Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2011-05. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

---

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

Attila Bernáth and Zoltán Király

---

April 2011

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

Attila Bernáth* and Zoltán Király**

## 1 Introduction

In this paper 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 with a $P$), paths with specified endvertices (denoted with $P_{st}$, where $s$ and $t$ are the prescribed endvertices), circuits (denoted with $C$), forests ($F$), spanning trees ($SpT$), (not necessarily spanning) trees ($T$), and cuts (denoted by $Cut$, where a cut means the set of edges leaving a nonempty proper subset of nodes). We restrict ourselves to undirected graphs. 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 $A \wedge B$): can we **find two edge-disjoint subgraphs** in $G$, one of type $A$ and the other of type $B$?
- **Covering problem** (denoted $A \cup B$): can we **cover the edge set** of $G$ with an object of type $A$ and an object of type $B$?
- **Partitioning problem** (denoted $A + B$): can we **partition the edge set** of $G$ into an object of type $A$ and an object of type $B$?

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 $SpT + F$. This problem is in $NP \cap \text{co}(NP)$ by the results of Nash-Williams [14], polynomial algorithms for deciding the problem were given by Kishi and Kajitani [12], and Kameda and Toida [11].

*MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary, H-1117. E-mail: bernath@cs.elte.hu. Supported by OTKA grant CK80124.

**Department of Computer Science and CNL, Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary. Research is supported by the MTA-ELTE Egerváry Research Group, OTKA grants CNK 77780 and CK 80124, and TÁMOP grant 4.2.1./B-09/1/KMR-2010-0003. E-mail: kiraly@cs.elte.hu

Table 1: 25 PARTITIONING PROBLEMS

| **Problem** | **Status** | **Reference** | **Remark** |
|---|---|---|---|
| $P + P$ | NPC | Theorem 2.5 | NPC for planar |
| $P + P_{st}$ | NPC | Theorem 2.5 | NPC for planar |
| $P + C$ | NPC | Theorem 2.5 | NPC for planar |
| $P + T$ | NPC | Theorem 2.5 (see also [4]) | NPC for planar |
| $P + SpT$ | NPC | Theorem 2.5 (see also [4]) | NPC for planar |
| $P + F$ | NPC | Theorem 2.3 (and Theorem 2.5) | NPC for subcubic planar |
| $P_{st} + P_{s't'}$ | NPC | Theorem 2.5 | NPC for planar |
| $P_{st} + C$ | NPC | Theorem 2.5 | NPC for planar |
| $P_{st} + T$ | NPC | Theorem 2.5 (see also [4]) | NPC for planar |
| $P_{st} + SpT$ | NPC | Theorem 2.5 (see also [4]) | NPC for planar |
| $P_{st} + F$ | NPC | Theorem 2.3 (and Theorem 2.5) | NPC for subcubic planar |
| $C + C$ | NPC | Theorem 2.5 | NPC for planar |
| $C + T$ | NPC | Theorem 2.5 (see also [4]) | NPC for planar |
| $C + SpT$ | NPC | Theorem 2.5 (see also [4]) | NPC for planar |
| $C + F$ | NPC | Theorem 2.3 (and Theorem 2.5) | NPC for subcubic planar |
| $T + T$ | NPC | Pálvölgyi [16] | planar graphs? |
| $T + SpT$ | NPC | Theorem 2.6 (see also [4]) | planar graphs? |
| $F + F$ | P | Kishi and Kajitani [12], Kameda and Toida [11] (Nash-Williams [14]) | in P for matroids: Edmonds [6] |
| $SpT + SpT$ | P | Kishi and Kajitani [12], Kameda and Toida [11], (Nash-Williams [15], Tutte [21]) | in P for matroids: Edmonds [6] |
| $Cut + Cut$ | P | if and only if bipartite (and $\|V\| \geq 3$) | |
| $Cut + F$ | NPC | Theorem 2.7 | planar graps? |
| $Cut + C$ | NPC | Theorem 2.3 | NPC for subcubic planar |
| $Cut + T$ | NPC | Theorem 2.3 | NPC for subcubic planar |
| $Cut + P$ | NPC | Theorem 2.3 | NPC for subcubic planar |
| $Cut + P_{st}$ | NPC | Theorem 2.3 | NPC for subcubic planar |

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 an edge? With our notations this is Problem $P_{st} \wedge P_{s't'}$. This problem is still solvable in polynomial time, as was shown by Thomassen [20] and Seymour [19].

