Theoretical Computer Science 562 (2015) 283–303

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Finding a subdivision of a digraph

Jørgen Bang-Jensen [a,1], Frédéric Havet [b,2], A. Karolinna Maia [b,2]

[a] Department of Mathematics and Computer Science, University of Southern Denmark, Odense DK-5230, Denmark
[b] Projet Coati, I3S (CNRS, UNSA) and INRIA, Sophia Antipolis, France

ARTICLE INFO

Article history:
Received 13 July 2012
Received in revised form 30 July 2014
Accepted 1 October 2014
Available online 7 October 2014
Communicated by G. Ausiello

Keywords:
Oriented graphs
Subdivision
Linkage problem

ABSTRACT

We consider the following problem for oriented graphs and digraphs: Given a directed graph D, does it contain a subdivision of a prescribed digraph F? We give a number of examples of polynomial instances, several NP-completeness proofs as well as a number of conjectures and open problems.

1. Introduction

Many interesting classes of graphs are defined by forbidding induced subgraphs, see [7] for a survey. This is why the detection of several kinds of induced subgraphs is interesting, see [16] where several such problems are surveyed. In particular, the problem of deciding whether a graph G contains, as an induced subgraph, some graph obtained after possibly subdividing prescribed edges of a prescribed graph H has been studied. This problem can be polynomial-time solvable or NP-complete according to H and to the set of edges that can be subdivided. The aim of the present work is to investigate various similar problems in digraphs, focusing only on the following problem: given a digraph H, is there a polynomial-time algorithm to decide whether an input digraph G contains a subdivision of H?

Of course the answer depends heavily on what we mean by "contain". Let us illustrate this by surveying what happens in the realm of undirected graphs. If the containment relation is the subgraph containment, then for any fixed H, detecting a subdivision of H in an input graph G can be performed in polynomial time by the Robertson and Seymour linkage algorithm [20] (for a short explanation of this see e.g. [3]). But, if we want to detect an *induced* subdivision of H, then the answer depends on H (assuming P $\neq$ NP). It is proved in [16] that detecting an induced subdivision of K_5 is NP-complete, and the argument can be reproduced for any H whose minimum degree is at least 4. Polynomial-time solvable instances trivially exist, such as detecting an induced subdivision of H when H is a path, or a graph on at most 3 vertices. But non-trivial polynomial-time solvable instances also exist, such as detecting an induced subdivision of $K_{2,3}$ which can be performed in $O(n^{11})$ time by Chudnovsky and Seymour's three-in-a-tree algorithm, see [8]. Note that for many graphs H, nothing is known about the complexity of detecting an induced subdivision of H: when H is cubic (in particular when $H = K_4$) or when H is a disjoint union of two triangles, and in many other cases.

E-mail addresses: jbj@imada.sdu.dk (J. Bang-Jensen), frederic.havet@inria.fr (F. Havet), karol.maia@inria.fr (A. Karolinna Maia).

[1] Most of this work was done while J. Bang-Jensen visited Projet Coati, I3S (CNRS, UNSA) and INRIA, Sophia Antipolis whose hospitality and financial support is gratefully acknowledged.
[2] Partly supported by ANR Blanc STINT and CAPES/Brazil.

http://dx.doi.org/10.1016/j.tcs.2014.10.004
0304-3975/

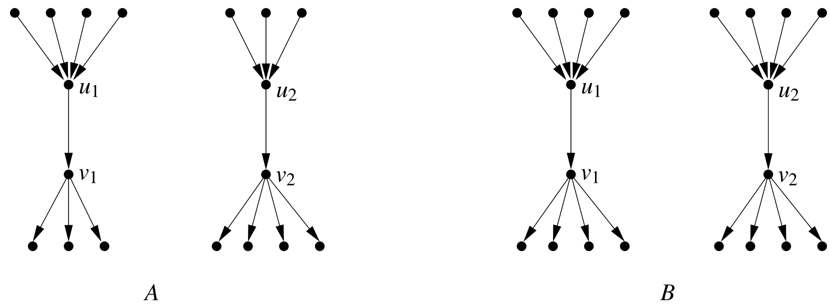

Fig. 1. Digraphs A and B such that A is a subdigraph of B, A-Subdivision is NP-complete, and B-Subdivision is polynomial-time solvable.

