Finding edge-disjoint subgraphs in graphs

Attila Bernáth* and Zoltán Király **

Abstract

In this note we show the hardness of many natural edge-disjoint subgraphs problems in undirected graphs. For example, given an undirected graph $G = (V, E)$ and two nodes $s, t \in V$, it is NP-complete to decide whether there exists an s-t path $P \subseteq E$ such that $G - P = (V,\ E - P)$ is connected.

1 Introduction

Let $G = (V, E)$ be a graph. For a subset of edges $F \subseteq E$ let $V(F)$ denote the subset of nodes incident to the edges of F, i.e., $V(F) = \{v \in V : \text{there exists an } f \in F \text{ with } v \in f\}$. In this note we show the NP-completeness of the following problems, where we are always given an undirected graph $G = (V, E)$ and in many cases two additional nodes $s, t \in V$. We generally use the notation of the form $\boldsymbol{A} + \boldsymbol{B}$ for the following problem: can the edge set E of G be partitioned into $E_1 \cup E_2$ such that E_1 is of type A and E_2 is of type B?

Problem 1 ($\boldsymbol{P_{st}} + \boldsymbol{Conn}$). Decide whether there exists an s-t path $P \subseteq E$ such that $G - P = (V,\ E - P)$ is connected.

Problem 2 ($\boldsymbol{P_{st}} + \boldsymbol{SpT}$). Decide whether there exists an s-t path $P \subseteq E$ such that $G - P = (V,\ E - P)$ is a spanning tree.

Problem 3 ($\boldsymbol{P} + \boldsymbol{SpT}$). Decide whether there exists a path $P \subseteq E$ such that $G - P = (V,\ E - P)$ is a spanning tree.

Problem 4 ($\boldsymbol{T} + \boldsymbol{SpT}$). Decide whether there exists a subset $T \subseteq E$ such that $(V(T), T)$ is a tree and $G - T = (V,\ E - T)$ is a spanning tree.

Problem 5 ($\boldsymbol{P_{st}} + \boldsymbol{T}$). Decide whether there exists an s-t path $P \subseteq E$ such that $(V(E - P),\ E - P)$ is a tree.

*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 grants K60802 and CK 80124.

**Department of Computer Science and CNL, Eötvös University, Budapest, Pázmány P. s. 1/C, Hungary H-1117. E-mail: kiraly@cs.elte.hu. Research is supported by EGRES group (MTA-ELTE) and OTKA grants CNK 77780, K 60802, CK 80124 and NK 67867.

Problem 6 (*P* + *T*). Decide whether there exists a path $P \subseteq E$ such that $(V(E-P),\ E-P)$ is a tree.

Problem 7 (*C* + *Conn*). Decide whether there exists a circuit $C \subseteq E$ such that $G-C=(V,\ E-C)$ is connected.

Problem 8 (*C* + *SpT*). Decide whether there exists a circuit $C \subseteq E$ such that $G-C=(V,\ E-C)$ is a spanning tree.

Problem 9 (*C* + *T*). Decide whether there exists a circuit $C \subseteq E$ such that $(V(E-C),\ E-C)$ is a tree.

Problems P_{st} + *Conn* and T + *SpT* were posed in the open problem portal called "EGRES Open"[1] of the Egerváry Research Group. Let us mention a related problem.

Problem 10 (*T* + *T*). Decide whether there exists an edge set $T \subseteq E$ such that both $(V(T),\ T)$ and $(V(E-T),\ E-T)$ are trees.

Problem $T+T$ was proved to be NP-complete in [3]; our reductions here are similar to the one used there.