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.

Table 2: 9 PACKING PROBLEMS

| Problem | Status | Reference | Remark |
|---|---|---|---|
| $P_{st} \wedge P_{s't'}$ | P | Seymour [19], Thomassen [20] | |
| $P_{st} \wedge C$ | P | see Section 3 | |
| $P_{st} \wedge SpT$ | NPC | Theorem 2.6 (see also [4]) | planar graphs? |
| $C \wedge C$ | P | Bodlaender [5] (see also Section 3) | matroids? |
| $C \wedge SpT$ | NPC | Theorem 2.6 (see also [4]) | planar graphs? |
| $SpT \wedge SpT$ | P | Imai [10], (Nash-Williams [15], Tutte [21]) | in P for matroids: Edmonds [6] |
| $Cut \wedge Cut$ | P | always, except for a complete graph | matroids? ($\Leftrightarrow C \wedge C$) |
| $Cut \wedge P_{st}$ | P | always, except if the graph is an $s-t$ path | |
| $Cut \wedge C$ | P | always, except if the graph is a tree or a circuit | matroids? |

Table 3: 10 COVERING PROBLEMS

| Problem | Status | Reference | Remark |
|---|---|---|---|
| $P \cup P$ | NPC | Theorem 2.5 | NPC for planar |
| $P \cup P_{st}$ | NPC | Theorem 2.5 | NPC for planar |
| $P \cup C$ | NPC | Theorem 2.5 | NPC for planar |
| $P_{st} \cup P_{s't'}$ | NPC | Theorem 2.5 | NPC for planar |
| $P_{st} \cup C$ | NPC | Theorem 2.5 | NPC for planar |
| $C \cup C$ | NPC | Theorem 2.5 | NPC for planar |
| $Cut \cup Cut$ | NPC | if and only if 4-colourable | always in planar Appel et al. [2], [1] |
| $Cut \cup C$ | NPC | Theorem 2.3 | NPC for subcubic planar |
| $Cut \cup P$ | NPC | Theorem 2.3 | NPC for subcubic planar |
| $Cut \cup P_{st}$ | NPC | Theorem 2.3 | NPC for subcubic planar |

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 $A + B$**. 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 $A \cup B$**.

The setting outlined above gives us 84 problems. Note however that some of these can be omitted. For example $P \wedge A$ is trivial for each possible type $A$ in question, because $P$ may consist of one vertex only. By the same reason, $T \wedge A$ and $F \wedge A$ type problems are also trivial. Furthermore, observe that the edge-set $E(G)$ of a graph $G$ is $F + A \Leftrightarrow E(G)$ is $F \cup A \Leftrightarrow E(G)$ is $T \cup A \Leftrightarrow E(G)$ is $SpT \cup A$: therefore we will only consider the problems of form $F + A$ among these for any $A$. Similarly, the edge set $E(G)$ is $F + F \Leftrightarrow E(G)$ is $T + F \Leftrightarrow E(G)$ is $SpT + F$: again we choose to deal with $F + F$. We can also omit the problems $Cut + SpT$ and $Cut \wedge SpT$ 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.

Problems $P_{st} + SpT$ and $T + SpT$ were posed in the open problem portal called "EGRES Open"[7] of the Egerváry Research Group. Most of the NP-complete problems remain NP-complete for planar graphs, though we don't know yet the status of Problems $T + T$, $T + SpT$, $P_{st} \wedge SpT$, $C \wedge C$, $C \wedge SpT$ and $Cut + F$ for planar graphs, as indicated in the table.

