On the tractability of some natural packing, covering and partitioning problems

Attila Bernáth∗ Zoltán Király†

July 21, 2014

Abstract

In this paper we fix 7 types of undirected graphs: paths, paths with prescribed endvertices, circuits, forests, spanning trees, (not necessarily spanning) trees and cuts. Given an undirected graph \( G = (V,E) \) and two “object types” \( A \) and \( B \) chosen from the alternatives above, we consider the following questions.

Packing problem: can we find an object of type \( A \) and one of type \( B \) in the edge set \( E \) of \( G \), so that they are edge-disjoint? Partitioning problem: can we partition \( E \) into an object of type \( A \) and one of type \( B \)? Covering problem: can we cover \( E \) with an object of type \( A \), and an object of type \( B \)? This framework includes 44 natural graph theoretic questions. Some of these problems were well-known before, for example covering the edge-set of a graph with two spanning trees, or finding an \( s \)-\( t \) path \( P \) and an \( s' \)-\( t' \) path \( P' \) that are edge-disjoint. However, many others were not, for example can we find an \( s \)-\( t \) path \( P \subseteq E \) and a spanning tree \( T \subseteq E \) that are edge-disjoint? Most of these previously unknown problems turned out to be NP-complete, many of them even in planar graphs. This paper determines the status of these 44 problems. For the NP-complete problems we also investigate the planar version, for the polynomial problems we consider the matroidal generalization (wherever this makes sense).

1 Introduction

In this paper we consider undirected graphs. The node set of a graph \( G = (V,E) \) is sometimes also denoted by \( V(G) \), and similarly, the edge set is sometimes denoted by \( E(G) \). A subgraph of a graph \( G = (V,E) \) is a pair \( (V',E') \) where \( V' \subseteq V \) and \( E' \subseteq E \cap (V' \times V') \). A graph is called subcubic if every node is incident to at most 3 edges, and it is called subquadratic if every node is incident to at most 4 edges. By a cut in a graph we mean the set of edges leaving a nonempty proper subset \( V' \) of the nodes (note that we do not require that \( V' \) and \( V - V' \) induces a connected graph). We use standard terminology and refer the reader to [9] for what is not defined here.

We consider 3 types of decision problems with 7 types of objects. The three types of problems are: packing, covering and partitioning, and the seven types of objects are the following: paths (denoted by \( P \)), paths with specified endvertices (denoted by \( P_{st} \), where \( s \) and \( t \) are the prescribed endvertices), (simple) circuits (denoted by \( C \) by that we mean a closed walk of length at least 2, without edge- and node-repetition), forests (\( F \)), spanning trees (SpT), (not necessarily spanning) trees (T), and cuts (denoted by Cut). Let \( G = (V,E) \) be a connected undirected graph (we assume connectedness in order to avoid trivial case-checkings) and \( A \) and \( B \) two (not necessarily different) object types from the 7 possibilities above. The general questions we ask are the following:

- **Packing problem** (denoted by \( A \wedge B \)): can we find two edge-disjoint subgraphs in \( G \), one of type \( A \) and the other of type \( B \)?
- **Covering problem** (denoted by \( A \cup B \)): can we cover the edge set of \( G \) with an object of type \( A \) and an object of type \( B \)?
- **Partitioning problem** (denoted by \( A + B \)): can we partition the edge set of \( G \) into an object of type \( A \) and an object of type \( B \)?

∗Hungarian Academy of Sciences, Institute for Computer Science and Control (MTA-SZTAKI), Budapest, Hungary. This work was supported by MTA-ELTE Egerváry Research Group, by the OTKA grant CK80124 and by the ERC StG project PAAno. 2599515. E-mail: bernath@cs.elte.hu.
†Department of Computer Science and Egerváry Research Group (MTA-ELTE), Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary. Research was supported by grants (no. CNK 77780 and no. K 109240) from the National Development Agency of Hungary, based on a source from the Research and Technology Innovation Fund. E-mail: kiraly@cs.elte.hu
Let us give one example of each type. A typical partitioning problem is the following: decide whether the edge set of $G$ can be partitioned into a spanning tree and a forest. Using our notations this is Problem $\text{SpT}+F$. This problem is in $\text{NP} \cap \text{co-NP}$ by the results of Nash-Williams [18], polynomial algorithms for deciding the problem were given by Kishi and Kajitani [15], and Kameda and Toida [13].

A typical packing problem is the following: given four (not necessarily distinct) vertices $s, t, s’, t’ \in V$, decide whether there exists an $s$-$t$ path $P$ and an $s’$-$t’$-path $P’$ in $G$, such that $P$ and $P’$ do not share any edge. With our notations this is Problem $P_{st} \land P’_{s’t’}$. This problem is still solvable in polynomial time, as was shown by Thomassen [24] and Seymour [23].

A typical covering problem is the following: decide whether the edge set of $G$ can be covered by a path and a circuit. In our notations this is Problem $P \cup C$. Interestingly we found that this simple-looking problem is NP-complete.

Let us introduce the following short formulation for the partitioning and covering problems. If the edge set of a graph $G$ can be partitioned into a type $A$ subgraph and a type $B$ subgraph then we will also say that the edge set of $G$ is $A \cup B$. Similarly, if there is a solution of Problem $A \cup B$ for a graph $G$ then we say that the edge set of $G$ is $A \cup B$.

| Problem | Status | Reference | Remark |
|---------|--------|-----------|--------|
| $P + P$ | NPC    | Theorem 5 | NPC for subquadratic planar |
| $P + P_{st}$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P + C$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P + T$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P + \text{SpT}$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P + F$ | NPC | Theorem 5 | NPC for subcubic planar |
| $P_{st} + P’_{s’t’}$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P_{st} + C$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P_{st} + T$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P_{st} + \text{SpT}$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $P_{st} + F$ | NPC | Theorem 5 (and Theorem 6) | NPC for subcubic planar |
| $C + C$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $C + T$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $C + \text{SpT}$ | NPC | Theorem 5 | NPC for subquadratic planar |
| $C + F$ | NPC | Theorem 5 (and Theorem 6) | NPC for subcubic planar |
| $T + T$ | NPC | Pálvölgyi [20] | planar graphs? |
| $T + \text{SpT}$ | NPC | Theorem 6 | planar graphs? |
| $F + F$ | P | Kishi and Kajitani [15], Kameda and Toida [13] (Nash-Williams [18]) | in $P$ for matroids: Edmonds [7] |
| $\text{SpT} + \text{SpT}$ | P | Kishi and Kajitani [15], Kameda and Toida [13] (Nash-Williams [19], Tutte [25]) | in $P$ for matroids: Edmonds [7] |
| $\text{Cut} + \text{Cut}$ | P | if and only if bipartite (and $|V| \geq 3$) | planar graphs? |
| $\text{Cut} + F$ | NPC | Theorem 7 | |
| $\text{Cut} + C$ | NPC | Theorem 3 | NPC for subcubic planar |
| $\text{Cut} + T$ | NPC | Theorem 3 | NPC for subcubic planar |
| $\text{Cut} + P$ | NPC | Theorem 3 | NPC for subcubic planar |
| $\text{Cut} + P_{st}$ | NPC | Theorem 3 | NPC for subcubic planar |

