Discrete Applied Mathematics 168 (2014) 88–107

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Digraph width measures in parameterized algorithmics

Robert Ganian [a,1], Petr Hliněný [a,*], Joachim Kneis [b], Alexander Langer [b], Jan Obdržálek [a], Peter Rossmanith [b]

[a] Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic

[b] Theoretical Computer Science, RWTH Aachen, Ahornstraße 55, 52056 Aachen, Germany

ARTICLE INFO

Article history:
Received 15 March 2012
Received in revised form 25 October 2013
Accepted 30 October 2013
Available online 23 November 2013

Keywords:
Digraph
Parameterized complexity
Tree-width
DAG-width
DAG-depth
Clique-width

ABSTRACT

In contrast to undirected width measures such as tree-width, which have provided many important algorithmic applications, analogous measures for digraphs such as directed tree-width or DAG-width do not seem so successful. Several recent papers have given some evidence on the negative side. We confirm and consolidate this overall picture by thoroughly and exhaustively studying the complexity of a range of directed problems with respect to various parameters, and by showing that they often remain NP-hard even on graph classes that are restricted very beyond having small DAG-width. On the positive side, it turns out that clique-width (of digraphs) performs much better on virtually all considered problems, from the parameterized complexity point of view.

1. Introduction

The very successful concept of graph *tree-width* was introduced in the context of the Graph Minors project by Robertson and Seymour [55,56], and it turned out to be very useful for efficiently solving many graph problems (including NP-hard ones). In a nutshell, tree-width measures tree-likeness of a graph. Trees themselves, for example, have tree-width one and series–parallel graphs have tree-width two. Many graphs occurring in practical applications have small tree-width. This comes as no big surprise as one often deals with hierarchical structures that are inherently similar to trees. Examples include problems in VLSI design, evolution theory, interval routing, and the control-flow graphs of structured programs. See [6–8,38] for surveys.

Tree-width is a property of *undirected graphs*. In this paper we will be interested in *digraphs* (directed graphs). Naturally, a width measure specifically tailored to digraphs with all the nice properties of tree-width would be tremendously useful. We can, of course, apply the concept of tree-width to digraphs, too, if we just forget the direction of all edges for the computation of the tree-width and the resulting *tree decomposition*. With such an approach we, however, seem to ignore too much: for example, in directed acyclic graphs (DAGs) it is easy to find a longest path while the problem is NP-complete in general. Nevertheless, DAGs have unbounded tree-width if we forget the directions.

On the search for a "truly directed" width measure inspired by tree-width, several suggestions were made, starting with directed tree-width [41], and being complemented recently with several new approaches including directed path-width [3], entanglement [5], D-width [57], DAG-width [4] and Kelly-width [39].

* Corresponding author. Tel.: +420 721320618.
E-mail addresses: rganian@gmail.com (R. Ganian), hlineny@fi.muni.cz (P. Hliněný), kneis@cs.rwth-aachen.de (J. Kneis), langer@cs.rwth-aachen.de (A. Langer), obdrzalek@fi.muni.cz (J. Obdržálek), rossmani@cs.rwth-aachen.de (P. Rossmanith).

1 Currently at Vienna University of Technology, Austria.

0166-218X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dam.2013.10.038

Some positive results were encouraging: the Hamiltonian path problem can be solved in polynomial time if the directed tree width, the DAG-width, or the Kelly-width are bounded by a constant [41]. More recently, it has been shown that parity games (Section 4.9) can be solved in polynomial time on digraphs of bounded entanglement [5], DAG-width [4] or Kelly-width [39].

Unfortunately, as encouraging as the first positive results are, there is also the negative side. For undirected graphs, the existence of a Hamiltonian path can be tested in *linear* time if the tree-width is bounded by a constant; only the constant hidden in the "big-O" increases with the tree-width. So this problem is *fixed-parameter tractable* (captured in the complexity class FPT, Section 2.3) for the parameter tree-width. While Hamiltonian path on digraphs is indeed solvable in polynomial time for bounded DAG-width, the degree of the polynomial in the running time increases with the DAG-width (in the complexity class XP, Section 2.3). This likely cannot be improved (unless the Exponential-time hypothesis fails) since Lampis, Kaouri, and Mitsou showed that Hamiltonian path is W[2]-hard for the parameter DAG-width [48].