We point out to an interesting thing: 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 (an inclusionwise minimal cut is called a simple cut). In contrast, the edge set of a graph is $Cut + Cut$ if and only if the graph is bipartite, that is $Cut + Cut$ is polinomially solvable.

Some of the problems can be formulated as a problem in the graphic matroid and therefore have a natural matroidal generalization, too. For example the matroidal generalization of $C \wedge C$ 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 is unknown (at least for us): the best example being the (above mentioned) matroidal version of $C \wedge C$. In the table above we indicate these matroidal generalizations, too, where the meaning of the problem is straightforward (for example the problem of type $A \wedge Cut$ in graphs is equivalent to the problem of packing $A$ and a simple cut in the graph, therefore the matroidal generalization is understandable).

We note that in our NP-completeness proofs we always show that the considered problem is NP-complete even for simple graphs. On the other hand, the polynomial algorithms given here always work for multigraphs, too.

# 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 2.1** (PLANAR3REGHAM)**.** Given a 3-regular planar graph $G = (V, E)$, decide whether there is a Hamiltonian circuit in $G$.

**Problem 2.2** (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 [8]).

## 2.1 NP-completeness proofs in subcubic planar graphs

**Theorem 2.3.** *The following problems are NP-complete, even if restricted to planar graphs of maximum degree three:* $Cut \cup C$, $Cut + C$, $C + F$, $Cut \cup P$, $Cut \cup P_{st}$, $Cut + P$, $Cut + P_{st}$, $Cut + T$, $P + F$, $P_{st} + F$.

*Proof.* All the problems are clearly in NP. First we prove the completeness of $Cut \cup C$, $Cut + C$ and $C + F$ using a reduction from Problem Planar3RegHam. Given an instance of the ProblemPlanar3RegHam with the 3-regular planar graph $G$, construct the following graph $G'$. First subdivide each edge $e = uv \in E(G)$ with 3 new nodes $x_e^u, x_e, x_e^v$ such that they form a path in the order $u, x_e^u, x_e, x_e^v, v$. 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: $G'$ is obtained from $G$ by blowing a triangle into every node of $G$ and subdividing each original edge with a new node. 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 $Cut + C$). It is easy to see 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 $C \cup Cut$) if and only if the edge-set of $G'$ is $C + Cut$ if and only if $G'$ contains a circuit covering all the circuits (i.e., the edge set of $G'$ is $C + F$).

For the rest of the problems we use Planar3RegHam–e. 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 $x_e^{v_1}, x_e, x_e^{v_1}$ are the nodes of $G'$ belonging to $e$ then let $G'' = (G' - x_e) + \{x_e^{v_i}a_i, a_ib_i, b_ic_i, c_ia_i : i = 1, 2\}$, where $a_i, b_i, c_i (i = 1, 2)$ are 6 new nodes (i.e., "cut" the path $x_e^{v_1}, x_e, x_e^{v_1}$ at $x_e$ and substitute the arising two vertices of degree 1 with two triangles). Let $s = b_1$ and $t = b_2$. The following chain of equivalences settles the NP-completeness of the rest of the problems promised in the theorem. The proof is left to the reader.

There exists a Hamiltonian circuit in $G$ using the edge $e \Leftrightarrow$ the edge set of $G''$ is $Cut + P_{st} \Leftrightarrow$ the edge set of $G''$ is $Cut + P \Leftrightarrow$ the edge set of $G''$ is $Cut + T \Leftrightarrow$ the edge set of $G''$ is $Cut \cup P_{st} \Leftrightarrow$ the edge set of $G''$ is $Cut \cup P \Leftrightarrow$ the edge set of $G''$ is $P_{st} + F \Leftrightarrow$ 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 [13].

**Theorem 2.4.** *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 [18]. Another useful and well known observation is the following: the line graph of a planar 3-regular graph is 4-regular and planar.

