) by border contractions of X a solution, if there is any. In a case when a border contraction instance has no solution, output $\emptyset$.

Thus an output for Border Contractions is a family of edge sets, where the total number of edges in the solution is at most $k \cdot (4k)^{4k} \cdot 2^{4k} = 2^{O(k \log k)}$. . Notice also that to solve Minimal Terminal Cut , we can apply an algorithm for Border Contractions for the special case $X = \emptyset$.

4.2 High-connectivity Phase

In this section, we construct an algorithm for Border Contractions for the case when an input graph is (pq, p)-unbreakable for $p = 2k$ and $q = k^2 \cdot 2^{4k + 4k \log 4k} + 4k + 1$; we fix the values of p and q for the remaining part of Section 4. First, we solve Minimal Terminal Cut and then explain how to obtain the algorithm for Border Contractions.

Lemma 4.3. *Let G be a graph with an edge weight function $w : E(G) \to \mathbb{N}$, $T \subseteq E(G)$ and let k be a positive integer. It can be decided in time $2^{O(k)} \cdot n^{O(1)}$ whether there is a cut (A, B) of G such that $T \subseteq E(A, B)$, and $w(E(A, B) \setminus T) \leq k$.*

Proof. We show the lemma by the reduction of the problem to the Odd Cycle Transversal (OCT) problem. Let us remind that in the OCT problem we are given a graph G and a positive integer k, the task is to decide whether there is a set of at most k vertices S such that $G - S$ is bipartite. Since OCT is known to be solvable in time $2^{O(k)} \cdot n^{O(1)}$, this will prove the lemma.

Let G be a graph with an edge weight function $w : E(G) \to \mathbb{N}$, $T \subseteq E(G)$, and let k be a positive integer. Recall that we allow loops and multiple edges. To slightly simplify reduction, we first exhaustively apply two simple reduction rules.

If $e \in T$ is a loop, then $e \notin E(A, B)$ for any cut (A, B). If a loop $e \notin T$, then e is irrelevant. Hence we have the following reduction rule.

Reduction Rule 4.1 (Loop reduction rule). *Let $e \in E(G)$ be a loop. If $e \notin T$, then delete e. Otherwise (if $e \in T$), report that there is no required cut (A, B).*

Clearly, any two parallel edges are either both included in a cut-set or both are excluded from it. Notice also that the weights of terminals are irrelevant. Hence, we can safely apply the following rule.

Reduction Rule 4.2 (Parallel terminal reduction rule). *If there are two parallel edges $e_1, e_2 \in T$, then delete one of them and change the weight of the remaining edge to 1.*

From now on we assume that the rules cannot be applied. We construct (unweighted) graph G' from G as follows.

- Subdivide each edge $uv \notin T$, that is, add a new vertex z_{uv} and replace uv by uz_{uv} and vz_{uv}; we call the new vertices *subdivision* vertices.
- Replace each subdivision vertex z_{uv} by $r = \min\{w(uv), k + 1\}$ false twins, i.e., we replace z_{uv} by r vertices adjacent to u and v; denote by Z_{uv} the set of obtained vertices.
- Replace each vertex v of $V(G)$ by $k + 1$ false twins, i.e., we replace v by $k + 1$ vertices with the same neighbors as v; denote by U_v the set of obtained vertices.

Notice that because of reduction rules, G' is a simple graph. We claim that there is a cut (A, B) of G such that $T \subseteq E(A, B)$, and $w(E(A, B) \setminus T) \leq k$ if and only of (G', k) is a yes-instance of OCT.

Suppose that (A, B) is a cut of G such that $T \subseteq E(A, B)$, and $w(E(A, B) \setminus T) \leq k$. We construct the set $S \subseteq V(G')$ by including in S the set of vertices Z_{uv} for each $uv \in E(A, B) \setminus T$. Then $G' - S$ is bipartite.

Suppose that there is $S \subseteq V(G')$ of size at most k such that $G' - S$ is bipartite. Without loss of generality we assume that S is an inclusion minimal set with this property. Because S is minimal, if x and y are false twins of G, then either $x, y \in S$, or $x, y \notin S$. Let (X, Y) be a bipartition of $G' - S$. Since $|U_v| > k$, we have that $U_v \cap S = \emptyset$ for $v \in V(G)$. Notice also that we can assume that either $U_v \subseteq X$ or $U_v \subseteq Y$ for $v \in V(G)$, as otherwise, if there is $v \in V(G)$ such that $U_v \cap X \neq \emptyset$ and $U_v \cap Y \neq \emptyset$, then the vertices of U_v are isolated vertices of $G' - S$. Let $A = \{v \in V(G) \mid U_v \subseteq X\}$ and $B = \{v \in V(G) \mid U_v \subseteq Y\}$. Clearly, (A, B) is a cut of G. Let $uv \in T$. Assume that $U_u \subseteq X$. Then $U_v \subseteq Y$ and, therefore, $uv \in E(A, B)$. Let $uv \in E(A, B) \setminus T$ and assume that $u \in A$ and $v \in B$. Then $U_u \subseteq X$ and $U_v \subseteq Y$. Hence, $Z_{uv} \subseteq S$. Since $|Z_{uv}| = \min\{w(uv), k + 1\}$ and $|S| \leq k$, the total weight of the edges of $E(A, B) \setminus T$ is at most k.

This proves the correctness of the reduction. Since OCT can be solved in time $2.3146^k \cdot n^{O(1)}$ by the results of Lokshtanov et al. [23], we get the claim of the lemma. ◻

Let (A_1, B_1) and (A_2, B_2) be cuts of a graph G. We define the *distance* between these cuts as

$$\text{dist}((A_1, B_1), (A_2, B_2)) = \min\{|A_1 \triangle A_2|, |A_1 \triangle B_2|\}.$$

The following structural lemmata are crucial for our algorithm.

Lemma 4.4. *Let G be a graph with an edge weight function $w : E(G) \to \mathbb{N}$, set of terminals $T \subseteq E(G)$, and let k be a positive integer. Let (A_1, B_1) and (A_2, B_2) be cuts of G such that $T \subseteq E(A_i, B_i)$ and $w(E(A_i, B_i) \setminus T) \leq k$ for $i \in \{1, 2\}$. Then $w(E(A_1 \triangle A_2, A_1 \triangle B_2)) \leq 2k$.*

Proof. Notice that $(A_1 \triangle A_2, A_1 \triangle B_2)$ is a cut of G. For each $i \in \{1, 2\}$, we have that $T \subseteq E(A_i, B_i)$. Therefore, the set

$$E(A_1 \cap A_2, A_1 \cap B_2) \cup E(B_1 \cap A_2, B_1 \cap B_2) \cup E(A_2 \cap A_1, A_2 \cap B_1) \cup E(B_2 \cap A_1, B_2 \cap B_1)$$

does not contain edges from T.

Hence,

$$E(A_1 \cap A_2, A_1 \cap B_2) \cup E(A_2 \cap B_1, B_1 \cap B_2) \subseteq E(A_2, B_2) \setminus T,$$

and therefore,

$$w(E(A_1 \cap A_2, A_1 \cap B_2) \cup E(A_2 \cap B_1, B_1 \cap B_2) \leq k.$$

Symmetrically,

$$w(E(A_1 \cap A_2, A_2 \cap B_1) \cup E(A_1 \cap B_2, B_1 \cap B_2)) \leq k.$$

Since

$$A_1 \triangle A_2 = (A_1 \cap B_2) \cup (A_2 \cap B_1) \text{ and } A_1 \triangle B_2 = (A_1 \cap A_2) \cup (B_1 \cap B_2),$$

the claim follows. □

Let us recall that in this section we fix $p = 2k$ and $q = k2^{4k+4k\log 4k} + 4k + 1$.

Lemma 4.5. *Let G be a connected (pq, p)-unbreakable graph with an edge weight function $w : E(G) \to \mathbb{N}$, $T \subseteq E(G)$ and let k be a positive integer. Let (A_1, B_1) and (A_2, B_2) be cuts of G such that $T \subseteq E(A_i, B_i)$ and $w(E(A_i, B_i) \setminus T) \leq k$ for $i \in \{1, 2\}$. Then*

$$\text{dist}((A_1, B_1), (A_2, B_2)) \leq pq.$$

Proof. Aiming toward a a contradiction, we assume that $\text{dist}((A_1, B_1), (A_2, B_2)) > pq$. Let us note that $(A_1 \triangle A_2, A_1 \triangle B_2)$ is a partition of $V(G)$. Since $\text{dist}((A_1, B_1), (A_2, B_2)) > pq$, we have that $|A_1 \triangle A_2| > pq$ and $|A_1 \triangle B_2| > pq$. By Lemma 4.4, $w(E(A_1 \triangle A_2, A_1 \triangle B_2)) \geq |A_1 \triangle A_2, A_1 \triangle B_2)| \leq 2k = p$; contradicting the assumption that G is (pq, p)-unbreakable. □

Our algorithm for Minimal Terminal Cut uses the *random separation* technique proposed by Cai, Chan and Chan [5]. For derandomization, we use the following lemma proved by Chitnis et al. [7].

Lemma 4.6 ([7]). *Given a set U of size n, and integers $0 \leq a, b \leq n$, one can in time $2^{O(\min\{a,b\}\log(a+b))} \cdot n\log n$ construct a family $\mathcal{F}$ of at most $2^{O(\min\{a,b\}\log(a+b))} \cdot \log n$ subsets of U, such that the following holds: for any sets $A, B \subseteq U$, $A \cap B = \emptyset$, $|A| \leq a$, $|B| \leq b$, there exists a set $S \in \mathcal{F}$ with $A \subseteq S$ and $B \cap S = \emptyset$.*

Now we are ready to give the algorithm for Minimal Terminal Cut for unbreakable graphs.

Lemma 4.7. *Minimal Terminal Cut can be solved in $2^{O(k^2\log k)}n^{O(1)}$ time for (pq, p)-unbreakable graphs.*

Proof. Let (G, w, T, R_1, R_2, k) be an instance of Minimal Terminal Cut, where G is (pq, p)-unbreakable. If $n \leq pq$, then we solve the problem by the brute-force selection of at most k edges in time $2^{O(k^2\log k)}n^{O(1)}$. From now we assume that $n > pq$.

Using Lemma 4.3, we find a cut (A, B) of G such that $T \subseteq E(A, B)$ and $w(E(A, B) \setminus T) \leq k$. If such a cut does not exist, then we conclude that we are a given a no-instance.

Let (G, w, T, R_1, R_2, k) be a yes-instance and let $C = E(A', B')$ be a solution. Without loss of generality, we assume that $\mathrm{dist}((A, B), (A', B')) = |A \triangle A'|$. By Lemma 4.5, $|A \triangle A'| \leq pq$. It means that to solve the problem, we can either find a cut (A', B') or, equivalently, $A \triangle A'$ with these properties or conclude correctly that such a cut does not exist. First, we describe a randomized algorithm that finds $A \triangle A'$ and then explain how to derandomize it.

We randomly color the vertices of $V(G) \setminus (R_1 \cup R_2)$ by two colors *red* and *blue* with the probabilities $1 - \frac{1}{pq}$ and $\frac{1}{pq}$, respectively. We are looking for a set $X \subseteq V(G)$ such that the following holds:

(i) $|X| \leq pq$.
(ii) For $A' = A \triangle X$ and $B' = V(G) \setminus A'$, $C = E(A', B')$ is a solution for (G, w, T, R_1, R_2, k).
(iii) The vertices of X are red and the vertices of $N_G(X)$ are blue.

We say that $C = E(A', B')$ is a *colorful solution*.

The vertices of G are colored in red and in blue induce subgraphs that we call *red* and *blue* correspondingly. We also say that H is a *red component* if H is a connected component of the red (respectively, blue) subgraph of G. Because of (i)–(iii), we have the following properties:

- if H is a red component, then either $V(H) \subseteq X$ or $V(H) \cap X = \emptyset$,
- if $v \in V(G)$ is colored blue, then $v \notin X$.

We use (i)–(iii) and these properties to obtain reduction rules that recolor red components in blue, that is, each vertex of such a component becomes blue. We apply these rules exhaustively.

Since $T \subseteq E(A', B')$ if $C = E(A', B')$ is a solution for (G, w, T, R_1, R_2, k), we get the following rule.

Reduction Rule 4.3 (T-reduction Rule). *If there is $uv \in T$ such that u is red and v is blue, then recolor the red component H containing u in blue.*

We say that $uv \in E(G)$ is a *crossing edge for a red component H* if $u \in V(H)$, $v \notin V(H)$, and either $u \in A$ and $v \in B$ or $u \in B$ and $v \in A$. Notice that v is colored blue. Notice also that if H is a red component without crossing edges and $V(H) \subseteq X$, then for $A' = A \triangle X$ and $B' = V(G) \setminus A'$, $V(H) \cap A'$ and $V(H) \cap B'$ induce components of $G[A']$ and $G[B']$, respectively. If $|V(H)| \leq pq$, then we have that $G[A']$ or $G[B']$ is not connected, because $|V(G)| > pq$. Hence, $V(H) \cap X = \emptyset$ if $C = E(A', B')$ is a solution for (G, w, T, R, k). It gives us the next rule.

Reduction Rule 4.4 (Crossing reduction rule). *If there is a red component H without crossing edges, then recolor H blue.*

After the exhaustive applications of Rules 4.3 and 4.4, each red component H has crossing edges and these crossing edges are not in T. Since $w(E(A, B) \setminus C) \leq k$, the total number of crossing edges is at most k and, therefore, there are at most k red components. Because X is a union of some red components, we check all possibilities for X (the number of all possibilities is at most 2^k), and for each choice, we check whether $C = E(A', B')$ is a solution for (G, w, T, R_1, R_2, k). If we do not succeed in finding a solution for at least one of the choices, then we return that there is no solution.

Since Rules 4.3 and 4.4 can be run in polynomial time, a colorful solution for (G, w, T, R_1, R_2, k) can be found in time $2^k \cdot n^{O(1)}$.

Our next aim is to evaluate the probability of existence of a colorful solution for (G, w, T, R_1, R_2, k) if (G, w, T, R_1, R_2, k) is a yes-instance of Minimal Terminal Cut. Assume that (G, w, T, R_1, R_2, k) is a yes-instance and $C = E(A', B')$ is a solution, where (A', B') is a cut of G. We assume that $\mathrm{dist}((A, B), (A', B')) = |A \triangle A'|$. Let $X = A \triangle A'$. By Lemma 4.5, $|X| = \mathrm{dist}((A, B), (A', B')) \leq pq$. By Lemma 4.4, $|E(X, V(G) \setminus X)| \leq 2k$ and, therefore, $|N_G(X)| \leq 2k$.

Then the probability that the vertices of X are colored red and the vertices of $N_G(X)$ are colored blue is at least $(1 - \frac{1}{pq})^{pq} \cdot \frac{1}{(pq)^{2k}} \geq \frac{1}{4(pq)^{2k}}$ if $pq \geq 2$. If we run our randomized algorithm $N = 4(pq)^{2k}$ times, then the probability that we do not have a colorful solution for each of the N random colorings, is at most $(1 - \frac{1}{4(pq)^{2k}})^N \leq e^{-1}$. It means, that it is sufficient to run the algorithm N times to claim that if we do not find a solution for N random colorings, then with probability at least $1 - e^{-1} > 0$, (G, w, T, R_1, R_2, k) is a no-instance. In other words, we have a true-biased Monte-Carlo algorithm that runs in time $N \cdot 2^k \cdot n^{O(1)}$ if the initial partition (A, B) is given. Since $p = 2k$ and $q = k2^{4k+4k\log 4k} + 4k + 1$ and the initial partition (A, B) can be found in time $2^{O(k)} \cdot n^{O(1)}$, see Lemma 4.3, the total running time is $2^{O(k^2 \log k)} \cdot n^{O(1)}$.

