Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 29 (2007) 207–211
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

Detecting induced subgraphs

Benjamin Lévêque [1]

Laboratoire G-SCOP
Grenoble, France

David Y. Lin [2]

Princeton University
Princeton, New Jersey

Frédéric Maffray [3]

Laboratoire G-SCOP
Grenoble, France

Nicolas Trotignon [4]

Centre d'Economie de la Sorbonne
Paris, France

Abstract

An *s-graph* is a graph with two kind of edges: *subdivisible* edges and *real* edges. A *realisation* of an s-graph B is any graph obtained by subdividing subdivisible edges of B into paths of length at least one. Given an s-graph B, we study the decision problem Π_B. Its instance is any graph G, its question is "Does G contains a realisation of B as an induced subgraph ?". For several B's, the complexity is known and here we give the complexity for several more. We also provide results on the problem of detecting an induced cycle through two prescribed vertices.

Keywords: detecting, induced, subgraphs.

doi:10.1016/j.endm.2007.07.036

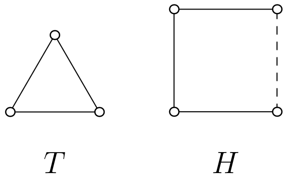

Fig. 1. S-graphs yielding trivially polynomial problems

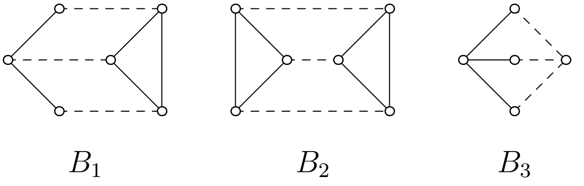

Fig. 2. Pyramids, prisms and thetas

1 Introduction

In this paper graphs are simple and finite. A *subdivisible graph* (*s-graph* for short) is a triple $B = (V, D, F)$ such that $B' = (V, D \cup F)$ is a graph and $D \cap F = \emptyset$. The edges in D are said to be *real edges of* B while the edges in F are said to be *subdivisible edges of* B. A *realisation* of B is a graph obtained from B by subdividing edges of F into paths of length at least one. The problem Π_B is the decision problem whose input is a graph G and whose question is "Does G contain a realisation of B as an induced subgraph ?". On figures, we depict real edges of an s-graph with straight lines, and subdivisible edges with dashed lines.

Several Π_B problems of interest are studied in the litterature. For some of them, the existence of a polynomial time algorithm is trivial, but efforts are devoted toward optimized algorithms. For example, Alon, Yuster and Zwick solve Π_T in time $O(m^{1.41})$ (instead of the obvious $O(n^3)$ algorithm), where T is the s-graph depicted on Figure 1. This problem is known as *triangle detection*. Tarjan and Yannakakis [9] solve P_H in time $O(n + m)$ where H is the s-graph depicted on Figure 1.

[1] Email: benjamin.leveque@g-scop.inpg.fr
[2] Email: dylin@Princeton.EDU
[3] Email: Frederic.Maffray@g-scop.inpg.fr
[4] Email: nicolas.trotignon@univ-paris1.fr

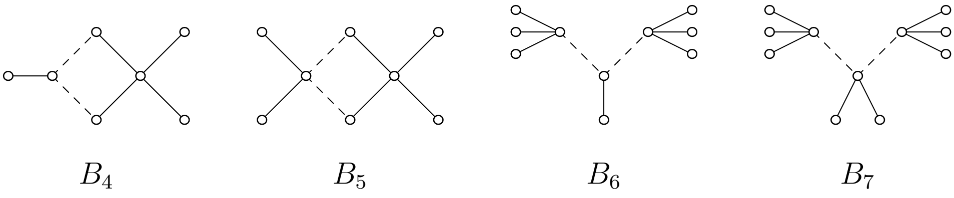

Fig. 3. Some s-graphs with pending edges

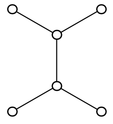

Fig. 4. I_1

But for some Π_B's, the existence of a polynomial time algorithm is non-trivial. A *pyramid* (resp. *prism*, *theta*) is any graph that is a realisation of the s-graph B_1 (resp. B_2, B_3) depicted on figure 2. Chudnovsky and Seymour [5] gave an $O(n^9)$-time algorithm for Π_{B_1} (or equivalently, for detecting a pyramid). As far as we know, that is the first example of a solution to a Π_B whose complexity is non-trivial to settle. In contrast, Maffray and Trotignon [7] proved that Π_{B_2} (or detecting a prism) is NP-complete. Chudnovsky and Seymour [4] gave an $O(n^{11})$-time algorithm for P_{B_3} (or detecting a theta). Their algorithm relies on the solution of a problem called "three-in-a-tree". Note that the algorithm for three-in-tree is quite general since it can be used to solve a lot of Π_B problems, including the detection of pyramids.

These facts are a motivation for a systematic study of Π_B. A further motivation is that very similar s-graphs can lead to a drasticly different complexity. The following example is maybe more striking than pyramid/prism/theta : Π_{B_4}, Π_{B_6} are polynomial and Π_{B_5}, Π_{B_7} are NP-complete, where $B_4, \dots, B_7$ are the s-graphs depicted on figure 3.

