EuroComb 2005 DMTCS proc. **AE**, 2005, 67–72

Directed One-Trees[†]

William Evans and Mohammad Ali Safari

Dept. of Computer Science, University of British Columbia, Vancouver, BC, Canada. $\{$*will,safari*$\}$*@cs.ubc.ca*

We identify the class of directed one-trees and prove the so-called min-max theorem for them. As a consequence, we establish the equality of directed tree-width and a new measure, d-width, on this class of graphs. In addition, we prove a property of all directed one-trees and use this property to create an $O(n^2)$ recognition algorithm and an $O(n^2)$ algorithm for solving the Hamiltonian cycle problem on directed one-trees.

Keywords: tree-width, tree-decomposition, d-width, d-decomposition, and haven order.

1 Introduction

In 1996, Reed et al. (5) proved Youngers's conjecture (7) roughly saying that every directed graph has either a large set of disjoint directed circuits or a small set of vertices that cover all directed circuits in that digraph. They considered an analogous definition of *well-linked* sets for directed graphs and, regarding the fact that the size of the largest well-linked set in undirected graphs has close relationship with tree-width (4), they suggested the idea that the analogous definition of tree-width for directed graphs might have many applications (3).

Following that suggestion, Johnson, Robertson, Seymour, and Thomas (2) introduced the first formal definition for directed tree-decomposition called *arboreal decomposition*, and, as evidence that their definition is a good measure of digraph connectivity, they proved the following theorem relating the minimum width of an arboreal decomposition, tree-width(D), and the haven order, $H(D)$, of a digraph D.

Theorem 1 (Johnson et al. (2)) $H(D) - 1 \leq$ *tree-width*$(D) \leq 3H(D) - 2$ *for digraphs* D.

Safari (6) considered a restricted class of arboreal decompositions, called *d-decompositions*, that more closely resemble the undirected version of tree-decomposition. The minimum width of a d-decomposition of a digraph D is called the *d-width* of D. Since d-decompositions are a subset of tree-decompositions:

Theorem 2 (Safari (6)) *For any digraph* D*, tree-width*$(D) \leq$ *d-width*(D).

Johnson et al. (2) conjectured that for any digraph D, tree-width(D) equals $H(D) - 1$, i.e. a min-max equality holds between haven order (minus one) and tree-width. Safari (6) conjectured that d-width equals tree-width for all digraphs.

This paper is devoted to the class of directed one-trees, i.e. the class of digraphs whose d-width is one. We show, in Section 3, that d-width, tree-width, and haven order minus one, are equal for this class.

[†]Research supported by an NSERC Discovery Grant.

This establishes the equality of d-width and tree-width for a non-trivial class of digraphs, and it confirms the conjecture of Johnson et al. for this class as well. We also provide, in Section 4, fast algorithms for recognition and the Hamiltonian cycle problem on directed one-trees.

2 Preliminaries

For general concepts of tree-decomposition and tree-width on undirected graphs including various related terms and algorithms the reader is referred to Bodlaender (1). Here we define some related concepts for digraphs that we use in this paper.

A *haven* of order w in D (for integer w) is a function β that assigns to every subset X of less than w vertices of D a (non-empty) strongly connected component of $D \setminus X$ with the extra condition that if X and Y are two subsets of size less than w and X is a subset of Y, then $\beta(Y)$ is a subset of $\beta(X)$. The haven order of a digraph D, represented by $H(D)$, is the maximum w such that D has a haven of order w.

A tree-decomposition of an *undirected* graph G is a pair (T, W) where T is a tree and $W = \{W_i | i \in V(T)\}$ is a family of subsets of $V(G)$ such that

1. For every edge $(u, v) \in E(G)$ there exists node $i \in V(T)$ such that $u, v \in W_i$.
2. If i, j, k are three nodes of T and j lies on the unique path from i to k, then $W_i \cap W_k \subset W_j$.

The width of a tree-decomposition is the minimum w for which $|W_i| \leq w + 1$ for all nodes $i \in V(T)$ and the tree-width of G is the minimum width over all tree-decompositions of G.