Theorem 1. *Problems 1-9 are NP-complete even for graphs with maximum degree 4.*

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 [2]). Let φ be a 3-CNF formula with variable set $\{x_1, x_2, \dots, x_n\}$ and clause set $\mathcal{C}=\{C_1, C_2, \dots, C_m\}$. Assume that literal x_j appears in k_j clauses $C_{a_1^j}, C_{a_2^j}, \dots, C_{a_{k_j}^j}$, and literal $\overline{x_j}$ occurs in l_j clauses $C_{b_1^j}, C_{b_2^j}, \dots, C_{b_{l_j}^j}$. Construct the following graph $G_\varphi=(V,E)$. For any clause $C \in \mathcal{C}$ and any literal y occurring in C introduce a node $v(y,C)$. Furthermore, for every variable x_j introduce two new nodes z_1^j and z_2^j. For every variable x_j, let G_φ contain a circuit on the k_j+l_j+2 nodes $z_1^j, v(x_j, C_{a_1^j}), v(x_j, C_{a_2^j}), \dots, v(x_j, C_{a_{k_j}^j})$, $z_2^j, v(\overline{x_j}, C_{b_{l_j}^j}), v(\overline{x_j}, C_{b_{l_j-1}^j}), \dots, v(\overline{x_j}, C_{b_1^j})$ in this order. The nodes of this circuit will be called the **nodes associated to variable** x_j. Identify the nodes z_2^j and z_1^{j+1} for every $j=1,2,\dots,n-1$. Furthermore, add two nodes s and t and introduce two parallel copies of the edges sz_1^1 and $z_2^n t$ (see Figure 1).

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 2 more nodes $w(y,C)$ and $w'(y,C)$ (these nodes will be called the **nodes associated to clause** C). Introduce the edges $u_C w(y,C)$, $w(y,C)v(y,C)$ and $w(y,C)w'(y,C)$ for every clause C and every literal y in C (i.e., the nodes $w(y,C)$ will have degree 3 and the ones $w'(y,C)$ will have degree 1). The construction of the graph G_φ is finished. An illustration can be found in Figure 1. We mention that for most of the NP-completeness proofs below the construction could be simplified: for Problems 1, 2 and 5 we could contract the sets $\{s, z_1^1\}$ and $\{z_2^n, t\}$ (and delete the

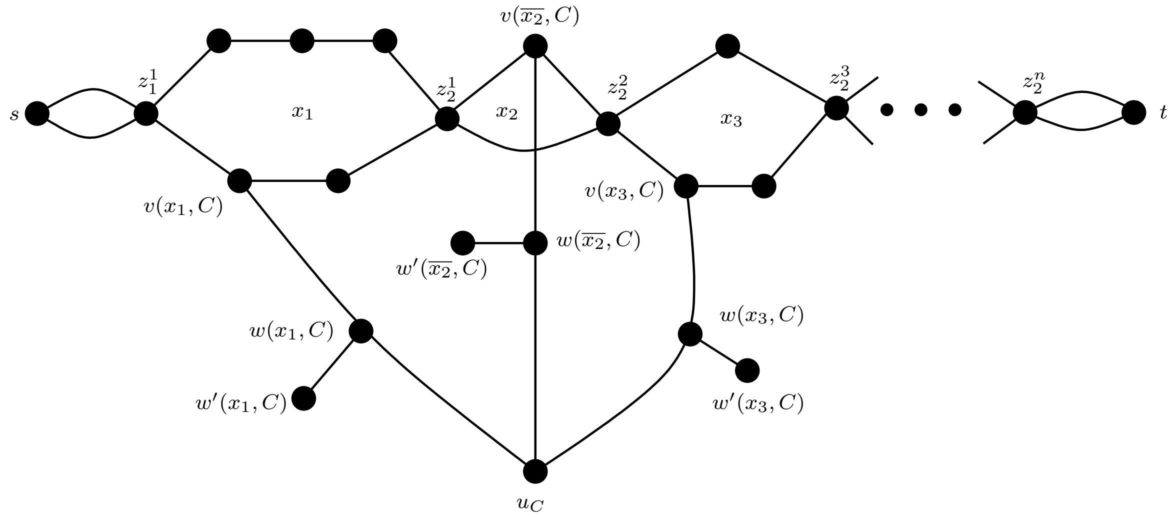

Figure 1: Part of the construction of graph G_φ for clause $C = \{x_1, \overline{x_2}, x_3\}$.

arising loops), and the nodes of form $w'(y, C)$ are only needed for Problems 5, 6 and 9, in every other case we could delete them. The reader can check the validity of these simplifications, we will not use them here.