To derandomize the algorithm, we use Lemma 4.6 for $a = q$, $b = p$ and $U = V(G)$. We construct the family $\mathcal{F}$ of subsets of $V(G)$ described in Lemma 4.6, and instead of random colorings, for each $S \in \mathcal{F}$, we consider the coloring of G such that the vertices of S are colored red and the vertices of $V(G) \setminus S$ are blue. Lemma 4.6 guarantees that (G, w, T, R_1, R_2, k) is a yes-instance of Minimal Terminal Cut if and only if we have a colorful solution for at least one of $|\mathcal{F}|$ colorings. Since $\mathcal{F}$ can be constructed in time $2^{O(k^2 \log k)} \cdot n^{O(1)}$ and $|\mathcal{F}| = 2^{O(k^2 \log k)} \cdot n^{O(1)}$, the running time of the derandomized algorithm is $2^{O(k^2 \log k)} \cdot n^{O(1)}$. □

We use Lemma 4.7, to solve Border Contractions.

Lemma 4.8. *Border Contractions can be solved in time $2^{O(k^2 \log k)} \cdot n^{O(1)}$ for (pq, p)-unbreakable graphs.*

Proof. Let $(G, w, T, R_1, R_2, k, X)$ be an instance of Border Contractions. Let us recall that the output of Border Contractions consists of solutions of Minimal Terminal Cut for all possible border contractions $(\hat{G}, w, T, \hat{R}_1, \hat{R}_2, \hat{k})$ of $(G, w, T, R_1, R_2, k, X)$. Notice that if G is (pq, p)-unbreakable, then each graph $\hat{G}$ is (pq, p)-unbreakable as well, because contractions of sets do not violate this property. We apply Lemma 4.7 for each instance $(\hat{G}, w, T, \hat{R}, \hat{k})$ of Minimal Terminal Cut. Since the number of all possible border contractions is in $2^{O(k \log k)}$, the total running time required to output the family of edge sets for Border Contractions is in $2^{O(k^2 \log k)} \cdot n^{O(1)}$. □

4.3 Proof of Theorem 3

We are ready to proceed with the proof of Theorem 3, which says that Minimal Terminal Cut is solvable in time $2^{O(k^2 \log k)} \cdot n^{O(1)}$. We give a recursive algorithm solving the more general Border Contractions. Then to solve Minimal Terminal Cut we solve the special case $X = \emptyset$ of Border Contractions. Recall that we fixed $p = 2k$ and $q = k2^{4k+4k\log 4k} + 4k + 1$.

Let $(G, w, T, R_1, R_2, k, X)$ be an instance of Border Contractions.

It is convenient to sort out a trivial case first. Notice that if for $(G, w, T, R_1, R_2, k, X)$ the set of terminal edges is a cut-set of the input graph but not a minimal cut-set, then this is a no-instance. It gives us the following rule.

Reduction Rule 4.5 (Stopping rule). *If graph $G - T$ has at least two components without border terminals, then output the empty set for every partition $(X_1, \ldots, X_t)$ and every partition (I_1, I_2) of $\{1, \ldots, t\}$.*

From now we assume that Stopping rule is not applicable to the given instance.

We apply Lemma 4.2 on G. If G is (pq, p)-unbreakable, then we apply Lemma 4.8 to solve the problem. Otherwise, the algorithm from Lemma 4.2 returns a (q, p)-good edge separation (U, W) of G.

The set of border terminals X has size at most $4k = 2p$. Hence, $|X \cap U| \leq p$ or $|C \cap W| \leq p$. Assume without loss of generality that $|X \cap U| \leq p$. Let $T' = T \cap E(G[U])$, $R'_1 = R_1 \cap U$, $R'_2 = R_2 \cap U$, and denote by w' the restriction of w on $E(G[U])$. We also define the set of border terminals

$$X' = (X \cap U) \cup \{v \in V(G[U]) \mid v \text{ is incident with an edge of } E(U,W)\};$$

observe that $|X'| \leq 2p = 4k$, because $|E(U,W)| \leq p$. We consider the instance $(G[U], w', T', R_1', R_2', k, X')$ of Border Contractions and solve the problem recursively.

Recall that the output of Border Contractions for $(G[U], w', T', R_1', R_2', k, X')$ is a family of solutions C for all possible border contractions. In other words, this is a family of solutions for instances $(\hat{G}', w', T', \hat{R}_1', \hat{R}_2', \hat{k})$ for all $\hat{k} \leq k$ such that a solution exist, and $\emptyset$ if there is no solutions. Each $\hat{G}'$ and $\hat{R}_i'$ is constructed as follows: For every partition $(X_1, \dots, X_t) \in \mathcal{P}(X')$ and every partition (I_1, I_2) of $\{1, \dots, t\}$, where I_i can be empty, we construct $\hat{G}'$ by consecutively contracting $X_1, \dots, X_t$, and set $\hat{R}_i' = R_i' \cup \{x_j \mid j \in I_i\}$ for $i \in \{1,2\}$, where each x_j is the vertex obtained from X_j by the contraction. For each of the subproblems, solution C is a set of edges of $G[U]$.

Denote by L the union of all sets generated by the algorithm for $(G[U], w', T', R_1', R_2', k, X')$. Let G'' be the graph obtained from G by contracting the edges of $E(G[U]) \setminus (L \cup T)$. Denote by $\alpha : V(G) \to V(G'')$ the mapping that maps each vertex $v \in V(G)$ to the vertex obtained from v by edge contractions. Let $R_1'' = \alpha(R_1)$, $R_2'' = \alpha(R_2)$ and $X'' = \alpha(X)$. Notice that the edges of T are not contracted. Denote by T'' the edges of G'' obtained from T; clearly, for each $uv \in T$, we have $\alpha(u)\alpha(v) \in T''$. For every $uv \in E(G)$ that was not contracted, the weight of the obtained edge $\alpha(u)\alpha(v)$ is $w''(\alpha(u)\alpha(v)) = w(uv)$. We obtain a new instance $(G'', w'', T'', R_1'', R_2'', k, X'')$ of Border Contractions. As before, we do not distinguish between the edges obtained by contracting edges or the original edges; thus $T'' = T$.

We claim that the original $(G, w, T, R_1, R_2, k, X)$ and new $(G'', w'', T'', R_1'', R_2'', k, X'')$ instances are equivalent in the following sense: there is a solution (in fact, every solution) for $(G'', w'', T'', R_1'', R_2'', k, X'')$ that is a solution for $(G, w, T, R_1, R_2, k, X)$, and there is a solution for $(G, w, T, R_1, R_2, k, X)$ that is a solution for $(G'', w'', T'', R_1'', R_2'', k, X'')$.

Lemma 4.9. *For every partition $(X_1, \dots, X_t) \in \mathcal{P}(X)$, every partition (I_1, I_2) of $\{1, \dots, t\}$, and every nonnegative $\hat{k} \leq k$, the instances $(\hat{G}, w, T, \hat{R}_1, \hat{R}_2, \hat{k})$ and $(\hat{G}'', w'', T'', \hat{R}_1'', \hat{R}_2'', \hat{k})$ of Minimal Terminal Cut are equivalent, where $\hat{G}$ is constructed from G by consecutive contracting $X_1, \dots, X_t$, $\hat{R}_i = R_i \cup \{x_j \mid j \in I_i\}$ for $i \in \{1,2\}$, where each x_j is the vertex obtained from X_j by contraction, and, respectively, $\hat{G}''$ is constructed from G'' by consecutive contracting $\alpha(X_1), \dots, \alpha(X_t)$, $\hat{R}_i'' = R_i'' \cup \{x_j \mid j \in I_i\}$ for $i \in \{1,2\}$, where each x_j is the vertex obtained from $\alpha(X_j)$ by contraction.*

Proof. Let $P = (X_1, \dots, X_t) \in \mathcal{P}(X)$, (I_1, I_2) be a partition of $\{1, \dots, t\}$ and $\hat{k} \leq k$.

Suppose that $(\hat{G}, w, T, \hat{R}_1, \hat{R}_2, \hat{k})$ is a yes-instance and denote by C a corresponding solution. Denote by (A, B) the cut of $\hat{G}$ such that $C = E(A, B)$ and assume that $\hat{R}_1 \subseteq A$ and $\hat{R}_2 \subseteq B$. Let $C' = C \cap E(G[U])$ and $k' = w(C' \setminus T)$.

We construct the partition $P' \in \mathcal{P}(X')$ in two stages. Recall that some of the border terminals in instance $(G[U], w', T', R_1', R_2', k, X')$ could be also border terminals in the original instance. We include two such border terminals in the same set of P' if they are in the same set of P. This way we obtain partition $(Y_1, \dots, Y_s)$ of X'. Then we replace two distinct sets Y_i and Y_j, $i, j \in \{1, \dots, s\}$, by their union if they can be "connected" in $\hat{G}$ by a path avoiding $G[U]$ and C. More precisely, if there are vertices $x \in Y_i$ and $y \in Y_j$ such that $\hat{G}$ contains an (x', y')-path, where x' and y' are the vertices of $\hat{G}$ that are x or y, or obtained by contracting set containing x or y, respectively, such that this path does not contain edges of $G[U]$ and C. Notice that for any pair of such vertices x an y, either $x', y' \in A$ or $x', y' \in B$, i.e., we never contract two vertices from different parts of the cut (A, B). Denote by $(X_1' \dots, X_r')$ the obtained partition P' of X'. We define the partition (I_1', I_2') of $\{1, \dots, r\}$ by including $j \in \{1, \dots, r\}$ in I_1 if X_j' is obtained by contracting vertices of A and we put j

in I_2 otherwise. Consider the instance $(\hat{G}', w', T', \hat{R}'_1, \hat{R}'_2, \hat{k})$ of Minimal Terminal Cut , where $\hat{G}'$ is constructed form $G[U]$ by contracting $X'_1, \ldots, X'_r$, and $\hat{R}'_i = R'_i \cup \{x_j \mid j \in I'_i\}$ for $i \in \{1, 2\}$, where each x_j is the vertex obtained from X'_j by contraction.

By the construction of P' and (I'_1, I'_2), we have that $(\hat{G}', w', T', \hat{R}'_1, \hat{R}'_2, \hat{k})$ is a yes-instance. Hence, for instance $(G[U], w', T', R'_1, R'_2, k, X')$ the output of Border Contractions contains a solution C'' for this choice of P' and (I'_1, I'_2), and $w(C'' \setminus T) \le k'$. Again, by the construction, we have that $S = (C \setminus C') \cup C''$ is a solution for $(\hat{G}'', w'', T'', \hat{R}''_1, \hat{R}''_2, \hat{k})$. Hence $(\hat{G}'', w'', T'', \hat{R}''_1, \hat{R}''_2, \hat{k})$ is a yes-instance of Minimal Terminal Cut.

Finally, if $(\hat{G}'', w'', T'', \hat{R}''_1, \hat{R}''_2, \hat{k})$ is a yes-instance, then $(\hat{G}, w, T, \hat{R}_1, \hat{R}_2, \hat{k})$ is a yes-instance, because G'' is obtained from G by contracting nonterminal edges, and every solution for $(\hat{G}'', w'', T'', \hat{R}''_1, \hat{R}''_2, \hat{k})$ is a solution for $(\hat{G}, w, T, \hat{R}_1, \hat{R}_2, \hat{k})$. □

By Lemma 4.9, that instead of deciding whether instance $(G, w, T, R_1, R_2, k, X)$ is a yes-instance of Border Contractions, we can solve the problem on instance $(G'', w'', T'', R''_1, R''_2, k, X'')$. What remains is to bound the size of G'', and this is what the next lemma does.

Lemma 4.10. $|V(G'')| < |V(G)|$.

Proof. Recall that G'' is the graph obtained from G by contracting the edges of $E(G[U]) \setminus (L \cup T)$, where L is the union of all sets generated by the algorithm for Border Contractions for $(G[U], w', T', R'_1, R'_2, k, X')$. Notice that for any C in a solution for $(G[U], w', T', R'_1, R'_2, k, X')$, $w(C \setminus T) \le k$. Hence, $|C \setminus T| \le k$. Since $|X'| \le 4k$, the total number of sets in the output is at most $k \cdot 2^{4t} \cdot (4t)^{4t}$. Therefore, the graph H obtained from $G[U]$ by contracting the edges of $E(G[U]) \setminus (L \cup T)$ has at most $k^2 \cdot 2^{4t} \cdot (4t)^{4t}$ nonterminal edges. Notice that $G[U] - T$ has at most $4k + 1$ components, because of Rule 4.5. Hence, H has at most $k^2 \cdot 2^{4t} \cdot (4t)^{4t} + 4k + 1 \le q$ vertices. Since (U, W) is a (q, p)-good edge separation, $|V(H)| < |U|$. As we replace $G[U]$ by H to construct G'', the claim follows. □

Lemma 4.10 shows that we reduce the size of an input graph at each iterative step. Together with Lemma 4.8, it implies that Border Contractions is solvable in time $2^{O(k^2 \log k)} \cdot n^{O(1)}$. This concludes the proof of Theorem 3.

5 SOLVING MINIMUM SPANNING CIRCUIT ON REGULAR MATROIDS

This section is devoted to the proof of the first main result of the article.

Theorem 4. *Minimum Spanning Circuit is solvable in time $2^{O(k^2 \log k)} \cdot n^{O(1)}$ on regular n-element matroids, where $k = \ell - w(T)$.*

The remaining part of the section contains the proof of the theorem. For technical reasons, in our algorithm we solve a special variant of Minimum Spanning Circuit. The technicalities are due to the difficulties of handling 3-sums in the decomposition of the input matroid. We need the following technical definition.

Definition 5.1 (Circuit Constraints and Extensions). Let M be a binary matroid given together with a set of terminals $T \subseteq E(M)$, and a family $\mathcal{X}$ of pairwise disjoint circuits of M of size 3, which are also disjoint with T. Then a *circuit constraint* for M, T, and $\mathcal{X}$ is an 8-tuple $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, where

- P is a mapping assigning to each $X \in \mathcal{X}$ a nonempty set $P(X)$ of subsets of X of size 1 or 2,
- $\mathcal{Z}$ is either the empty set, or is a pair of the form (Z, t), where Z is a circuit of size 3 disjoint with the circuits of $\mathcal{X}$ and with terminals T, and t is an element of Z,

- w is a weight function, $w : E \setminus L \to \mathbb{N}$, where $L = \bigcup_{X \in \mathcal{X}} X$,
- $\mathcal{W} = \{w_X \mid X \in \mathcal{X}\}$ is a family of weight functions, where $w_X : P(X) \to \mathbb{N}$ for each $X \in \mathcal{X}$, and
- k is an integer.

We say that a circuit C of M is a *feasible extension satisfying circuit constraint* $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ (or just feasible when it is clear from the context) if

- $C \cap X \in P(X)$ for each $X \in \mathcal{X}$,
- if $\mathcal{Z} \neq \emptyset$, then $C \triangle Z$ is a circuit of M and $Z \cap C = \{t\}$, and
- $w(C \setminus (T \cup L)) + \sum_{X \in \mathcal{X}} w_X(C \cap X) \leq k$.

We consider the following auxiliary problem.

Extended Minimum Circuit parameterized by k

Input:	A circuit constraint $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$.
Task:	Decide whether there is an extension satisfying the circuit constraint.

Notice that Minimum Spanning Circuit parameterized by $k = \ell - w(T)$ is the special case of Extended Minimum Circuit for $\mathcal{X} = \emptyset$ and $\mathcal{Z} = \emptyset$. We call a circuit C satisfying the requirements of the problem, i.e., which is an extension satisfying the corrsponding circuit constraint, by a *solution*. We also refer to the value $\omega(C) = w(C \setminus (T \cup L)) + \sum_{X \in \mathcal{X}} w_X(C \cap X)$ as to the *weight* of C.