Table 1: 25 PARTITIONING PROBLEMS

The setting outlined above gives us 84 problems. Note however that some of these can be omitted. For example $P \land A$ is trivial for each possible type $A$ in question, because $P$ may consist of only one vertex. By the same reason, $T \land A$ and $F \land A$ type problems are also trivial. Furthermore, observe that the edge-set $E(G)$ of a graph $G$ is $F + A \iff E(G) = F \cup A \iff E(G) = T \cup A \iff E(G) = \text{SpT} \cup A$: therefore we will only consider the problems of form $F + A$ among these for any $A$. Similarly, the edge set $E(G)$ is $F + F \iff E(G)$ is $T + F \iff E(G)$ is $\text{SpT} + F$: again we choose to deal with $F + F$. We can also omit the problems $\text{Cut} + \text{SpT}$ and $\text{Cut} \land \text{SpT}$ because a cut and a spanning tree can never
be disjoint.

The careful calculation gives that we are left with 44 problems. We have investigated the status of these. Interestingly, many of these problems turn out to be NP-complete. Our results are summarized in Tables 1-3. We note that in our NP-completeness proofs we always show that the considered problem is NP-complete even if the input graph is simple. On the other hand, the polynomial algorithms given here always work also for multigraphs (we allow parallel edges, but we forbid loops).

Some of the results shown in the tables were already proved in the preliminary version [5] of this paper: namely we have already shown the NP-completeness of Problems $P_{st} + \text{T}$, $P_{st} + \text{SpT}$, $P_{st} + \text{T}$, $P_{st} + \text{SpT}$, $C + \text{T}$, $C + \text{SpT}$, $T + \text{SpT}$, $P_{st} \land \text{SpT}$, and $C \land \text{SpT}$ there.

Table 2: 9 PACKING PROBLEMS

| Problem     | Status | Reference | Remark                  |
|-------------|--------|-----------|-------------------------|
| $P_{st} \land P_{st}'$ | P      | Seymour [23], Thomassen [24] |                       |
| $P_{st} \land C$      | P      | see Section 3 |                       |
| $P_{st} \land \text{SpT}$ | NPC    | Theorem 6  | planar graphs?          |
| $C \lor C$            | NPC    | Bodlaender [6] (see also Section 3) | NPC in linear matroids (Theorem 10) |
| $C \land \text{SpT}$  | NPC    | Theorem 6  | polynomial in planar graphs, [4] |
| $\text{SpT} \land \text{SpT}$ | P      | Imai [12], (Nash-Williams [19], Tutte [25]) | in $P$ for matroids; Edmonds [7] |
| $\text{Cut} \land \text{Cut}$ | P      | always, if $G$ has two non-adjacent vertices | NPC in linear matroids (Corollary 12) |
| $\text{Cut} \land P_{st}$ | P      | always, except if the graph is an s-t path (with multiple copies for some edges) |                       |
| $\text{Cut} \land C$   | P      | always, except if the graph is a tree, a circuit, or a bunch of parallel edges | NPC in linear matroids ($\leftrightarrow$ the matroid is not uniform, Theorem 9) |

Table 3: 10 COVERING PROBLEMS

| Problem     | Status | Reference | Remark                  |
|-------------|--------|-----------|-------------------------|
| $P \cup P$  | NPC    | Theorem 5  | NPC for subquadratic planar |
| $P \cup P_{st}$ | NPC    | Theorem 5  | NPC for subquadratic planar |
| $P \cup C$  | NPC    | Theorem 5  | NPC for subquadratic planar |
| $P_{st} \cup P_{st}'$ | NPC    | Theorem 5  | NPC for subquadratic planar |
| $P_{st} \cup C$ | NPC    | Theorem 5  | NPC for subquadratic planar |
| $C \cup C$  | NPC    | Theorem 5  | NPC for subquadratic planar |
| $\text{Cut} \cup \text{Cut}$ | NPC    | if and only if 4-colourable | always in planar Appel et al. [11], [2] |
| $\text{Cut} \cup C$ | NPC    | Theorem 3  | NPC for subcubic planar |
| $\text{Cut} \cup P$ | NPC    | Theorem 3  | NPC for subcubic planar |
| $\text{Cut} \cup P_{st}$ | NPC    | Theorem 3  | NPC for subcubic planar |

Problems $P_{st} + \text{SpT}$ and $T + \text{SpT}$ were posed in the open problem portal called “EGRES Open” [8] of the Egerváry Research Group. Most of the NP-complete problems remain NP-complete for planar graphs, though we do not know yet the status of Problems $T + T$, $T + \text{SpT}$, Cut+$F$ $P_{st} \land \text{SpT}$, and $C \land \text{SpT}$ for planar graphs, as indicated in the table.

We point out to an interesting phenomenon: planar duality and the NP-completeness of Problem $C + C$ gives that deciding whether the edge set of a planar graph is the disjoint union of two simple cuts is NP-complete (a simple cut, or bond of a graph is an inclusionwise minimal cut). In contrast, the edge set of a graph is Cut+$\text{Cut}$ if and only if the graph is bipartite on at least 3 nodes [4] that is

---

1It is easy to see that the edge set of a connected bipartite graph on at least 3 nodes is Cut+$\text{Cut}$. On the other hand, the intersection of a cut and a circuit contains an even number of edges, therefore the edge set of a non-bipartite graph cannot be Cut+$\text{Cut}$.
Cut+Cut is polynomially solvable even for non-planar graphs.