Clearly, G_φ has maximum degree four. Note that the simplicity of G_φ can be recovered by subdividing one of the parallel edges sz_1^1 and $z_2^n t$.

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

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 any nodes associated to clauses. In other words, P is assignment-defining if and only if $V(P)$ contains exactly the following nodes: s, t, nodes $z_1^1, z_2^1, z_2^2, \dots, z_2^n$ and for every variable x_j it either contains every node $v(x_j, C_{a_1^j}), v(x_j, C_{a_2^j}), \dots,$ or it contains every node $v(\overline{x_j}, C_{b_1^j}), v(\overline{x_j}, C_{b_2^j}), \dots$. In this case let us define the truth assignment τ_P defined by P the following way: for every literal y, if $v(y, C) \in V(P)$ then let y be FALSE. The following claim will be frequently used later.

Claim 2. *If $P \subseteq E$ is an s-t path such that $(V(E - P),\ E - P)$ is connected, then P is an assignment-defining path.*

Proof. If P used the edge $v(y, C)w(y, C)$ for some clause C and literal $y \in C$ then it would also use the edge $w(y, C)u_C$, but then the edge $w(y, C)w'(y, C)$ would be a component of $(V(E - P),\ E - P)$. □

Claim 3. *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 1 is NP-complete.*

Proof. If τ is a truth assignment showing that $\varphi \in 3SAT$ then P_τ 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 is assignment-defining by Claim 2 and τ_P shows $\varphi \in 3SAT$. □

Claim 4. *The following statements are equivalent.*

(i) $\varphi \in$ ONE-IN-THREE 3SAT, *i.e., there is an evaluation of the variables of* φ *such that every clause of* φ *contains exactly one true literal,*

(ii) there is an s-t *path* $P \subseteq E$ *such that* $(V,\ E-P)$ *is a spanning tree,*

(iii) there is a path $P \subseteq E$ *such that* $(V,\ E-P)$ *is a spanning tree,*

(iv) there is an s-t *path* $P \subseteq E$ *such that* $(V(E-P),\ E-P)$ *is a tree,*

(v) there is a path $P \subseteq E$ *such that* $(V(E-P),\ E-P)$ *is a tree.*

Proof. First we show (i)⇒(ii); given a truth assignment τ to the variables $x_1, x_2, \dots, x_n$ as promised in (i), one can check that the path P_τ satisfies the requirements of (ii).

Obviously, (ii) ⇒(iii)⇒(v). Similarly, (ii) ⇒(iv)⇒(v). Therefore we only need to show (v)⇒(i). Assume that (v) holds. Since neither P, nor $E-P$ contains a circuit, P is in fact an s-t path. By Claim 2, P is assignment-defining, and the truth-assignment τ_P satisfies φ. As $E-P$ contains another assignment-defining path P' (that contains the TRUE literals), if τ_P gave two TRUE values for any clause C (two literals of C are on P') then $E-P$ would contain a circuit. Thus $\varphi \in$ ONE-IN-THREE 3SAT. □

Let $G' := (V, E')$ where $E' := E \cup \{st\}$. If C is a circuit in G' and $G'-C$ does not contain any circuit then C consists of the edge st and an s-t path P, because C must contain exactly one of the parallel edges between s and z_1^1 and exactly one of the parallel edges between z_2^n and t. Consequently, there is a circuit C in G' such that $G'-C$ is a spanning tree (or $(V(E'-C),\ E'-C)$ is a tree) if and only if there is an s-t path P in G such that $G-P$ is a spanning tree (or $(V(E-P),\ E-P)$ is a tree resp.). Therefore Problems 8 and 9 are also NP-complete.