To see the intuition behind the definition of Minimum Spanning Circuit , recall that we are going to apply Theorem 2 that allows to obtain a tree $\mathcal{T}$ whose nodes are basic matroids and the input matroid M is defined by $\mathcal{T}$. Recall also that we assume that $\mathcal{T}$ is rooted and perform the bottom-up transversal of $\mathcal{T}$ with respect to its root. Note that we can choose the root matroid in such a way that it contains a terminal. For a node M_i of $\mathcal{T}$, assume that $\mathcal{T}_i$ is the subtree of $\mathcal{T}$ rooted in M_i. Let M_i be a node of $\mathcal{T}$ and assume that $M_{j_1}, \ldots, M_{j_r}$ are its children in $\mathcal{T}$. Assume also that we are able to obtain partial solutions for the matroids $\hat{M}_1, \ldots, \hat{M}_r$ defined by $\mathcal{T}_{j_1}, \ldots, \mathcal{T}_{j_r}$, respectively. Consider a child M_{j_h} of M_i. If $M_i M_{j_h} \in E(\mathcal{T})$ corresponds to the 1-sum in the decomposition of M, then handling of $\hat{M}_h$ is trivial as we explained in Section 3. If $M_i M_{j_h}$ corresponds to the 2-sum, then we are able to incorporate the information about the partial solution for $\hat{M}_h$ or about $\hat{M}_h$ if $\hat{M}_h$ has no terminals by assigning a special weight to the unique edge of $E(M_i) \cap E(M_{j_h})$ as was discussed in Section 3. Assume that $M_i M_{j_h}$ corresponds to the 3-sum. If $\hat{M}_h$ has no terminals, then we are able to encode the information about $\hat{M}_h$ by a weight assignment to the edges of $X = E(M_i) \cap E(M_{j_h})$. Let $E(\hat{M}_h) \cap T \neq \emptyset$. Then we should overcome the technical difficulties corresponding to this case explained in Section 3. In particular, we obtain at most six partial solutions C, that is, circuits of $\hat{M}_h$, depending on the choice of an edge of X to be in C and on the property whether $C \triangle X$ is a circuit or not. We encode these partial solutions by including X in $\mathcal{X}$ and defining $P(X)$ and the function w_X. In particular, we create an one-element subset of $P(X)$ for a partial solution C such that $C \triangle X$ is a circuit and we create a two-element subset for a partial solution without this restriction. This explains the appearance of $\mathcal{X}$, P and $\mathcal{W}$ in Minimum Spanning Circuit. Now recall that unless M_h is the root, it has a parent M_s in $\mathcal{T}$. If $M_s M_h$ correspond to the 3-sum in the decomposition, then we should be able to handle the case when we require $C \triangle Z$ to be a circuit for a partial solution C and $Z = E(M_s) \cap E(M_h)$. We also have to fix $t \in E(M_h)$ that we use as an additional terminal in this case. This is the reason for the inclusion of $\mathcal{Z}$ in the input of Minimum Spanning Circuit.

In what follows, we construct an algorithm for Extended Minimum Circuit. In Section 5.1, we solve Extended Minimum Circuit on matroids of basic types, and in Section 5.2, we construct the algorithm for regular matroids.

5.1 Solving Minimum Spanning Circuit on Basic Matroids

First, we consider matroids obtained from R_{10} by deleting elements and adding parallel elements.

Lemma 5.1. *Extended Minimum Circuit can be solved in polynomial time on the class of matroids that can be obtained from R_{10} by adding parallel elements and deleting some elements.*

Proof. Let $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ be an instance of Extended Minimum Circuit, where M is a matroid obtained from R_{10} by adding parallel elements and deleting some elements. Since M has no circuit of odd size, $\mathcal{X} = \emptyset$ and $\mathcal{Z} = \emptyset$. If $e_1, e_2 \in E \setminus T$ are parallel, then any circuit C contains at most one of the elements e_1, e_2, and if $e_1 \in C$, then $C' = (C \setminus \{e_1\}) \cup \{e_2\}$ is a circuit by Observation 2.1. It means that we can apply the following reduction rule:

Reduction Rule 5.1. *If there are parallel $e_1, e_2 \in E \setminus T$ and $w(e_1) \leq w(e_2)$, then delete e_2.*

The matroid obtained from M by the exhaustive application of the rule has at most 10 nonterminal elements. Hence, the problem can be solved by brute force: for each set S of nonterminal elements we check whether $S \cup T$ is a circuit and find a circuit of minimum weight it it exists. □

To construct an algorithm for Extended Minimum Circuit for graphic matroids, we consider the following parameterized problem:

Cycle Through Terminals parameterized by k	
Input:	A graph G, a weight function $w \colon E(G) \to \mathbb{N}$, a set of terminals $T \subseteq E(G)$, and a positive integer k.
Task:	Does G have a cycle C with $T \subseteq E(C)$ such that $w(E(C)) - w(T) \leq k$?

We show that Cycle Through Terminals is FPT. This problem can be solved in time $2^k n^{O(1)}$ by making use of the randomized algorithm of Björklund et al. [3]. As the running time of our algorithms for Minimum Spanning Circuit is dominated by the running time of the algorithm for cographic matroids, we give here a *deterministic* algorithm with a worse constant in the base of the exponent. The algorithm is based of the color coding technique of Alon et al. [1].

Lemma 5.2. *Cycle Through Terminals is solvable in time $2^{O(k)} \cdot n^{O(1)}$.*

Proof. Let (G, w, T, k) be an instance of Cycle Through Terminals. First, we exhaustively apply the following reduction rules.

Reduction Rule 5.2 (Stopping Rule). *If $G[T]$ is not a disjoint union of paths or $G[T]$ has at least $k + 1$ components, then return a no-answer and stop.*

Reduction Rule 5.3 (Dissolving Rule). *If there is a vertex v incident to two distinct edges $vx, vy \in T$, then do the following:*

- *delete each edge $e \in E(G) \setminus T$ incident to v;*
- *delete v and replace vx, vy by an edge xy and set $w(xy) = 1$; set $T = (T \setminus \{vx, vy\}) \cup \{xy\}$.*

It is straightforward to see that the rules are safe. Assume that we do not stop when applying Rule 5.2, and, to simplify notations, we use (G, w, T, k) to denote the instance obtained after applying Dissolving Rule. Let $T = \{x_1y_1, \ldots, x_ry_r\}$. Notice that the edges of T are independent, i.e., have no common end-vertices, and $r \le k$. If $r = 1$, then we find a shortest (x_1, y_1)-path in $G - x_1y_1$ using the Dijkstra's algorithm [10]. If the weight of the path is at most k, then we are done. Otherwise, we have a no-instance.

We assume from now that $r \ge 2$. Let $U = \{x_1, \dots, x_r\} \cup \{y_1, \dots, y_r\}$ and denote $h = k - r$. Observe that any cycle C such that $T \subseteq E(C)$ and $w(E(C) \setminus T) \le k$ has at most k vertices and, therefore, at most h vertices in $V(G) \setminus U$. We use the color coding technique [1] to find a cycle C of minimum weight with at most k vertices such that $T \subseteq E(C)$. First, we describe a randomized true-biased Monte-Carlo algorithm and then explain how to derandomize it.

We color the vertices of $V(G) \setminus U$ by h colors uniformly at random. Denote by $c(v)$ the color of $v \in V(G) \setminus U$. Our aim is to find a *colorful* cycle C in G of minimum weight such that $T \subseteq E(C)$ and the vertices of $V(C) \setminus U$ have distinct colors.

First, for each set of colors $X \subseteq \{1, \dots, h\}$, for each pair $\{i, j\}$ of distinct $i, j \in \{1, \dots, r\}$ and each $u \in \{x_i, y_i\}$ and $v \in \{x_j, y_j\}$, we find a (u, v)-path P of minimum weight such that $V(P) \cap U = \{u, v\}$ and the internal vertices of P are colored by distinct colors from X. It can be done in a standard way by dynamic programming across subsets (see [1, 8]). For completeness, we sketch how to find the weight of such a path.

Denote for $z \in V(G) \setminus \{x_i, y_i\}$, by $s(X, u, z)$ the minimum weight of a (u, z)-path P in G with all internal vertices in $V(G) \setminus U$ such that $V(P) \setminus U$ are colored by distinct colors from X; we assume that $s(X, u, z) = +\infty$ if such a path does not exist. We also assume slightly abusing notations that $s(X, u, u) = 0$ for any $X \subseteq \{1, \dots, h\}$. Clearly,

$$s(\emptyset, u, z) = \begin{cases} w(uz) & \text{if } uz \in E(G) \text{ and } z \in U \setminus \{x_i, y_i\}, \\ +\infty & \text{otherwise.} \end{cases}$$

If $X \neq \emptyset$, then it is straightforward to verify that for $z \in V(G) \setminus U$, $s(X, u, z)$

$$= \begin{cases} \min\{s(X \setminus \{c(z)\}, u, x) + w(xz) \mid xz \in E(G), x \in (V(G) \setminus U) \cup \{u\}\} & \text{if } c(z) \in X, \\ +\infty & \text{if } c(z) \notin X, \end{cases}$$

and for $z \in U \setminus \{x_i, y_i\}$,

$$s(X, u, z) = \min\{s(X, u, x) + w(xz) \mid xz \in E(G), x \in (V(G) \setminus U) \cup \{u\}\}.$$

Using these recurrences, we compute $s(X, u, v)$ for all $X \in \{1, \dots, h\}$ and $v \in U \setminus \{x_i, y_i\}$ in time $2^h \cdot n^{O(1)}$. We do these computations for all $u \in \{x_i, y_i\}$ for every $i \in \{1, \dots, r\}$.

Using the table of values of $s(X, u, v)$, we compute the table of values of $c'(X, Y, v)$ for $v \in \{x_i, y_i\}$, where $i \in \{2, \dots, r\}$, $X \subseteq \{1, \dots, h\}$ and $Y \in \{2, \dots, r\} \setminus \{i\}$, where $c'(X, Y, v)$ is a minimum weight of a (y_1, v)-path P in G such that $E(P) \cap T = \{x_jy_j \mid j \in X\}$ and the vertices $V(P) \setminus U$ are colored by distinct colors from X. Notice that $c'(\{1, \dots, h\}, \{2, \dots, r\}, y_1)$ is the minimum weight of a cycle C containing the edges of T with $|V(C) \setminus U| \le h$. For $Y = \emptyset$,

$$c'(X, Y, v) = c(X, y_1, v).$$

For $Y \neq \emptyset$, we have that

$$\begin{aligned} c'(X, Y, v) = \min\{\min\{&c'(X \setminus X', Y \setminus \{j\}, x_j) + w(x_jy_j) + c(X', y_j, v), \\ &c'(X \setminus X', Y \setminus \{j\}, y_j) + w(x_jy_j) + c(X', x_j, v)\} \mid X' \subseteq X, j \in \{1, \dots, r\}\}. \end{aligned}$$

The correctness of the recurrence is proved by the standard arguments. We obtain that the table of values of $c'(X, Y, v)$ can be constructed in time $2^h 2^r \cdot n^{O(1)}$. Hence, $c'(\{1, \dots, h\}, \{2, \dots, r\}, y_1)$ can be computed in time $2^k \cdot n^{O(1)}$.

We have that in time $2^k \cdot n^{O(1)}$ we can check whether we have a *colorful solution*, i.e., a cycle C of weight at most $w(T) + k$ such that $T \subseteq E(C)$ and the vertices of $V(C) \setminus U$ are colored by distinct colors. If we have a colorful solution, then we return it.

Notice that if C is a solution for (G, w, T, k), that is, $T \subseteq E(C)$ and $w(E(C) \setminus T) \leq k$, then the probability that the vertices of $V(C) \setminus U$ are colored by distinct colors from the set $\{1, \ldots, h\}$ is at least $h!/h^h \geq e^{-k}$. Hence, it is sufficient to repeat the algorithm for e^k random colorings to claim that the probability that (G, w, T, k) has a solution but our algorithm returns a no-answer for e^k random colorings is at most $(1 - 1/e^k)^{e^k} \leq 1/e$, that is, we have a true biased Monte-Carlo FPT algorithm that runs in time $(2e)^k \cdot n^{O(1)}$.

This algorithm can be derandomized by the standard tools [1, 8] by replacing the random colorings by *perfect hash functions*. The currently best family of perfect hash functions is constructed by Naor et al. [26]. □

Lemma 5.3. *Extended Minimum Circuit can be solved in time $2^{O(k)} \cdot |E(M)|^{O(1)}$ on graphic matroids.*

Proof. Let $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ be an instance of Extended Minimum Circuit, where M is a graphic matroid. We find G such that M is isomorphic to $M(G)$ in polynomial time using the results of Seymour [31]. Clearly, the choice of G such that M is isomorphic to $M(G)$ is not unique and, in particular, we can assume that G is connected. The only property of G, besides connectivity, that we need is that a set of edges of G composes a cycle of G is and only if the corresponding set of elements of M is a circuit of M. Hence, we assume that $M = M(G)$. We reduce the problem to Cycle Through Terminals. If $|\mathcal{X}| > k$, then we have a trivial no-instance. Assume from now that $|\mathcal{X}| \leq k$ and let $\mathcal{X} = \{X_1, \ldots, X_r\}$.

First, we solve the problem for the case $\mathcal{Z} = \emptyset$. If C is a solution, then $C \cap X \in P(X)$ for $X \in \mathcal{X}$. For each $X_i \in \mathcal{X}$, we guess a set $Y_i \in P(X_i)$ such that $C \cap X_i = Y_i$ for a hypothetic solution C. Since Y_i has size 1 or 2, we have at most 6^k possibilities to guess $Y_1, \ldots, Y_r$. If $\sum_{i=1}^{r} w_{X_i}(Y_i) > k$, then we discard the guess. Assume that $\sum_{i=1}^{r} w_{X_i}(Y_i) \leq k$. We define the graph $G' = G - \bigcup_{i=1}^{r}(X_i \setminus Y_i)$, $T' = T \cup (\bigcup_{i=1}^{r} Y_i)$ and $k' = k - \sum_{i=1}^{r} w_{X_i}(Y_i)$. We also define $w'(e) = w(e)$ for $e \in E(G') \setminus T'$ and set $w'(e) = 1$ for $e \in T'$. Then we solve Cycle Through Terminals for (G', w', T', k') using Lemma 5.2. It is straightforward to see that we have a solution C for the considered instance of Extended Minimum Circuit such that $C \cap X_i = Y_i$ for $i \in \{1, \ldots, r\}$ if and only if (G', w', T', k') is a yes-instance of Cycle Through Terminals.

Assume now that $\mathcal{Z} = (Z, t)$. Clearly, Z induces a cycle in G. Let u be a vertex of this cycle that is not incident to the edge t. We construct the instances of Cycle Through Terminals for every guess of $Y_1, \ldots, Y_r$ in almost the same way as before. The difference is that we delete u from the obtained graph, define t to be a terminal and reduce the parameter by $w(t)$. Notice that if a terminal is incident to u, we have a no-instance for the considered guess. Lemma 3.4 (i) immediately implies the correctness of the reduction.

Since Cycle Through Terminals can be solved in time $2^{O(k)} \cdot n^{O(1)}$ by Lemma 5.2 for each constructed instance, and we consider at most 6^k instances, and each instance is constructed in polynomial time; the total running time is $2^{O(k)} \cdot n^{O(1)}$. Because G is connected, we can write the running time as $2^{O(k)} \cdot |E(M)|^{O(1)}$. □

We use Theorem 3 to solve Extended Minimum Circuit on cographic matroids.

Lemma 5.4. *Extended Minimum Circuit can be solved in time $2^{O(k^2 \log k)} \cdot |E(M)|^{O(1)}$ on cographic matroids.*

Proof. Let $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ be an instance of Extended Minimum Circuit, where M is a cographic matroid. We find G such that M is isomorphic to $M^*(G)$ in polynomial time using the results of Seymour [31]. Notice that we can assume without loss of generality that G is connected. Because we only need the property that a set of edges of G is a minimal cut-set if and only if the corresponding set of elements is a cocircuit of M^*, we can assume that $M = M^*(G)$. We reduce the problem to Minimal Terminal Cut.