Some of the problems can be formulated as a problem in the graphic matroid and therefore also have a natural matroidal generalization. For example the matroidal generalization of $C \cap C$ is the following: can we find two disjoint circuits in a matroid (given with an independence oracle, say)? Of course, such a matroidal question is only interesting here if it can be solved for graphic matroids in polynomial time. Some of these matroidal questions is known to be solvable (e.g., the matroidal version of SpT+SpT), and some of them was unknown (at least for us): the best example being the (above mentioned) matroidal version of $C \cap C$. In the table above we indicate these matroidal generalizations, too, where the meaning of the problem is understandable. The matroidal generalization of spanning trees, forests, circuits is straightforward. We do not want to make sense of matroidal generalizations, too, where the meaning of the problem is understandable. The matroidal generalization of trees, paths, or $s$-$t$-paths in matroids. On the other hand, cuts deserve some explanation. In matroid theory, a cut (also called bond in the literature) of a matroid is defined as an inclusionwise minimal subset of elements that intersects every base. In the graphic matroid this corresponds to a simple cut of the graph defined above. So we will only consider packing problems for cuts in matroids: for example the problem of type $A \cup C$ in graphs is equivalent to the problem of packing $A$ and a simple cut in the graph, therefore the matroidal generalization is understandable. We will discuss these matroidal generalizations in Section 4.

2 NP-completeness proofs

A graph $G = (V, E)$ is said to be subcubic if $d_G(v) \leq 3$ for every $v \in V$. In many proofs below we will use Problem Planar3RegHam and Problem Planar3RegHam-e given below.

**Problem 1** (Planar3RegHam). Given a 3-regular planar graph $G = (V, E)$, decide whether there is a Hamiltonian circuit in $G$.

**Problem 2** (Planar3RegHam-e). Given a 3-regular planar graph $G = (V, E)$ and an edge $e \in E$, decide whether there is a Hamiltonian circuit in $G$ through edge $e$.

It is well-known that Problems Planar3RegHam and Planar3RegHam-e are NP-complete (see Problem [GT37] in [10]).

2.1 NP-completeness proofs for subcubic planar graphs

![Figure 1: An illustration for the proof of Theorem 3](image)

**Theorem 3.** The following problems are NP-complete, even if restricted to subcubic planar graphs: Cut+C, Cut+C, C+F, Cut+P, Cut+Pst, Cut+P, Cut+Pst, Cut+T, P+F, Pst+F.

**Proof.** All the problems are clearly in NP. First we prove the completeness of Cut+C, Cut+C and C+F using a reduction from Problem Planar3RegHam. Given an instance of the Problem Planar3RegHam with the 3-regular planar graph $G$, construct the following graph $G'$. First subdivide each edge $e = v_1v_2 \in E(G)$ with 3 new nodes $x_e^1$, $x_e$, $x_e^2$ such that they form a path in the order $v_1$, $x_e^1$, $x_e$, $x_e^2$, $v_2$. Now for any node $u \in V(G)$ and any pair of edges $e, f \in E(G)$ incident to $u$ connect $x_e^u$ and $x_f^u$ with a new edge. Finally, delete all the original nodes $v \in V(G)$ to get $G'$. Informally speaking, $G'$ is obtained from $G$ by blowing a triangle into every node of $G$ and subdividing each original edge with a new node: see Figure 1 a, b, for an illustration. Note that by contracting these triangles in $G'$ and undoing the subdivision vertices of form $x_e$ gives back $G$. Clearly, the resulting graph $G'$ is still planar and has maximum degree 3 (we mention that the subdivision nodes of form $x_e$ are only needed for the Problem Cut+C). We will prove that $G$ contains a Hamiltonian circuit if and only if $G'$ contains a circuit covering odd circuits (i.e., the edge-set of $G'$ is C+Cut) if and only if
the edge-set of $G'$ is $C + \text{Cut}$ if and only if $G'$ contains a circuit covering all the circuits (i.e., the edge set of $G'$ is $C + F$). First let $C$ be a Hamiltonian circuit in $G$. We define a circuit $C'$ in $G'$ as follows. For any $v \in V(G)$, if $C$ uses the edges $e, f$ incident to $v$ then let $C'$ use the 3 edges $e, x_e', x_f', e_f$ (see Figure 1 for an illustration). Observe that $G' - C'$ is a forest, proving that the edge-set of $G'$ is $C + F$. Similarly, the edge set of $G' - C'$ is a cut of $G'$, proving that the edge-set of $G'$ is $C + \text{Cut}$.

Finally we show that if the edge set of $G'$ is $C \cup F$ then $G$ contains a Hamiltonian circuit: this proves the sequence of equivalences stated above (the remaining implications being trivial). Assume that $G'$ has a circuit $C'$ that intersects the edge set of every odd circuit of $G'$. Contract the triangles of $G'$ and undo the subdivision nodes of form $x$, and observe that $C'$ becomes a Hamiltonian circuit of $G$.

For the rest of the problems we use PLANAR3REGHAM. Given the 3-regular planar graph $G$ and an edge $e = v_1v_2 \in E(G)$, first construct the graph $G'$ as above. Next modify $G'$ the following way: if $x_{e_1}, x_e, x_{e_2}$ are the nodes of $G'$ arising from the subdivision of the original edge $e \in E(G)$ then let $G'' = (G' - x_e) + (x_{e_1}a_i, a_i b_i, b_i c_i, c_i a_i : i = 1, 2)$, where $a_i, b_i, c_i (i = 1, 2)$ are 6 new nodes (informally, “cut off” the path $x_{e_1}, x_e, x_{e_2}$ at $x_e$ and substitute the arising two vertices of degree 1 with two triangles). An illustration can be seen in Figure 1.

Let $s = c_1$ and $t = c_2$. The following chain of equivalences settles the NP-completeness of the rest of the problems promised in the theorem. The proof is similar to the one above and is left to the reader.

There exists a Hamiltonian circuit in $G$ using the edge $e \iff$ the edge set of $G''$ is $\text{Cut} + P_{st} \iff$ the edge set of $G''$ is $\text{Cut} + P \iff$ the edge set of $G''$ is $\text{Cut} + P + \text{Cut} \iff$ the edge set of $G''$ is $\text{Cut} + P \iff$ the edge set of $G''$ is $P + F \iff$ the edge set of $G''$ is $P + F$. □

### 2.2 NP-completeness proofs based on Kotzig’s theorem

Now we prove the NP-completeness of many other problems in our collection using the following elegant result proved by Kotzig [16].

**Theorem 4.** A 3-regular graph contains a Hamiltonian circuit if and only if the edge set of its line graph can be decomposed into two Hamiltonian circuits.

This theorem was used to prove NP-completeness results by Pike in [22]. Another useful and well known observation is the following: the line graph of a planar 3-regular graph is 4-regular and planar.

**Theorem 5.** The following problems are NP-complete, even if restricted to subquadratic planar graphs: $P + F, P + P_{st}, P + C, P + T, P + SpT, P + F, P_{st} + P_{st'}', P_{st} + C, P_{st} + F, P_{st} + T, P_{st} + SpT, C + C, C + T, C + SpT, C + F, P \cup P, P \cup P_{st}, P \cup C, P_{st} \cup P_{st'}, P_{st} \cup C, C \cup C$.