**Theorem 2.5.** *The following problems are NP-complete, even if restricted to planar graphs of maximum degree four:* $P+P$, $P+P_{st}$, $P+C$, $P+T$, $P+SpT$, $P+F$, $P_{st}+P_{s't'}$, $P_{st}+C$, $P_{st}+F$, $P_{st}+T$, $P_{st}+SpT$, $C+C$, $C+T$, $C+SpT$, $C+F$, $P\cup P$, $P\cup P_{st}$, $P\cup C$, $P_{st}\cup P_{s't'}$, $P_{st}\cup C$, $C\cup C$.

*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 $s = e_{xy}$ 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. □

Observe 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 2.3.

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

First we show the NP-completeness of Problems $P_{st} \wedge SpT$, $T + SpT$, and $C \wedge SpT$. Problem $T+T$ was proved to be NP-complete by Pálvölgyi in [16] (the NP-completeness of this problem with the additional requirement that the two trees have to be of equal size was proved by Pferschy, Woeginger and Yao [17]). Our reductions here are similar to the one used by Pálvölgyi in [16]. We remark that our first proof for the NP-completeness of Problems $P+T$, $P+SpT$, $P_{st}+T$, $P_{st}+SpT$, $C+T$ and $C+SpT$ used a variant of the construction below (this can be found in [4]), but later we found that using Kotzig's result (Theorem 2.4) a simpler proof can be given for these.

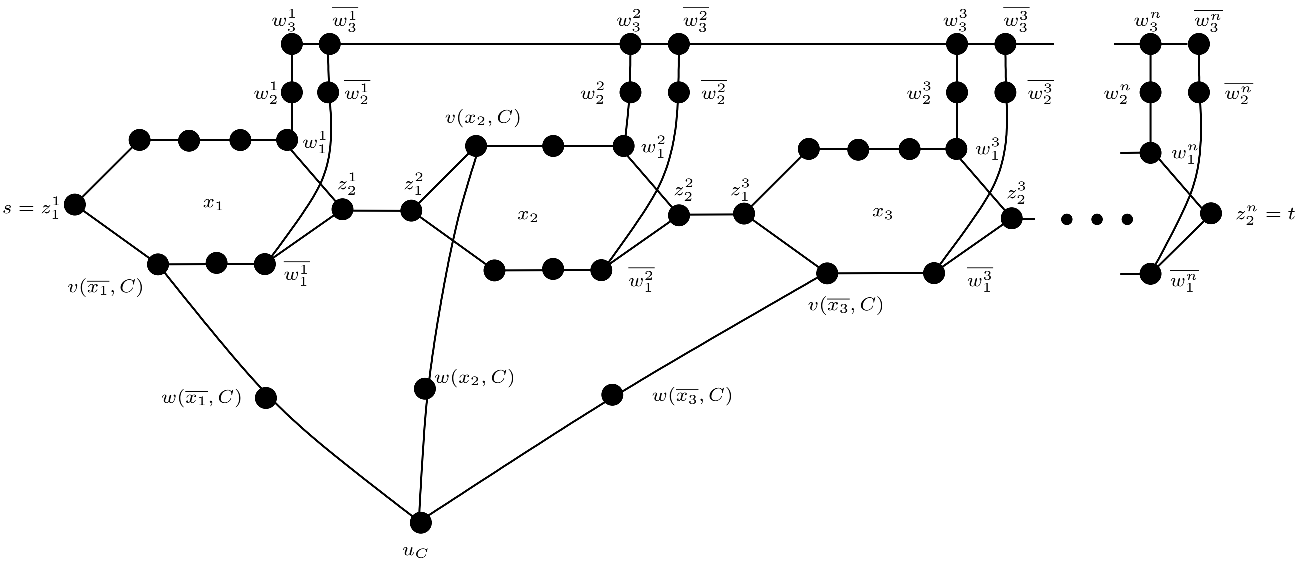


Figure 1: Part of the construction of graph $G_\varphi$ for clause $C = \{\overline{x_1}, x_2, \overline{x_3}\}$.