If T is a tree-decomposition of an undirected graph G then for every connected subgraph S of G, the nodes of T that contain vertices from S must form a connected subtree in T. More formally we can replace conditions 1 and 2 in the above definition by the single codition:

> For every connected subset $S \subset V(G)$ let $E' = \{(i, j) | W_i \cap W_j \cap S \neq \emptyset\}$ and $V' = \{i | i \in V(T) \text{ and } W_i \cap S \neq \emptyset\}$. Then $E' \cup V'$ must form a connected subtree in T.

By $E' \cup V'$ we mean the subgraph of G whose edges are E' and whose vertices are V' plus the end points of edges of E'. Condition 1 is the special case where $S = \{u\}$, $S = \{v\}$, and $S = \{u, v\}$. Condition 2 is the special case where S contains any single vertex from $W_i \cap W_k$.

The definition of d-decomposition is exactly the same except that *connected set* is changed to *strongly connected set* in the above definition: A d-decomposition for a directed graph D is a pair (T, W) where T is a tree and $W = \{W_i | i \in V(T)\}$ is a family of subsets of $V(D)$ such that

> For every **strongly** connected subset $S \subset V(D)$ let $E' = \{(i, j) | W_i \cap W_j \cap S \neq \emptyset\}$ and $V' = \{i | i \in V(T) \text{ and } W_i \cap S \neq \emptyset\}$. Then $E' \cup V'$ must form a connected subtree in T.

The width of T is the minimum w for which $|W_i| \leq w + 1$ for all nodes $i \in V(T)$. The d-width of D is the minimum width over all d-decompositions of D. Fig.1 shows a digraph with d-width one with its optimum d-decomposition. One can verify the above condition for all strongly connected subsets: $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{a, b, c\}$, $\{a, c, d\}$, $\{a, c, d, e\}$, $\{a, b, c, d\}$, and $\{a, b, c, d, e\}$.

3 A Min-Max Theorem on Directed One-Trees

In this section we prove the following theorem relating haven order and d-width of directed one-trees.

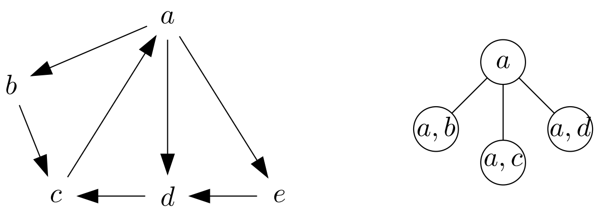

Fig. 1: A digraph (left) with its d-decomposition of width one (right).

Theorem 3 *A digraph D has d-width one if and only if it has haven order two.*

Proof: If D has d-width one then, by Theorems 1 and 2, it has haven order at most two. On the other hand, D has a cycle (otherwise its d-width would be zero) and, thus, D has haven order at least two. (Simply set $\beta(\emptyset) = C$ and let $\beta(\{x\})$ be a strongly connected component the contains a vertex of $C - \{x\}$, where C is a cycle in D.) Consequently D has haven order exactly two.

Next, we show if D has haven order two then its d-width is one. It suffices to prove this for strongly connected D since if D has haven order two and d-width d, it contains a strongly connected component with haven order two and d-width d.

The proof is by induction on the number of vertices of D. By Lemma 1, D contains a vertex u with out-degree (or in-degree) one. Suppose u has out-degree one (the in-degree one case is handled similarly) with edge (u, v) being its only outgoing edge. Contract the edge (u, v), by removing u and connecting all u's incoming edges to v, to obtain a new digraph D'. By Lemma 2, D' has haven order at most two and, according to the induction hypothesis, has d-width at most one‡. Let T' be a d-decomposition of D' with width at most one. Add a new node r to T' with $W_r = \{u, v\}$ and attach it to a node of T' that contains v. It is easy to verify that the resulting d-decomposition is a proper one for D and has d-width one. □

Lemma 1 *If D is strongly connected and has haven order two then D contains a vertex with in-degree or out-degree one.*

Proof: For any vertex u, a strongly connected component C of $D \setminus \{u\}$ is called *u-left* if there is no edge from a vertex in another component of $D \setminus \{u\}$ to a vertex in C. Similarly we say a component C is *u-right* if there is no edge from C to any other component. Let *leftright(u)* be the minimum size over all u-left and u-right components of $D \setminus \{u\}$.