**Proof.** The problems are clearly in NP. Let $G$ be a planar 3-regular graph. Since $L(G)$ is 4-regular, it is decomposable to two circuits if and only if it is decomposable to two Hamiltonian circuits. This together with Kotzig’s theorem shows that $C + C$ is NP-complete. For every other problem of type $C + A$ use $L = L(G) - st$ with an arbitrary edge $st$ of $L(G)$. Let $C$ be a circuit of $L$ and observe that (by the number of edges of $L$ and the degree conditions) $L - C$ is circuit-free if and only if $C$ is a Hamiltonian circuit and $L - C$ is a Hamiltonian path connecting $s$ and $t$.

For the rest of the partitioning type problems we need one more trick. Let us be given a 3-regular planar graph $G = (V, E)$ and an edge $e = xy \in E$. We construct another 3-regular planar graph $G'' = (V', E')$ as follows. Delete edge $xy$, add vertices $x', y'$, and add edges $xx', yy'$ and add two parallel edges between $x'$ and $y'$, namely $e_{xy'}$ and $f_{xy}$ (note that $G''$ is planar, too). Clearly $G$ has a Hamiltonian circuit through edge $xy$ if and only if $G''$ has a Hamiltonian circuit. Now consider $L(G'')$, the line graph of $G''$, it is a 4-regular planar graph. Note, that in $L(G'')$ there are two parallel edges between nodes $s = c_{xy}$ and $t = f_{xy}$, call these edges $g_1$ and $g_2$. Clearly, $L(G'')$ can be decomposed into two Hamiltonian circuits if and only if $L' = L(G'') - g_1 - g_2$ can be decomposed into two Hamiltonian paths. Let $P$ be a path in $L'$ and notice again that the number of edges of $L'$ and the degrees of the nodes in $L'$ imply that $L' - P$ is circuit free if and only if $P$ and $L' - P$ are two Hamiltonian paths in $L'$.

Finally, the NP-completeness of the problems of type $A \cup B$ is an easy consequence of the NP-completeness of the corresponding partitioning problem $A + B$: use the same construction and observe that the number of edges enforce the two objects in the cover to be disjoint. □

We remark that the above theorem gives a new proof of the NP-completeness of Problems $C + F, P + F$ and $P_{st} + F$, already proved in Theorem 4.

### 2.3 NP-completeness of Problems $P_{st} \land SpT, T + SpT, C \land SpT,$ and $\text{Cut} + F$

First we show the NP-completeness of Problems $P_{st} \land SpT, T + SpT,$ and $C \land SpT$. Problem $T + T$ was proved to be NP-complete by Pálvolgyi in [20] (the NP-completeness of this problem with the
Proof. It is clear that the problems are in NP. Their completeness will be shown by a reduction from the well known NP-complete problems 3SAT or the problem ONE-IN-THREE 3SAT (Problems LO2 and LO4 in [10]). Let \( \varphi \) be a 3-CNF formula with variable set \( \{x_1, x_2, \ldots, x_n\} \) and clause set \( C = \{C_1, C_2, \ldots, C_m\} \) (where each clause contains exactly 3 literals). Assume that literal \( x_i \) appears in \( k_j \) clauses \( C_{a_{ij}}^1, C_{a_{ij}}^2, \ldots, C_{a_{ij}}^{k_j} \), and literal \( \tau_j \) occurs in \( l_j \) clauses \( C_{b_{lj}}^1, C_{b_{lj}}^2, \ldots, C_{b_{lj}}^{l_j} \). Construct the following graph \( G_{\varphi} = (V, E) \).

For an arbitrary clause \( C \in C \) we will introduce the new node \( u_C \), and for every literal \( y \) in \( C \) we introduce two more nodes \( v(y, C), w(y, C) \). Introduce the edges \( u_C w(y, C), w(y, C)v(y, C) \) for every clause \( C \) and every literal \( y \) in \( C \) (the nodes \( w(y, C) \) will have degree 2).

For every variable \( x_j \), introduce 8 new nodes \( z_{j1}^1, z_{j2}^1, w_{j1}^1, w_{j2}^1, w_{j3}^1, w_{j4}^1, \) for every variable \( x_j \), let \( G_{\varphi} \) contain a circuit on the \( k_j + l_j + 4 \) nodes \( z_{j1}^1, v(x_j, C_{a_{lj}}^1), v(x_j, C_{a_{lj}}^2), \ldots, v(x_j, C_{a_{lj}}^{k_j}), w_{j1}^1, z_{j2}^1, w_{j2}^1, (v(\tau_j, C_{b_{lj}}^1), v(\tau_j, C_{b_{lj}}^2), \ldots, v(\tau_j, C_{b_{lj}}^{l_j})) \) in this order. We say that this circuit is associated to variable \( x_j \). Connect the nodes \( z_{j1}^1 \) and \( z_{j+1}^1 \) with an edge for every \( j = 1, 2, \ldots, n - 1 \). Introduce furthermore a path on nodes \( w_{j1}^1, w_{j2}^1, w_{j3}^1, w_{j4}^1 \) in this order and add the edges \( w_{j1}^1w_{j2}^1, w_{j2}^1w_{j3}^1, w_{j3}^1w_{j4}^1, w_{j4}^1w_{j1}^1 \) for every \( j = 1, 2, \ldots, n \). Let \( s = z_1^1 \) and \( t = z_n^1 \).

The construction of the graph \( G_{\varphi} \) is finished. An illustration can be found in Figure 2.

Clearly, \( G_{\varphi} \) is simple and has maximum degree three.

If \( \tau \) is a truth assignment to the variables \( x_1, x_2, \ldots, x_n \) then we define an \( s-t \) path \( P_\tau \) as follows: for every \( j = 1, 2, \ldots, n \), if \( x_j \) is set to TRUE then let \( P_\tau \) go through the nodes \( z_{j1}^1, v(\tau_j, C_{a_{lj}}^1), v(\tau_j, C_{a_{lj}}^2), \ldots, v(\tau_j, C_{b_{lj}}^{l_j}), w_{j1}^1, z_{j2}^1 \), otherwise (i.e., if \( x_j \) is set to FALSE) let \( P_\tau \) go through \( z_{j1}^1, v(x_j, C_{a_{lj}}^1), v(x_j, C_{a_{lj}}^2), \ldots, v(x_j, C_{a_{lj}}^{k_j}), w_{j1}^1, z_{j2}^1 \).

We need one more concept. An \( s-t \) path \( P \) is called an assignment-defining path if \( v \in V(P) \), \( d_C(v) = 2 \) implies \( v \in \{s, t\} \). For such a path \( P \) we define the truth assignment \( \tau_P \) such that \( P_{\tau_P} = P \).