If $|\mathcal{X}| > k$, then we have a trivial no-instance. Assume from now that $|\mathcal{X}| \leq k$ and let $\mathcal{X} = \{X_1, \ldots, X_r\}$. If C is a solution, then $C \cap X \in P(X)$ for $X \in \mathcal{X}$. For each $X_i \in \mathcal{X}$, we guess a set $Y_i \in P(X_i)$ such that $C \cap X_i = Y_i$ for a hypothetic solution C. Since Y_i has size 1 or 2, we have at most 6^k possibilities to guess $Y_1, \ldots, Y_r$. If $s = \sum_{i=1}^{r} w_{X_i}(Y_i) > k$, then we discard the guess. If $\mathcal{Z} = (Z, t)$ and $s + w(t) > k$, then we also can discard the guess. Assume that it is not the case. We construct G' by contracting the edges of $\bigcup_{i=1}^{r}(X_i \setminus Y_i)$; for simplicity, we do not distinguish the edges of G' obtained by contractions from the edges of the original graph. If $\mathcal{Z} = \emptyset$, then we set $T' = T \cup (\bigcup_{i=1}^{r} Y_i)$, $R_1 = \emptyset$ and $k' = k - s$, and if $\mathcal{Z} = (Z, t)$, then $T' = T \cup \bigcup_{i=1}^{r} Y_i \cup \{t\}$, R_1 is defined to be the set containing the end-vertices of the edges of $Z \setminus \{t\}$ and $k' = k - s - w(t)$. We also define $w'(e) = w(e)$ for $e \in E(G') \setminus T'$ and set $w'(e) = 1$ for $e \in T'$. Then we solve Minimal Terminal Cut for $(G', w', T', R_1, \emptyset, k')$ using Theorem 3. If $\mathcal{Z} = \emptyset$, then it is straightforward to see that we have a solution C for the considered instance of Extended Minimum Circuit such that $C \cap X_i = Y_i$ for $i \in \{1, \ldots, r\}$ if and only if (G', w', T', k') is a yes-instance of Cycle Through Terminals. If $\mathcal{Z} = (Z, t)$, then correctness follows from Lemma 3.4 (ii).

Since Minimal Terminal Cut can be solved in time $2^{O(k^2 \log k)} \cdot n^{O(1)}$ by Theorem 3 for each constructed instance and we consider at most 6^k instances, and each instance is constructed in polynomial time, the total running time is $2^{O(k^2 \log k)} \cdot n^{O(1)}$. Because G is connected, we can write the running time as $2^{O(k^2 \log k)} \cdot |E(M)|^{O(1)}$. □

5.2 Proof of Theorem 4

Now we are ready to give an algorithm for Minimum Spanning Circuit parameterized by $k = \ell - w(T)$ on regular matroids. Let (M, w, T, ℓ) be an instance of Minimum Spanning Circuit, where M is regular. We consider it to be an instance $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ of Extended Minimum Circuit, where $\mathcal{X} = \emptyset$ and $\mathcal{Z} = \emptyset$. If M can be obtained from R_{10} by the addition of parallel elements or is graphic or cographic, then we solve the problem directly using Lemmas 5.1–5.4. Assume that it is not the case. Using Theorem 2, we find a conflict tree $\mathcal{T}$. Recall that the set of nodes of $\mathcal{T}$ is the collection of basic matroids $\mathcal{M}$ and the edges correspond to extended 1-, 2-, and 3-sums. We select a node r of $\mathcal{T}$ containing an element of T as a root.

We say that an instance $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ of Extended Minimum Circuit is *consistent (with respect to* $\mathcal{T}$*)* if $\mathcal{Z} = \emptyset$, and for any $X \in \mathcal{X}$, $X \in E(M')$ for some $M' \in \mathcal{M}$. Clearly, the instance obtained from the original input instance (M, w, T, ℓ) of Minimum Spanning Circuit is consistent. We use reduction rules that remove leaves keeping this property.

Let $M_\ell \in \mathcal{M}$ be a matroid that is a leaf of $\mathcal{T}$. We construct reduction rules depending on whether M_ℓ is 1-, 2-, or 3-leaf. Denote by M_s its neighbor in $\mathcal{T}$. Let also $\mathcal{T}'$ be the tree obtained from $\mathcal{T}$ be the deletion of M_ℓ, and let M' be the matroid defined by $\mathcal{T}'$. Clearly, $M = M' \oplus M_\ell$.

Throughout this section, we say that a reduction rule is *safe* if it either correctly solves the problem or returns an equivalent instance of Extended Minimum Circuit together with corresponding conflict tree of the obtained matroid that is consistent and the value of the parameter does not increase.

From now onward, let $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ be a consistent instance of Extended Minimum Circuit. Denote $L = \bigcup_{X \in \mathcal{X}} X$.

Reduction Rule 5.4 (1-Leaf reduction rule). *If M_ℓ is a 1-leaf, then do the following.*

(i) If $T \cap E(M_\ell) \neq \emptyset$ or there is $X \in \mathcal{X}$ such that $X \subseteq E(M_\ell)$, then stop and return a no-answer,
(ii) Otherwise, return the instance $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$, where w' is the restriction of w on $E(M') \setminus L$, and solve it using the conflict tree $\mathcal{T}'$.

Since the root matroid contains at least one terminal, Lemma 3.3 (i) immediately implies the following lemma.

Lemma 5.5. *Reduction Rule 5.4 is safe and can be implemented to run in time polynomial in $|E(M)|$.*

Reduction Rule 5.5 (2-Leaf reduction rule). *If M_ℓ is a 2-leaf, then let $\{e\} = E(M_\ell) \cap E(M_s)$ and do the following.*

(i) If $T \cap E(M_\ell) = \emptyset$ and there is no $X \in \mathcal{X}$ such that $X \subseteq E(M_\ell)$, then find a circuit C_ℓ of M_ℓ containing e with minimum $w(C_\ell \setminus \{e\}) \le k$. If there is no such a circuit, then set $w'(e) = k+1$, and let $w'(e) = w(C_\ell \setminus \{e\}$ otherwise. Assume that $w'(e') = w(e')$ for $e' \in E(M') \setminus L$. Return the instance $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ and solve it using the conflict tree $\mathcal{T}'$.

(ii) Otherwise, if $T \cap E(M_\ell) \neq \emptyset$ or there is $X \in \mathcal{X}$ such that $X \subseteq E(M_\ell)$, then consider $T_\ell = (T \cap E(M_\ell)) \cup \{e\}$ and $\mathcal{X}_\ell = \{X \in \mathcal{X} \mid X \subseteq E(M_\ell)\}$. Define $P_\ell, w_\ell, \mathcal{W}_\ell$ by restricting the corresponding functions by $E(M_\ell)$ assuming additionally that $w_\ell(e) = 1$ (the value is, in fact, irrelevant here). Find the minimum $k_\ell \le k$ such that $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell)$ is a yes-instance of Extended Minimum Circuit. Stop and return a no-answer if such k_ℓ does not exist. Otherwise, do the following. Set $T' = (T \cap E(M')) \cup \{e\}$ and $\mathcal{X}' = \{X \in \mathcal{X} \mid X \subseteq E(M')\}$. Define P', w', $\mathcal{W}'$ by restricting the corresponding functions by $E(M')$ assuming additionally that $w'(e) = 1$. Return the instance $(M', T', \mathcal{X}', P', \emptyset, w', \mathcal{W}', k - k_\ell)$ and solve it using the conflict tree $\mathcal{T}'$.

Lemma 5.6. *Reduction Rule 5.5 is safe and can be implemented to run in time $2^{O(k^2 \log k)} \cdot |E(M)|^{O(1)}$.*

Proof. Clearly, if the rule returns a new instance, then it is consistent with respect to $\mathcal{T}'$ and the parameter does not increase.

We show that the rule either correctly solves the problem or returns an equivalent instance.

Suppose that $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ is a consistent yes-instance. We prove that the rule returns a yes-instance. Denote by C a circuit of M that is a solution for the instance. We consider two cases corresponding to the cases (i) and (ii) of the rule.

Case 1. $T \cap E(M_\ell) = \emptyset$ and there is no $X \in \mathcal{X}$ such that $X \in E(M_\ell)$. If $C \subseteq E(M')$, then by Lemma 3.3 (ii), C is a circuit of M'. It is straightforward to see that C is a solution for $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$. Suppose that $C \cap E(M_\ell) \neq \emptyset$. Then $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(M')$, $C_2 \in \mathcal{C}(M_2)$, and $e \in C_1 \cap C_2$ by Lemma 3.3 (ii). It remains to observe that C_1 is a feasible circuit for $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ and its weight is at most the weight of C. Hence, C_1 is a solution for $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$, and the algorithm returns is a yes-instance.

Case 2. $T \cap E(M_\ell) \neq \emptyset$ or there is $X \in \mathcal{X}$ such that $X \subseteq E(M_\ell)$. Then by Lemma 3.3 (ii), $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(M')$, $C_2 \in \mathcal{C}(M_2)$, and $e \in C_1 \cap C_2$. We have that C_2 is a solution for $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k')$, where $k' \le k$ is the weight of C_2 and the algorithm does not stop. Also we have that C_1 is a solution for $(M', T', \mathcal{X}', P', \emptyset, w', \mathcal{W}', k - k_\ell)$ as C_1 is feasible, and its weight is $\omega(C) - k' \le k - k_\ell$, i.e., the rule returns a yes-instance.

Suppose now that the instance constructed by the rule is a yes-instance with a solution C'. We show that the original instance $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ is a yes-instance. We again consider two cases.

Case 1*. The new instance is constructed by Rule 5.5 (i). If $e \notin C'$, then C' is a circuit of M by Lemma 3.3 (ii), and, therefore, C' is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, that is, the original instance is a yes-instance. Assume that $e \in C'$. In this case, $w'(e) \le k$. Hence, there is a circuit C''

of M_ℓ containing e with $w(C'' \setminus \{e\}) = w'(e)$. By Lemma 3.3 (ii), $C = C' \triangle C''$ is a circuit of M. We have that C is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, and it is a yes-instance.

Case 2*. The new instance is constructed by Rule 5.5 (ii). In this case, $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell)$ has a solution C'' of weight k_ℓ. Notice that $e \in C' \cap C''$. We have that $C = C' \triangle C''$ is a circuit of M by Lemma 3.3 (ii). We have that C is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ and, therefore, the original instance is a yes-instance.

We proved that the rule is safe. To evaluate the running time, notice first that we can find a circuit C_ℓ of M_ℓ containing e with minimum $w(C_\ell \setminus \{e\}) \leq k$ in Rule 5.5 (i) in time $2^{O(k^2 \log k)} \cdot |E(M_\ell)|^{O(1)}$ using the observation that we have an instance of Minimum Spanning Circuit with $T = \{e\}$ and can apply Lemmas 5.1–5.4 depending on the type of M_ℓ (in fact, it can be done in polynomial time for this degenerate case). We find k_ℓ in Rule 5.5 (ii) by solving $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell)$ for $k_\ell \leq k$ using Lemmas 5.1–5.4 depending on the type of M_ℓ in time $2^{O(k^2 \log k)} \cdot |E(M_\ell)|^{O(1)}$. □

Reduction Rule 5.6 (3-Leaf reduction rule). *If M_ℓ is a 3-leaf, then let $S = \{e_1, e_2, e_3\} = E(M_\ell) \cap E(M_s)$ and do the following.*

(i) If $T \cap E(M_\ell) = \emptyset$ and there is no $X \in \mathcal{X}$ such that $X \subseteq E(M_\ell)$, then for each $i \in \{1, 2, 3\}$, find a circuit $C_\ell^{(i)}$ of M_ℓ such that $C_\ell^{(i)} \cap S = \{e_i\}$ and $C_\ell^{(i)} \triangle S$ is a circuit of M_ℓ with minimum $w(C_\ell^{(i)} \setminus \{e_i\}) \leq k$. If there is no such a circuit, then set $w'(e_i) = k + 1$, and let $w'(e_i) = w(C_\ell^{(i)} \setminus \{e_i\})$ otherwise. Assume that $w'(e') = w(e')$ for $e' \in E(M') \setminus (L \cup S)$. Return the instance $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ and solve it using the conflict tree $\mathcal{T}'$.

(ii) If there is no $X \in \mathcal{X}$ such that $X \subseteq E(M_\ell)$, but $T_\ell = T \cap E(M_\ell) \neq \emptyset$ and there is $i \in \{1, 2, 3\}$ such that $C_\ell = T_\ell \cup \{e_i\}$ is a circuit of M_ℓ, then consider two cases.

—$C_\ell \triangle S$ is a circuit of M_ℓ. Set $w'(e_i) = 1$ and assume that $w'(e') = w(e')$ for $e' \in E(M') \setminus (S \cup L)$. For each $j \in \{1, 2, 3\} \setminus \{i\}$, do the following. Let $h \in \{1, 2, 3\} \setminus \{i, j\}$. Set $\mathcal{X}_\ell = \{S\}$, $P_\ell(S) = \{e_j\}$, $w_S(\{e_h\}) = 1$, and $\mathcal{W}_\ell = \{w_S\}$. Let w_ℓ be a restriction of w on $E(M_\ell)$. Find a minimum $k_\ell^{(h)} \leq k + 1$ such that $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell^{(h)})$ is a yes-instance of Extended Minimum Circuit. If there is no such $k_\ell^{(h)}$, then set $w'(e_j) = k + 1$ and set $w'(e_j) = k_\ell^{(h)} - 1$ otherwise. Set $T' = (T \cap E(M')) \cup \{e_i\}$. Return the instance $(M', T', \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ and solve it using the conflict tree $\mathcal{T}'$.

—$C_\ell \triangle S$ is not a circuit of M_ℓ. Set $w'(e_i) = k + 1$ and $w'(e_j) = 1$ for $j \in \{1, 2, 3\} \setminus \{i\}$. Assume that $w'(e') = w(e')$ for $e' \in E(M') \setminus (L \cup S)$. Set $T' = (T \cap E(M')) \cup (S \setminus \{e_i\})$. Return the instance $(M', T', \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$, and solve it using the conflict tree $\mathcal{T}'$.

(iii) Otherwise, let $T_\ell = T \cap E(M_\ell)$ and $\mathcal{X}_\ell = \{X \in \mathcal{X} \mid X \subseteq E(M_\ell)\}$. Define $P_\ell, w_\ell, \mathcal{W}_\ell$ by restricting the corresponding functions by $E(M_\ell)$. Construct the set Y of subsets of S and the function $w_S : Y \to \mathbb{N}$ as follows. Initially, set $Y = \emptyset$.

—Define $w'_\ell(e_i) = 1$ for $i \in \{1, 2, 3\}$ and let $w'_\ell(e) = w_\ell(e)$ for $e \in E(M_\ell) \setminus (L \cup S)$. For $i \in \{1, 2, 3\}$, find the minimum $k_\ell^{(i)} \leq k + 1$ such that $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, (S, e_i), w'_\ell, \mathcal{W}_\ell, k_\ell^{(i)})$ is a yes-instance of Extended Minimum Circuit. If such $k_\ell^{(i)}$ exists, then add $\{e_i\}$ in Y and set $w_S(\{e_i\}) = k_\ell^{(i)} - 1$.

