Theoretical Computer Science 562 (2015) 365–376

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Approximation algorithms for digraph width parameters ☆

Shiva Kintali [a], Nishad Kothari [b,*], Akash Kumar [c]

[a] Department of Computer Science, Princeton University, NJ, USA
[b] Department of Combinatorics and Optimization, University of Waterloo, ON, Canada
[c] Department of Computer Science, Purdue University, IN, USA

ARTICLE INFO

Article history:
Received 24 December 2012
Received in revised form 18 May 2014
Accepted 1 October 2014
Available online 13 October 2014
Communicated by G. Ausiello

Keywords:
Approximation algorithms
Arboreal decomposition
Directed treewidth
DAG-decomposition
DAG-width
Directed path decomposition
Directed pathwidth
Kelly decomposition
Kelly-width
Directed vertex separators

ABSTRACT

Several problems that are NP-hard on general graphs are efficiently solvable on graphs with bounded treewidth. Efforts have been made to generalize treewidth and the related notion of pathwidth to digraphs. Directed treewidth, DAG-width and Kelly-width are some such notions which generalize treewidth, whereas directed pathwidth generalizes pathwidth. Each of these digraph width measures have an associated decomposition structure.

In this paper, we present approximation algorithms for all these digraph width parameters. In particular, we give an $O(\sqrt{\log n})$-approximation algorithm for directed treewidth, and an $O(\log^{3/2} n)$-approximation algorithm for directed pathwidth, DAG-width and Kelly-width. Our algorithms construct the corresponding decompositions whose widths agree with the above mentioned approximation factors.

1. Introduction

The related notions of tree decompositions and path decompositions have been studied extensively by Robertson and Seymour in their seminal work on graph minors. These decompositions correspond to associated width measures for undirected graphs called treewidth and pathwidth, respectively. Besides playing a crucial role in structural graph theory, these width measures also proved to be very useful in the design of algorithms. Roughly speaking, treewidth of an undirected graph measures how close the graph is to being a tree. On the other hand, pathwidth measures how close the graph is to being a path. Several problems that are NP-hard on general graphs are solvable in polynomial time on graphs of bounded treewidth using dynamic programming techniques. These include classical problems such as Hamiltonian cycle, graph coloring, vertex cover, graph isomorphism and many more. We refer the reader to [19,10] and references therein for an introduction to treewidth.

One attempt at solving algorithmic problems on digraphs would be to consider the treewidth of the underlying undirected graph. However, this approach suffers from certain drawbacks if the problem being considered depends on the directions of the arcs. For instance, it is possible to orient the edges of a complete graph in order to obtain a directed

☆ This work was done when the authors were at Georgia Institute of Technology, Atlanta, GA, USA.

* Corresponding author.
E-mail addresses: kintali@cs.princeton.edu (S. Kintali), nkothari@uwaterloo.ca (N. Kothari), akumar@purdue.edu (A. Kumar).

http://dx.doi.org/10.1016/j.tcs.2014.10.009
0304-3975/

acyclic graph (DAG). Although the (undirected) treewidth of such a digraph is large, it is easy to solve the Hamiltonian cycle problem on such a digraph. Thus, it would be desirable to have a width measure for digraphs which would attain much lower values on such digraphs than the value of the (undirected) treewidth. Hence, efforts have been made to generalize treewidth and pathwidth to digraphs. Directed treewidth, DAG-width and Kelly-width are some such notions which generalize treewidth, whereas directed pathwidth generalizes pathwidth. Each of these digraph width measures have an associated decomposition structure as well.

Johnson et al. [18] introduced the first directed analogue of treewidth called directed treewidth. They demonstrated the algorithmic benefits of directed treewidth by providing efficient algorithms for NP-hard problems (such as Hamiltonian cycle) on digraphs of bounded directed treewidth. Reed [26] defined another directed analogue which is closely related to the one introduced by Johnson et al. Later on, Berwanger et al. [7] and independently Obdržálek [24] introduced DAG-width. They demonstrated the usefulness of DAG-width by showing that the winner of a parity game can be decided in polynomial time on digraphs of bounded DAG-width. Parity games are a certain form of combinatorial game played on digraphs. They also give an equivalent characterization of DAG-width in terms of a certain variant of the cops-and-robber game in which the robber is visible and dynamic. More recently, Hunter and Kreutzer [17] introduced Kelly-width. They presented several equivalent characterizations of Kelly-width such as elimination ordering, partial k-DAGs and another variant of the cops-and-robber game in which the robber is invisible and inert. We refer the reader to Appendix A for a discussion of cops-and-robber games.