Claim 1. There is an \( s-t \) path \( P \subseteq E \) such that \( (V, E - P) \) is connected if and only if \( \varphi \in 3\text{SAT} \). Consequently, Problem \( P_{st} \land \text{SpT} \) is NP-complete.
**Proof.** If \( \tau \) is a truth assignment showing that \( \varphi \in 3\text{SAT} \) then \( P_{\tau} \) is a path satisfying the requirements, as one can check. On the other hand, if \( P \) is an \( s\text{-}t \) path such that \((V, E \setminus P)\) is connected then \( P \) cannot go through nodes of degree 2, therefore \( P \) is assignment-defining, and \( \tau_P \) shows \( \varphi \in 3\text{SAT} \). \( \square \)

To show the NP-completeness of Problem T+SpT, modify \( G_{\varphi} \) the following way: subdivide the two edges incident to \( s \) with two new nodes \( s' \) and \( s'' \) and connect these two nodes with an edge. Repeat this with \( t \): subdivide the two edges incident to \( t \) with two new nodes \( t' \) and \( t'' \) and connect \( t' \) and \( t'' \). Let the graph obtained this way be \( G = (V, E) \). Clearly, \( G \) is subcubic and simple. Note that the definition of an assignment defining path and that of \( P_{\tau} \) for a truth assignment \( \tau \) can be obviously modified for the graph \( G \).

**Claim 2.** There exists a truth assignment \( \tau \) such that every clause in \( \varphi \) contains exactly one true literal if and only if there exists a set \( T \subseteq E \) such that \((V(T), T)\) is a tree and \((V, E \setminus T)\) is a spanning tree. Consequently, Problem T+SpT is NP-complete.

**Proof.** If \( \tau \) is a truth assignment as above then one can see that \( T = P_{\tau} \) is an edge set satisfying the requirements.

On the other hand, assume that \( T \subseteq E \) is such that \((V(T), T)\) is a tree and \( T^* = (V, E \setminus T) \) is a spanning tree. Since \( T^* \) cannot contain circuits, \( T \) must contain at least one of the 3 edges \( ss', ss'', s's \) (call it \( e \)), as well as at least one of the 3 edges \( tt', tt'', t't \) (say \( f \)). Since \((V(T), T)\) is connected, \( T \) contains a path \( P \subseteq T \) connecting \( e \) and \( f \) (note that since \((V, E \setminus T) \) is connected, \( |T \cap \{ss', ss'', s's\}| = |T \cap \{tt', tt'', t't\}| = 1 \)). Since \((V, E \setminus P)\) is connected, \( P \) cannot go through nodes of degree 2 (except for the endnodes of \( P \)), and the edges \( e \) and \( f \) must be the last edges of \( P \) (otherwise \( P \) would disconnect \( s \) or \( t \) from the rest). Thus, without loss of generality we can assume that \( P \) connects \( s \) and \( t \) (by locally changing \( P \) at its ends), and we get that \( P \) is assignment defining. Observe that in fact \( T \) must be equal to \( P \), since \( G \) is subcubic (therefore \( T \) cannot contain nodes of degree 3). Consider the truth assignment \( \tau_P \) associated to \( P \), we claim that \( \tau_P \) satisfies our requirements. Clearly, if a clause \( C \) of \( \varphi \) does not contain a true literal then \( u_{C'} \) is not reachable from \( s \) in \( G \), therefore every clause of \( \varphi \) contains at least one true literal. On the other hand assume that a clause \( C \) contains at least 2 true literals (say \( x_j \) and \( \tau_x \) for some \( j \neq k \)), then one can see that there exists a circuit in \( G \) (because \( v(x_j, C) \) is still reachable from \( v(\tau_x, C) \) in \( G - u_C \) via the nodes \( w^1_i, w^2_i, w^3_i \) and \( w^1_k, w^2_k, w^3_k \)). \( \square \)

Finally we prove the NP-completeness of Problem C\( \land \)SpT. For the 3CNF formula \( \varphi \) with variables \( x_1, x_2, \ldots, x_n \) and clauses \( C_1, C_2, \ldots, C_m \), let us associate the 3CNF formula \( \varphi' \) with the same variable set and clauses \( (x_1 \lor x_1 \lor \tau_x), (x_2 \lor x_2 \lor \tau_x), \ldots, (x_n \lor x_n \lor \tau_x) \), \( C_1, C_2, \ldots, C_m \). Clearly, \( \varphi \) is satisfiable if and only if \( \varphi' \) is satisfiable. Construct the graph \( G_{\varphi'} = (V, E) \) as above (the construction is clear even if some clauses contain only 2 literals), and let \( G = (V, E) \) be obtained from \( G_{\varphi'} \) by adding the edge \( st \).

**Claim 3.** The formula \( \varphi' \) is satisfiable if and only if there exists a set \( K \subseteq E \) such that \((V(K), K)\) is a circuit and \( G \setminus K = (V, E \setminus K) \) is connected. Consequently, Problem C\( \land \)SpT is NP-complete.

**Proof.** First observe that if \( \tau \) is a truth assignment satisfying \( \varphi' \) then \( K = P_{\tau} \cup \{st\} \) is an edge set satisfying the requirements. On the other hand, if \( K \) is an edge set satisfying the requirements then \( K \) cannot contain nodes of degree 2, since \( G \setminus K \) is connected. We claim that \( K \) can neither be a circuit associated to a variable \( x_i \), because in this case the node \( u_{C'} \) associated to clause \( C = (x_i \lor x_i \lor \tau_x) \) would not be reachable in \( G \setminus K \) from \( s \). Therefore \( K \) consists of the edge \( st \) and an assignment defining path \( P \). It is easy to check (analogously to the previous arguments) that \( \tau_P \) is a truth assignment satisfying \( \varphi' \).

As we have proved the NP-completeness of all three problems, the theorem is proved. \( \square \)

We note that the construction given in our original proof of the above theorem (see [5]) was used by Bang-Jensen and Yeo in [3]. They settled an open problem raised by Thomassé in 2005. They proved that it is NP-complete to decide SpA\( \land \)SpT in digraphs, where SpA denotes a spanning arborescence and SpT denotes a spanning tree in the underlying undirected graph.

We also point out that the planarity of the graphs in the above proofs cannot be assumed. We do not know the status of any of the Problems P\( \setminus \)SpT, T+SpT, and T+T in planar graphs. It was shown in [4] that Problem C\( \land \)SpT is polynomially solvable in planar graphs. We also mention that planar duality gives that Problem C\( \land \)SpT in a planar graph is equivalent to finding a cut in a planar graph that contains no circuit: by the results of [4], this problem is also polynomially solvable. However van den Heuvel [11] has shown that this problem is NP-complete for general (i.e., not necessarily planar) graphs.