—Let $\mathcal{X}'_\ell = \mathcal{X}_\ell \cup \{S\}$. For each $i \in \{1, 2, 3\}$, do the following. Set $P_\ell^{(i)}(X) = P_\ell(X)$ for $X \in \mathcal{X}_\ell$ and $P_\ell^{(i)}(Y) = \{x_i\}$, set $w_S^{(i)}(\{e_i\}) = 1$ and $\mathcal{W}_\ell^{(i)} = \mathcal{W}_\ell \cup \{w_S^{(i)}\}$. Find the minimum $k_\ell^{(i)} \leq k + 1$ such that $(M_\ell, T_\ell, \mathcal{X}'_\ell, P_\ell^{(i)}, \emptyset, w_\ell, \mathcal{W}_\ell^{(i)}, k_\ell^{(i)})$ is a yes-instance of Extended Minimum Circuit. If such $k_\ell^{(i)}$ exists, then add $S \setminus \{e_i\}$ in Y and set $w_S(S \setminus \{e_i\}) = k_\ell^{(i)} - 1$.

If $Y = \emptyset$, then return a no-answer and stop. Otherwise, set $T' = T \cap E(M')$, $\mathcal{X}' = \{X \in \mathcal{X} \mid X \subseteq E(M')\} \cup \{S\}$ and for $X \in \mathcal{X}'$, let $P'(X) = P(X)$ if $X \subseteq P(X)$ and $P'(S) = Y$. Also let

$\mathcal{W}' = \{w_X \mid X \in \mathcal{X}'\}$ and let w' be the restriction of w on $E(M')$. Return the instance $(M', T', \mathcal{X}', P', \emptyset, w', \mathcal{W}', k)$ and solve it using the conflict tree $\mathcal{T}'$.

Lemma 5.7. *Reduction Rule 5.6 is safe and can be implemented to run in time $2^{O(k^2 \log k)} \cdot |E(M)|^{O(1)}$.*

Proof. It is straightforward to see that if the rule returns a new instance, then it is consistent with respect to $\mathcal{T}'$ and the parameter does not increase. We show that the rule either correctly solves the problem or returns an equivalent instance.

Suppose that $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ is a consistent yes-instance. We prove that the rule returns a yes-instance. Denote by C a circuit of M that is a solution for the instance. We consider three cases corresponding to the cases (i)–(iii) of the rule.

Case 1. $T \cap E(M_\ell) = \emptyset$ and there is no $X \in \mathcal{X}$ such that $X \in E(M_\ell)$.

If $C \subseteq E(M')$, then by Lemma 3.3 (iii), C is a circuit of M', and C is a solution for the instance $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ returned by Rule 5.6 (i), that is, we get a yes-instance. Suppose that $C \cap E(M_\ell) \neq \emptyset$. Then, by Lemma 3.3 (iii), $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(M')$, $C_2 \in \mathcal{C}(M_\ell)$, $C_1 \cap S = C_2 \cap S = \{e_i\}$ for some $i \in \{1, 2, 3\}$, and $C_1 \triangle S$ is a circuit of M' or $C_2 \triangle S$ is a circuit of M_ℓ.

Suppose that $C_2 \triangle S$ is a circuit of M_ℓ. Then C_2 is a circuit of M_ℓ containing e_i such that $C_2 \triangle S$ is a circuit and $w(C_2 \setminus \{e_i\}) \leq k$. We have that $w'(e_i) \leq w(C_2 \setminus \{e_i\})$. Hence, C_1 is a solution for the instance $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ returned by Rule 5.6 (i) and, therefore, the rule returns a yes-instance.

Assume now that $C_2 \triangle S$ is a not circuit of M_ℓ. By Lemma 3.1, $C_2 \triangle S$ is a disjoint union of two circuits $C_2^{(1)}$ and $C_2^{(2)}$ of M_ℓ containing $e_h, e_j \in S \setminus \{e_i\}$, and $C_2^{(1)} \triangle S$ and $C_2^{(2)} \triangle S$ are circuits of M_ℓ. We obtain that $w'(e_h) \leq w(C_2^{(1)} \setminus \{e_h\})$ and $w'(e_j) \leq w(C_2^{(2)} \setminus \{e_j\})$. Consider $C_1' = C_1 \triangle S$. Because $C_2 \triangle S$ is not a circuit of M_ℓ, C_1' is a circuit of M'. Since $e_h, e_j \in E(M')$, we have that C_1' is a solution for $(M', T, \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ returned by Rule 5.6 (i). Hence, we get a yes-instance.

Case 2. There is no $X \in \mathcal{X}$ such that $X \subseteq E(M_\ell)$, but $T_\ell = T \cap E(M_\ell) \neq \emptyset$, and there is $i \in \{1, 2, 3\}$ such that $C_\ell = T_\ell \cup \{e_i\}$ is a circuit of M_ℓ.

Notice that $w'(e) \geq 1$ for $e \in E(M') \setminus L$, that is, the instance returned by 5.6 (ii) is a feasible instance of Extended Minimum Circuit. To prove it, observe that if $C_\ell \triangle S$ is a circuit of M_ℓ and $j \in \{1, 2, 3\} \setminus \{i\}$, then $k_\ell^{(h)} \geq 2$, because any solution C' for $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell^{(h)})$ contains at least one element of $E(M_\ell) \setminus (T_\ell \cup S)$. Otherwise, we get that $C_\ell \triangle C' = \{e_i, e_h\}$ is a cycle of M_ℓ contradicting that S is a circuit of M_ℓ.

By Lemma 3.3 (iii), $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(M')$, $C_2 \in \mathcal{C}(M_\ell)$, $C_1 \cap S = C_2 \cap S = \{e_h\}$ for some $h \in \{1, 2, 3\}$, and $C_1 \triangle S$ is a circuit of M' or $C_2 \triangle S$ is a circuit of M_ℓ.

Assume first that $C_\ell \triangle S$ is a circuit of M_ℓ. If $h = i$, then it is straightforward to verify that $C' = C_1 \triangle C_\ell$ is a solution for the instance $(M', T', \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ returned by Rule 5.6 (ii) and, therefore, the rule returns a yes-instance. Suppose that $h \in \{1, 2, 3\} \setminus \{i\}$. We have that C_2 is a solution for $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell^{(h)})$ constructed in Rule 5.6 (ii). Hence, $w'(e_j) = k_\ell^{(h)} - 1$, where $k_\ell^{(h)}$ is at most the weight of the solution C_2 for $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell^{(h)})$ and $j \in \{1, 2, 3\} \setminus \{i, h\}$. Notice that $C_\ell \subset C_2 \triangle S$, that is, $C_2 \triangle S$ is not a circuit of M_ℓ. Hence, $C_1' = C_1 \triangle S$ is a circuit of M'. We obtain that C_1' is a solution for the instance $(M', T', \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ returned by Rule 5.6 (ii). Hence, we get a yes-instance of the problem.

Suppose now that $C_\ell \triangle S$ is not a circuit of M_ℓ. We claim that $h = i$ and $C_2 = C_\ell$ in this case. If $h = i$, then $C_2 = C_\ell$ by minimality, because $T_\ell \subseteq C_2$. Suppose that $h \neq i$. By Lemma 3.1, $C_\ell \triangle S$ is disjoint union of two circuits $C_\ell^{(1)}$ and $C_\ell^{(2)}$ of M_ℓ containing e_h and e_j, respectively, where

$j \in \{1,2,3\} \setminus \{i,h\}$. Therefore, $C_\ell^{(1)} \subseteq C_2$ and, by minimality, $C_2 = C_\ell^{(1)}$, but at least one terminal of T_ℓ is in $C_\ell^{(2)}$ contradicting $T_\ell \subseteq C_2$. Hence, $h = i$ and $C_2 = C_\ell$. Then $C_1' = C_1 \triangle S$ is a circuit of M' and is a solution for the instance $(M', T', \mathcal{X}, P, \emptyset, w', \mathcal{W}, k)$ returned by Rule 5.6 (ii) and, therefore, the rule returns a yes-instance.

Case 3. Cases 1 and 2 do not apply, that is, we are in the conditions of Rule 5.6 (iii). By Lemma 3.3 (iii), $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(M')$, $C_2 \in \mathcal{C}(M_\ell)$, $C_1 \cap S = C_2 \cap S = \{e_i\}$ for some $i \in \{1,2,3\}$, and $C_1 \triangle S$ is a circuit of M' or $C_2 \triangle S$ is a circuit of M_ℓ. Notice that if Rule 5.6 (iii) returns an instance, then w_S has only positive values, because it always holds that $k_\ell^{(i)} \geq 2$, since the conditions of Rule 5.6 (ii) are not fulfilled.

Assume first that $C_2 \triangle S$ is a circuit of M_ℓ. Notice that C_2 is a feasible circuit for $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, (S, e_i), w_\ell', \mathcal{W}_\ell, k_\ell)$ for $k_\ell \leq k$ and its weight with respect to this instance is at most k. Hence, $\{e_i\} \in Y \neq \emptyset$. It means that we do not stop while executing Rule 5.6 (iii) and $w_S(\{e_i\})$ is at most the weight of C_2. It implies that C_1 is a solution for $(M', T', \mathcal{X}', P', \emptyset, w', \mathcal{W}', k)$ returned by Rule 5.6 (iii), i.e., we obtain a yes-instance.

Suppose that $C_2 \triangle S$ is not a circuit of M_ℓ. Then $C_1 \triangle S$ is a circuit of M'. We have that C_2 is a feasible circuit for $(M_\ell, T_\ell, \mathcal{X}_\ell', P_\ell^{(i)}, \emptyset, w_\ell, \mathcal{W}_\ell^{(i)}, k_\ell^{(i)})$ for $k_\ell \leq k$ and its weight with respect to this instance is at most k. Hence, $S \setminus \{e_i\} \in Y \neq \emptyset$. Therefore, we do not stop and $w_S(S \setminus \{e_i\})$ is at most the weight of C_2. It implies that $C_1' = C_1 \triangle S$ is a solution for $(M_\ell, T_\ell, \mathcal{X}_\ell', P_\ell^{(i)}, \emptyset, w_\ell, \mathcal{W}_\ell^{(i)}, k_\ell^{(i)})$ returned by Rule 5.6 (iii), that is, the rule returns a yes-instance.

Suppose now that the instance constructed by the rule is a yes-instance with a solution C'. We show that the original instance $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ is a yes-instance.

We consider three cases corresponding to the cases of the rule.

Case 1*. The new instance is constructed by Rule 5.6 (i). If $C' \cap S = \emptyset$, then it is straightforward to see that C' is a solution for the original instance, and, therefore, $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$ is a yes-instance. Suppose that $C' \cap S \neq \emptyset$. Clearly, $|C' \cap S| \leq 2$.

Assume that $C' \cap S = \{e_i\}$ for some $i \in \{1,2,3\}$. Clearly, $w'(e_i) \leq k$. Hence, M_ℓ has a circuit C'' with $C'' \cap S = \{e_i\}$ such that $w'(e_i) = w(C'' \setminus \{e_i\})$ and $C'' \triangle S$ is a circuit of M_ℓ. By Lemma 3.3 (iii), $C = C' \triangle C''$ is a circuit of M. We obtain that C is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, that is, it is a yes-instance.

Suppose that $C' \cap S = \{e_i, e_j\}$ for distinct $i, j \in \{1,2,3\}$. Let $h \in \{1,2,3\} \setminus \{i,j\}$. We have that $w'(e_i) \leq k$ and $w'(e_j) \leq k$. It means, that M_ℓ has circuits C_1'' and C_2'' such that $C_1'' \cap S = \{e_i\}$, $C_2'' \cap S = \{e_j\}$ and $w'(e_i) = w(C_1'' \setminus \{e_i\})$, $w'(e_j) = w(C_2'' \setminus \{e_i\})$. Consider $C'' = C_1'' \triangle C_2''$. By Lemma 3.1, C'' is a cycle of M_ℓ. Then there is a circuit $C''' \subseteq C''$ of M_ℓ such that $C''' \cap S = \{e_h\}$. Notice that $w(C''' \setminus \{e_h\} \leq w'(e_i) + w'(e_j)$. By Lemma 3.2, $C' \triangle S$ is a circuit of M. Let $C = (C' \triangle S) \triangle C'''$. By Lemma 3.3 (iii), C is a circuit of M. We have that C is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, and, therefore, it is a yes-instance.

Case 2*. The new instance is constructed by Rule 5.6 (ii). Recall that $C_\ell = T_\ell \cup \{e_i\}$ is a circuit of M_ℓ. Clearly, $1 \leq |C' \cap S| \leq 2$.

Suppose, first, that $C_\ell \triangle S$ is a circuit of M_ℓ. If $|C' \cap S| = 1$, then $C' \cap S = \{e_i\}$. Then we obtain that $C = C' \triangle C_\ell$ is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, and it is a yes-instance. Assume that $C' \cap S = \{e_i, e_j\}$ for $j \in \{1,2,3\} \setminus \{i\}$. Then $w'(e_j) \leq k$. Then there is a circuit C'' of M_ℓ such that $C'' \cap S = \{e_h\}$ for $h \in \{1,2,3\} \setminus \{i,j\}$ that is a solution of weight $w'(e_j) + 1$ for $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell^{(h)})$ considered by Rule 5.6 (ii). Notice that $C' \triangle S$ is a circuit of M' by Lemma 3.2. By Lemma 3.3 (iii), we obtain that $C = (C' \triangle S) \triangle C''$ is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, and, therefore, it is a yes-instance.

Assume now that $C_\ell \triangle S$ is a not circuit of M_ℓ. Then $C' \cap S = \{e_h, e_j\}$ for $\{h, j\} = \{1,2,3\} \setminus \{i\}$. By Lemma 3.2, $C' \triangle S$ is a circuit of M', and by Lemma 3.3 (iii), we obtain that $C = (C' \triangle S) \triangle C_\ell$ is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, that is, it is a yes-instance.

Case 3*. The new instance is constructed by Rule 5.6 (iii). We have that $C' \cap S \in Y$ for the set Y constructed by the rule.

Assume that $C' \cap S = \{e_i\}$ for $i \in \{1,2,3\}$. Then $w_S(\{e_i\}) \le k$ and, therefore, there is a solution C'' of weight $k_\ell^{(i)} = w_S(\{e_i\}) + 1$ for the instance $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, (S, e_i), w'_\ell, \mathcal{W}_\ell, k_\ell^{(i)})$ constructed by the rule. Notice that $C'' \triangle S$ is a circuit of M_ℓ. We obtain that $C = C' \triangle C''$ is a solution for $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$, and it is a yes-instance.

Suppose now that $C' \cap S = \{e_i, e_j\}$ for distinct $i, j \in \{1,2,3\}$. Let $h \in \{1,2,3\} \setminus \{i, j\}$. We have that $w_S(\{e_i, e_j\}) \le k$. Hence, there is a solution C'' of weight $k_\ell^{(h)} = w(\{e_i, e_j\}) + 1$ for the instance $(M_\ell, T_\ell, \mathcal{X}'_\ell, P_\ell^{(i)}, \emptyset, w_\ell, \mathcal{W}_\ell^{(i)}, k_\ell^{(h)})$. By Lemma 3.2, $C' \triangle S$ is a circuit of M', and by Lemma 3.3 (iii), $C = C' \triangle C''$ is a circuit of M. We have that C is a solution for the original instance $(M, T, \mathcal{X}, P, \mathcal{Z}, w, \mathcal{W}, k)$. Hence, it is a yes-instance.