Even worse, many other natural problems remain NP-hard on digraphs of low widths [14,15,46,48] and some of them are already NP-complete on DAGs—such as MaxDiCut [48] or oriented colouring [14]. This particularly implies that for DAG-width there cannot be a result similar to famous Courcelle's MSO_2 theorem. Therefore, one should perhaps look for other new directed measures providing a "finer resolution" on DAGs.

We will add many more natural directed problems to the list, but will go even further: one of the main goals of this paper is to show that not only many problems are hard on DAGs, but rather that they remain hard even if we very severely further restrict the digraphs' structure. To this end, we introduce *two new digraph measures*; K-width (Section 3.4) and DAG-depth (Section 3.3), with the intention to complete the full (and rather *negative*) picture of structural digraph width parameters with some more restrictive ones.

On the other hand, one width measure that fares much better is *clique-width* [13], an algebraic width measure that equally handles graphs and digraphs (and related bi-rank-width generalizing the rank-width of undirected graphs [42]). Nearly all of our problems are fixed-parameter tractable or at least in XP with respect to this parameter. Even better, unlike as for DAG-width or Kelly-width, finding an optimal bi-rank-decomposition is known to be in FPT [37,42].

2. Preliminaries

2.1. Digraphs

We assume that the readers are familiar with standard terms of undirected graphs, for example in Diestel [16].

A *directed graph* (or *digraph*) is a pair (V, E) of disjoint sets of *vertices* and *arcs*, together with two mappings *tail* : $E \to V$ and *head* : $E \to V$ assigning to every arc e its *starting vertex* $x = tail(e)$ and *terminal vertex* $y = head(e)$ (e is said to be *directed* from x to y). Note that a digraph may have several arcs between the same two vertices x, y. If two of them have the same direction (say from x to y), they are called *parallel*. If $x = y$, then e is called a *loop*. We will sometimes refer an arc from x to y as to (x, y).

A *directed path* is a digraph of the form: $V = \{x_0, x_1, x_2, \dots, x_k\}$, $E = \{(x_0, x_1), (x_1, x_2), \dots, (x_{k-1}, x_k)\}$ where x_i are all distinct; intuitively, x_0 and x_k are called the endpoints of the directed path. The length of a directed path is defined as the number of its arcs. Two directed paths P_1 and P_2 are *internally disjoint* if they are vertex disjoint except for their endpoints. A *directed cycle* is a directed path with an additional arc (x_k, x_0). A *directed acyclic graph (DAG)* is a digraph with no directed cycles.

Many terms of undirected graphs are naturally extended to digraphs, like those of subgraph/*subdigraph* and of *isomorphism*. Given two vertices x, y, we say that y is an *out-neighbour* (*in-neighbour*) of x if there exists an arc (x, y) ((y, x), respectively). We say that y is reachable from x if there exists a directed path from x to y. A digraph G is *strongly connected* if each of its vertices is reachable from any other one. *Strong components* of G are the equivalence classes defined by the relation $x \sim y$ meaning that x is reachable from y and y is reachable from x.

2.2. SAT and its variants

We define the Boolean Satisfiability problem (abbreviated as SAT). A *literal* is a positive propositional variable x or a negative variable $\neg x$. A *clause* is a finite set of literals, e.g., $C = x_1 \vee x_2 \vee \neg x_3$. A propositional formula ϕ in conjunctive normal form, or CNF formula for short, is a set of clauses $\phi = C_1 \wedge \cdots \wedge C_p$. A CNF formula is a c-CNF formula if each clause contains at most c literals. Let $\mathrm{var}(\phi)$ denote the set of variables of ϕ.

A CNF formula ϕ is *satisfiable* if there is a truth assignment of $\mathrm{var}(\phi)$ for which ϕ evaluates to true, otherwise F is *unsatisfiable*. SAT is the NP-complete problem of deciding whether a given CNF formula is satisfiable [11,49]. Analogously, c-SAT is the problem of deciding whether a given c-CNF formula is satisfiable. Based on the value of c we have the following distinction:

Theorem 2.1 (*[47]*)**.** *2-SAT can be solved in polynomial time.*

Theorem 2.2 (*[30], Folklore*)**.** *The 3-SAT problem remains NP-complete even if the input CNF formula satisfies all the following conditions*