We point out to an interesting connection towards the Graphic TSP Problem. This problem can be formulated as follows. Given a connected graph \( G = (V, E) \), find a connected Eulerian subgraph
of 2G spanning V with minimum number of edges (where 2G = (V, 2E) is the graph obtained from G by doubling its edges). The connection is the following. Assume that F ⊆ 2E is a feasible solution to the problem. A greedy way of improving F would be to delete edges from it, while maintaining the feasibility. It is thus easy to observe that this greedy improvement is possible if and only if the graph (V, F) contains an edge-disjoint circuit and a spanning tree (which is Problem C ∩ SpT in our notations). However, slightly modifying the proof above it can be shown that Problem C ∩ SpT is also NP-complete in Eulerian graphs (details can be found in [4]).

**Theorem 7.** Problem Cut+F is NP-complete.

**Proof.** The problem is clearly in NP. In order to show its completeness let us first rephrase the problem. Given a graph, Problem Cut+F asks whether we can colour the nodes of this graph with two colours such that no monochromatic circuit exists.

Consider the NP-complete Problem 2-COLOURABILITY OF A 3-UNIFORM HYPERGRAPH. This problem is the following: given a 3-uniform hypergraph H = (V, E), can we colour the set V with two colours (say red and blue) such that there is no monochromatic hyperedge in E (the problem is indeed NP-complete, since Problem GT6 in [10] is a special case of this problem). Given the instance H = (V, E) of this problem, construct the following graph G. The node set of G is V ∪ V̅, where V̅ is disjoint from V and it contains 6 nodes for every hyperedge in E: for an arbitrary hyperedge e = {v1, v2, v3} ∈ E, the 6 new nodes associated to it are x_{v1,e}, y_{v1,e}, x_{v2,e}, y_{v2,e}, x_{v3,e}, y_{v3,e}. The edge set of G contains the following edges: for the hyperedge e = {v1, v2, v3} ∈ E, x1 is connected with x_{v1,e} and y_{v1,e} for every i = 1, 2, 3, and among the 6 nodes associated to e every two is connected with an edge except for the 3 pairs of form x_{v1,e}, y_{v1,e} for i = 1, 2, 3 (i.e., |E(G)| = 18|E|). The construction of G is finished. An illustration can be found in Figure 3. Note that in any two-colouring of V ∪ V̅ the 6 nodes associated to the hyperedge e = {v1, v2, v3} ∈ E do not induce a monochromatic circuit if and only if there exists a permutation i, j, k of 1, 2, 3 so that they are coloured the following way: x_{v1,e}, y_{v1,e} is blue, x_{vj,e}, y_{vj,e} is red and x_{vk,e}, y_{vk,e} are of different colour.

One can check that V can be coloured with 2 colours such that there is no monochromatic hyperedge in E if and only if V ∪ V̅ can be coloured with 2 colours such that there is no monochromatic circuit in G.

We note that we do not know the status of Problem Cut+F in planar graphs.

### 3 Algorithms

**Algorithm for P_{st} ∩ C.** Assume we are given a connected multigraph G = (V, E) and two nodes s, t ∈ V, and we want to decide whether an s-t-path P ⊆ E and a circuit C ⊆ E exists with P ∩ C = ∅. We may even assume that both s and t have degree at least two. If v ∈ V \{s, t\} has degree at most
two then we can eliminate it. If there is a cut-vertex \( v \in V \) then we can decompose the problem into smaller subproblems by checking whether \( s \) and \( t \) fall in the same component of \( G - v \), or not. If they do then \( P \) should lie in that component, otherwise \( P \) has to go through \( v \).

If there is a non-trivial two-edge \( s-t \)-cut (i.e., a set \( X \subseteq V - \{s, t\} \) and \( \delta_0(X) = 2 \)), then we can again reduce the problem in a similar way: the circuit to be found cannot use both edges entering \( X \) and we have to solve two smaller problems obtained by contracting \( X \) for the first one, and contracting \( V - X \) for the second one.

So we can assume that \( |E| \geq n + [n/2] - 1 \), and that \( G \) is 2-connected and \( G \) has no non-trivial two-edge \( s-t \)-cuts. Run a BFS from \( s \) and associate levels to vertices (\( s \) gets 0). If \( t \) has level at most \( [n/2] - 1 \) then we have a path of length at most \( [n/2] - 1 \) from \( s \) to \( t \), after deleting its edges, at least \( n \) edges remain, so we are left with a circuit.

So we may assume that the level of \( t \) is at least \( [n/2] \). As \( G \) is 2-connected, we must have at least two vertices on each intermediate level. Consequently \( n \) is even, \( t \) is on level \( n/2 \), and we have exactly two vertices on each intermediate level, and each vertex \( v \in V - \{s, t\} \) has degree 3, or, otherwise for a minimum \( s-t \) path \( P \) we have that \( G - P \) has at least \( n \) edges, i.e., it contains a circuit. We have no non-trivial two-edge \( s-t \)-cuts, consequently there can only be two cases: either \( G \) equals to \( K_4 \) with edge \( st \) deleted, or \( G \) arises from a \( K_4 \) such that two opposite edges are subdivided (and these subdivision nodes are \( s \) and \( t \)). In either cases we have no solution.

Algorithm for \( C \land C \). We give a simple polynomial time algorithm for deciding whether two edge-disjoint circuits can be found in a given connected multigraph \( G = (V,E) \). We note that a polynomial (but less elegant) algorithm for this problem was also given in [8].

If any vertex has degree at most two, we can eliminate it, so we may assume that the minimum degree is at least 3. If \( G \) has at least 16 vertices, then it has a circuit of length at most \( n/2 \) (simply run a BFS from any node and observe that there must be a non-tree edge between some nodes of depth at most \( n/2 \), giving us a circuit of length at most \( 2 \log(n) \leq n/2 \), and after deleting the edges of this circuit, at least \( n \) edges remain, so we are left with another circuit. For smaller graphs we can check the problem in constant time.

### 4 Matroidal generalizations

In this section we will consider the matroidal generalizations for the problems that were shown to be polynomially solvable in the graphic matroid. In fact we will only need linear matroids, since it turns out that the problems we consider are already NP-complete in them. We will use the following result of Khachyan.

**Theorem 8** (Khachyan [11]). Given a \( D \times N \) matrix over the rationals, it is NP-complete to decide whether there exist \( D \) linearly dependent columns.