If leftright(u) $= 1$ for some vertex u, then there is a single vertex v with either out-degree or in-degree one (to or from u). Otherwise, leftright(u) is at least two, for all u.

We show that D has haven order at least three, a contradiction. Let u be the vertex with minimum leftright(u) and C_u be the component that minimizes leftright(u), i.e. $|C_u| =$ leftright(u). Assume, without loss of generality, that C_u is a u-left component.

Let $\beta(x)$§, for any single vertex x, be the strongly connected component of $D \setminus \{x\}$ that contains C_u, if $x \notin C_u$, and the strongly connected component of $D \setminus \{x\}$ that contains u, otherwise. Moreover, let

‡ If D' has haven order one then it is acyclic and trivially has d-width zero.

§ In what follows we use $\beta(x)$ for $\beta(\{x\})$.

$\beta(\{x,y\})$ be the strongly connected component of $D \setminus \{x,y\}$ that contains $\beta(x) \cap \beta(y)$. We argue that β is a haven of order three.

It is sufficient to show that $\beta(x) \cap \beta(y) \neq \emptyset$ for all x and y. If x and y are both in C_u, then both $\beta(x)$ and $\beta(y)$ have vertex u in common. Similarly, if both are in $D \setminus C_u$, then both $\beta(x)$ and $\beta(y)$ share C_u.

For the final case, $x \in C_u$ and $y \in D \setminus C_u$, it suffices to show that $\beta(x)$ contains at least one vertex from C_u. Let $S_1, S_2, \ldots S_k$ be the strongly connected components of $D \setminus \{x\}$ that contain at least one vertex from C_u, in topological order. (At least one such component must exist because $|C_u| \geq 2$.) If any S_i contains u then we're done. Otherwise, each S_i contains only vertices from C_u because every path from $v \in D \setminus C_u$ to a vertex in C_u contains u (a consequence of C_u being u-left). Thus, $|S_i| < |C_u|$. We show that some S_i is an x-left or x-right component. This is a contradiction since C_u is supposedly the smallest such component.

For all $y \in C_u \setminus \{x\}$, there exists a path from u to y that contains only vertices in $C_u \setminus \{x\}$ (in particular, that doesn't contain x). If not, then the first component S_i (smallest i) that contains such a y is x-left, a contradiction.

Since C_u is a u-left component and $x \in C_u$, for all $z \in D \setminus C_u$, there exists a path from z to u that doesn't contain x. Hence S_k cannot have an outgoing edge (y,z) to a vertex $z \in D \setminus C_u$, otherwise y, z, and u would be strongly connected via paths that don't contain x. This implies that S_k is x-right, a contradiction. □

Lemma 2 *If D' is a digraph obtained by contracting an edge (u,v), with u having out-degree one or v having in-degree one, in a digraph D then $H(D') \leq H(D)$.*

Note: The same statement regarding directed tree-width of D and D' was noted by Johnson et al. (2).
Proof: Let β' be a haven of order w for D'. We construct a haven, β, of order w for D. Assume u has out-degree one. For any subset Z of vertices in D, if $u \in Z$, let $U(Z) = (Z - \{u\}) \cup \{v\}$, otherwise let $U(Z) = Z$. For Z with $|Z| < w$, let $\beta(Z)$ be the strongly connected component of D that contains $\beta'(U(Z))$. (Note: $|U(Z)| \leq |Z|$ so β is well-defined.) If C is a strongly connected component of $D' \setminus Z$ for some Z and $u \notin Z$ then either C or $C \cup \{u\}$ is a strongly connected component of $D \setminus Z$. Thus, $\beta(Z)$ equals either $\beta'(U(Z))$ or $\beta'(U(Z)) \cup \{u\}$. Therefore, for any two subsets $Z_1 \subset Z_2$ of less than w vertices of D, $U(Z_1) \subseteq U(Z_2)$, so $\beta(Z_1) \cap \beta(Z_2) \supseteq \beta'(U(Z_1)) \cap \beta'(U(Z_2)) = \beta'(U(Z_2)) \neq \emptyset$. Thus, $\beta(Z_2) \subseteq \beta(Z_1)$. Notice that if C_1 and C_2 are two strongly connected componets of $D \backslash Z_1$ and $D \backslash Z_2$, respectively, then either $C_2 \subseteq C_1$ or $C_1 \cap C_2 = \emptyset$. The case when v has in-degree one is similar. □