For a subset of edges $E' \subseteq E$ in a graph $G = (V, E)$, let $V(E')$ denote the subset of nodes incident to the edges of $E'$, i.e., $V(E') = \{v \in V : \text{there exists an } f \in E' \text{ with } v \in f\}$.

**Theorem 2.6.** *Problems $P_{st} \wedge SpT$, $T + SpT$ and $C \wedge SpT$ are NP-complete even for graphs with maximum degree 3.*

*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 [8]). Let $\varphi$ be a 3-CNF formula with variable set $\{x_1, x_2, \ldots, x_n\}$ and clause set $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ (where each clause contains exactly 3 literals). Assume that literal $x_j$ appears in $k_j$ clauses $C_{a_1^j}, C_{a_2^j}, \ldots, C_{a_{k_j}^j}$, and literal $\overline{x_j}$ occurs in $l_j$ clauses $C_{b_1^j}, C_{b_2^j}, \ldots, C_{b_{l_j}^j}$. Construct the following graph $G_\varphi = (V, E)$.

For an arbitrary clause $C \in \mathcal{C}$ we will introduce a 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 $C$ (i.e., the nodes $w(y, C)$ will have degree 2).

For every variable $x_j$ introduce 8 new nodes $z_1^j$, $z_2^j$, $w_1^j$, $\overline{w_1^j}$, $w_2^j$, $\overline{w_2^j}$, $w_3^j$, $\overline{w_3^j}$. For every variable $x_j$, let $G_\varphi$ contain a circuit on the $k_j + l_j + 4$ nodes $z_1^j$, $v(x_j, C_{a_1^j})$, $v(x_j, C_{a_2^j})$, $\ldots$, $v(x_j, C_{a_{k_j}^j})$, $w_1^j$, $z_2^j$, $\overline{w_1^j}$, $v(\overline{x_j}, C_{b_{l_j}^j})$, $v(\overline{x_j}, C_{b_{l_j-1}^j})$, $\ldots$, $v(\overline{x_j}, C_{b_1^j})$ in this order. We say that this circuit is **associated to variable** $x_j$. Connect the nodes $z_2^j$ and $z_1^{j+1}$ with an edge for every $j = 1, 2, \ldots, n-1$. Introduce furthermore a path on nodes $w_3^1, \overline{w_3^1}, w_3^2, \overline{w_3^2}, \ldots, w_3^n, \overline{w_3^n}$ in this order and add the edges $w_1^j w_2^j$, $w_2^j w_3^j$, $\overline{w_1^j}\,\overline{w_2^j}$, $\overline{w_2^j}\,\overline{w_3^j}$ for every $j = 1, 2, \ldots, n$. Let $s = z_1^1$ and $t = z_2^n$.

The construction of the graph $G_\varphi$ is finished. An illustration can be found in Figure 2.3.

Clearly, $G_\varphi$ is simple and has maximum degree three.

If $\tau$ is a truth assignment to the variables $x_1, x_2, \dots, x_n$ then we define an $s$-$t$ path $P_\tau$ as follows: for every $j = 1, 2, \dots, n$, if $x_j$ is set to TRUE then let $P_\tau$ go through the nodes $z_1^j, v(\overline{x_j}, C_{b_1^j}), v(\overline{x_j}, C_{b_2^j}), \dots, v(\overline{x_j}, C_{b_{l_j}^j}), \overline{w_1^j}, z_2^j$, otherwise (i.e., if $x_j$ is set to FALSE) let $P_\tau$ go through $z_1^j, v(x_j, C_{a_1^j}), v(x_j, C_{a_2^j}), \dots, v(x_j, C_{a_{k_j}^j}), w_1^j, z_2^j$.

We need one more concept. An $s$-$t$ path $P$ is called an *assignment-defining path* if its node set $V(P)$ does not contain nodes of degree 2. 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 $\varphi \in 3SAT$. Consequently, Problem $P_{st} \wedge SpT$ is NP-complete.