First we consider the matroidal generalization of Problem \( C \land C \).

**Theorem 9.** It is NP-complete to decide whether an (explicitly given) linear matroid contains a cut and a circuit that are disjoint.

**Proof.** Observe that there is no disjoint cut and circuit in a matroid if and only if every circuit contains a base, that is equivalent with the matroid being uniform. Khachyan’s Theorem 8 is equivalent with the uniformness of the linear matroid determined by the columns of the matrix in question, proving our theorem.

Finally we consider the matroidal generalization of Problem \( C \land C \) and \( Cut \land Cut \).

**Theorem 10.** The problem of deciding whether an (explicitly given) linear matroid contains two disjoint circuits is NP-complete.

**Proof.** We will prove here that Khachyan’s Theorem 8 is true even if \( N = 2D + 1 \), which implies our theorem, since there are two disjoint circuits in the linear matroid represented by this \( D \times (2D + 1) \) matrix if and only if there are \( D \) linearly dependent columns in it.

Khachyan’s proof of Theorem 8 was simplified by Vardy [26], we will follow his line of proof. Consider the following problem.

**Problem 11.** Given different positive integers \( a_1, a_2, \ldots, a_n, b \) and a positive integer \( d \), decide whether there exist \( d \) indices \( 1 \leq i_1 < i_2 < \cdots < i_d \leq n \) such that \( b = a_i_1 + a_i_2 + \cdots + a_i_d \).

Note that Problem 11 is very similar to the SUBSET-SUM Problem (Problem SP13 in [10]), the only difference being that in the SUBSET-SUM problem we do not specify \( d \), and the numbers \( a_1, a_2, \ldots, a_n \) need not be different. On the other hand, here we will strongly need that the numbers \( a_1, a_2, \ldots, a_n \) are all different. Vardy has shown the following claim (we include a proof for sake of completeness).
Claim 4. There is solution to Problem \([P]\) if and only if there are \(d+1\) linearly dependent columns (above the rationals) in the \((d+1) \times (n+1)\) matrix

\[
\begin{pmatrix}
1 & 1 & \cdots & 1 & 0 \\
 a_1 & a_2 & \cdots & a_n & 0 \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 a_i^{d-2} & a_i^{d-2} & \cdots & a_i^{d-2} & 0 \\
 a_i^{d-1} & a_i^{d-1} & \cdots & a_i^{d-1} & 1 \\
 a_i^1 & a_i^1 & \cdots & a_i^1 & b \\
\end{pmatrix}
\]

Proof. We use the following facts about determinants. Given real numbers \(x_1, x_2, \ldots, x_k\), we have the following well-known relation for the Vandermonde determinant:

\[
\det \begin{pmatrix}
1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_k \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{k-2} & x_2^{k-2} & \cdots & x_k^{k-2} \\
x_1^k & x_2^k & \cdots & x_k^k \\
\end{pmatrix} = \prod_{i<j}(x_j - x_i).
\]

Therefore the Vandermonde determinant is not zero, if the numbers \(x_1, x_2, \ldots, x_k\) are different. Furthermore, we have the following relation for an alternant of the Vandermonde determinant (see Chapter V in [17], for example):

\[
\det \begin{pmatrix}
1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_k \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{k-2} & x_2^{k-2} & \cdots & x_k^{k-2} \\
x_1^k & x_2^k & \cdots & x_k^k \\
\end{pmatrix} = (x_1 + x_2 + \cdots + x_k) \prod_{i<j}(x_j - x_i).
\]

We include a proof of this last fact: given an arbitrary \(k \times k\) matrix \(X = ((x_{ij}))\) and numbers \(u_1, u_2, \ldots, u_k\), observe (by checking the coefficients of the \(u_i\)s on each side) that

\[
\det \begin{pmatrix}
 x_{11} & x_{12} & \cdots & x_{1k} \\
x_{21} & x_{22} & \cdots & x_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
x_{k1} & x_{k2} & \cdots & x_{kk} \\
\end{pmatrix} + \det \begin{pmatrix}
 x_{11} & x_{12} & \cdots & x_{1k} \\
x_{21} & x_{22} & \cdots & x_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
x_{k1} & x_{k2} & \cdots & x_{kk} \\
\end{pmatrix} + \cdots +
\]

\[
\det \begin{pmatrix}
 x_{11} & x_{12} & \cdots & x_{1k} \\
x_{21} & x_{22} & \cdots & x_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
x_{k1} & x_{k2} & \cdots & x_{kk} \\
\end{pmatrix} = (u_1 + u_2 + \ldots + u_k) \det \begin{pmatrix}
 x_{11} & x_{12} & \cdots & x_{1k} \\
x_{21} & x_{22} & \cdots & x_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
x_{k1} & x_{k2} & \cdots & x_{kk} \\
\end{pmatrix}
\]

Now apply this to the Vandermonde matrix \(X = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_k \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{k-1} & x_2^{k-1} & \cdots & x_k^{k-1} \end{pmatrix}\) and numbers \(u_i = x_i\) for every \(i = 1, 2, \ldots, k\).

We will use these two facts. The first one implies that if \(d+1\) columns of our matrix are dependent then they have to include the last column. By the second fact, if \(1 \leq i_1 < i_2 < \cdots < i_d \leq n\) are arbitrary indices then

\[
\det \begin{pmatrix}
1 & 1 & \cdots & 1 & 0 \\
a_{i_1} & a_{i_2} & \cdots & a_{i_d} & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_i^{d-2} & a_i^{d-2} & \cdots & a_i^{d-2} & 0 \\
a_i^{d-1} & a_i^{d-1} & \cdots & a_i^{d-1} & 1 \\
a_i^1 & a_i^1 & \cdots & a_i^1 & b \\
\end{pmatrix} = (b - a_{i_1} - a_{i_2} - \cdots - a_{i_d}) \prod_{k<l}(a_{i_k} - a_{i_l}).
\]

This implies the claim.

Vardy also claimed that Problem \([P]\) is NP-complete: our proof will be completed if we show that this is indeed the case even if \(n = 2d + 2\). Since we have not found a formal proof of this claim of Vardy, we will give a full proof of the following claim. For a set \(V\) let \(\binom{V}{3} = \{X \subseteq V : |X| = 3\}\).
Claim 5. Problem $[11]$ is NP-complete even if $n = 2d + 2$.