Corollary 1 *The three statements "D has d-width one", "D has tree-width one", and "D has haven order two" are equivalent.*

Proof: This follows from Theorems 1, 2, and 3. □

4 Algorithmic results

The nice property of directed one-trees is that they have a contractible edge. We can use this property to design recursive algorithms for certain problems on directed one-trees.

In the following algorithms, we assume that the contractible edge is (u,v) (with u having out-degree one). We contract the edge by removing u and connecting all u's incoming edges to v. The case when u has in-degree one is handled analogously.

Recognition To find a d-decomposition with width one, if it exists, of a digraph D:

0. If D is a single vertex return a single node containing the vertex.

1. Find a contractible edge (u, v), contract it, and obtain a new digraph D'. If no contractible edge exists then FAIL.

2. Recursively find a d-decomposition T' for D'.

3. Look for a node of T' that contains v, and add a new node r to it with $W_r = \{u, v\}$

If we keep the list of vertices ordered by in-degree and also by out-degree, we can perform steps 1, 2, and 3 in $O(n)$ time. Thus, the total running time is $O(n^2)$.

Hamiltonian cycle To find a Hamiltonian cycle, if it exists, in a directed one-tree D in $O(n^2)$ time:

0. If D is a single vertex return the vertex. If D is acyclic then FAIL.

1. Find a contractible edge (u, v), contract it, and obtain a new digraph D'. Also remove all edges (x, v) in D' where (x, v) is an edge in D. If no contractible edge exists then FAIL.

2. Recursively find a Hamiltonian cycle C in D'.

3. Replace the edge (x, v) in C with $(x, u), (u, v)$ to obtain a Hamiltonian cycle for D'.

5 Conclusions and Future Work

We show in this paper that d-width and tree-width are equivalent on directed one-trees, which can be seen as a generalization of directed acyclic graphs (DAGs). Various algorithmic problems that have efficient solutions on DAGs may also be solvable efficiently for directed one-trees. The main graph-theoretic problem regarding directed tree-width is whether the min-max equality between haven order (minus one) and d-width (or tree-width) holds. Identifying the class of directed k-trees, for $k > 1$, is another interesting problem. One main algorithmic problem is computing optimum (or approximately optimum) d-decompositions or tree-decompositions for bounded d-width graphs. Finding efficient algorithms for generally hard problems on the class of bounded d-width graphs is another interesting problem.

References

[1] H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–21, 1993.

[2] T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory (Series B)*, 82:128–154, 2001.

[3] B. Reed. Tree width and tangles: A new connectivity measure and some applications. *Survey in Combinatorics*, 241:87–158, 1997.

[4] B. Reed. Introducing directed tree width. In H.J. Broersma, U. Faigle, C. Hoede, and J.L. Hurink, editors, *Electronic Notes in Discrete Mathematics*, volume 3. Elsevier, 2000.

[5] B. Reed, N. Robertson, P. Seymour, and R. Thomas. On packing directed circuits. *Combinatorica*, 16:535–554, 1996.

[6] M.A. Safari. Directed tree-width. Master's thesis, School of Computer Science, University of Waterloo, 2003.

[7] D. H. Younger. Graphs with interlinked directed circuits. In *Proceedings of the Mid-west Symposium on Circuit Theory*, volume 2, pages XVI 2.1 – XVI 2.7, 1973.