*Proof.* If $\tau$ is a truth assignment showing that $\varphi \in 3SAT$ then $P_\tau$ is a path satisfying the requirements. On the other hand, if $P$ is an $s$-$t$ path such that $(V,\ E - P)$ is connected then $P$ cannot go through nodes of degree 2, therefore $P$ is assignment-defining, and $\tau_P$ shows $\varphi \in 3SAT$. □

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 $(V,\ E - T)$ 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 $T^* = (V,\ E - T)$ is a spanning tree. Since $T^*$ cannot contain circuits, $T$ must contain at least one of the 3 edges $ss', s's'', s''s$ (say $e$), as well as at least one of the 3 edges $tt', t't'', t''t$ (say $f$). Since $(V(T), T)$ is connected, $T$ contains a path $P \subseteq T$ connecting $e$ and $f$ (note that since $(V, E - T)$ is connected, $|T \cap \{ss', s's'', s''s\}| = |T \cap \{tt', t't'', t''t\}| = 1$). Since $(V,\ E - P)$ is connected, this path is assignment defining (and we can assume without loss of generality that it connects $s$ and $t$). Observe that in fact $T$ must be equal to $P$, since every node on $P$ has degree 3 in $G$. 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 $G - T$, 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 $\overline{x_k}$ for some $j \neq k$), then one can see that there exists a circuit in $G - T$ (because $v(x_j, C)$ is still reachable from $v(\overline{x_k}, C)$ in $G - u_C$ via the nodes $w_j^1, w_j^2, w_j^3$ and $\overline{w_k^1}, \overline{w_k^2}, \overline{w_k^3}$). □

Finally we prove the NP-completeness of Problem $C \wedge SpT$. 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 formula $\varphi'$ with the same variable set and clauses $\{x_1, \overline{x_1}\}$, $\{x_2, \overline{x_2}\}$, $\ldots$, $\{x_n, \overline{x_n}\}$, $C_1, C_2, \ldots, C_m$. Clearly, $\varphi$ is satisfyable if and only if $\varphi'$ is satisfyable. 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 satisfyable if and only if there exists a set $K \subseteq E$ such that $(V(K), K)$ is a circuit and $G - K = (V, E - K)$ is connected. Consequently, Problem $C \wedge SpT$ 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 $G - K$ is connected.We claim that $K$ neither can be a circuit associated to a variable $x_i$, because in this case the node $u_C$ associated to clause $C = \{x_i, \overline{x_i}\}$ would not be reachable in $G - K$ from $s$. Therefore $K$ consists of the edge $st$ and an assignment defining path $P$. It is easy to check that – as in the previous claim – $\tau_P$ is a truth assignment giving exactly one TRUE literal in each clause. □

As we have proved the NP-completeness of all three problems, the theorem is proved. □

We note that the construction given in our original proof of the above theorem (see [4]) was used recently 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 $SpA \wedge SpT$ 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 don't know the status of any of the Problems $P_{st} \wedge SpT$, $T + SpT$, $C \wedge SpT$, and $T + T$ in planar graphs. We also mention that planar duality gives that Problem $C \wedge SpT$ in a planar graph is eqivalent to finding a cut in a planar graph that contains no circuit. By the previous remark the status of this problem is also unknown for planar graphs, however van den Heuvel [9] has shown that this problem is also NP-complete in general (i.e., for not necessarily planar graphs).

**Theorem 2.7.** *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 (also known as MONOTONE NAE EXACT 3-SAT). This problem is indeed NP-complete, since Problem GT6 in [8] is a special case of this problem. This problem is the following: given a 3-uniform hypergraph $H = (V, \mathcal{E})$, can we colour the set $V$ with two colours (say red and blue) such that there is no monochromatic hyperedge in $\mathcal{E}$. Given the instance $H = (V, \mathcal{E})$ of this problem, construct the following graph $G$. The node set of $G$ is $V \cup V_{\mathcal{E}}$, where $V_{\mathcal{E}}$ is disjoint from $V$ and it contains 6 nodes for every hyperedge in $\mathcal{E}$: for an arbitrary hyperedge $e = \{v_1, v_2, v_3\} \in \mathcal{E}$, the 6 new nodes associated to it are $x^1_{1,e}, x^2_{1,e}, x^1_{2,e}, x^2_{2,e}, x^1_{3,e}, x^2_{3,e}$. The edge set of $G$ contains the following edges: for the hyperedge $e = \{v_1, v_2, v_3\} \in \mathcal{E}$, $v_i$ is connected with $x^1_{i,e}$ and $x^2_{i,e}$ 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 $x^1_{i,e}, x^2_{i,e}$ for $i = 1, 2, 3$ (i.e., $|E(G)| = 18|\mathcal{E}|$). The construction of $G$ is finished.