When we move to digraphs, the situation becomes more complicated, even for the subdigraph containment relation. In this paper, by digraph we mean a simple digraph, that is a digraph with no parallel arcs nor loops. Sometimes however, multiple arcs are possible. In such cases, we write multidigraph. We rely on [1] for classical notation and concepts. A few things need to be stated here though. Unless otherwise stated the letters n and m will always denote the number of vertices and arcs (edges) of the input digraph (graph) of the problem in question. By *linear time*, we mean $O(n+m)$ time. If D is a digraph, then we denote by $UG(D)$ the underlying (multi)graph of D, that is, the (multi)graph we obtain by replacing each arc by an edge. A digraph D is *connected* if $UG(D)$ is a connected graph. If xy is an arc from x to y, then we say that x *dominates* y. When H, H' are digraphs we denote by $H+H'$ the disjoint union of H and H' (no arcs between disjoint copies of these).

A *subdivision of a digraph* F, also called an F-*subdivision*, is a digraph obtained from F by replacing each arc ab of F by a directed (a,b)-path.

In this paper, we consider the following problem for a fixed digraph F.

F-Subdivision
Input: A digraph D.
Question: Does D contain a subdivision of F as a subgraph?

In [2] the problem Induced-F-Subdivision of finding an induced subdivision of a prescribed digraph F in a given digraph D was studied. It turns out that here there is a big difference in the complexity of the problem depending on whether or not D is an oriented graph or it may contain 2-cycles. In the latter case Induced-F-Subdivision is NP-complete for every oriented digraph F which is not the disjoint union of spiders (see definition of these digraphs below) and also in [2] it was conjectured that Induced-F-Subdivision is NP-complete unless F is the disjoint union of spiders and at most one 2-cycle.

Let $x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_k$ be distinct vertices of a digraph D. A k-*linkage* from $(x_1, x_2, \dots, x_k)$ to $(y_1, y_2, \dots, y_k)$ in D is a system of disjoint directed paths $P_1, P_2, \dots, P_k$ such that P_i is an (x_i, y_i)-path in D.

Similarly to the situation for undirected graphs, the D-Subdivision problem is related to the following k-Linkage problem.

k-Linkage
Input: A digraph D and $2k$ distinct vertices $x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_k$.
Question: Is there a k-linkage from $(x_1, x_2, \dots, x_k)$ to $(y_1, y_2, \dots, y_k)$ in D?

However, contrary to graphs, unless P = NP, k-Linkage cannot be solved in polynomial time in general digraphs. Fortune, Hopcroft and Wyllie [10] showed that already 2-Linkage is NP-complete. Using this result, we show that for lots of F, the F-Subdivision problem is NP-complete. We also give some digraphs F for which we prove that F-Subdivision is polynomial-time solvable. We believe that there is a dichotomy between NP-complete and polynomial-time solvable instances.

Conjecture 1. *For every digraph F, the F-Subdivision problem is polynomial-time solvable or NP-complete.*

To prove such a conjecture, a first idea would be to try to establish for any digraph G and subdigraph F, that if F-Subdivision is NP-complete, then G-Subdivision is also NP-complete, and conversely, if G-Subdivision is polynomial-time solvable, then F-Subdivision is polynomial-time solvable. However, these two statements are false as shown by the two digraphs depicted in Fig. 1. The NP-completeness of A-Subdivision follows Theorem 12. The fact that B-Subdivision is polynomial-time solvable is proved in Theorem 27.

The paper is organized as follows. We start by giving some general lemmas which allow to extend NP-completeness results of F-Subdivision for some digraphs F to much larger classes of digraphs. Next we give a powerful tool, based on a reduction from the NP-complete 2-linkage problem in digraphs, which can be applied to conclude the NP-completeness of