To show the NP-completeness of Problems 4 and 7 we need one more trick. For the 3CNF formula φ with variables $x_1, x_2, \dots, x_n$ and clauses $C_1, C_2, \dots, C_m$, let us associate the formula φ' with the same variable set and clauses $\{x_1, \overline{x_1}\}, \{x_2, \overline{x_2}\}, \dots,$ $\{x_n, \overline{x_n}\}, C_1, C_1, C_2, C_2, \dots, C_m, C_m, \{x_1, \overline{x_1}\}, \{x_2, \overline{x_2}\}, \dots, \{x_n, \overline{x_n}\}$, i.e., every clause of φ is twice in φ' and we also introduced the clauses of form $\{x_j, \overline{x_j}\}$ also twice. Clearly, for any truth assignment τ, every clause in φ contains exactly one true literal if and only if every clause in φ' contains exactly one true literal. Construct the graph $G_{\varphi'} = (V, E)$ as above (where the order of the clauses does matter here: for every variable x_j the first and the last clause containing this variable is $\{x_j, \overline{x_j}\}$), and observe the following.

Claim 5. *There exists a truth assignment* τ *such that every clause in* φ' *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.*

Proof. If τ 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 one of the parallel edges sz_1^1, as well as one of the parallel pair $z_2^n t$. Since $(V(T), T)$ is connected, T contains a path $P \subseteq T$ between s and t. By Claim 2 and since $(V,\ E - P)$ is connected, this path is assignment defining. Consider the truth assignment τ_P associated to P, we claim that τ_P satisfies our requirements. Clearly, if a clause C of φ' does not contain a true literal then u_C is not reachable from s in $G_{\varphi'} - T$ (not even in $G_{\varphi'} - P$), therefore every clause of φ' contains at least one true literal. In order to show that none can contain more than one, we claim that all the edges of $T - P$ are incident with a node z_b^j. As T^* is spanning, it contains at least one edge incident with every node of P, so edges of $T - P$ can be incident with only the degree four vertices of P, i.e., with vertices z_b^j. Such an edge is e.g. $z_2^i v(x_i, \{x_i, \overline{x_i}\})$. To prove the claim, it is enough to show that $v(x_i, \{x_i, \overline{x_i}\})$ is a leave of T. As observed, node $v(\overline{x_i}, \{x_i, \overline{x_i}\})$ is a leave of T^*, consequently, as T^* is a spanning tree, edges $w(x_i, \{x_i, \overline{x_i}\})v(x_i, \{x_i, \overline{x_i}\})$ and $v(x_i, \{x_i, \overline{x_i}\})v(x_i, C_{a^i_{k_i}})$ also must belong to T^*.

If a clause C would contain two true literals in τ_P then $E - T$ would contain a circuit through the nodes associated to the two copies of clause C, a contradiction. □

It remained to prove the NP-completeness of Problem 7. We redefine $G_{\varphi'} = (V, E)$: contract s to z_1^1 and t to z_2^n, and delete the arising loops.

Claim 6. *There exists a truth assignment τ such that every clause in φ' contains exactly one true literal if and only if there exists a set $C \subseteq E$ such that $(V(C), C)$ is a circuit and $G - C = (V,\ E - C)$ is connected.*

Proof. First observe that due to the facts that $G - C$ is connected and contains node $w(y, C)$ for every clause C and literal $y \in C$, circuit C cannot use any node associated to a clause. We claim that $V(C)$ neither can be the node set associated to a variable x_i, because in this case the nodes associated to clause $\{x_i, \overline{x_i}\}$ would not be reachable in $G - C$ from s. Therefore C consist of edge st and an assignment defining path P. It is easy to check that – as in the previous claim – τ_P is a truth assignment giving exactly one TRUE value to each clause. On the other hand, as before, if τ is a truth assignment as defined in the claim then one can see that $C = P_\tau \cup \{st\}$ is an edge set satisfying the requirements. □

As we proved the NP-completeness of all nine problems, we are done. □

References

[1] 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”.

[2] 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.

[3] Dömötör Pálvölgyi, *Partitionability to two trees is NP-complete*, 2006, Egres Quick Proofs QP-2006-06, www.cs.elte.hu/egres.