Proof. We will reduce the well-known NP-complete problem Exact-Cover-by-3-Sets (Problem SP2 in [10]) to this problem. Problem Exact-Cover-by-3-Sets is the following: given a 3-uniform family $E \subseteq \binom{V}{3}$, decide whether there exists a subfamily $E' \subseteq E$ such that every element of $V$ is contained in exactly one member of $E'$. We assume that $3$ divides $|V|$, and let $d = |V|/3$, so Problem Exact-Cover-by-3-Sets asks whether there exist $d$ disjoint members in $E$. First we show that this problem remains NP-complete even if $|E| = 2d + 2$. Indeed, if $|E| \neq 2d + 2$ then let us introduce $3k$ new nodes $\{u_i, v_i, w_i : i = 1, 2, \ldots, k\}$ where

- $k$ is such that $3k > 2d + 2 - |E|$ if $|E| < 2d + 2$, and
- $k = |E| - (2d + 2)$, if $|E| > 2d + 2$.

Let $V^* = V \cup \{u_i, v_i, w_i : i = 1, 2, \ldots, k\}$ and let $E^* = E \cup \{\{u_i, v_i, w_i\} : i = 1, 2, \ldots, k\}$ (note that $|V^*| = 3(d + k)$). If $|E| < 2d + 2$ then include furthermore $2d(k + 2) + (|E| + k)$ arbitrary new sets of size 3 to $E^*$ from $V \setminus V^*$, but so that $E^*$ does not contain a set twice (this can be done by the choice of $k$). It is easy to see that $|E^*| = 2|V^*|/3 + 2$, and $V$ can be covered by disjoint members of $E$ if and only if $V^*$ can be covered by disjoint members of $E^*$.

Finally we show that Exact-Cover-by-3-Sets is a special case of Problem $[11]$ in disguise. Given an instance of Exact-Cover-by-3-Sets by a 3-uniform family $E \subseteq \binom{V}{r}$, consider the characteristic vectors of these 3-sets as different positive integers written in base 2 (that is, assume a fixed ordering of the maximal number of ones in the binary representation of a sum of positive integers.) This together with the previous observation proves our claim. \[ \square \]

By combining Claims $[4]$ and $[5]$ we obtain the proof of Theorem $[10]$ as follows. Consider an instance of Problem $[11]$ with $n = 2d + 2$ and let $D = d + 1$. Claim $[4]$ states that this instance has a solution if and only if the $(d + 1) \times (n + 1) = D \times (2D + 1)$ matrix defined in the claim has $D$ linearly dependent columns, which must be NP-hard to decide by Claim $[5]$. \[ \square \]

Corollary 12. The problem of deciding whether an (explicitly given) linear matroid contains two disjoint cuts is NP-complete.

Proof. Since the dual matroid of the linear matroid is also linear, and we can construct a representation of this dual matroid from the representation of the original matroid, this problem is equivalent to the problem of deciding whether a linear matroid contains two disjoint circuits, which is NP-complete by Theorem $[10]$. \[ \square \]

References

[1] K. Appel and W. Haken, Every planar map is four colorable. I. Discharging, Illinois J. Math. 21 (1977), no. 3, 429–490.
[2] K. Appel, W. Haken, and J. Koch, Every planar map is four colorable. II. Reducibility, Illinois J. Math. 21 (1977), no. 3, 491–567.
[3] J. Bang-Jensen and A. Yeo, Arc-disjoint spanning sub(di)graphs in digraphs, Theoretical Computer Science 438 (2012), 48 – 54.
[4] A. Bernáth and M. Kamiński, Nonseparating cycles in planar and eulerian graphs, Tech. Report TR-2014-05, Egerváry Research Group, Budapest, 2014, www.cs.elte.hu/egres.
[5] A. Bernáth and Z. Király, Finding edge-disjoint subgraphs in graphs, Tech. Report QP-2010-04, Egerváry Research Group, Budapest, 2010, www.cs.elte.hu/egres.
[6] H. Bodlaender, On disjoint cycles, Graph-Theoretic Concepts in Computer Science (Gunther Schmidt and Rudolf Berghammer, eds.), Lecture Notes in Computer Science, vol. 570, Springer Berlin / Heidelberg, 1992, 10.1007/3-540-55121-2_24, pp. 230–238.
[7] J. Edmonds, Minimum partition of a matroid into independent sets, J. Res. Nat. Bur. Standards B69 (1965), 67–72.
[8] Egres Open Problems www.cs.elte.hu/egres/open, Problems “Partitionability to a tree and a spanning tree” and “Edge-disjoint spanning trees and paths”
[9] A. Frank, Connections in combinatorial optimization, Oxford University Press, 2011.
10. M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.

11. J. van den Heuvel, *Private communication*.

12. H. Imai, *Network-flow algorithms for lower-truncated transversal polymatroids*, Journal of the Operations Research Society of Japan 26 (1983), no. 3, 186–210.

13. T. Kameda and S. Toida, *Efficient algorithms for determining an extremal tree of a graph (extended abstract)*, FOCS, IEEE, 1973, pp. 12–15.

14. L. Khachiyan, *On the complexity of approximating extremal determinants in matrices*, J. Complexity 11 (1995), no. 1, 138–153.

15. G. Kishi and Y. Kajitani, *Maximally distant trees and principal partition of a linear graph*, Circuit Theory, IEEE Transactions on 16 (1969), no. 3, 323 – 330.

16. A. Kotzig, *Aus der Theorie der endlichen regulären Graphen dritten und vierten Grades*, Časopis Pěst. Mat. 82 (1957), 76–92.

17. T. Muir, *Treatise on the theory of determinants*, New York: Dover, 1960.

18. C. St. J. A. Nash-Williams, *Decomposition of finite graphs into forests*, J. London Math. Soc. 39 (1964), 12.

19. C. St. J. A. Nash-Williams, *Edge-disjoint spanning trees of finite graphs*, Journal of the London Mathematical Society s1-36 (1961), no. 1, 445–450.

20. D. Pálvölgyi, *Partitionability to two trees is NP-complete*, Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae - Sectio mathematica 52 (2009), no. 1, 131–135.

21. U. Pferschy, Woeginger G. J., and Yao E., *Partitioning graphs into two trees*, Acta Cybern. 11 (1994), no. 3, 233–240.

22. D. A. Pike, *Hamilton decompositions of graphs*, Ph.D. thesis, Auburn University, 1996.

23. P. D. Seymour, *Four-terminus flows*, Networks 10 (1980), no. 1, 79–86.

24. C. Thomassen, *2-linked graphs*, European J. Combin. 1 (1980), no. 4, 371–378.

25. W. T. Tutte, *On the problem of decomposing a graph into n connected factors*, J. London Math. Soc. 36 (1961), 221–230.

26. A. Vardy, *Algorithmic complexity in coding theory and the minimum distance problem*, STOC (F. T. Leighton and P. W. Shor, eds.), ACM, 1997, pp. 92–109.