All of the above mentioned width measures are generalizations of undirected treewidth. More precisely, for a graph G with treewidth k, let $\overline{G}$ be the *digraph* obtained from G by replacing each edge $\{u, v\}$ of G by two arcs (u, v) and (v, u), then: (i) the directed treewidth of $\overline{G}$ is equal to k [18, Theorem 2.1], (ii) the DAG-width of $\overline{G}$ is equal to $k+1$ [7, Proposition 5.2], and, (iii) the Kelly-width of $\overline{G}$ is equal to $k+1$ [17]. Similarly, directed pathwidth introduced by Reed, Seymour and Thomas is a generalization of undirected pathwidth [5, Lemma 1]. Computing the *treewidth* (or *pathwidth*) of an undirected graph is NP-complete [2]. Moreover, Bodlaender et al. [8, Theorem 23] showed that unless P = NP, neither treewidth nor pathwidth can be approximated within an additive constant or term of the form n^{ϵ} for $\epsilon < 1$ of optimal. It follows that computing any of these digraph width parameters is also NP-complete, and futhermore a similar approximation hardness applies.

All the algorithms proposed for approximating treewidth rely on the relation between treewidth and balanced vertex separators (which we discuss in more detail shortly). In their seminal work, Leighton and Rao [23] gave an $O(\log n)$-pseudo approximation algorithm for computing balanced vertex separators. Bodlaender et al. [8] gave an $O(\log n)$-approximation algorithm for computing treewidth. Their algorithm made use of the small vertex separators obtained using the results of [23]. Moreover, their techniques imply that any ρ-approximation algorithm for balanced vertex separators can be used to obtain a ρ-approximation algorithm for treewidth. Now, let k denote the treewidth of a graph. Bouchitté et al. [9] gave an $O(\log k)$-approximation algorithm for treewidth using different techniques. Independently, Amir [3] gave another approximation algorithm with the same guarantee for treewidth, and this again relies on the algorithms of [23].

The approximation algorithm for balanced vertex separators was improved to $O(\sqrt{\log k})$ by Feige, Hajiaghayi and Lee [13]. As per the above discussion and as noted by Feige et al. [13], this gives an $O(\sqrt{\log k})$-approximation algorithm for treewidth. Kloks [19] described a procedure to transform a tree decomposition to a path decomposition whose width is at most $\log n$ times the width of the original tree decomposition. It follows that the result of Feige et al. [13] implies an $O(\sqrt{\log k} \cdot \log n)$-approximation algorithm for pathwidth.

To the best of our knowledge, no (non-trivial) approximation algorithms are known for any of the above mentioned digraph width parameters. We take a step in this direction. Our algorithms are similar to the above mentioned approximation algorithms for treewidth in the sense that they rely on the approximation algorithms for balanced directed vertex separators (see Definition 13). Leighton and Rao [23] observed that their algorithm can be extended to work on directed graphs as well. This leads to an $O(\log n)$-approximation algorithm for balanced directed vertex separators using the algorithm for directed edge separators as a black box. This was further improved to $O(\sqrt{\log n})$ by Agarwal et al. [1]. Our algorithms make use of their approximation algorithm as a subroutine.

1.1. Results and techniques

- We obtain an $O(\log^{3/2} n)$-approximation algorithm for directed pathwidth. This algorithm uses ideas similar to those of Bodlaender et al. [8] for approximating treewidth and pathwidth, which in turn builds on techniques developed by Lagergren [21] and Reed [25]. Let G be an undirected graph. Informally speaking, a balanced vertex separator is a set of vertices $S \subseteq V(G)$ such that $V(G) - S$ can be divided into two parts of roughly the same size. Their algorithm at a high level uses a divide-and-conquer approach to compute approximate path decompositions of the graphs induced by these two parts, and then uses these to construct an approximate path decomposition of G. We refer the reader to [19, Section 6.1] for a detailed description of this algorithm. The approximation guarantee of their algorithm crucially depends on the fact that every graph of treewidth k has a balanced vertex separator of size at most $k+1$. We first establish analogous relations between balanced directed vertex separators (see Definition 13) and all of the relevant digraph width parameters, and then use a similar divide-and-conquer approach to compute an approximate directed path decomposition.