One can check that $V$ can be coloured with 2 colours such that there is no monochromatic hyperedge in $\mathcal{E}$ if and only if $V \cup V_{\mathcal{E}}$ can be coloured with 2 colours such that there is no monochromatic circuit in $G$. The crucial observation is that the 6 nodes associated to the hyperedge $e = \{v_1, v_2, v_3\} \in \mathcal{E}$ do not induce a monochromatic circuit if and only if for some permutation $i, j, k$ of $1, 2, 3$ they are coloured the following way: $x^1_{i,e}, x^2_{i,e}$ is blue, $x^1_{j,e}, x^2_{j,e}$ is red and $x^1_{k,e}, x^2_{k,e}$ are of different colour. □

Again we point out that we don't know the status of Problem $Cut + F$ in planar graphs.

# 3 Algorithms

**Algorithm for $P_{st} \wedge C$.** Assume we are given a connected multigraph $G = (V, E)$ and two nodes $s, t \in V$, and we want to decide whether an $s-t$-path $P \subseteq E$ and a circuit $C \subseteq E$ exists with $P \cap C = \emptyset$. We may even assume that both $s$ and $t$ have degree at least two. If $v \in V - \{s, t\}$ has degree at most two, we may eliminate it. If there is a cut-vertex $v \in V$ then we can reduce the problem by carefully checking whether or not $s$ and $t$ fall in different components of $G - v$.

If there is a non-trivial two-edge $s-t$-cut (i.e., a set $X$ with $\{s\} \subsetneq X \subsetneq V - t$, and $d_G(X) = 2$), 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 $|E| \geq n + \lceil n/2 \rceil - 1$, 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 $\lceil n/2 \rceil - 1$ then we have a path of length at most $\lceil n/2 \rceil - 1$ 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 $\lceil n/2 \rceil$. 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 $K_4$ 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 $C \wedge C$.** 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 is also given in [5].

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 $\log(n)$, 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.

# 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*, (manuscript).

[4] 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`.

[5] 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.

[6] J. Edmonds, *Minimum partition of a matroid into independent sets*, J. Res. Nat. Bur. Standards **B69** (1965), 67–72.

[7] Egres Open Problems (`http://lemon.cs.elte.hu/egres/open`), Problems `http://lemon.cs.elte.hu/egres/open/Partitionability_to_a_tree_and_a_spanning_tree` and `http://lemon.cs.elte.hu/egres/open/Edge-disjoint_spanning_trees_and_paths`.

[8] 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.

[9] J. van den Heuvel, *Private communication.*

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

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

[12] 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.

[13] A. Kotzig, *Aus der Theorie der endlichen regulären Graphen dritten und vierten Grades*, Časopis Pěst. Mat. **82** (1957), 76–92. MR 0090815 (19,876c)

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

[15] 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.

[16] D. Pálvölgyi, *Partitionability to two trees is NP-complete*, Tech. Report QP-2006-06, Egerváry Research Group, Budapest, 2006, `www.cs.elte.hu/egres`.

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

[18] D. A. Pike, *Hamilton decompositions of graphs*, Ph.D. thesis, Auburn University, 1996.

[19] P. D. Seymour, *Four-terminus flows*, Networks **10** (1980), no. 1, 79–86. MR 565327 (81c:90042)

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

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