To complete the proof, it remains to evaluate the running time. Rule 5.6 (i) can be executed in time $2^{O(k^2 \log k)} \cdot |E(M)|^{O(1)}$.[2] To see it, observe that to compute $w'(e_i)$ for $i \in \{1,2,3\}$, we can solve Extended Minimum Circuit for $(M_\ell, \emptyset, \emptyset, \emptyset, (S, e_i), w_\ell, k_\ell^{(i)})$ for $k_\ell^{(i)} \le k$, where $w_\ell(e) = w(e)$ for $e \in E(M_\ell) \setminus (L \cup S)$ and $w_\ell(e_i) = 1$ for $i \in \{1,2,3\}$, using Lemmas 5.1–5.4 depending on the type of M_ℓ. For Rule 5.6 (ii), observe that it can be checked in polynomial time whether $C_\ell = T_\ell \cup \{e_i\}$ and $C_\ell \triangle S$ are circuits of M for $i \in \{1,2,3\}$. Then we can solve the problem for each $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, \emptyset, w_\ell, \mathcal{W}_\ell, k_\ell^{(h)})$ in time $2^{O(k^2 \log k)} \cdot |E(M)|^{O(1)}$ by Lemmas 5.1–5.4. Finally, the problem for every auxiliary instance $(M_\ell, T_\ell, \mathcal{X}_\ell, P_\ell, (S, e_i), w'_\ell, \mathcal{W}_\ell, k_\ell^{(i)})$ and every $(M_\ell, T_\ell, \mathcal{X}'_\ell, P_\ell^{(i)}, \emptyset, w_\ell, \mathcal{W}_\ell^{(i)}, k_\ell^{(i)})$ can be solved in time $2^{O(k^2 \log k)} \cdot |E(M)|^{O(1)}$ by Lemmas 5.1–5.4. □

Now we can complete the proof of Theorem 4. Observe that $\mathcal{M}$ and the corresponding conflict tree $\mathcal{T}$ can be constructed in polynomial time by Theorem 2, and then we apply the reduction rules at most $|V(\mathcal{T})| - 1$ times until we obtain an instance of Extended Minimum Circuit for a matroid of one of basic types and solve the problem using Lemmas 5.1–5.4.

6 SOLVING SPANNING CIRCUIT ON REGULAR MATROIDS

In this section, we prove the following theorem.

Theorem 5. *Spanning Circuit is* FPT *on regular matroids when parameterized by* $|T|$.

The remaining part of the section contains the proof of the theorem. Similarly to the proof of Theorem 4, we solve a special variant of Spanning Circuit. We redefine a simplified variant of *circuit constraint* that we need in this section as follows.

Definition 6.1 (Circuit Constraints and Extensions). Let M be a binary matroid given together with a set $\mathcal{X}$ of nonempty pairwise disjoint subsets of $E(M)$ of size at most 3. Then a *circuit constraint* for M and $\mathcal{X}$ is a 4-tuple $(M, \mathcal{X}, P, \mathcal{Z})$, where

- P is a mapping assigning to each $X \in \mathcal{X}$ a nonempty set $P(X)$ of subsets of X of size 1 or 2,
- $\mathcal{Z}$ is either the empty set, or is a pair of the form (Z, t), where Z is a circuit of size 3 disjoint with the sets of $\mathcal{X}$ and t is an element of Z.

[2] In fact, it can be done in polynomial time for this degenerate case.

We say that a circuit C of M is a *feasible extension satisfying circuit constraint* $(M, \mathcal{X}, P, \mathcal{Z})$ (or just feasible when it is clear from the context) if

- $C \cap X \in P(X)$ for each $X \in \mathcal{X}$, and
- If $\mathcal{Z} \neq \emptyset$, then $C \triangle Z$ is a circuit of M and $Z \cap C = \{t\}$.

Extended Spanning Circuit

Input:	A circuit constraint $(M, \mathcal{X}, P, \mathcal{Z})$.
Task:	Decide whether there is an extension satisfying the circuit constraint.

We also say that a circuit C is a *feasible extension satisfying circuit constraint* $(M, \mathcal{X}, P, \mathcal{Z})$ is a *solution* for an instance of Extended Spanning Circuit. Clearly, Spanning Circuit is a special case of Extended Spanning Circuit for $\mathcal{X} = \{\{t\} \mid t \in T\}$, $P(\{t\}) = \{t\}$ for $t \in T$, and $\mathcal{Z} = \emptyset$. The intuition behind the definition of Extended Spanning Circuit is essentially the same as for Extended Minimum Circuit. In fact, the absence of weights makes the problem less technical. In Section 6.1, we construct algorithms for Extended Spanning Circuit for basic matroids, and in Section 6.2 we explain how to use these results to solve Spanning Circuit on regular matroids.

6.1 Solving Extended Spanning Circuit on Basic Matroids

First, we consider matroids obtained from R_{10} by deleting elements and adding parallel elements. Notice that, in fact, such matroids that occur in decompositions have bounded size but, formally, we have to deal with the possibility that the number of parallel elements added to R_{10} can be arbitrary.

Lemma 6.1. *Extended Spanning Circuit can be solved in polynomial time on the class of matroids that can be obtained from R_{10} by adding parallel elements and deleting some elements.*

Proof. Let $(M, \mathcal{X}, P, \mathcal{Z})$ be an instance of Extended Spanning Circuit, where M is a matroid with a ground set E that is obtained from R_{10} be adding parallel elements and deleting some elements. Notice that $\mathcal{Z} = \emptyset$, because M has no circuits of odd size.

Notice that if e and e' are parallel elements of M, then for any circuit C of M, either $C = \{e, e'\}$ or $|C \cap \{e, e'\}| \leq 1$. It implies that if $|\mathcal{X}| > 10$, then $(M, \mathcal{X}, P, \mathcal{Z})$ is a no-instance, because for any selection of sets $S(X) \in P(X)$, $\bigcup_{X \in \mathcal{X}} S(X)$ contains two parallel elements. Suppose that this does not occur. Let $Y = \bigcup_{X \in \mathcal{X}} X$. Let M' be the matroid obtained from M by the exhaustive deletions of elements of $E \setminus Y$ that are parallel to some other remaining element of $E \setminus Y$. We claim that $(M, \mathcal{X}, P, \mathcal{Z})$ is a yes-instance if and only if $(M', \mathcal{X}, P, \mathcal{Z})$ is a yes-instance. If C is a circuit of M' such that $T \subseteq C$, then C is a circuit of M as well. Hence, if $(M', \mathcal{X}, P, \mathcal{Z})$ is a yes-instance, then $(M, \mathcal{X}, \mathcal{P}, Z)$ is a yes-instance of Extended Spanning Circuit. Suppose that $(M, \mathcal{X}, \mathcal{P}, Z)$ is a yes-instance and let a circuit C of M be a solution for the instance such that $|C \setminus E(M')|$ is minimum. If $C \subseteq E(M')$, then C is a circuit of M' and $(M', \mathcal{X}, P, \mathcal{Z})$ is a yes-instance. Assume that there is $e \in C \setminus E(M')$. Then there is $e' \in E(M')$ that is parallel to e in M such that $e' \notin Y$. Consider $C' = C \triangle \{e, e'\}$. By Observation 2.1, C' is a circuit of M. We obtain that C' is a solution such that $|C' \setminus E(M')| < |C \setminus E(M')|$; a contradiction.

It remains to to observe that M' has at most 40 elements. Hence, Extended Spanning Circuit can be solved for $(M', \mathcal{X}, P, \mathcal{Z})$ in time $O(1)$ by brute force. □

Next, we consider graphic matroids. Recall that Björklund, Husfeldt, and Taslaman [3] proved that a shortest cycle that goes through a given set of k vertices or edges in a graph can be found in time $2^k \cdot n^{O(1)}$. The currently best deterministic algorithm that finds a cycle that goes through a given set of k vertices or edges was given by Kawarabayashi [20]. We show that these results can be applied to solve Extended Spanning Circuit.

Lemma 6.2. *Extended Spanning Circuit is* FPT *on graphic matroids when parameterized by* $|\mathcal{X}|$.

Proof. Let $(M, \mathcal{X}, P, \mathcal{Z})$ be an instance of Extended Spanning Circuit, where M is a graphic matroid. We find G such that M is isomorphic to $M(G)$ using the results of Seymour [31] and assume that $M = M(G)$.

First, we show how to solve the problem for the case $\mathcal{Z} = \emptyset$ and then explain how to modify the algorithm if $\mathcal{Z} \neq \emptyset$. Because the sets of $\mathcal{X}$ have sizes 2 or 3, $|P(X)| \leq 6$ for $X \in \mathcal{X}$ and there is at most $6^{|\mathcal{X}|}$ possibilities to guess sets $S(X) \in P(X)$ of representatives of the elements $X \in \mathcal{X}$ in C. For each guess, let $T = \bigcup_{X \in \mathcal{X}} S(X)$. Consider the graph G' obtained from G by the deletion of the elements of $(\bigcup_{X \in \mathcal{X}} X) \setminus T$. Clearly, $(M, \mathcal{X}, P, \mathcal{Z})$ has a solution corresponding to the considered guess of sets $S(X)$ if and only if G' has a cycle that goes through all the edges of T. To find such a cycle, we can apply the results of [3, 20]. If $\mathcal{Z} = (Z, t)$, then we use Lemma 3.4 (i). We additionally find a vertex v of the cycle of G induced by Z that is not incident to the specified element t. By Lemma 3.4 (i), $(M, \mathcal{X}, P, \mathcal{Z})$ has a solution corresponding to the considered guess of sets $S(X)$ if and only if G' has a cycle that goes through all the edges of $T \cup \{t\}$ and avoids v. To find such a cycle, we again can apply the results of [3, 20].

Since we consider at most $6^{|\mathcal{X}|}$ guesses of sets $S(X) \in P(X)$ and, for each guess, $|T| \leq 2|\mathcal{X}|$, we conclude that the algorithm runs in FPT time. □

For cographic matroids, we obtain the following lemma using the results of Robertson and Seymour [29].

Lemma 6.3. *Extended Spanning Circuit is* FPT *on cographic matroids when parameterized by* $|\mathcal{X}|$.

Proof. Let $(M, \mathcal{X}, P, \mathcal{Z})$ be an instance of Extended Spanning Circuit, where M is a cographic matroid. Using the results of Seymour [31], we can in polynomial time find a graph G such that M is isomorphic to the bond matroid $M^*(G)$. We assume that $M = M^*(G)$. We can assume without loss of generality that G is connected. Recall that to solve Extended Spanning Circuit, we have to check whether there is a cut (A, B) of G such that $G[A]$ and $G[B]$ are connected and $C = E(A, B)$ satisfies the requirements of Extended Spanning Circuit.

Because the sets of $\mathcal{X}$ have sizes 2 or 3, $|P(X)| \leq 6$ for $X \in \mathcal{X}$ and there is at most $6^{|\mathcal{X}|}$ possibilities to guess sets $S(X) \in P(X)$ of representatives of the elements $X \in \mathcal{X}$ in C. For each guess, let $T = \bigcup_{X \in \mathcal{X}} S(X)$. If $\mathcal{Z} = (Z, t)$, then we additionally include t in T. Consider the graph G' obtained from G by the contraction of the elements of $(\bigcup_{X \in \mathcal{X}} X) \setminus T$.

If there is $e \in T$ that is a loop of G', then $(M, \mathcal{X}, Z, \mathcal{P})$ is a no-instance for the guess, since there is no minimal cut containing e. Assume that the edges of T are not loops. We guess the placement of the end-vertices of the edges of T in A and B considering at most $2^{|T|}$ possibilities. Let T_A be the set of end-vertices guessed to be in A, and let T_B be the set of end-vertices in B. If $\mathcal{Z} = (Z, t)$, then we additionally put the end-vertices of the edges of $Z \setminus \{t\}$ in T_B using Lemma 3.4 (ii). Now we have to check whether there is a partition (A, B) of $V(G)$ such that $T_A \subseteq A$, $T_B \subseteq B$, and $G[A]$ and $G[B]$ are connected. By the celebrated results of Robertson and Seymour [29] about disjoint paths, one can find in FPT -time with the parameter $|T_A| + |T_B|$ disjoint sets of vertices A' and B' containing T_A and T_B, respectively, such that $G[A']$ and $G[B']$ are connected if such sets exist. If there are no such sets A' and B', then we conclude that there is no partition (A, B) with the required properties for the considered guess of T_A and T_B. Otherwise, we extend A' and B' to the partition of $V(G)$ by the exhaustive applying the following rule: If there is $v \in V(G) \setminus (A' \cup B')$ that is adjacent to a vertex of A' or B', then put v in A' or B', respectively. Clearly, we always obtain a partition of $V(G)$, because G is connected.

Since we consider at most $6^{|\mathcal{X}|}$ guesses of sets $S(X) \in P(X)$ and, for each guess, $|T| \leq 2|\mathcal{X}|$ and $|T_A| + |T_B| \leq 4|\mathcal{T}| + 4$, we conclude that the algorithm runs in FPT time. □

6.2 Proof of Theorem 5

Now we are ready to give an algorithm for Spanning Circuit on regular matroids. Let (M, T) be an instance of Spanning Circuit, where M is regular. We consider it to be an instance $(M, \mathcal{X}, P, \mathcal{Z})$ of Extended Spanning Circuit, where $\mathcal{X} = \{\{t\} \mid t \in T\}$, $P(X) = X$ for $X \in \mathcal{X}$, and $\mathcal{Z} = \emptyset$. If M can be obtained from R_{10} by the addition of parallel elements or is graphic or cographic, then we solve the problem directly using Lemmas 6.1–6.3. Assume that it is not the case. Using Theorem 2, we find a conflict tree $\mathcal{T}$. Recall that the set of nodes of $\mathcal{T}$ is the collection of basic matroids $\mathcal{M}$ and the edges correspond to extended 1-, 2-, and 3-sums. The key observation is that M can be constructed from $\mathcal{M}$ by performing the sums corresponding to the edges of $\mathcal{T}$ in an arbitrary order. We select an arbitrarily node r of $\mathcal{T}$ containing an element of T as a root. Our algorithm is based on performing bottom-up traversal of the tree $\mathcal{T}$. We exhaustively apply reduction rules that remove leaves of $\mathcal{T}$ until we obtain a basic case for which we can apply Lemmas 6.1–6.3.

We say that an instance $(M, \mathcal{X}, P, \mathcal{Z})$ of Extended Spanning Circuit is *consistent (with respect to $\mathcal{T}$)* if $\mathcal{Z} = \emptyset$ and for any $X \in \mathcal{X}$, $X \in E(M')$ for some $M' \in \mathcal{M}$. Clearly, the instance obtained from the original input instance (M, T) of Spanning Circuit is consistent. Our reduction rules keep this property.

Let $M_\ell \in \mathcal{M}$ be a matroid that is a leaf of $\mathcal{T}$. Denote by M_s its adjacent sub-leaf. We construct reduction rules depending on whether M_ℓ is 1-, 2-, or 3-leaf.

Throughout this section, we say that a reduction rule is *safe* if it either correctly solves the problem or returns an equivalent instance of Extended Spanning Circuit together with corresponding conflict tree of the obtained matroid that is consistent and the value of the parameter does not increase.

Reduction Rule 6.1 (1-Leaf reduction rule). *If M_ℓ is a 1-leaf, then do the following.*

(i) If there is $X \in \mathcal{X}$ such that $X \in E(M_\ell)$, then stop and return a no-answer,
(ii) Otherwise, delete M_ℓ from $\mathcal{T}$ and denote by T' the obtained conflict tree. Return the instance $(M', \mathcal{X}, P, \emptyset)$ and solve it using the conflict tree $\mathcal{T}'$, where M' is the matroid defined by $\mathcal{T}'$.

Since the root matroid contains at least one set of $\mathcal{X}$, Lemma 3.3 (i) immediately implies the following lemma.