Notation

By C_k ($k \geq 3$) we denote the cycle on k vertices, by K_l ($l \geq 1$) the clique on l vertices. By I_l ($l \geq 1$) we denote the tree on $l+5$ vertices that we obtain by taking a path of length l with end a, b, and by adding four vertices, two of them adjacent to a, the two others two b, see Figure 4. When a graph G contains a graph isomorphic to H as an induced subgraph, we will often say "G contain an H".

2 Detection of holes with prescribed vertices

Let $\Delta(G)$ be the maximum degree of G. Let $\mathcal{I}$ be a set of graphs and k be an integer. Let $\Gamma^k_{\mathcal{I}}$ be the problem whose instance is (G, x, y) where G is a graph such that $\Delta(G) \leq k$, with no induced subgraph in $\mathcal{I}$ and $x, y \in V(G)$ are two non-adjacent vertices of degree 2. The question is "Does G contain a hole passing through x, y ?". For simplicity, we write $\Gamma_{\mathcal{I}}$ instead of $\Gamma^{+\infty}_{\mathcal{I}}$ (so, the graph in the instance of $\Gamma_{\mathcal{I}}$ has unbounded degree). Also we write Γ^k instead of $\Gamma^k_{\emptyset}$ (so the graph in the instance of Γ^k has no restriction on its induced subgraphs). Bienstock [3] proved that $\Gamma = \Gamma_{\emptyset}$ is NP-complete. For $I = \{K_3\}$ and $I = \{K_{1,4}\}$, $\Gamma_{\mathcal{I}}$ can be shown to be NP-complete, and a consequence is the NP-completeness of several problems of interest: see [7] and [8].

We try to settle $\Gamma^k_{\mathcal{I}}$ for as many $\mathcal{I}$'s and k's as we can because we need this in the proofs of the results in the next section. In particular, we give the complexity of $\Gamma_{\mathcal{I}}$ when $\mathcal{I}$ contains only one connected graph and of Γ^k for all k. We also settle $\Gamma^k_{\mathcal{I}}$ for some cases when I is a set of cycles. The polynomial cases are either trivial, or are a direct consequence of the algorithm three-in-a-tree of Chudnovsky and Seymour that we have already mentionned. The NP-complete cases follow from several extensions of Bienstock's construction.

Theorem 2.1 *Let H be a connected graph. Then either :*

- *H is a path or a subdivision of a claw and $\Gamma_{\{H\}}$ is polynomial.*
- *H contains one of $K_{1,4}$, I_k for some $k \geq 1$, or C_l for some $l \geq 3$ as an induced subgraph and $\Gamma_{\{H\}}$ is NP-complete.*

Interestingly, a similar theorem has been proved by Alekseev:

Theorem 2.2 (Alekseev, [1]) *Let H be a connected graph that is not a path nor a subdivided claw. Then the problem of finding a maximum stable set in H-free graphs is NP-hard.*

But the complexity of the maximum stable set problem is not known in general for H-free graphs when H is a path or a subdivided claw. See [6] for a survey.

Theorem 2.3 *The following statements hold.*

- *For any $k \in \mathbb{Z}$ with $k \geq 2$, the problem Γ^k is NP-complete when $k \geq 3$ and polynomial when $k = 2$.*
- *If $\mathcal{H}$ is any finite list of cycles $C_{k_1}, C_{k_2}, \ldots, C_{k_m}$ such that $C_6 \notin \mathcal{H}$, then $\Gamma^3_{\mathcal{H}}$ is NP-complete.*

3 Π_B for some special s-graphs

The s-graphs $B_4, \ldots, B_7$ are depicted on Figure 3.

Theorem 3.1 *There is an $O(n^{13})$-time algorithm for Π_{B_4}, an $O(n^{14})$-time algorithm for Π_{B_6}, but Π_{B_5}, Π_{B_7} are NP-complete.*

We put : $sK_5 = (\{a,b,c,d,e\}, \emptyset, \binom{\{a,b,c,d,e\}}{2})$. So sK_5 is the s-graph on five vertices with all its edges subdivisible. The following theorem is the only NP-hardness result known for an s-graph with no real edges.

Theorem 3.2 *Π_{sK_5} is NP-complete.*

References

[1] V.E. Alekseev. On the local restrictions effect on the complexity of finding the graph independence number. *Combinatorial-algebraic methods in applied mathematics*, 132:3–13, 1983. Gorky University Press, Gorky, in Russian.

[2] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. In *Proceedings of the 2nd European Symposium on Algorithms. Utrecht, The Netherlands*, pages 354–364, 1994.

[3] D. Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Math.*, 90:85–92, 1991. See also Corrigendum by B. Reed, *Discrete Math.*, 102, (1992), p. 109.

[4] M. Chudnovsky and P. Seymour. The three-in-a-tree problem. Manuscript.

[5] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković. Recognizing Berge graphs. *Combinatorica*, 25:143–186, 2005.

[6] A. Hertz and V. V. Lozin. The maximum independent set problem and augmenting graphs. Manuscript.

[7] F. Maffray and N. Trotignon. Algorithms for perfectly contractile graphs. *SIAM Jour. of Discrete Math.*, 19(3):553–574, 2005.

[8] F. Maffray, N. Trotignon, and K. Vušković. Algorithms for square-$3PC(\cdot,\cdot)$-free Berge graphs. *SIAM Journal on Discrete Mathematics*. To appear.

[9] R.E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13:566–579, 1984.