Lemma 6.4. *Reduction Rule 6.1 is safe and can be implemented to run in time polynomial in $|E(M)|$.*

Reduction Rule 6.2 (2-Leaf reduction rule). *If M_ℓ is a 2-leaf, then let $\{e\} = E(M_\ell) \cap E(M_s)$ and do the following.*

(i) If there is no $X \in \mathcal{X}$ such that $X \in E(M_\ell)$, then check whether there is a circuit of M_ℓ containing e. If there is no such a circuit, then delete e from M_s. Delete M_ℓ from $\mathcal{T}$ and denote by T' the obtained conflict tree. Return the instance $(M', \mathcal{X}, P, \emptyset)$ and solve it using the conflict tree $\mathcal{T}'$, where M' is the matroid defined by $\mathcal{T}'$.
(ii) Otherwise, if there is $X \in \mathcal{X}$ such that $X \in E(M_\ell)$, consider $\mathcal{X}_\ell = \{X \in \mathcal{X} \mid X \subseteq E(M_\ell)\} \cup \{\{e\}\}$. Set $P_\ell(X) = P(X)$ for $X \in \mathcal{X}_\ell$ such that $X \neq \{e\}$, and set $P_\ell(\{e\}) = \{e\}$. Solve Extended Spanning Circuit for $(M_\ell, \mathcal{X}_\ell, P_\ell, \emptyset)$. If $(M_\ell, \mathcal{X}_\ell, P_\ell, \emptyset)$ is a no-instance, then stop and return a no-answer. Otherwise, do the following. Set $\mathcal{X}' = \{X \in \mathcal{X} \mid X \not\subseteq E(M_\ell)\} \cup \{\{e\}\}$. Set $P'(X) = P(X)$ for $X \in \mathcal{X}'$ such that $X \neq \{e\}$, and set $P'(\{e\}) = \{e\}$. Delete M_ℓ from $\mathcal{T}$ and denote the obtained conflict tree by $\mathcal{T}'$. Let M' be the matroid defined by $\mathcal{T}'$. Return the instance $(M', \mathcal{X}', P', \emptyset)$ and solve it using the conflict tree $\mathcal{T}'$.

Lemma 6.5. *Reduction Rule 6.2 is safe and can be implemented to run in time $f(\mathcal{X}) \cdot n^{O(1)}$ for some function f of $\mathcal{X}$ only.*

Proof. Clearly, if the rule returns a new instance, then it is consistent with respect to $\mathcal{T}'$ and the parameter does not increase.

We show that the rule either correctly solves the problem or returns an equivalent instance. Denote by $\hat{M}$ the matroid defined by the conflict tree obtained from $\mathcal{T}$ by the deletion of the node M_ℓ. Clearly, $M = \hat{M} \oplus_2 M_\ell$.

Suppose that $(M, \mathcal{X}, P, \mathcal{Z})$ is a consistent yes-instance. We prove that the rule returns a yes-instance. Denote by C a circuit of M that is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$. We consider two cases corresponding to the cases (i) and (ii) of the rule.

Case 1. There is no $X \in \mathcal{X}$ such that $X \in E(M_\ell)$. If $C \subseteq E(\hat{M})$, then by Lemma 3.3 (ii), C is a circuit of M' constructed by the rule that is either $\hat{M}$ or the matroid obtained by from $\hat{M}$ by the deletion of e, because $e \notin C$. Suppose that $C \cap E(M_\ell) \neq \emptyset$. Then $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(\hat{M})$, $C_2 \in \mathcal{C}(M_2)$, and $e \in C_1 \cap C_2$ by Lemma 3.3 (ii). Because C_2 is a circuit of M_2 containing e, we do not delete e from M_s and, therefore, C_1 is a circuit of $M' = \hat{M}$ constructed by the rule in this case. It remains to observe that C_1 is a solution for $(M', \mathcal{X}, P, \emptyset)$. Hence, $(M', \mathcal{X}, P, \emptyset)$ is a yes-instance.

Case 2. There is $X \in \mathcal{X}$ such that $X \in E(M_\ell)$. Then by Lemma 3.3 (ii), $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(\hat{M})$, $C_2 \in \mathcal{C}(M_2)$, and $e \in C_1 \cap C_2$. We have that C_2 is a solution for $(M_\ell, \mathcal{X}_\ell, P_\ell, \emptyset)$ and the algorithm does not stop. Also we have that C_1 is a solution for $(M', \mathcal{X}', P', \emptyset)$, i.e., the rule returns a yes-instance.

Suppose now that the instance constructed by the rule is a yes-instance with a solution C'. We show that the original instance $(M, \mathcal{X}, P, \mathcal{Z})$ is a yes-instance. We again consider two cases.

Case 1*. The new instance is constructed by Rule 6.2 (i). If $e \notin C'$, then C' is a circuit of M by Lemma 3.3 (ii) and, therefore, C' is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$, that is, $(M, \mathcal{X}, P, \mathcal{Z})$ is a yes-instance. Assume that $e \in C'$. In this case, e was not deleted by the rule from M_s. Hence, there is a circuit C'' of M_ℓ containing e. By Lemma 3.3 (ii), $C = C' \triangle C''$ is a circuit of M. We have that C is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$ and it is a yes-instance.

Case 2*. The new instance is constructed by Rule 6.2 (ii). In this case, $(M_\ell, \mathcal{X}_\ell, P_\ell, \emptyset)$ is a yes-instance and there is a solution C'' for it. Notice that $e \in C' \cap C''$. We have that $C = C' \triangle C''$ is a circuit of M by Lemma 3.3 (ii). We have that C is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$ and, therefore, $(M, \mathcal{X}, P, \mathcal{Z})$ is a yes-instance.

We proved that the rule is safe. To evaluate the running time, notice first that we can check existence of a circuit of M_ℓ containing e in Rule 6.2 (i) in polynomial time either directly or using the straightforward observation that we have an instance of Spanning Circuit with $T = \{e\}$ and can apply Lemmas 6.1–6.3 depending on the type of M_ℓ. The problem for $(M_\ell, \mathcal{X}_\ell, P_\ell, \emptyset)$ in Rule 6.2 (ii) can be solved in FPT time by Lemmas 6.1–6.3 depending on the type of M_ℓ, because $|\mathcal{X}_\ell| \leq |\mathcal{X}|$. □

Reduction Rule 6.3 (3-Leaf Reduction Rule). *If M_ℓ is a 3-leaf, then let $Z = E(M_\ell) \cap E(M_s) = \{e_1, e_2, e_3\}$ and do the following.*

(i) If there is no $X \in \mathcal{X}$ such that $X \in E(M_\ell)$, then for each $i \in \{1, 2, 3\}$, solve Extended Spanning Circuit for the instance $(M_\ell, \emptyset, \emptyset, (Z, e_i))$, and if $(M_\ell, \emptyset, \emptyset, (Z, e_i))$ is a no-instance, then delete e_i from M_s. Delete M_ℓ from $\mathcal{T}$ and denote by T' the obtained conflict tree. Return the instance $(M', \mathcal{X}, P, \emptyset)$ and solve it using the conflict tree $\mathcal{T}'$, where M' is the matroid defined by $\mathcal{T}'$.

(ii) Otherwise, if there is $X \in \mathcal{X}$ such that $X \in E(M_\ell)$, set $\mathcal{X}_\ell = \{X \in \mathcal{X} \mid X \subseteq E(M_\ell)\}$ and $P_\ell(X) = P(X)$ for $X \in \mathcal{X}_\ell$. We construct the set R of subsets of Z as follows. Initially, $R = \emptyset$.

—*For $i \in \{1, 2, 3\}$, solve* Extended Spanning Circuit *for the instance $(M_\ell, \mathcal{X}_\ell, P_\ell, (Z, e_i))$, and if we get a yes-instance, then add $\{e_i\}$ in R.*
—*For $i \in \{1, 2, 3\}$, solve* Extended Spanning Circuit *for the instance $(M_\ell, \mathcal{X}'_\ell, P^{(i)}_\ell, \emptyset)$, where $\mathcal{X}'_\ell = \mathcal{X}_\ell \cup \{Z\}$ and $P^{(i)}_\ell(X) = P_\ell(X)$ for $X \in \mathcal{X}_\ell$ and $L^{(i)}_\ell(Z) = \{e_i\}$. If we get a yes instance, then add $Z \setminus \{e_i\}$ in R.*

If $R = \emptyset$, then stop and return a no-answer. Otherwise, do the following. Set $\mathcal{X}' = \{X \in \mathcal{X} \mid X \nsubseteq E(M_\ell)\} \cup \{Z\}$. Set $P'(X) = P(X)$ for $X \in \mathcal{X}'$ such that $X \neq Z$, and set $P'(Z) = R$. Delete M_ℓ from $\mathcal{T}$ and denote the obtained conflict tree by $\mathcal{T}'$. Let M' be the matroid defined by $\mathcal{T}'$. Return the instance $(M', \mathcal{X}', P', \emptyset)$ and solve it using the conflict tree $\mathcal{T}'$.

Lemma 6.6. *Reduction Rule 6.3 is safe and and can be implemented to run in time $f(\mathcal{X}) \cdot n^{O(1)}$ for some function f of $\mathcal{X}$ only.*

Proof. Clearly, if the rule returns a new instance, then it is consistent with respect to $\mathcal{T}'$ and the parameter does not increase.

We show that the rule either correctly solves the problem or returns an equivalent instance. Denote by $\hat{M}$ the matroid defined by the conflict tree obtained from $\mathcal{T}$ by the deletion of the node M_ℓ. Clearly, $M = \hat{M} \oplus_2 M_\ell$.

Suppose that $(M, \mathcal{X}, P, \mathcal{Z})$ is a consistent yes-instance. We prove that the rule returns a yes-instance. Denote by C a circuit of M that is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$. We consider two cases corresponding to the cases (i) and (ii) of the rule.

Case 1. There is no $X \in \mathcal{X}$ such that $X \in E(M_\ell)$. If $C \subseteq E(\hat{M})$, then by Lemma 3.3 (iii), C is a circuit of M' constructed by the rule that is obtained by from $\hat{M}$ by the deletion of some elements of Z, because $Z \cap C = \emptyset$. Suppose that $C \cap E(M_\ell) \neq \emptyset$. Then, by Lemma 3.3 (iii), $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(\hat{M})$, $C_2 \in \mathcal{C}(M_\ell)$, $C_1 \cap Z = C_2 \cap Z = \{e_i\}$ for some $i \in \{1, 2, 3\}$, and $C_1 \triangle Z$ is a circuit of $\hat{M}$ or $C_2 \triangle Z$ is a circuit of M_ℓ.

Suppose that $C_2 \triangle Z$ is a circuit of M_ℓ. Then $(M_\ell, \emptyset, \emptyset, (Z, e_i))$ is a yes-instance and, therefore, $e_i \in E(M')$. Hence, C_1 is a circuit of M' constructed by the rule. We have that C_1 is a solution for $(M', \mathcal{X}, P, \emptyset)$. Hence, $(M', \mathcal{X}, P, \emptyset)$ is a yes-instance.

Assume now that $C_2 \triangle Z$ is a not circuit of M_ℓ. By Lemma 3.1, $C_2 \triangle Z$ is a disjoint union of two circuits $C^{(1)}_2$ and $C^{(2)}_2$ of M_2 containing $e_h, e_j \in Z \setminus \{e_i\}$, and $C^{(1)}_2 \triangle Z$ and $C^{(2)}_2 \triangle Z$ are circuits of M_ℓ. Then $(M_\ell, \emptyset, \emptyset, (Z, e_h))$ and $(M_\ell, \emptyset, \emptyset, (Z, e_h))$ are yes-instances and, therefore, $e_h, e_j \in E(M')$. Consider $C'_1 = C_1 \triangle Z$. Because $C_2 \triangle Z$ is a not circuit of M_ℓ, C'_1 is a circuit of $\hat{M}$. Since $e_h, e_j \in E(M')$, we have that C'_1 is a solution for $(M', \mathcal{X}, P, \emptyset)$. Hence, $(M', \mathcal{X}, P, \emptyset)$ is a yes-instance.

Case 2. There is $X \in \mathcal{X}$ such that $X \in E(M_\ell)$. We have that $C = C_1 \triangle C_2$, where $C_1 \in \mathcal{C}(\hat{M})$, $C_2 \in \mathcal{C}(M_\ell)$, $C_1 \cap Z = C_2 \cap Z = \{e_i\}$ for some $i \in \{1, 2, 3\}$, and $C_1 \triangle Z$ is a circuit of $\hat{M}$ or $C_2 \triangle Z$ is a circuit of M_ℓ.

Suppose that $C_2 \triangle Z$ is a circuit of M_ℓ. Then $(M_\ell, \mathcal{X}_\ell, P_\ell, (Z, e_i))$ is a yes-instance and, therefore, $\{e_i\} \in R$. Since $R \neq \emptyset$, the algorithm does not stop. Also we have that C_1 is a solution for $(M', \mathcal{X}', P', \emptyset)$, i.e., the rule returns a yes-instance.

Assume now that $C_2 \triangle Z$ is not a circuit of M_ℓ. Then $(M_\ell, \mathcal{X}'_\ell, P^{(i)}_\ell, \emptyset)$ is a yes-instance and, therefore, $Z \setminus \{e_i\} \in R$. Since $R \neq \emptyset$, the algorithm does not stop. Consider $C'_1 = C_1 \triangle Z$. Notice that C'_1 is a circuit of $\hat{M}$. We obtain that C'_1 is a solution for $(M', \mathcal{X}', P', \emptyset)$, i.e., the rule returns a yes-instance.

Suppose now that the instance constructed by the rule is a yes-instance with a solution C'. We show that the original instance $(M, \mathcal{X}, P, \mathcal{Z})$ is a yes-instance. We again consider two cases.

Case 1*. The new instance is constructed by Rule 6.3 (i).

If $C' \cap Z = \emptyset$, then C' is a circuit of M by Lemma 3.3 (iii) and, therefore, C' is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$, that is, $(M, \mathcal{X}, P, \mathcal{Z})$ is a yes-instance.

Suppose that $C' \cap Z = \{e_i\}$ for some $i \in \{1, 2, 3\}$. Then, by the construction of the rule, there is a circuit C'' of M_ℓ such that $C'' \cap Z = \{e_i\}$ and $C'' \triangle Z$ is a circuit. By Lemma 3.3 (iii), $C = C' \triangle C''$ is a circuit of M. We have that C is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$ and it is a yes-instance.

Assume that $C' \cap Z = \{e_h, e_j\}$ for some distinct $h, j \in \{1, 2, 3\}$. Let e_i be the element of Z distinct from e_h and e_j. We have that M_ℓ has two circuits C_h and C_j such that $C_h \cap Z = \{e_h\}$, $C_j \cap Z = \{e_j\}$. Then $C_h \triangle C_j \triangle Z$ is a cycle of M_ℓ by Lemma 3.1, and this cycle contains a circuit C_i such that $C_i \cap Z = \{e_i\}$. Consider $C'' = C' \triangle Z$. By Lemma 3.2, C'' is a circuit of $\hat{M}$ and $C'' \triangle Z$ is a circuit. By Lemma 3.3 (iii), we conclude that $C = C'' \triangle C_i$ is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$ and, therefore, $(M, \mathcal{X}, P, \mathcal{Z})$ is a yes-instance.

Case 2*. The new instance is constructed by Rule 6.2 (ii). In this case, $C'' \cap Z \in P'(Z) = R$. Recall that R contains sets of size 1 or 2.

Suppose that $C' \cap Z = \{e_i\}$ for some $i \in \{1, 2, 3\}$. Then, by the construction of the rule, there is a solution C'' for the instance $(M_\ell, \mathcal{X}_\ell, P_\ell, (Z, e_i))$. Notice that $C'' \cap Z = \{e_i\}$ and $C'' \triangle Z$ is a circuit of M_ℓ. By Lemma 3.3 (iii), $C = C' \triangle C''$ is a circuit of M. We have that C is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$ and it is a yes-instance.

Assume that $C' \cap Z = \{e_h, e_j\}$ for some distinct $h, j \in \{1, 2, 3\}$. Let e_i be the element of Z distinct from e_h and e_j. There is a solution C'' for $(M_\ell, \mathcal{X}'_\ell, P^{(i)}_\ell, \emptyset)$. Recall that $C'' \cap Z = \{e_i\}$. Consider $C''' = C' \triangle Z$. By Lemma 3.2, C''' is a circuit of $\hat{M}$ and $C''' \triangle Z$ is a circuit. Since $C''' \cap Z = \{e_i\}$, we obtain that $C = C''' \triangle C''$ is a circuit of M. It remains to observe that C is a solution for $(M, \mathcal{X}, P, \mathcal{Z})$ and it is a yes-instance.

We proved that the rule is safe. To evaluate the running time, notice first that we can check existence of a circuit of M_ℓ containing each e_i in Rule 6.3 (ii) in polynomial time using Lemmas 6.1–6.3 depending on the type of M_ℓ. The problems for $(M_\ell, \mathcal{X}_\ell, P_\ell, (Z, e_i))$ and $(M_\ell, \mathcal{X}'_\ell, P^{(i)}_\ell, \emptyset)$ in Rule 6.3 (i) can be solved in FPT time by Lemmas 6.1–6.3 depending on the type of M_ℓ, because $|\mathcal{X}_\ell| < |\mathcal{X}'_\ell| \leq |\mathcal{X}|$. □

To complete the proof of Theorem 5, it remains to observe that $\mathcal{M}$ and the corresponding conflict tree $\mathcal{T}$ can be constructed in polynomial time by Theorem 2, and then we apply the reduction rules at most $|V(\mathcal{T})| - 1$ times until we obtain an instance of Extended Spanning Circuit for a matroid of one of basic types and solve the problem using Lemmas 6.1–6.3.

7 LOWER BOUNDS AND OPEN QUESTIONS

In this article, we gave FPT algorithms for Minimum Spanning Circuit and Spanning Circuit for regular matroids. We conclude with a number of open algorithmic questions about circuits in matroids. We also discuss here certain algorithmic limitations for extending our results.

Larger matroid classes. The first natural question is whether our results can be extended to other classes of matroids? There is no hope (of course up to certain complexity assumptions) that our results can be extended to binary matroids. Downey et al. proved in [15] that the following problem is W[1]-hard being parameterized by k. (We refer to the book of Downey and Fellows [14] for the definition of W-hierarchy.) In the Maximum-Likelihood Decoding problem we are given a binary $n \times m$ matrix A, a target binary n-element vector $\vec{s}$, and a positive integer k. The question is whether there is a set of at most k columns of A that sum to $\vec{s}$? As it was observed by Gavenciak et al. [17], the result of Downey et al. immediately implies the following proposition.

Proposition 7.1 ([17]). *Minimum Spanning Circuit is* W[1]*-hard on binary matroids with unit-weights elements when parameterized by* ℓ *even when* $|T| = 1$.

Let us note that Minimum Spanning Circuit with $|T| = 0$ on binary matroids is equivalent to Even Set, which parameterized complexity is a long-standing open question, see, e.g., [14].

However Proposition 7.1 does not rule out a possibility that our results can be extended from the class of regular matroids to any proper minor-closed class of binary, and even more generally, representable over some finite field, matroids. We doubt that our techniques tailored for regular matroids could be directly used but it is very likely that the powerful structural theorems obtained by Geelen et al. that were used to settle Rota's conjecture, see [18] for further discussions, can shed some light on this question.

Solving both problems on transversal matroids is another interesting problem.

Stronger parameterization. Björklund et al. [3] gave a randomized algorithm that finds a shortest cycle through a given set T of vertices or edges in a graph in time $2^{|T|} \cdot n^{O(1)}$. Hence Minimum Spanning Circuit parameterized by $w(T)$ is (randomized) FPT on graphic matroids if the weights are encoded in unary. Unfortunately, it is possible to show that Minimum Spanning Circuit is W[1]-hard already on cographic matroids for this parameterization.

Theorem 6. *Minimum Spanning Circuit is* W[1]*-hard on cographic matroids with unit-weights elements when parameterized by* $|T|$.

Proof. We reduce the following variant of the Multicolored Clique problem. In the Regular Multicolored Clique we are given a regular graph G, a positive integer parameter k, and a partition $V_1, \dots, V_k$ of $V(G)$. The task is to decide whether G have a clique K such that $|V_i \cap K| = 1$ for $i \in \{1, \dots, k\}$. Regular Multicolored Clique parameterized by k was shown to be W[1]-hard by Cai [4].

Let $(G, k, V_1, \dots, V_k)$ be an instance of Regular Multicolored Clique, and assume that G is a d-regular n-vertex graph. Assume without loss of generality that $k < d < n - 1$. We construct the graph H as follows.

- Construct a copy of G.
- For each $i \in \{1, \dots, k\}$, construct a vertex v_i and edges $v_i u$ for $u \in V_i$.
- Construct n pairwise adjacent vertices $x_1, \dots, x_n$ and make them adjacent to the vertices of G.
- Construct $p = 2n^2$ pairwise adjacent vertices $y_1, \dots, y_p$ and make each of them adjacent to $x_1, \dots, x_n$.
- Construct edges $y_1 v_1, \dots, y_1 v_k$ and set $T = \{y_1 v_1, \dots, y_1 v_k\}$.

We put $\ell = n + (n + d - k + 1)k$.

We claim that $(G, k, V_1, \dots, V_k)$ is a yes-instance of Regular Multicolored Clique if and only if H has a minimal cut-set C of size at most ℓ such that $T \subseteq C$.

Suppose that K is a clique in G with $|V_i \cap K| = 1$ for $i \in \{1, \dots, k\}$. Consider the partition $(A, \overline{A})$ of $V(G)$ with $A = \{v_1, \dots, v_k\} \cup K$. It is straightforward to verify that $H[A]$ and $H[\overline{A}]$ connected. Therefore $C = E(A, \overline{A})$ is a minimal cut-set. The vertices $v_1, \dots, v_k$ have $n - k$ neighbors in $V(G) \cap \overline{A}$ in total and all their neighbors are distinct. Also each v_i is adjacent to $y_1 \in \overline{A}$. Since G is d-regular, each vertex $u \in K$ has $d - k + 1$ neighbors in $V(G) \cap \overline{A}$ and n neighbors $x_1, \dots, x_n$ among the remaining vertices of $\overline{A}$. Hence, $|C| = (n - k) + k + (n + d - k + 1)k = \ell$.

Assume now that H has a minimal cut-set C of size at most ℓ such that $T \subseteq C$. Let $(A, \overline{A})$ be the partition of $V(H)$ with $E(A, \overline{A}) = C$. We also assume that $y_1 \in \overline{A}$. Then $v_1, \dots, v_k \in A$.

First, we show that $x_i, y_j \in \overline{A}$ for $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$. To obtain a contradiction, assume that at least one of these vertices is in A. Because $\{x_1, \ldots, x_n\} \cup \{y_1, \ldots, y_n\}$ is a clique of size $2n^2 + n$ and $T \subseteq E(A, \overline{A})$, we have that $|E(A, \overline{A})| \geq 2n^2 + n - 1 + k > n + (n + d - k + 1)k = \ell$ contradicting $|E(A, \overline{A})| \leq \ell$.

Because $H[A]$ is connected and $v_1, \ldots, v_k \in A$, there is $u_i \in V_i$ such that $u_i \in A$ for each $i \in \{1, \ldots, k\}$. Let $A' = \{v_1, \ldots, v_k\} \cup \{u_1, \ldots, u_k\}$. The vertices $v_1, \ldots, v_k$ have $n - k$ neighbors in total in $V(G) \cap \overline{A'}$ and all their neighbors are distinct. Also each v_i is adjacent to $y_1 \in \overline{A'}$. Since G is d-regular, each vertex u_i has at least $d - k + 1$ neighbors in $V(G) \cap \overline{A'}$, and all the vertices $u_1, \ldots, u_k$ are incident to $(d - k + 1)d$ edges of G with exactly one end-vertex in A' if and only if $\{u_1, \ldots, u_k\}$ is a clique of G. Also each vertex u_i is adjacent to $x_1, \ldots, x_k$. Therefore, $|E(A', \overline{A'})| \geq (n - k) + k + (n + d - k + 1)k = \ell$, and $|E(A', \overline{A'})| = \ell$ if and only if $\{u_1, \ldots, u_k\}$ is a clique of G.

Since $x_i, y_j \in \overline{A}$ for $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$, $A' \setminus A \subseteq V(G)$. Because G is d-regular and each vertex of G is adjacent to exactly one vertex v_i and the vertices $x_1, \ldots, x_n$, $\ell = |E(A, \overline{A})| \geq |E(A', \overline{A'})| + |A' \setminus A|(n - d - 1) \geq |E(A', \overline{A'})| \geq \ell$. As $d < n - 1$, we obtain that $A = A'$. Hence, $\{u_1, \ldots, u_k\}$ is a clique of G.

To complete the proof, we observe that H has a minimal cut-set C of size at most ℓ such that $T \subseteq C$ if and only if $(M(H), w, T, \ell)$ is a yes-instance of Minimum Spanning Circuit with the weight function $w(e) = 1$ for $e \in E(H)$. ◻

Interestingly, Theorem 6 does not rule out a possibility that for a fixed numbers of terminals Minimum Spanning Circuit is still solvable in polynomial time or, in other words, that it is in XP parameterized by $|T|$. We conjecture that this is not the case. More precisely, is Minimum Spanning Circuit NP-complete on cographic matroids for a fixed number, say, $|T| = 3$, terminal elements?

ACKNOWLEDGMENTS

The authors thank the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

[1] Noga Alon, Raphael Yuster, and Uri Zwick. 1995. Color-coding. *J. ACM* 42, 4 (1995), 844–856.

[2] Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. 2018. Parameterized intractability of even set and shortest vector problem from gap-ETH. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP'18)*, Vol. 107. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 17:1–17:15. DOI: https://doi.org/10.4230/LIPIcs.ICALP.2018.17

[3] Andreas Björklund, Thore Husfeldt, and Nina Taslaman. 2012. Shortest cycle through specified elements. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'12)*. SIAM, 1747–1753.

[4] Leizhen Cai. 2008. Parameterized complexity of cardinality constrained optimization problems. *Comput. J.* 51, 1 (2008), 102–121. DOI: https://doi.org/10.1093/comjnl/bxm086

[5] Leizhen Cai, Siu Man Chan, and Siu On Chan. 2006. Random separation: A new method for solving fixed-cardinality optimization problems. In *Proceedings of the 2nd International Workshop in Parameterized and Exact Computation (IWPEC'06)*. Vol. 4169. Springer, 239–250.

[6] Chandra Chekuri and Julia Chuzhoy. 2016. Polynomial bounds for the grid-minor theorem. *J. ACM* 63, 5 (2016), 40:1–40:65. DOI: https://doi.org/10.1145/2820609

[7] Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. 2016. Designing FPT algorithms for cut problems using randomized contractions. *SIAM J. Comput.* 45, 4 (2016), 1171–1229. DOI: https://doi.org/10.1137/15M1032077

[8] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer. DOI: https://doi.org/10.1007/978-3-319-21275-3

[9] Tristan Denley and Haidong Wu. 2001. A generalization of a theorem of Dirac. *J. Combin. Theory Ser. B* 82, 2 (2001), 322–326.

[10] E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1 (1959), 269–271.

[11] Michael Dinitz and Guy Kortsarz. 2014. Matroid secretary for regular and decomposable matroids. *SIAM J. Comput.* 43, 5 (2014), 1807–1830. DOI: https://doi.org/10.1137/13094030X
[12] Irit Dinur. 2016. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electr. Colloq. Comput. Complex.* 23 (2016), 128.
[13] Gabriel Andrew Dirac. 1960. In abstrakten Graphen vorhandene vollständige 4-Graphen und ihre Unterteilungen. *Math. Nachr.* 22 (1960), 61–85.
[14] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity.* Springer. DOI: https://doi.org/10.1007/978-1-4471-5559-1
[15] Rodney G. Downey, Michael R. Fellows, Alexander Vardy, and Geoff Whittle. 1999. The parametrized complexity of some fundamental problems in coding theory. *SIAM J. Comput.* 29, 2 (1999), 545–570. DOI: https://doi.org/10.1137/S0097539797323571
[16] Herbert Fleischner and Gerhard J. Woeginger. 1992. Detecting cycles through three fixed vertices in a graph. *Inform. Process. Lett.* 42, 1 (1992), 29–33. DOI: https://doi.org/10.1016/0020-0190(92)90128-I
[17] Tomás Gavenciak, Daniel Král, and Sang-il Oum. 2012. Deciding first order properties of matroids. In *Proceedings of the 39th International Colloquium of Automata, Languages and Programming (ICALP'12)*, Lecture Notes in Computer Science, Vol. 7392. Springer, 239–250.
[18] Jim Geelen, Bert Gerards, and Geoff Whittle. 2014. Solving Rota's conjecture. *Not. Am. Math. Soc.* 61, 7 (2014), 736–743. DOI: https://doi.org/10.1090/noti1139
[19] Ken-ichi Kawarabayashi. 2002. One or two disjoint circuits cover independent edges: Lovász-woodall conjecture. *J. Comb. Theory, Ser. B* 84, 1 (2002), 1–44. DOI: https://doi.org/10.1006/jctb.2001.2059
[20] Ken-ichi Kawarabayashi. 2008. An improved algorithm for finding cycles through elements. In *Proceedings of the 13th International Conference on Integer Programming and Combinatorial Optimization (IPCO'08)*, Lecture Notes in Computer Science, Vol. 5035. Springer, 374–384.
[21] Ken-ichi Kawarabayashi and Mikkel Thorup. 2011. The minimum k-way cut of bounded size is fixed-parameter tractable. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS'11)*. IEEE Computer Society, 160–169.
[22] Andrea S. LaPaugh and Ronald L. Rivest. 1980. The subgraph homeomorphism problem. *J. Comput. Syst. Sci.* 20, 2 (1980), 133–149. DOI: https://doi.org/10.1016/0022-0000(80)90057-4
[23] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. 2014. Faster parameterized algorithms using linear programming. *ACM Trans. Algor.* 11, 2 (2014), 15:1–15:31.
[24] Pasin Manurangsi and Prasad Raghavendra. 2017. A birthday repetition theorem and complexity of approximating dense CSPs. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, Vol. 80. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 78:1–78:15. DOI: https://doi.org/10.4230/LIPIcs.ICALP.2017.78
[25] Sean McGuinness. 2009. Ore-type and Dirac-type theorems for matroids. *J. Combin. Theory Ser. B* 99, 6 (2009), 827–842.
[26] Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. 1995. Splitters and near-optimal derandomization. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*. IEEE, 182–191.
[27] James G. Oxley. 1992. *Matroid Theory.* Oxford University Press.
[28] James G. Oxley. 1997. A matroid generalization of a result of Dirac. *Combinatorica* 17, 2 (1997), 267–273. DOI: https://doi.org/10.1007/BF01200909
[29] Neil Robertson and Paul D. Seymour. 1995. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B* 63, 1 (1995), 65–110.
[30] Paul D. Seymour. 1980. Decomposition of regular matroids. *J. Comb. Theory Ser. B* 28, 3 (1980), 305–359. DOI: https://doi.org/10.1016/0095-8956(80)90075-1
[31] Paul D. Seymour. 1981. Recognizing graphic matroids. *Combinatorica* 1, 1 (1981), 75–78. DOI: https://doi.org/10.1007/BF02579179
[32] P. D. Seymour. 1986. Triples in matroid circuits. *Eur. J. Combin.* 7, 2 (1986), 177–185. DOI: https://doi.org/10.1016/S0195-6698(86)80044-0
[33] Klaus Truemper. 1992. *Matroid Decomposition.* Academic Press.
[34] Alexander Vardy. 1997. Algorithmic complexity in coding theory and the minimum distance problem. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC'97)*. ACM, 92–109.
[35] Hassler Whitney. 1935. On the abstract properties of linear dependence. *Am. J. Math.* 57, 3 (1935), 509–533. DOI: https://doi.org/10.2307/2371182

Received February 2017; revised October 2018; accepted July 2019